Command Line Interface

Administration Guide

Version NGX R65

Check Point®
SOFTWARE TECHNOLOGIES LTD.
We Secure the Internet.

January 2007

For third party notices, see: THIRD PARTY TRADEMARKS AND COPYRIGHTS.

Contents

Preface

Chapter 1 **Introduction to the CLI**

Chapter 2 **SmartCenter and Firewall Commands**

Chapter 3 VPN-1 Commands

Chapter 4 SmartView Monitor Commands

Chapter 5 SecureClient Commands

Preface

In This Chapter

Who Should Use This Guide

This guide is intended for administrators responsible for maintaining network security within an enterprise, including policy management and user support.

This guide assumes a basic understanding of

- System administration.
- The underlying operating system.
- Internet protocols (IP, TCP, UDP etc.).

Summary of Contents

This guide describes the VPN components of VPN-1 Power. It contains the following sections and chapters:

This section describes the basic components of a VPN and provides the background for the technology that comprises the VPN infrastructure.

Chapter	Description
Chapter 1, "Introduction to the CLI"	Purpose of this guide, and how to debug SmartConsole clients.
Chapter 2, "SmartCenter and Firewall Commands"	Commands for controlling the SmartCenter server and the firewall components of the SmartCenter server and of Check Point gateways.
Chapter 3, "VPN-1 Commands"	The vpn command and its subcommands, used for controlling the VPN component of Check Point gateways.
Chapter 4, "SmartView Monitor Commands"	The rtm command its subcommands, used to execute SmartView Monitor operations.
Chapter 5, "SecureClient Commands"	The scc command and its subcommands are VPN commands that are executed on SecureClient. They are used to generate status information, stop and start services, or connect to defined sites using specific user profiles.
Chapter 6, "ClusterXL Commands"	Commands used for controlling, monitoring and troubleshooting ClusterXL gateway clusters.

Related Documentation

The NGX R65 release includes the following documentation

TABLE P-1 VPN-1 Power documentation suite documentation

Title	Description
Internet Security Product Suite Getting Started Guide	Contains an overview of NGX R65 and step by step product installation and upgrade procedures. This document also provides information about What's New, Licenses, Minimum hardware and software requirements, etc.
Upgrade Guide	Explains all available upgrade paths for Check Point products from VPN-1/FireWall-1 NG forward. This guide is specifically geared towards upgrading to NGX R65.
SmartCenter Administration Guide	Explains SmartCenter Management solutions. This guide provides solutions for control over configuring, managing, and monitoring security deployments at the perimeter, inside the network, at all user endpoints.
Firewall and SmartDefense Administration Guide	Describes how to control and secure network access; establish network connectivity; use SmartDefense to protect against network and application level attacks; use Web Intelligence to protect web servers and applications; the integrated web security capabilities; use Content Vectoring Protocol (CVP) applications for anti-virus protection, and URL Filtering (UFP) applications for limiting access to web sites; secure VoIP traffic.
Virtual Private Networks Administration Guide	This guide describes the basic components of a VPN and provides the background for the technology that comprises the VPN infrastructure.

TABLE P-1 VPN-1 Power documentation suite documentation (continued)

Title	Description
Eventia Reporter Administration Guide	Explains how to monitor and audit traffic, and generate detailed or summarized reports in the format of your choice (list, vertical bar, pie chart etc.) for all events logged by Check Point VPN-1 Power, SecureClient and SmartDefense.
SecurePlatform™/ SecurePlatform Pro Administration Guide	Explains how to install and configure SecurePlatform. This guide will also teach you how to manage your SecurePlatform and explains Dynamic Routing (Unicast and Multicast) protocols.
Provider-1/SiteManager-1 Administration Guide	Explains the Provider-1/SiteManager-1 security management solution. This guide provides details about a three-tier, multi-policy management architecture and a host of Network Operating Center oriented features that automate time-consuming repetitive tasks common in Network Operating Center environments.

TABLE P-2 Integrity Server documentation

Title	Description
Integrity Advanced Server Installation Guide	Explains how to install, configure, and maintain the Integrity Advanced Server.
Integrity Advanced Server Administrator Console Reference	Provides screen-by-screen descriptions of user interface elements, with cross-references to relevant chapters of the Administrator Guide. This document contains an overview of Administrator Console navigation, including use of the help system.
Integrity Advanced Server Administrator Guide	Explains how to managing administrators and endpoint security with Integrity Advanced Server.
Integrity Advanced Server Gateway Integration Guide	Provides information about how to integrating your Virtual Private Network gateway device with Integrity Advanced Server. This guide also contains information regarding deploying the unified SecureClient/Integrity client package.

TABLE P-2 Integrity Server documentation (continued)

Title	Description
Integrity Advanced Server System Requirements	Provides information about client and server requirements.
Integrity Agent for Linux Installation and Configuration Guide	Explains how to install and configure Integrity Agent for Linux.
Integrity XML Policy Reference Guide	Provides the contents of Integrity client XML policy files.
Integrity Client Management Guide	Explains how to use of command line parameters to control Integrity client installer behavior and post-installation behavior.

More Information

- For additional technical information about Check Point products, consult Check Point’s SecureKnowledge at https://secureknowledge.checkpoint.com/.
- See the latest version of this document in the User Center at http://www.checkpoint.com/support/technical/documents

Feedback

Check Point is engaged in a continuous effort to improve its documentation. Please help us by sending your comments to:

cp_techpub_feedback@checkpoint.com

Chapter 1

Introduction to the CLI

In This Chapter

Introduction

This guide documents the Command Line Interface (CLI) commands across different Check Point Products and features. The commands are documented according to the product for which they are used.

Within each product chapter, the commands are arranged alphabetically.

For Provider-1/SiteManager-1 CLI commands, see the Provider-1/SiteManager-1 Administration Guide.

General Information

Debugging SmartConsole Clients

It is possible to obtain debugging information on any of the SmartConsole clients by running these clients in a debug mode. You can save the debug information in a default text file, or you can specify another file in which this information should be saved.

Usage: `<fwpolicy.exe> -d -o <Debug-Output-File-Name.txt>`

Syntax:

parameter	meaning
`-d`	enter the debug mode. If `-o` is omitted, debug information is saved into a file with the default name: `<ROLE_STR>_debug_output.txt.`
-o	This optional parameter, followed by a file name indicates in which text file debug information should be saved.

Chapter 2

SmartCenter and Firewall Commands

In This Chapter

comp_init_policy

Description Use the `comp_init_policy` command to generate and load, or to remove, the Initial Policy.

Usage

```
$FWDIR/bin/comp_init_policy [-u | -g]
```

Syntax

Argument	Description
`-u`	Removes the current Initial Policy, and ensures that it will not be generated in future when `cpconfig` is run.

`-g`	Can be used if there is no Initial Policy. If there is, make sure that after removing the policy, you delete the `$FWDIR\state\local\FW1\` folder. Generates the Initial Policy and ensures that it will be loaded the next time a policy is fetched (at `cpstart`, or at next boot, or via the `fw fetch localhost` command). After running this command, `cpconfig` will add an Initial Policy when needed.
	The `comp_init_policy -g` command will only work if there is no previous Policy. If you perform the following commands:
	`comp_init_policy -g + fw fetch localhost`
	`comp_init_policy -g + cpstart`
	`comp_init_policy -g + reboot`
	The original policy will still be loaded.

cpca_client

Description This command and all its derivatives are used to execute operations on the ICA.

Usage `cpca_client`

In This Section

cpca_client create_cert

Description This command prompts the ICA to issue a SIC certificate for the SmartCenter server.

Usage

```
cpca_client [-d] create_cert [-p <ca_port>] -n "CN=<common
name>" -f <PKCS12 filename>
```

Syntax

Argument	Description
-d	Debug flag
-p <ca_port>	Specifies the port which is used to connect to the CA (if the CA was not run from the default port 18209)
-n "CN=<common name>"	sets the CN
-f <PKCS12 filename>	specifies the file name where the certificate and keys are saved.

cpca_client revoke_cert

Description This command is used to revoke a certificate issued by the ICA.

Usage

```
cpca_client [-d] revoke_cert [-p <ca_port>] -n "CN=<common
name>"
```

Syntax

Argument	Description
-d	debug flag
-p <ca_port>	specifies the port which is used to connect to the CA (if the CA was not run from the default port 18209)
-n "CN=<common name>"	sets the CN

cpca_client set_mgmt_tools

Description This command is used to invoke or terminate the ICA Management Tool.

Usage

```
cpca_client [-d] set_mgmt_tools on|off  [-p <ca_port>]
[-no_ssl] [-a|-u "administrator|user DN" -a|-u
"administrator|user DN" ... ]
```

Syntax

Argument	Description
`-d`	debug flag
set_mgmt_tools onloff	• `on` - Start the ICA Management tool • `off` - Stop the ICA Management tool
-p <ca_port>	Specifies the port which is used to connect to the CA (if the appropriate service was not run from the default port 18265)
-no_ssl	Configures the server to use clear http rather than https.
-al-u"administratorluser DN"	Sets the DNs of the administrators or user that permitted to use the ICA Management tool

Comments Note the following:

1. If the command is ran without `-a` or `-u` the list of the permitted users and administrators isn’t changed. The server can be stopped or started with the previously defined permitted users and administrators.
2. If two consecutive start operations are initiated the ICA Management Tool will not respond, unless you change the ssl mode. Once the ssl mode has been modified, the server can be stopped and restarted.

cp_conf

Description This command is used to configure/reconfigure a VPN-1 installation via the cli internafce. The configuration options shown depend on the installed configuration and products.

Usage `cp_conf`

In This Section

cp_conf sic

Description Enables the user to manage SIC.

Usage

```
cp_conf sic state # Get the current Trust state
cp_conf sic init <Activation Key> [norestart] # Initialize
SIC
cp_conf sic cert_pull <SmartCenter Server name/IP> <module
object name> # Pull certificate (DAIP only)
```

cp_conf admin

Description Use this command to manage the Check Point Administrator.

Usage

```
cp_conf admin get # Get the list of administrators.
cp_conf admin add <user> <passw> <permissions> # Add
administrator
where permissions:
w - read/write
r - read only
cp_conf admin del <admin1> <admin2>... # Delete
administrators.
```

cp_conf ca

Description Use this command to initialize the Certificate Authority

Usage

```
cp_conf ca init # Initializes Internal CA.
cp_conf ca fqdn <name> # Sets the name of the Internal CA.
```

cp_conf finger

Description Displays the fingerprint which will be used on first-time launch to verify the identity of the SmartCenter server being accessed by the SmartConsole. This fingerprint is a text string derived from the SmartCenter server’s certificate

Usage

```
cp_conf finger get # Get Certificate's Fingerprint.
```

cp_conf lic

Description Use this command to enable the administrator to add a license manually and to view the license installed.

Usage

```
cp_conf lic get # Get licenses installed.
cp_conf lic add -f <file name> # Add license from file.
cp_conf lic add -m <Host> <Date> <Signature Key>
<SKU/Features> # Add license manually.
cp_conf lic del <Signature Key> # Delete license.
```

cp_conf client

Description Use this command to manage the GUI Clients allowed to connect to the management.

Usage

```
cp_conf client get # Get the GUI Clients list
cp_conf client add < GUI Client > # Add one GUI Client
cp_conf client del < GUI Client 1> < GUI Client 2>... #
Delete GUI Clients
cp_conf client createlist < GUI Client 1> < GUI Client
2>... # Create new list.
```

cp_conf ha

Description Use this command to enable or disable the High Availability module.

Usage

```
cp_conf ha enable/disable [norestart] # Enable/Disable
HA\n",
```

cp_conf snmp

Description Use this command activate or deactivate SNMP.

Usage

```
cp_conf snmp get # Get SNMP Extension status.
cp_conf snmp activate/deactivate [norestart] # Deactivate
SNMP Extension.
```

cp_conf auto

Description Use this command to determine whether or not the Firewall/SmartCenter starts automatically after the machine restarts.

Usage

```
cp_conf auto get [fw1] [fg1] [rm] [all] # Get the auto
state of products.
cp_conf auto <enable|disable> <product1> <product2>... #
Enable/Disable auto start.
```

cp_conf sxl

Description Use this command to enable or disable the SecureXL acceleration module.

Usage

```
cp_conf sxl <enable|disable> # Enable/Disable SecureXL.
```

cpconfig

Description This command is used to run a Command Line version of the Check Point Configuration Tool. This tool is used to configure/reconfigure a VPN-1 installation. The configuration options shown depend on the installed configuration and products. Amongst others, these options include:

- Licenses - modify the necessary Check Point licenses
- Administrators - modify the administrators authorized to connect to the SmartCenter server via the SmartConsole
- GUI Clients - modify the list of GUI Client machines from which the administrators are authorized to connect to a SmartCenter server
- Certificate Authority - install the Certificate Authority on the SmartCenter server in a first-time installation
- Key Hit Session - enter a random seed to be used for cryptographic purposes.

- Secure Internal Communication - set up trust between the gateway on which this command is being run and the SmartCenter server
- Fingerprint - display the fingerprint which will be used on first-time launch to verify the identity of the SmartCenter server being accessed by the SmartConsole. This fingerprint is a text string derived from the SmartCenter server's certificate.

Usage

```
cpconfig
```

Further Info. See the *Getting Started* Guide and the *SmartCenter* Administration Guide.

cplic

Description This command and all its derivatives relate to the subject of Check Point license management. All `cplic` commands are located in `$CPDIR/bin`. License Management is divided into three types of commands:

- *Local licensing commands* are executed on local machines.
- *Remote licensing commands* are commands which affect remote machines are executed on the SmartCenter server.
- *License repository commands* are executed on the SmartCenter server.

Usage

```
cplic
```

In This Section

cplic check

Description Use this command to check whether the license on the local machine will allow a given feature to be used.

Usage

```
cplic check [-p <product name>] [-v <product version>] [-c
count] [-t <date>] [-r routers] [-S SRusers] <feature>
```

Syntax

Argument	Description
-p <product name>	The product for which license information is requested. For example `fw1`, `netso`.
-v <product version>	The product version for which license information is requested. For example `4.1`, `5.0`
-c count	Count the licenses connected to this feature
-t <date>	Check license status on future date. Use the format *ddmmmyyyy*. A given feature may be valid on a given date on one license, but invalid in another.
-r routers	Check how many routers are allowed. The `feature` option is not needed.
-S SRusers	Check how many SecuRemote users are allowed. The `feature` option is not needed
<feature>	The `<feature>` for which license information is requested.

cplic db_add

Description The `cplic db_add` command is used to add one or more licenses to the license repository on the SmartCenter server. When local license are added to the license repository, they are automatically attached to its intended Check Point gateway, central licenses need to undergo the attachment process.

Usage

```
cplic db_add < -l license-file | host expiration-date
signature SKU/features >
```

Syntax

Argument	Description
`-l license-file`	adds the license(s) from `license-file`. The following options are **NOT** needed: Host Expiration-Date Signature SKU/feature

Comments

This command is a license repository command, it can only be executed on the SmartCenter server.

Copy/paste the following parameters from the license received from the User Center. More than one license can be added.

- `host` - the target hostname or IP address
- `expiration date` - The license expiration date.
- `signature` -The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (Case sensitive. The hyphens are optional)
- `SKU/features` - The SKU of the license summarizes the features included in the license. For example: `CPSUITE-EVAL-3DES-vNG`

Example

If the file `192.168.5.11.lic` contains one or more licenses, the command: `cplic db_add -l 192.168.5.11.lic` will produce output similar to the following:

```
Adding license to database ...
Operation Done
```

cplic db_print

Description

The `cplic db_print` command displays the details of Check Point licenses stored in the license repository on the SmartCenter server.

Usage

```
cplic db_print <object name | -all> [-n noheader] [-x print
signatures] [-t type] [-a attached]
```

Syntax

Argument	**Description**
`Object name`	Print only the licenses attached to `Object name`. `Object name` is the name of the Check Point gateway object, as defined in SmartDashboard.
`-all`	Print all the licenses in the license repository
`-noheader` (or `-n`)	Print licenses with no header.
`-x`	Print licenses with their signature

`-t`
`(or -type)`
	Print licenses with their type: Central or Local.

`-a` (or `-attached`)	Show which object the license is attached to. Useful if the `-all` option is specified.

Comments **This command is a license repository command, it can only be executed on the SmartCenter server.**

cplic db_rm

Description The `cplic db_rm` command removes a license from the license repository on the SmartCenter server. It can be executed ONLY after the license was detached using the `cplic del` command. Once the license has been removed from the repository, it can no longer be used.

Usage

```
cplic db_rm <signature>
```

Syntax

Argument	**Description**
`Signature`	The signature string within the license.

Example

```
cplic db_rm 2f540abb-d3bcb001-7e54513e-kfyigpwn
```

Comments **This command is a license repository command, it can only be executed on the SmartCenter server.**

cplic del

Description Use this command to delete a single Check Point license on a host, including unwanted evaluation, expired, and other licenses. This command is used for both local and remote machines

Usage

```
cplic del [-F <output file>] <signature> <object name>
```

Syntax

Argument	Description
-F <output file>	Send the output to `<output file>` instead of the screen.
<signature>	The signature string within the license.

cplic del <object name>

Description Use this command to detach a Central license from a Check Point gateway. When this command is executed, the license repository is automatically updated. The Central license remains in the repository as an unattached license. This command can be executed only on a SmartCenter server.

Usage

```
cplic del <Object name> [-F outputfile] [-ip dynamic ip]
<Signature>
```

Syntax

Argument	Description
`object name`	The name of the Check Point gateway object, as defined in SmartDashboard.

Argument	Description
`-F outputfile`	Divert the output to `outputfile` rather than to the screen.
`-ip dynamic ip`	Delete the license on the Check Point gateway with the specified IP address. This parameter is used for deleting a license on a DAIP Check Point gateway **Note -** If this parameter is used, then object name must be a DAIP gateway.
`Signature`	The signature string within the license.

Comments This is a *Remote Licensing Command* which affects remote machines that is executed on the SmartCenter server.

cplic get

Description The `cplic get` command retrieves all licenses from a Check Point gateway (or from all Check Point gateways) into the license repository on the SmartCenter server. Do this to synchronize the repository with the Check Point gateway(s). When the command is run, all local changes will be updated.

Usage

```
cplic get <ipaddr | hostname | -all> [-v41]
```

Syntax

Argument	Description
`ipaddr`	The IP address of the Check Point gateway from which licenses are to be retrieved.

Argument	Description
`hostname`	The name of the Check Point gateway object (as defined in SmartDashboard) from which licenses are to be retrieved.
`-all`	Retrieve licenses from all Check Point gateways in the managed network.
`-v41`	Retrieve version 4.1 licenses from the NF Check Point gateway. Used to upgrade version 4.1 licenses.

Example If the Check Point gateway with the object name `caruso` contains four Local licenses, and the license repository contains two other Local licenses, the command: `cplic get caruso` produces output similar to the following

```
Get retrieved 4 licenses.
Get removed 2 licenses.
```

Comments This is a *Remote Licensing Command* which affects remote machines that is executed on the SmartCenter server.

cplic put

Description The `cplic put` command is used to install one or more Local licenses on a local machine.

Usage

```
cplic put [-o overwrite] [-c check-only] [-s select] [-F
<output file>]
[-P Pre-boot] [-k kernel-only] <-l license-file | host
expiration date signature SKU/feature>
```

Syntax

Argument	Description
`-overwrite` (or `-o`)	On a SmartCenter server this will erase all existing licenses and replace them with the new license(s). On a Check Point gateway this will erase only Local licenses but not Central licenses, that are installed remotely.
`-check-only` (or `-c`)	Verify the license. Checks if the IP of the license matches the machine, and if the signature is valid
`select` (or `-s`)	Select only the Local licenses whose IP address matches the IP address of the machine.
`-F outputfile`	Outputs the result of the command to the designated file rather than to the screen.
`-Preboot` (or `-P`)	Use this option after upgrading to VPN-1/FireWall-1 NG FP2 and before rebooting the machine. Use of this option will prevent certain error messages.
-kernel-only (or -k)	Push the current valid licenses to the kernel. For Support use only.
`-l license-file`	Installs the license(s) in `license-file`, which can be a multi-license file. The following options are NOT needed: *`host expiration-date signature SKU/features`*

Comments

Copy and paste the following parameters from the license received from the User Center.

- `host` - One of the following:

All platforms - The IP address of the external interface (in dot notation); last part cannot be 0 or 255.

Sun OS4 and Solaris2 - The response to the `hostid` command (beginning with 0x).

HP-UX - The response to the `uname -i` command (beginning with 0d).

AIX - The response to the `uname -l` command (beginning with 0d), or the response to the `uname -m` command (beginning and ending with 00).

- `expiration date` - The license expiration date. Can be `never`
- `signature` -The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (Case sensitive. The hyphens are optional)
- `SKU/features` - A string listing the SKU and the Certificate Key of the license. The SKU of the license summarizes the features included in the license. For example: `CPMP-EVAL-1-3DES-NG CK0123456789ab`

Example `cplic put -l 215.153.142.130.lic` produces output similar to the following:

```
Host             Expiration SKU
215.153.142.130  26Dec2001  CPMP-EVAL-1-3DES-NG
CK0123456789ab
```

cplic put <object name> ...

Description Use the `cplic put` command to attach one or more central or local license remotely.When this command is executed, the license repository is also updated.

Usage

```
cplic put <object name> [-ip dynamic ip] [-F <output file>]
< -l license-file | host expiration-date signature
SKU/features >
```

Argument	Description
`Object name`	The name of the Check Point gateway object, as defined in SmartDashboard.
`-ip dynamic ip`	Install the license on the Check Point gateway with the specified IP address. This parameter is used for installing a license on a DAIP Check Point gateway. **NOTE**: If this parameter is used, then object name must be a DAIP Check Point gateway.
`-F outputfile`	Divert the output to `outputfile` rather than to the screen.
`-l license-file`	Installs the license(s) from `license-file`. The following options are **NOT** needed: `Host Expiration-Date Signature SKU/features`

Comments This is a *Remote Licensing Command* which affects remote machines that is executed on the SmartCenter server.

This is a Copy and paste the following parameters from the license received from the User Center. More than one license can be attached

- `host` - the target hostname or IP address
- `expiration date` - The license expiration date. Can be `never`
- `signature` -The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (Case sensitive. The hyphens are optional)
- `SKU/features` - A string listing the SKU and the Certificate Key of the license. The SKU of the license summarizes the features included in the license. For example: `CPMP-EVAL-1-3DES-NG CK0123456789ab`

cplic print

Description The `cplic print` command (located in `$CPDIR/bin`) prints details of Check Point licenses on the local machine.

Usage

```
cplic print [-n noheader][-x prints signatures][-t type][-F
<outputfile>] [-p preatures]
```

Syntax

Argument	Description
`-noheader` (or `-n`)	Print licenses with no header.
`-x`	Print licenses with their signature
`-type` (or -t)	Prints licenses showing their type: Central or Local.
`-F` <outputfile>	Divert the output to `outputfile`.
`-preatures` (or `-p`)	Print licenses resolved to primitive features.

Comments On a Check Point gateway, this command will print all licenses that are installed on the local machine — both Local and Central licenses.

cplic upgrade

Description Use the `cplic upgrade` command to upgrade licenses in the license repository using licenses in a license file obtained from the User Center.

Usage

```
cplic upgrade <-l inputfile>
```

Syntax

Argument	Description
`-l inputfile`	Upgrades the licenses in the license repository and Check Point gateways to match the licenses in `<inputfile>`

Example The following example explains the procedure which needs to take place in order to upgrade the licenses in the license repository.

- Upgrade the SmartCenter server to the latest version.

  Ensure that there is connectivity between the SmartCenter server and the remote workstations with the version 4.1 products.

- Import all licenses into the license repository. This can also be done *after* upgrading the products on the remote workstations to NG
- Run the command: `cplic get -all`. For example

```
Getting licenses from all modules ...

count:root(su) [~] # cplic get -all
golda:
Retrieved 1 licenses.
Detached  0 licenses.
Removed  0 licenses.
count:
Retrieved 1 licenses.
Detached  0 licenses.
Removed   0 licenses.
```

- To see all the licenses in the repository, run the command:

  `cplic db_print -all -a`

```
count:root(su) [~] # cplic db_print -all -a

Retrieving license information from database ...

The following licenses appear in the database:
==================================================


Host          Expiration Features
192.168.8.11  Never      CPFW-FIG-25-41       CK-49C3A3CC7
121 golda
192.168.5.11  26Nov2002  CPSUITE-EVAL-3DES-NG CK-123456789
0 count
```

- Upgrade the version 4.1 products on the remote Check Point gateways.
- In the User Center (http://www.checkpoint.com/usercenter), view the licenses for the products that were upgraded from version 4.1 to NG and create new upgraded licenses.
- Download a file containing the upgraded NG licenses. Only download licenses for the products that were upgraded from version 4.1 to NG.
- If you did not import the version 4.1 licenses into the repository in step •, import the version 4.1 licenses now using the command `cplic get -all -v41`
- Run the license upgrade command: `cplic upgrade –l <inputfile>`

- The licenses in the downloaded license file and in the license repository are compared.

- If the certificate keys and features match, the old licenses in the repository and in the remote workstations are updated with the new licenses.

- A report of the results of the license upgrade is printed.

In the following example, there are two NG licenses in the file. One does not match any license on a remote workstation, the other matches a version 4.1 license on a remote workstation that should be upgraded:

Comments This is a *Remote Licensing Command* which affects remote machines that is executed on the SmartCenter server.

Further Info. See the *SmartUpdate* chapter of the *SmartCenter* Administration Guide*.*

cp_merge

Description The cp_merge utility has two main functionalities

- Export and import of policy packages
- Merge of objects from a given file into SmartCenter database

Usage `cp_merge help`

Syntax

Argument	Description
`help`	Displays the usage for `cp_merge`.

In This Section

cp_merge delete_policy

Description This command provides the options of deleting an existing policy package. Note that the default policy can be deleted by delete action.

Usage

```
cp_merge delete_policy [-s <db server>] [-u <user> | -c
<certificate file>] [-p <password>] -n <package name>
```

Syntax

Argument	Description
-s <db server>	Specify the database server IP Address or DNS name.[2]
-u <user>	The administrator's name.[1,2]
-c <certificate file>	The path to the certificate file.[1]
-p <password>	The administrator's password.[1]
-n <policy package name>	The policy package to export.[2,3]

Comments

Further considerations:

1. Either use certificate file or user and password

2. Optional

Example

Delete the policy package called `standard`.

```
cp_merge delete_policy -n Standard
```

cp_merge export_policy

Description

This command provides the options of leaving the policy package in the active repository, or deleting it as part of the export process. The default policy cannot be deleted during the export action.

Usage

```
cp_merge export_policy [-s <db server>] [-u <user> | -c
<certificate file>] [-p <password>][-n <policy package
name> | -l <policy name>] [-d <output directory>] [-f
<outputfile>] [-r]
```

Syntax

Argument	Description
-s <db server>	Specify the database server IP Address or DNS name.[2]
-u <user>	The database administrator's name.[1]
-c <certificate file>	The path to the certificate file.[1]
-p <password>	The administrator's password.[1]
-n <policy package name	The policy package to export.[2,3]

Argument	Description
-l <policy name>	Export the policy package which encloses the policy name.[2,3,4]
-d <output directory>	Specify the output directory.[2]
-f <outputfile>	Specify the output file name (where the default file name is `<policy name>.pol`).[2]
-r	Remove the original policy from the repository.[2]

Comments Further considerations:

1. Either use certificate file or user and password
2. Optional
3. If both `-n` and `-l` are omitted all policy packages are exported.
4. If both `-n` and `-l` are present `-l` is ignored.

Example Export policy package Standard to file

```
cp_merge export_policy -n Standard -f
StandardPolicyPackageBackup.pol -d C:\bak
```

cp_merge import_policy and cp_merge restore_policy

Description This command provides the options to overwrite an existing policy package with the same name, or preventing overwriting when the same policy name already exists

Usage

```
cp_merge import_policy|restore_policy [-s <db server>] [-u
<user> | -c <certificate file>] [-p <password>][-n <package
name>] [-d <input directory>] -f <input file> [-v]
```

Syntax

Argument	Description
-s <db server>	Specify the database server IP address or DNS name.[2]
-u <user>	The administrator’s name.[1,2]
-c <certificate file>	The path to the certificate file.[1]
-p <password>	The administrator’s password.[1,2]

Argument	Description
-n <policy package name	Rename the policy package to <policy package name> when importing.[2]
-d <input directory>	Specify the input directory.[2]
-f <inputfile>	Specify the input file name.
-v	Override an existing policy if found.[2]

Comments Further considerations

1. Either use certificate file or user and password

2. Optional

The `cp_merge restore_policy` works only locally on the SmartCenter server and it will not work from remote machines.

Caution: A FireWall-1 policy from `<policy>.W` file can be restored using this utility; however, important information may be lost when the policy is translated into `.W` format. This restoration should be used only if there is no other backup of the policy.

Example Import the policy package saved in file `Standard.pol` into the repository and rename it to `StandardCopy`.

```
cp_merge import_policy -f Standard.pol -n StandardCopy
```

cp_merge list_policy

Usage

```
cp_merge list_policy [-s <db server>] [-u <user> | -c
<certificate file>] [-p <password>]
```

Syntax

Argument	Description
-s <db server>	Specify the database server IP Address or DNS name.[2]
-u <user>	The administrator’s name.[1,2]
-c <certificate file>	The path to the certificate file.[1,2]
-p <password>	The administrator’s password.[1,2]

Comments Further considerations:

1. Either use certificate file or user and password

2. Optional

Example List all policy packages which reside in the specified repository:

```
cp_merge list -s localhost
```

cppkg

Description This command is used to manage the product repository. It is always executed on the SmartCenter server.

In This Section

cppkg add

Description The `cppkg add` command is used to add a product package to the product repository. Only SmartUpdate packages can be added to the product repository.

Products can be added to the Repository as described in the following procedures, by importing a file downloaded from the Download Center web site at http://www.checkpoint.com/techsupport/downloads/downloads.html. The package file can be added to the Repository directly from the CD or from a local or network drive.

Usage

```
cppkg add <package-full-path | CD drive>
```

Syntax

Argument	Description
`package-full-path`	If the package to be added to the repository is on a local disk or network drive, type the full path to the package.

`CD drive`	If the package to be added to the repository is on a CD:
	For Windows machines type the CD drive letter, e.g.
	`d:\`
	For UNIX machines, type the CD root path, e.g.
	`/caruso/image/CPsuite-NG/FP2`
	You will be asked to specify the product and appropriate Operating System (OS).

Comments

`cppkg add` does not overwrite existing packages. To overwrite existing packages, you must first delete existing packages.

Example

```
[d:\winnt\fw1\ng\bin]cppkg add l:\CPsuite-NG_FP2\
Enter package name:
----------------------
(1) SVNfoundation
(2) firewall
(3) floodgate
(4) rtm

(e) Exit
Enter you choice : 1
Enter package OS :
----------------------
(1) win32
(2) solaris
(3) linux
```

```
(4) hpux

(5) ipso

(6) aix


(e) Exit

Enter your choice : 1

You choose to add 'SVNfoundation' for 'win32' OS. Is this
correct? [y/n] : y

Adding package from CD ...

Package added to repository.
```

cppkg delete

Description The command is used to delete a product package from the repository. To delete a product package you must specify a number of options. To see the format of the options and to view the contents of the product repository, use the `cppkg print` command.

Usage `cppkg delete [<vendor> <product> <version> <os> [sp]]`

Syntax

Argument	**Description**
vendor	Package vendor (e.g. `checkpoint`).
product	Package name Options are: `SVNfoundation`, `firewall`, `floodgate`.
version	Package version (e.g. NG`)`.
os	Package Operating System. Options are: `win32` for Windows NT and Windows 2000, `solaris`, `hpux`, ipso, `aix`, `linux`.
sp	Package service pack (e.g. `fcs` for NG R54 initial release, `FP1`, `FP2` etc.) This parameter is optional. Its default is `fcs`.

Comments It is not possible to undo the cppkg del command.

Example

```
[d: \winnt\fw1\ng\bin]cppkg del

Getting information from package repository. Please wait...

Select package:

----------------------

(1) checkpoint SVNfoundation NG win32 FCS_FP1

(2) checkpoint SNVfoundation NG win32 FP1

(e) Exit

Enter your choice : 2

You choose to delete ‘checkpoint SVNfoundation NG win32 FP1’
Is this correct? [y/n] : y

Package removed from repository.
```

cppkg get

Description This command synchronizes the Package Repository database with the content of the actual package repository under $SUROOT.

Usage

```
cppkg get
```

cppkg getroot

Description The command is used to find out the location of the product repository. The default product repository location on Windows machines is C:\SUroot. On UNIX it is /var/SUroot

Usage

```
cppkg getroot
```

Example

```
# cppkg getroot
Current repository root is set to : /var/suroot/
```

cppkg print

Description The command is used to list the contents of the product repository.

Use `cppkg print` to see the product and OS strings required to install a product package using the `cprinstall` command, or to delete a package using the `cppkg delete` command.

Usage `cppkg print`

Example

```
[d:\winnt\fw1\ng\bin]cppkg print

Getting information from package repository. Please wait
...

Vendor     Product       Version OS      SP       Descript
ion
-----------------------------------------------------------
---
checkpoint SVNfoundation NG      win32   FCS_FP1  SVN
foundation NG Feature Pack 1 for 4.1 upgrade
checkpoint SVNfoundation NG      win32   FP1      SVN
foundation Feature Pack 1 for NG upgrade
```

cppkg setroot

Description The command is used to create a new repository root directory location, and to move existing product packages into the new repository.

The default product repository location is created when the SmartCenter server is installed. On Windows machines the default location is `C:\SUroot` and on UNIX it is `/var/SUroot`. Use this command to change the default location.

When changing repository root directory:

- The contents of the old repository is copied into the new repository.
- The `$SUROOT` environment variable gets the value of the new root path.
- A product package in the new location will be overwritten by a package in the old location, if the packages are the same (that is, they have the same ID strings).

The repository root directory should have at least 200 Mbyte of free disk space.

Usage `cppkg setroot <repository-root-directory-full-path>`

Syntax

Argument	Description
repository-root-directory-full-path	The desired location for the product repository.

Comments It is important to reboot the SmartCenter server after performing this command, in order to set the new `$SUROOT` environment variable.

Example

```
# cppkg setroot /var/new_suroot
Repository root is set to : /var/new_suroot/

Note: When changing repository root directory :

1. Old repository content will be copied into the new
repository.

2. A package in the new location will be overwritten by a
package in the old location, if the packages have the same
name.

Change the current repository root ? [y/n] : y

The new repository directory does not exist. Create it ?
[y/n] : y

Repository root was set to : /var/new_suroot

Notice : To complete the setting of your directory, reboot
the machine!
```

cpridrestart

Description Stops and starts the Check Point Remote Installation Daemon (`cprid`). This is the daemon that is used for remote upgrade and installation of products. It is part of the SVN Foundation. In Windows it is a service.

cpridstart

Description Start the Check Point Remote Installation Daemon (`cprid`). This is the service that allows for the remote upgrade and installation of products. It is part of the SVN Foundation. In Windows it is a service.

Usage `cpridstart`

cpridstop

Description Stop the Check Point Remote installation Daemon (`cprid`). This is the service that allows for the remote upgrade and installation of products. It is part of the SVN Foundation. In Windows it is a service.

Usage

```
cpridstop
```

cprinstall

Description Use `cprinstall` commands to perform remote installation of product packages, and associated operations.

On the SmartCenter server, `cprinstall` commands require licenses for SmartUpdate

On the remote Check Point gateways the following are required:

- Trust must be established between the SmartCenter server and the Check Point gateway.
- `cpd` must run.
- `cprid` remote installation daemon must run. `cprid` is available on VPN-1/FireWall-1 4.1 SP2 and higher, and as part of SVN Foundation for NG and higher.

In This Section

cprinstall boot

Description The command is used to boot the remote computer.

Usage

```
cprinstall boot <Object name>
```

Syntax

Argument	Description
Object name	Object name of the Check Point gateway defined in SmartDashboard.

Example

```
# cprinstall boot harlin
```

cprinstall cprestart

Description This command enables `cprestart` to be run remotely.

All products on the Check Point gateway must be of the same version of NG.

Usage

```
cprinstall cprestart <object name>
```

Syntax

Argument	Description
Object name	Object name of the Check Point gateway defined in SmartDashboard.

cprinstall cpstart

Description This command enables `cpstart` to be run remotely.

All products on the Check Point gateway must be of the same version of NG.

Usage

```
cprinstall cpstart <object name>
```

Syntax

Argument	Description
Object name	Object name of the Check Point gateway defined in SmartDashboard.

cprinstall cpstop

Description This command enables `cpstop` to be run remotely.

All products on the Check Point gateway must be of the same version of NG.

Usage

```
cprinstall cpstop <-proc | -nopolicy> <object name>
```

Syntax

Argument	Description
Object name	Object name of the Check Point gateway defined in SmartDashboard.
-proc	Kills Check Point daemons and Security servers while maintaining the active Security Policy running in the kernel. Rules with generic allow/reject/drop rules, based on services continue to work.
-nopolicy	

cprinstall get

Description The `cprinstall get` command is used to obtain details of the products and the Operating System installed on the specified Check Point gateway, and to update the database.

Usage

```
cprinstall get <Object name>
```

Syntax

Argument	Description
Object name	The name of the Check Point gateway object defined in SmartDashboard.

Example

```
[c:\winnt\fw1\5.0\bin]cprinstall get fred

Getting information from fred...

Operating system    Version              SP
-----------------------------------------------------------
solaris              5.7                 fcs

Vendor              Product             Version       SP
----------------------------------------------------------
CheckPoint          VPN-1 Power         NG            fcs
CheckPoint          SVNfoundation       NG            fcs
```

cprinstall install

Description

The `cprinstall install` command is used to install Check Point products on remote Check Point gateways. To install a product package you must specify a number of options. Use the `cppkg print` command and copy the required options.

Usage

```
cprinstall install [-boot] <Object name> <vendor> <product>
<version> [sp]
```

Syntax

Argument	**Description**
-boot	Boot the remote computer after installing the package. Only boot after ALL products have the same version, either NG or NG FP1. Boot will be cancelled in certain scenarios. See the Release Notes for details.
Object name	Object name of the Check Point gateway defined in SmartDashboard.
vendor	Package vendor (e.g. `checkpoint`)

Argument	Description
product	Package name Options are: SVNfoundation, firewall, floodgate.
version	Package version (e.g. NG FP2)
sp	Package service pack (e.g. fcs for NG FP2 initial release, FP1 for NG Feature Pack 1.)

Comments

Before transferring any files, this command runs the `cprinstall verify` command to verify that the Operating System is appropriate and that the product is compatible with previously installed products.

Example

```
# cprinstall install -boot fred checkpoint firewall NG FP1

Installing firewall NG FP1 on fred...
Info : Testing Check Point Gateway
Info : Test completed successfully.
Info : Transferring Package to Check Point Gateway
Info : Extracting package on Check Point Gateway
Info : Installing package on Check Point Gateway
Info : Product was successfully applied.
Info : Rebooting the Check Point Gateway
Info : Checking boot status
Info : Reboot completed successfully.
Info : Checking Check Point Gateway
Info : Operation completed successfully.
```

cprinstall stop

Description

This command is used to stop the operation of other `cprinstall` commands. In particular, this command stops the remote installation of a product - even during transfer of files, file extraction, and pre-installation verification. The operation can be stopped at any time up to the actual installation.

`cprinstall stop` can be run from one command prompt to stop a running operation at another command prompt.

Usage

```
cprinstall stop <Object name>
```

Syntax

Argument	Description
object name	Object name of the Check Point gateway, defined in SmartDashboard.

Example

```
[c:\winnt\fw1\5.0\bin]  cprinstall stop Check Point
Gateway01
Info : Stop request sent
```

cprinstall uninstall

Description The `cprinstall uninstall` command is used to uninstall products on remote Check Point gateways. To uninstall a product package you must specify a number of options. Use the `cppkg print` command and copy the required options.

Usage

```
cprinstall uninstall [-boot] <Object name> <vendor>
<product> <version> [sp]
```

Syntax

Argument	Description
-boot	Boot the remote computer after installing the package. Only boot after ALL products have the same version, either NG or NG FP1. Boot will be cancelled in certain scenarios. See the Release Notes for details.
Object name	Object name of the Check Point gateway defined in SmartDashboard.
vendor	Package vendor (e.g. `checkpoint`)
product	Package name Options are: `SVNfoundation`, `firewall`, `floodgate`.
version	Package version (e.g. NG FP2)
sp	Package service pack (e.g. `fcs` for NG FP2 initial release, `FP1` for NG Feature Pack 1.)

Comments

Before uninstalling any files, this command runs the `cprinstall verify` command to verify that the Operating System is appropriate and that the product is installed.

After uninstalling, retrieve the Check Point gateway data by running `cprinstall get`.

Example

```
# cprinstall uninstall fred checkpoint firewall NG FP1

Uninstalling firewall NG FP1 from fred...
Info : Removing package from Check Point Gateway
Info : Product was successfully applied.
Operation Success.Please get network object data to complete the
operation.
```

cprinstall upgrade

Description

Use the `cprinstall upgrade` command to upgrade all products on a Check Point gateway to the latest version.

All products on the Check Point gateway must be of the same version of NG.

Usage

`cprinstall upgrade [-boot] <object name>`

Syntax

Argument	Description
-boot	Boot the remote Check Point gateway after completing the remote installation.
object name	Object name of the Check Point gateway, defined in SmartDashboard.

Comments

When `cprinstall upgrade` is run, the command first verifies which products are installed on the Check Point gateway, and that there is a matching product package in the product repository with the same OS, and then installs the product package on the remote Check Point gateway.

cprinstall verify

Description

The `cprinstall verify` command is used to verify:

- If a specific product can be installed on the remote Check Point gateway.
- That the Operating System and currently installed products are appropriate for the package.
- That there is enough disk space to install the product.
- That there is a `CPRID` connection.

Usage

```
cprinstall verify <Object name> <vendor> <product>
<version> [sp]
```

Syntax

Argument	Description
Object name	Object name of the Check Point gateway defined in SmartDashboard.
vendor	Package vendor (e.g. `checkpoint`).
product	Package name Options are: `SVNfoundation`, `firewall`, `floodgate`.
version	Package version (e.g. NG) .
sp	Package service pack (e.g. `fcs` for `NG with Application Intelligence` initial release, `FP1`, `FP2` etc.) This parameter is optional. Its default is `fcs`.

Example

The following examples show a successful and a failed verify operation:

Verify succeeds:

```
cprinstall verify harlin checkpoint SVNfoundation NG_FP4

Verifying installation of SVNfoundation NG FP4 on harlin...
Info : Testing Check Point Gateway.
Info : Test completed successfully.
Info : Installation Verified, The product can be installed.
```

Verify fails:

```
cprinstall verify harlin checkpoint SVNfoundation NG
FCS_FP4

Verifying installation of SVNfoundation NG FCS_FP4 on
harlin...
Info : Testing Check Point Gateway
Info : SVN Foundation NG is already installed on
192.168.5.134
Operation Success.Product cannot be installed, did not pass
dependency check.
```

cprinstall verify_upgrade

Description Use the `cprinstall verify_upgrade` command to verify the success of the upgrade of all products on a Check Point gateway to the latest version, before performing the upgrade. This command is automatically performed by the `cprinstall upgrade` command**.**

All products on the Check Point gateway must be of the same version of NG.

Usage `cprinstall verify_upgrade <object name>`

Syntax

Argument	Description
object name	Object name of the Check Point gateway, defined in SmartDashboard.

Comments When the command is run, the command verifies which products are installed on the Check Point gateway, and that there is a matching product package in the product repository with the same OS.

cpstart

Description This command is used to start all Check Point processes and applications running on a machine.

Usage `cpstart`

Comments This command cannot be used to start `cprid`. `cprid` is invoked when the machine is booted and it runs independently.

cpstat

Description `cpstat` displays the status of Check Point applications, either on the local machine or on another machine, in various formats.

Usage `cpstat [-h host][-p port][-f flavour][-d] application_flag`

Syntax

Argument	Description
`-h host`	A resolvable hostname, or a dot-notation address (for example,192.168.33.23). The default is localhost.
`-p port`	Port number of the AMON server. The default is the standard AMON port (18192)
`-f flavour`	The flavor of the output (as appears in the configuration file). The default is to use the first flavor found in configuration file.
`-d`	debug flag
`application_flag`	One of: • `fwm` — FireWall-1 • `vpn` — VPN-1 • `fg` — FloodGate-1 (QoS) • `ha` — ClusterXL (High Availability) • `os` — SVN Foundation and OS Status • `mg` — for SmartCenter

Where the flavors are:

- `fwm — "fw", with flavours: "default", "all", "policy", "performance", "hmem", "kmem", "inspect", "cookies", "chains", "fragments", "totals", "ufp_caching", "http_stat", "ftp_stat", "telnet_stat", "rlogin_stat", "ufp_stat", "smtp_stat"`
- `vpn — “product”, “general”, “IKE”, “ipsec”, “fwz”, “accelerator”, “all”`
- `fg — “all”`
- `mg — “default”`

- `os` — “default”, “routing”
- `ha` — “default”, “all”

Example

```
> cpstat fw

Policy name:  Standard
Install time: Wed Nov  1 15:25:03 2000

Interface table
--------------------------------------------------------------
---
|Name|Dir|Total *|Accept**|Deny|Log|
--------------------------------------------------------------
---
|hme0|in |739041*|738990**|51 *|7**|
--------------------------------------------------------------
---
|hme0|out|463525*|463525**| 0 *|0**|
--------------------------------------------------------------
---
*********|1202566|1202515*|51**|7**|
```

cpstop

Description This command is used to terminate all Check Point processes and applications, running on a machine.

Usage

```
cpstop

cpstop -fwflag [-proc | -default]
```

Syntax

Argument	Description
-fwflag -proc	Kills Check Point daemons and Security servers while maintaining the active Security Policy running in the kernel. Rules with generic allow/reject/drop rules, based on services continue to work.
-fwflag -default	Kills Check Point daemons and Security servers. The active Security Policy running in the kernel is replaced with the default filter..

Comments This command cannot be used to terminate `cprid`. `cprid` is invoked when the machine is booted and it runs independently.

cpwd_admin

Description `cpwd` (also known as WatchDog) is a process that invokes and monitors critical processes such as Check Point daemons on the local machine, and attempts to restart them if they fail. Among the processes monitored by Watchdog are `cpd`, `fwd`, `fwm`. `cpwd` is part of the SVN Foundation.

`fwd` does not work in a Management Only machine. To work with `fwd` in a Management Only machine add `-n` (for example, `fwd -n`).

`cpwd` writes monitoring information to the `$CPDIR/log/cpwd.elg log` file. In addition, monitoring information is written to the console on UNIX platforms, and to the Windows Event Viewer.

The `cpwd_admin` utility is used to show the status of processes, and to configure `cpwd`.

Usage

```
cpwd_admin
```

In This Section

cpwd_admin start

Description Start a new process by `cpwd`.

Usage

```
cpwd_admin start -name <process name> -path <"full path">
-command <"executable name">
```

Syntax

Argument	Description
-name <process name>	A name for the process to be watched by WatchDog.
-path <"full path">	The full path to the executable including the executable name
-command <"executable name & arguments">	The name of the executable file.

Example To start and monitor the `fwm` process.

```
cpwd_admin start -name FWM -path "$FWDIR/bin/fwm" -command
"fwm"
```

cpwd_admin stop

Description Stop a process which is being monitored by `cpwd`.

Usage

```
cpwd_admin stop -name <process name> [-path <"full path">
-command <"executable name">]
```

Syntax

Argument	Description
-name <process name>	A name for the process to be watched by WatchDog.
-path <"full path">	Optional: the full path to the executable (including the executable name) that is used to stop the process.
-command <"executable name & arguments">	Optional: the name of the executable file mentioned in `-path`

Comments If `-path` and `-command` are not stipulated, cpwd will abrupbtly terminate the process.

Example stop the `FWM` process using `fw kill`.

```
cpwd_admin stop -name FWM -path "$FWDIR/bin/fw" -command "fw
kill fwm"
```

cpwd_admin list

Description This command is used to print a status of the selected processes being monitored by `cpwd`.

Usage

```
cpwd_admin list
```

Output The status report output includes the following information:

- `APP` — Application. The name of the process.
- `PID` — Process Identification Number.
- `STAT` — Whether the process Exists (E) or has been Terminated (T).
- `#START` —How many times the process has been started since `cpwd` took control of the process.
- `START TIME` — The last time the process was run.
- `COMMAND` — The command that `cpwd` used to start the process.

For example:

```
#cpwd_admin list
APP  PID   STAT #START    START_TIME                  COMMAND
CPD  463     E      1      [20:56:10] 21/5/2001       cpd
FWD  440     E      1      [20:56:24] 21/5/2001       fwd
FWM  467     E      1      [20:56:25] 21/5/2001       fwm
```

cpwd_admin exist

Description This command is used to check whether cpwd is alive.

Usage

```
cpwd_admin exist
```

cpwd_admin kill

Description This command is used to kill cpwd.

Usage

```
cpwd_admin kill
```

cpwd_admin config

Description This command is used to set cpwd configuration parameters. When parameters are changed, these changes will not take affect until cpwd has been stopped and restarted.

Usage

```
cpwd_admin config -p

cpwd_admin config -a <value=data value=data...>

cpwd_admin config -d <value value...>

cpwd_admin config -r
```

Syntax

Argument	Description
config -p	Shows the cpwd parameters added using the config -a option.
config -a	Add one or more monitoring parameters to the cpwd configuration.
config -d	Delete one or more parameters from the cpwd configuration
config -r	Restore the default cpwd parameters.

Where the values are as follows:

Argument	**Description**
timeout (any value in seconds)	If rerun_mode=1, how much time passes from process failure to rerun. The default is 60 seconds.
no_limit (any value in seconds)	Maximum number of times that cpwd will try to restart a process. The default is 5.
zero_timeout (any value in seconds)	After failing no_limit times to restart a process, cpwd will wait zero_timeout seconds before retrying. The default is 7200 seconds. Should be greater than timeout.
sleep_mode	• 1 - wait timeout • 0 - ignore timeout. Rerun the process immediately
dbg_mode	• 1 - Accept pop-up error messages (with exit-code#0) displayed when a process terminates abruptly (Windows NT only). • 0 -Do not receive pop-up error messages. This is useful if pop-up error messages freeze the machine. This is the default (Windows NT only).

Argument	Description
rerun_mode	- 1 - Rerun a failed process. This is the default. - 0 - Do not rerun a failed process. Perform only monitoring.
stop_timeout	The time in seconds that the cpwd will wait for a stop command to be completed. Default is 60 seconds.
reset_startups	Indicates the time in seconds that the cpwd waits after the process begins before it resets the startup_counter. Default value is 1 hour, meaning that an hour after the process begins its startup counter is reset to 0.

Example The following example shows two configuration parameters being changed:
timeout to 120 seconds, and no_limit to 10.

```
# C:\>cpwd_admin config -p
WD doesn't have configuration parameters

C:\>cpwd_admin config -a timeout=120 no_limit=12

C:\>cpwd_admin config -p
WD Configuration parameters are:
timeout : 120
no_limit : 12cpwd_admin config -a timeout=120 no_limit=10
```

config -a and cpwd_admin config -d have no effect if cpwd is running. They will affect cpwd the next time it is run.

dbedit

Description This command is used by administrators to edit the objects file on the SmartCenter server. From version NG, there is an objects file on the gateway and a new file, objects_5_0.C on the SmartCenter server. A new objects.C file is created on the gateway (based on the objects_5_0.C on the SmartCenter server) whenever a Policy is installed. Editing the `objects.C` file on the gateway is no longer required or desirable, since it will be overwritten the next time a Policy is installed.

Usage

```
dbedit [-s server] [- u user | -c certificate] [-p password]
[-f filename] [-r db-open-reason] [-help]
```

Syntax

Argument	Description
`-s server`	The SmartCenter server on which the `objects_5_0.C` file to be edited is located. If this is not specified in the command line, then the user will be prompted for it. If the server is not localhost, the user will be required to authenticate.

`-u user |`
`-c certificate`
	The user's name (the name used for the SmartConsole) or the full path to the certificate file.

`-p password`	The user's password (the password used for the SmartConsole).
`-f filename`	The name of the file containing the commands. If *filename* is not given, then the user will be prompted for commands.
`-r db-open-reason`	A non-mandatory flag used to open the database with a string that states the reason. This reason will be attached to audit logs on database operations.
`-help`	Print usage and short explanation.

dbedit commands:

Argument	Description
create [object_type] [object_name]	Create an object with its default values. The create command may use an extended (or “owned”) object. Changes are committed to the database only by an `update` or `quit` command.
modify [table_name] [object_name] [field_name] [value]	Modify fields of an object which is: • stored in the database (the command will lock the object in such case). • newly created by `dbedit` Extended Formats for owned objects can be used: For example, `[field_name] = Field_A:Field_B`
update [table_name] [object_name]	Update the database with the object. This command will check the object validity and will issue an error message if appropriate.
delete [table_name] [object_name]	Delete an object from the database and from the client implicit database.
addelement [table_name] [object_name] [field_name] [value]	Add an element (of type string) to a multiple field.

Argument	Description
rmelement [table_name] [object_name] [field_name] [value]	Remove an element (of type string) from a multiple field.
rename [table_name][object_name] [new_object_name]	Assign a new name for a given object. The operation also performs an update. Example: Rename network object London to Chicago. `rename network_objects london chicago`
quit	Quit `dbedit` and update the database with modified objects not yet committed.

Example Replace the owned object with a new null object, where NULL is a reserved word specifying a null object:

```
modify network_objects my_obj firewall_setting NULL
```

Example **Extended Format**

`firewall_properties` owns the object `floodgate_preferences`.

`floodgate_preferences` has a Boolean attribute `turn_on_logging`, which will be set to `true`.

```
modify properties firewall_properties
floodgate_preferences:turn_on_logging true
```

`comments` is a field of the owned object contained in the ordered container. The 0 value indicates the first element in the container (zero based index).

```
modify network_objects my_networkObj interfaces:0:comments
my_comment
```

Replace the owned object with a new one with its default values.

```
modify network_objects my_net_obj interfaces:0:security
interface_security
```

dbver

Description The `dbver` utility is used to *export* and *import* different revisions of the database. The properties of the revisions (last time created, administrator responsible for, etc) can be reviewed. The utility can be found in `$FWDIR/bin`.

Usage

```
export <version_numbers> <delete | keep>

import <exported_version_in_server>

create <version_name> <version_comment>

delete <version_numbers>

print <version_file_path>

print_all
```

dbver create

Description Create a revision from the current state of `$fwdir/conf`, including current objects, rule bases, etc.

Usage

```
create <version_name> <version_comment>
```

Syntax

Argument	Description
`version_name`	the name of the revision
`version_comment`	append a comment to the revision

dbver export

Description Archive the revision as an archive file in the revisions repository: `$fwdir/conf/db_versions/export`.

Usage `export <version_numbers> <delete | keep>`

Syntax

Argument	Description
update [table_name] [object_name]	Update the database with the object. This command will check the object validity and will issue an error message if appropriate.
delete [table_name] [object_name]	Delete an object from the database and from the client implicit database.
addelement [table_name] [object_name] [field_name] [value]	Add an element (of type string) to a multiple field.
`version_numbers`	the file name of the exported version
`delete	keep`	- `delete` removes the revision from the revisions repository. - `keep` maintains the revision in the revisions repository.

dbver import

Description Add an exported revision to the repository a version from `$fwdir/conf/db_versions/export`. Give filename of revision as input.

Usage `import <exported_version_in_server>`

Syntax

Argument	Description
exported_version_in_server	The file name of the exported version.

dbver print

Description Print the properties of the revision.

Usage `print <version_file_path>`

Syntax

Argument	Description
version_file_path	The full name and path on the local machine of the revision.

Output

```
dbver> print c:\rwright_2002-04-01_160810.tar.gz

Version Id: 1

Version Date: Mon Apr  1 16:08:10 2002

Version Name: save

Created by Administrator: jbrown

Major Version: NG

Minor Version: FP2
```

dbver print_all

Description Print the properties of all revisions to be found on the server side: `$fwdir/conf/db_versions`

Usage `print_all`

dynamic_objects

Description `dynamic_objects` specifies an IP address to which the dynamic object will be resolved on this machine.

This command cannot be executed when the VPN-1 gateway is running.

Usage

```
dynamic_objects -o <object_name> [-r [fromIP toIP] ...] [-s]
[-a] [-d] [-l] [-n <object_name> ] [-c]
```

Syntax

Argument	Description
`-o <object_name>`	The name of the object, as defined in SmartDashboard.
`-r [fromIP toIP] ...`	address ranges — one or more "from IP address to IP address" pairs
`-a [fromIP toIP] ...`	add ranges to object
`-d [fromIP toIP] ...`	delete range from object
`-l`	list dynamic objects
`-n object_name`	create new object (if VPN-1 Power gateway is not running)
`-c`	compare the objects in the dynamic objects file and in `objects.C`.
`-do object_name`	delete object

Example

Create a new dynamic object named "bigserver" and add to it the IP address range 190.160.1.1-190.160.1.40: `dynamic_objects -n bigserver -r 190.160.1.1 190.160.1.40 -a`

fw

Description

The `fw` commands are used for working with various aspects of FireWall-1. All `fw` commands are executed on the FireWall-1 gateway.

Typing `fw` at the command prompt sends a list of available fw commands to the standard output.

Usage

```
fw
```

In This Section

fw ctl

Description The fw ctl command controls the FireWall-1 kernel module.

Usage

```
fw ctl <install|uninstall>

fw ctl debug [-x] [-m <module>] [+|-]
<options | all | 0>

fw ctl debug -buf [buffer size]

fw ctl kdebug

fw ctl pstat [-h][-k][-s][-n][-l]

fw ctl iflist

fw ctl arp [-n]

fw ctl block <on|off>

fw ctl chain

fw ctl conn
```

Argument	Description

`<Install|`
`Uninstall>`
	• `Uninstall` — tells the operating system to stop passing packets to FireWall-1, and unloads the Security Policy. The networks behind it become unprotected.
	• `Install` — tells the operating system to start passing packets to FireWall-1. The command `fw ctl install` runs automatically when `cpstart` is performed.
	Note - If you run `fw ctl uninstall` followed by `fw ctl install`, the Security Policy is not restored.

`debug`	Generate debug messages to a buffer. `fw ctl debug [-m module] [+ | -] <options|all|0>` Sets or resets debug flags for the requested gateway (default is fw). • If `+` is used, the specified flags are set, and the rest remain as they were. • If - is used, the specified flags are reset, and the rest remain as they were. • If neither `+` nor `-` are used, the specified flags are set and the rest are reset. `fw ctl debug 0` Returns all flags in all gateways to their default values, releases the debug buffer (if there was one).
debug -buf [buffer size]	Allocates a buffer of size kilobytes (default 128) and starts collecting messages there.
debug -h	Print a list of modules and flags.
debug -x	Do not use.
kdebug	Reads the debug buffer and obtains the debug messages. If there is no debug buffer, the command will fail. If `-f` is used, the command will read the buffer every second and print the messages, until `Ctrl-C` is pressed. Otherwise, it will read the current buffer contents and end. If -T is added the time will be printed in microseconds.

Argument	Description
pstat [-h] [-k] [-s] [-n] [-l]	Displays Firewall-1 internal statistics: -h — Generates additional hmem details. -k — Generates additional kmem details. -s — Generates additional smem details. -n — Generates NDIS information (Windows only). -l — Generates general FireWall-1 statistics.
iflist	Displays the IP interfaces known to the kernel, by name and internal number
arp [-n]	Displays ARP proxy table. -n — Do not perform name resolution.
block <on|off>	on — Blocks all traffic. off — Restores traffic and the Security Policy.
chain	Prints the names of internal FireWall-1 gateways that deal with packets. Use to ensure that a gateway is loaded. The names of these gateways can be used in the fw monitor -p command.
conn	Prints the names of the connection modules.

fw expdate

Description This command is used to modify the expiration date of all users and administrators.

Usage

```
fw expdate dd-mmm-1976
```

Comments The date can be modified using a filter.

Example

```
fw expdate 02-mar-2003 -f 01-mar-2003
```

fw fetch

Description This command fetches the Inspection Code from the specified host and installs it to the kernel.

Usage

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

Syntax

Argument	Description
-n	Fetch the Security Policy from the SmartCenter server to the local `state` directory, and install the Policy only if the fetched Policy is different from the Policy already installed.
-f <filename>	Fetch the Security Policy from the SmartCenter server listed in <filename>. If filename is not specified, the list in `conf/masters` is used.
-c	Cluster mode, get policy from one of the cluster members, from the Check Point High Availability (CPHA) kernel list
-i	Ignore SIC information (for example, SIC name) in the database and use the information in `conf/masters`. This option is used when a Security Policy is fetched for the first time by a DAIP gateway from a SmartCenter server with a changed SIC name.
master1	Execute command on the designated master. The name of the SmartCenter server from which to fetch the Policy. You may specify a list of one or more SmartCenter servers, such as `master1  master2` which will be searched in the order listed. If no `targets` is not specified, or if `targets` is inaccessible, the Policy is fetched from `localhost`.

fw fetchlogs

Description `fw fetchlogs` fetches Log Files from a remote machine. You can use the `fw fetchlogs` command to transfer Log Files to the machine on which the `fw fetchlogs` command is executed. The Log Files are read from and written to the directory `$FWDIR/log`.

Usage

```
fw fetchlogs [[-f file name] ... ] module
```

Syntax

Argument	Description
-f filename	The Log Files to be transferred. The file name can include wildcards. In Solaris, any file containing wildcards should be enclosed in quotes. The default parameter is `*.log`. Related pointer files will automatically be fetched.
module	The name of the remote machine from where you transfer the Log Files.

Comments The files transferred by the fw fetchlogs command are MOVED from the source machine to the target machine. This means that they are deleted from the source machine once they have been successfully copied.

Fetching Current Log Data

The active Log File (`fw.log`) cannot be fetched. If you want to fetch the most recent log data, proceed as follows:

- Run `\`to close the currently active Log File and open a new one.
- Run `fw lslogs` to see the newly-generated file name.
- Run `fw fetchlogs -f` *filename* to transfer the file to the machine on which the `fw fetchlogs` command is executed. The file is now available for viewing in the SmartView Tracker.

After a file has been fetched, it is renamed. The gateway name and the original Log File name are concatenated to create a new file name. The new file name consists of the gateway name and the original file name separated by two (underscore) _ _ characters.

Example The following command:

```
fw fetchlogs -f 2001-12-31_123414.log module3
```

fetches the Log File `2001-12-31_123414.log` from `Module3`.

After the file has been fetched, the Log File is renamed:

```
module3_ _2001-12-31_123414.log
```

Further Info. See the *SmartCenter* Administration Guide.

fw isp_link

Description This comnand takes down (or up) a redundant ISP link.

Usage

```
fw isp_link [target] link-name {up|down}
```

Syntax

Argument	Description
targe	The name of the enforcement module.
link-name	The name of the ISP link as defined in the ISP-redundancy tab.

Comments This command can be executed locally on the enforcement module or remotely from the management module. In the latter case, the target argument must be supplied. For this command to work, the enforcement module should be using the ISP redundancy feature.

fw kill

Description This command prompts the kernel to shut down all FireWall-1 daemon processes. The command is located in the `$FWDIR/bin directory` on the SmartCenter server or gateway machine.

The FireWall-1 daemons and Security servers write their `pids` to files in the `$FWDIR/tmp directory` upon startup. These files are named `$FWDIR/tmp/daemon_name.pid`. For example, the file containing the `pid` of the FireWall-1 `snmp` daemon is `$FWDIR/tmp/snmpd.pid`.

Usage

```
fw kill [-t sig_no] proc-name
```

Syntax

Argument	Description
-t sig_no	This Unix only command specifies that if the file `$FWDIR/tmp/proc-name.pid` exists, send `signal sig_no` to the `pid` given in the file. If no signal is specified, signal 15 (`sigterm` or the terminate command) is sent.
proc-name	Prompt the kernel to shut down specified FireWall-1 daemon processes.

Comments In Windows, only the default syntax is supported: `fw kill proc_name`. If the `-t` option is used it is ignored.

fw lea_notify

Description This command should be run from the SmartCenter server. It sends a `LEA_COL_LOGS` event to all connected lea clients, see the *LEA Specification* documentation. It should be used after new log files have been imported (manually or automatically) to the `$FWDIR/log` directory in order to avoid the scheduled update which takes 30 minutes.

Usage

```
fw lea_notify
```

fw lichosts

Description This command prints a list of hosts protected by VPN-1 products. The list of hosts is in the file `$fwdir/database/fwd.h`

Usage

```
fw lichosts [-x] [-l]
```

Syntax

Argument	Description
-x	Use hexadecimal format.
-l	Use long format.

fw log

Description `fw log` displays the content of Log files.

Usage

```
fw log [-f [-t]] [-n] [-l] [-o] [-c action] [-h host] [-s
starttime] [-e endtime] [-b starttime endtime] [-u
unification_scheme_file]
[-m unification_mode(initial|semi|raw)] [-a] [-k
(alert_name|all)] [-g] [logfile]
```

Syntax

Argument	Description
`-f [-t]`	After reaching the end of the currently displayed file, do not exit (the default behavior), but continue to monitor the Log file indefinitely and display it while it is being written. The `-t` parameter indicates that the display is to begin at the end of the file, in other words, the display will initially be empty and only new records added later will be displayed. `-t` must come with a `-f` flag. These flags are relevant only for active files.
-n	Do not perform DNS resolution of the IP addresses in the Log file (the default behavior). This option significantly speeds up the processing.
-l	Display both the date and the time for each log record (the default is to show the date only once above the relevant records, and then specify the time per log record).
-o	Show detailed log chains (all the log segments a log record consists of)

Argument	**Description**
`-c action`	Display only events whose action is `action`, that is, `accept`, `drop`, `reject`, `authorize`, `deauthorize`, `encrypt` and `decrypt`. Control actions are always displayed.
`-h host`	Display only log whose origin is the specified IP address or name.
`-s starttime`	Display only events that were logged after the specified time (see format below). `starttime` may be a date, a time, or both. If date is omitted, then today's date is assumed.
`-e endtime`	Display only events that were logged before the specified time (see format below). `endtime` may be a date, a time, or both.
`-b starttime endtime`	Display only events that were logged between the specified start and end times (see format below), each of which may be a date, a time, or both. If date is omitted, then today's date is assumed. The start and end times are expected after the flag.
-u `unification_scheme_file`	Unification scheme file name.

Argument	**Description**
`-m unification_mode`	This flag specifies the unification mode. • `initial` - the default mode, specifying complete unification of log records; that is, output one unified record for each id. This is the default. When used together with `-f`, no updates will be displayed, but only entries relating to the start of new connections. To display updates, use the `semi` parameter. • `semi` - step-by-step unification, that is, for each log record, output a record that unifies this record with all previously-encountered records with the same id. • `raw` - output all records, with no unification.
-a	Output account log records only.
-k alert_name	Display only events that match a specific alert type. The default is `all`, for any alert type.
-g	Do not use a delimited style. The default is: • `:` after field name • `;` after field value
`logfile`	Use `logfile` instead of the default Log file. The default Log File is `$FWDIR/log/fw.log`.

Where the full date and time format is: `MMM DD, YYYY HH:MM:SS`. For example: `May 26, 1999 14:20:00`

It is possible to specify date only in the format `MMM DD, YYYY`, or time only, in the format: `HH:MM:SS`, where time only is specified, the current date is assumed.

Example

```
fw log
fw log | more
fw log -c reject
fw log -s "May 26, 1999"
fw log -f -s 16:00:00
```

Output

```
[<date>] <time> <action> <origin> <interface dir and name>
[alert] [field name: field value;] ...
```

Each output line consists of a single log record, whose fields appear in the format shown above.

Example

```
14:56:39 reject jam.checkpoint.com >daemon alert src:
veredr.checkpoint.com; dst: jam.checkpoint.com; user: a;
rule: 0; reason: Client Encryption: Access denied - wrong
user name or password  ; scheme: IKE; reject_category:
Authentication error; product: VPN-1 Power
14:57:49 authcrypt jam.checkpoint.com >daemon src:
veredr.checkpoint.com; user: a; rule: 0; reason: Client
Encryption: Authenticated by Internal Password; scheme: IKE;
methods: AES-256,IKE,SHA1; product: VPN-1 Power;

14:57:49 keyinst jam.checkpoint.com >daemon src:
veredr.checkpoint.com; peer gateway: veredr.checkpoint.com;
scheme: IKE; IKE: Main Mode completion.; CookieI:
32f09ca38aeaf4a3; CookieR: 73b91d59b378958c; msgid: 47ad4a8d;
methods: AES-256 + SHA1, Internal Password; user: a;  product:
VPN-1 Power;
```

fw logswitch

Description

`fw logswitch` creates a new active Log File. The current active Log File is closed and renamed by default `$FWDIR/log/`*current_time_stamp*`.log` unless you define an alternative name that is unique. The format of the default name *current_time_stamp*`.log` is `YYYY-MM-DD_HHMMSS.log`. For example: `2003-03-26_041200.log`

Warning:

- The Logswitch operation fails if a log file is given an pre-existing file name.

- The rename operation fails on Windows if the active log that is being renamed, is open at the same time that the rename operation is taking place; however; the Logswitch will succeed and the file will be given the default name `$FWDIR/log/current_time_stamp.log`.

The new Log File that is created is given the default name `$FWDIR/log/fw.log`. Old Log Files are located in the same directory.

A SmartCenter server can use `fw logswitch` to switch a Log File on a remote machine and transfer the Log File to the SmartCenter server. This same operation can be performed for a remote machine of version NG FP2 and higher, using “fw lslogs” on *page 100* and “fw fetchlogs” on *page 82*.

When a log file is sent to the SmartCenter server, the data is compressed.

Usage

```
fw logswitch [-audit] [filename]

fw logswitch -h hostname [+|-][filename]
```

Syntax

Argument	**Description**
-audit	Does logswitch for the SmartCenter audit file. This is relevant for local activation.
filename	The name of the file to which the log is saved. If no name is specified, a default name is provided.

Argument	Description
`-h hostname`	The resolvable name or IP address of the remote machine (running either a VPN-1 Power gateway or a SmartCenter server) on which the Log File is located. The SmartCenter server (on which the `fw logswitch` command is executed) must be defined as one of `host`'s SmartCenter servers. In addition, you must initialize SIC between the SmartCenter server and the `host`.
+	Switch a remote log and copy it to the local machine
-	Switch a remote log and move it to the local machine thereby deleting the log from the remote machine.

Comments Files are created in the `$FWDIR/log` directory on both `host` and the SmartCenter server when the `+` or `-` parameters are specified. Note that if `-` is specified, the Log File on the host is deleted rather than renamed.

`hostname` specified:

- `filename` specified - On `hostname`, the old Log File is renamed to `old_log`. On the SmartCenter server, the copied file will have the same name, prefixed by `hostname`'s name. For example, the command `fw logswitch -h venus +xyz` creates a file named `venus_xyz.log` on the SmartCenter server.
- `filename` not specified - On `hostname`, the new name is the current date, for example: `2003-03-26_041200.log`. On the SmartCenter server, the copied file will have the same name, but prefixed by `hostname_`. For example, `target_2003-03-26_041200.log`.

`hostname` not specified:

- `filename` specified - On the SmartCenter server, the old Log File is renamed to `old_log`.
- `filename` not specified - On the SmartCenter server, the old Log File is renamed to the current date.

If either the SmartCenter server or `hostname` is an NT machine, the files will be created using the NT naming convention.

Compression

When log files are transmitted from one machine to another, they are compressed using the zlib package, a standard package used in the Unix `gzip` command (see RFC 1950 to RFC 1952 for details). The algorithm is a variation of LZ77 method.

The compression ratio varies with the content of the log records and is difficult to predict. Binary data are not compressed, but string data such as user names and URLs are compressed.

fw mergefiles

Description This command merges several Log Files into a single Log File. The merged file can be sorted according to the creation time of the Log entries, and the times can be "fixed" according to the time zones of the origin Log servers.

Logs entries with the same Unique-ID are unified. If a Log switch was performed before all the segments of a specific log were received, this command will merge the records with the same Unique-ID from two different files, into one fully detailed record.

Usage

```
fw mergefiles [-s] [-t time_conversion_file]
log_file_name_1 [... log_file_name_n] output_file
```

Syntax

Argument	Description
`-s`	Sort merged file by log records time field.
`-t time_conversion_file`	Fix different GMT zone log records time in the event that the log files originated from Log Servers in different time zone. The time_conversion_file format is as follows: ip-address signed_date_time_in_seconds ip-address signed_date_time_in_seconds . .
log_file_name_n	Full pathnames of the Log File(s).
output_file	Full pathname of the output Log File.

Comments It is not recommended to merge the current active `fw.log file` with other Log Files. Instead, run the `fw logswitch` command and then run `fw mergefiles`.

fw monitor

Description Inspecting network traffic is an essential part of troubleshooting network deployments. `fw monitor` is a powerful built-in tool to simplify the task of capturing network packets at multiple capture points within the FireWall-1 chain. These packets can be inspected using industry-standard tools later on.

In many deployment and support scenarios capturing network packets is an essential functionality. tcpdump or snoop are tools normally used for this task. `fw monitor` provides an even better functionality but omits many requirements and risks of these tools.

- *No Security Flaws* — tcpdump and snoop are normally used with network interface cards in promiscuous mode. Unfortunately the promiscuous mode allows remote attacks against these tools. fw monitor does not use the promiscuous mode to capture packets. In addition most FireWalls' operating systems are hardened. In most cases this hardening includes the removal of tools like tcpdump or snoop because of their security risk.

- *Available on all FireWall-1 installations* — `fw monitor` is a built-in firewall tool which needs no separate installation in case capturing packets is needed. It is a functionality provided with the installation of the FireWall package.
- *Multiple capture positions within the FireWall-1 kernel module chain* — `fw monitor` allows you to capture packets at multiple capture positions within the FireWall-1 kernel module chain; both for inbound and outbound packets. This enables you to trace a packet through the different functionalities of the firewall.
- *Same tool and syntax on all platforms* — Another important fact is the availability of `fw monitor` on different platforms. Tools like snoop or tcpdump are often platform dependent or have specific "enhancements" on certain platforms. `fw monitor` and all its related functionality and syntax is absolutely identical across all platforms. There is no need to learn any new "tricks" on an unknown platform.

Normally the Check Point kernel modules are used to perform several functions on packets (like filtering, encrypting and decrypting, QoS ...). `fw monitor` adds its own modules to capture packets. Therefore fw monitor can capture all packets which are seen and/or forwarded by the FireWall.

Only one instance of `fw monitor` can be run at a time.

Use ^C (that is Control + C) to stop fw monitor from capturing packets.

Usage

```
fw monitor [-u|s] [-i] [-d] [-D] <{-e expr}+|-f
<filter-file|->> [-l len] [-m mask] [-x offset[,len]] [-o
<file>] <[-pi pos] [-pI pos] [-po pos] [-pO pos] | -p all
> [-a] [-ci count] [-co count] [-vs vsid or vsname] [-h] -T
```

Syntax

Argument	**Description**
-uls	**Printing the UUID or the SUUID:** The option –u or –s is used to print UUIDs or SUUIDs for every packet. Please note that it is only possible to print the UUID or the SUUID – not both.
-i	**Flushing the standard output:** Use to make sure that captured data for each packet is at once written to standard output. This is especially useful if you want to kill a running fw monitor process and want to be sure that all data is written to a file.
[-d] [-D]	**Debugging fw monitor:** The -d option is used to start fw monitor in debug mode. This will give you an insight into `fw monitor`'s inner workings. This option is only rarely used outside Check Point. It is also possible to use `-D` to create an even more verbose output.
<{-e expr}+l-f <filter-filel->>	**Filtering fw monitor packets:** fw monitor has the ability to capture only packets in which you are interested. fw monitor filters use a subset of INSPECT to specify the packets to be captured. Set the filter expression • on the command line using the `-e` switch • by reading it from a file using the `-f` switch. • by reading it from standard input using the `-f -` switch.

Argument	Description
-l len	**Limiting the packet length:** fw monitor allow you to limit the packet data which will be read from the kernel with -l. This is especially useful if you have to debug high sensitive communication. It allows you to capture only the headers of a packet (e.g. IP and TCP header) while omitting the actual payload. Therefore you can debug the communication without seeing the actual data transmitted. Another possibility is to keep the amount of data low. If you don't need the actual payload for debugging you can decrease the file site by omitting the payload. It’s also very useful to reduce packet loss on high-loaded machines. fw monitor uses a buffer to transfer the packets from kernel to user space. If you reduce the size of a single packet this buffer won’t fill up so fast.
-m mask	**Setting capture masks:** By default fw monitor captures packets before and after the virtual machine in both directions. These positions can be changed. This option allows you to specify in which of the four positions you are interested.
-x offset[,len]	**Printing packet/payload data:** In addition to the IP and Transport header fw monitor can also print the packets’ raw data using the –x option. Optionally it is also possible to send all data that is written only to the screen the data written.

Argument	Description
-o <file>	**Write output to file:** Save the raw packet data to a file in a standard (RFC 1761) format. The file can be examined using by tools like snoop, tcpdump or Ethereal. **Note -** The snoop file format is normally used to store Layer 2 frames. For "normal" capture files this means that the frame includes data like a source and a destination MAC address. fw monitor operates in the FireWall-1 kernel and therefore has no access to Layer 2 information like MAC addresses. Instead of writing random MAC addresses, fw monitor includes information like interface name, direction and chain position as "MAC addresses".
-T	Print time stamp in microseconds. -T is needed only when -o is not used. When -o is used the exact time is written to the snoop file by default as of Corsica.
<[-pi pos] [-pI pos] [-po pos] [-pO pos] | -p all >	**Insert fw monitor chain module at a specific position:** In addition to capture masks (which give the ability to look at packets in a specific position) fw monitor has the ability to define where exactly in the FireWall-1 chain the packets should be captured. This can be defined using these options.

Argument	Description
-a	**Use absolute chain positions:** If you use fw monitor to output the capture into a file (option –o), one of the fields written down to the capture file is the chain position of the fw monitor chain module. Together with a simultaneous execution of `fw ctl` chain you can determine where the packet was captured. Especially when using –p all you will find the same packet captured multiples times at different chain positions. The option –a changes the chain id from an relative value (which only makes sense with the matching `fw ctl` chain output) to an absolute value. These absolute values are known to CPEthereal and can be displayed by it.

Argument	Description
[-ci count] [-co count]	**Capture a specific number of packets:** fw monitor enables you to limit the number of packets being captured. This is especially useful in situations where the firewall is filtering high amounts of traffic. In such situations fw monitor may bind so many resources (for writing to the console or to a file) that recognizing the break sequence (Control-C) might take very long.
[-vs vsid or vsname]	**Capture on a specific Virtual Router or Virtual Machine:** FireWall-1 VSX enables you to run multiple Virtual Routers and FireWalls on one physical machine. Using the option –vs you can specify on which virtual component the packets should be captured. This option is only available on a FireWall-1 VSX module. Please refer to fw monitor on FireWall-1 VSX for more information.
-h	Displays the usage.

Example

The easiest way to use `fw monitor` is to invoke it without any parameter. This will output every packet from every interface that passes (or at least reaches) the enforcement module. Please note that the same packet is appearing several times (two times in the example below). This is caused by `fw monitor` capturing the packets at different capture points.

Output

```
cpmodule]# fw monitor
 monitor: getting filter (from command line)
 monitor: compiling
monitorfilter:
Compiled OK.
 monitor: loading
 monitor: monitoring (control-C to stop)
eth0:i[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285
id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:I[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285
id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:o[197]: 172.16.1.2 -> 172.16.1.133 (TCP) len=197
id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:O[197]: 172.16.1.2 -> 172.16.1.133 (TCP) len=197
id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:o[1500]: 172.16.1.2 -> 172.16.1.133 (TCP) len=1500
id=44600
TCP
^C
: 18190 -> 1050 ....A. seq=941b0659 ack=bf8bca83
monitor: caught sig 2
 monitor: unloading
```

The first line of the `fw monitor` output is

```
eth0:i[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285
id=1075
```

This packet was captured on the first network interface (eth0) in inbound direction before the virtual machine (lowercase i). The packet length is 285 bytes (in square parenthesis; repeated at the end of the line. Note that these two values may be different. The packets ID is 1075. The packet was sent from 172.16.1.133 to 172.16.1.2 and carries a TCP header/payload.

The second line of the fw monitor output is

```
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
```

The second line tells us that this is an TCP payload inside the IP packet which was sent from port 1050 to port 18190. The following element displays the TCP flags set (in this case PUSH and ACK). The last two elements are showing the sequence number (seq=bf8bc98e) of the TCP packet and the acknowledged sequence number (ack=941b05bc). You will see similar information for UDP packets.

You will only see a second line if the transport protocol used is known to fw monitor. Known protocols are for example TCP, UDP and ICMP. If the transport protocol is unknown or can not be analyzed because it is encrypted (e.g. ESP or encapsulated (e.g. GRE) the second line is missing.

Further Info. See the document *How to use fw monitor* at http://www.checkpoint.com/techsupport/downloadsng/utilities.html.

fw lslogs

Description This command displays a list of Log Files residing on a remote or local machine. You must initialize SIC between the SmartCenter server and the remote machine.

Usage

```
fw lslogs [[-f file name] ...] [-e] [-s name | size | stime
| etime] [-r] [machine]
```

Syntax

Argument	Description
`-f filename`	The list of files to be displayed. The file name can include wildcards. In Unix, any file containing wildcards should be enclosed in quotes. The default parameter is `*.log`.
-e	Display an extended file list. It includes the following data: • `Size` - The size of the file and its related pointer files together. • `Creation Time` - The time the Log File was created. • `Closing Time` - The time the Log File was closed. • `Log File Name` - The file name.
-s	Specify the sort order of the Log Files using one of the following sort options: • `name` - The file name. • `size` - The file size. • `stime` - The time the Log File was created. • `etime` - The time the Log File was closed. The default is `stime`.
-r	Reverse the sort order (descending order).
module	The name of the machine on which the files are located. It can be a gateway or a Log Server. The default is localhost.

Example

This example shows the extended file list you see when you use the `fw lslogs -e` command:

```
fw lslogs -e module3
Size  Creation Time       Closing Time         Log file name
99KB  10Jan2002 16:46:27  10Jan2002 18:36:05
2002-01-10_183752.log
16KB  10Jan2002 18:36:05    --                 fw.log
```

fw putkey

Description This command installs a VPN-1 & FireWall-1 authentication password on a host. This password is used to authenticate internal communications between VPN-1 gateways and between a Check Point gateway and its SmartCenter server. A password is used to authenticate the control channel the first time communication is established. This command is required for backward compatibility scenarios.

Usage

```
fw putkey [-opsec] [-no_opsec] [-ssl] [-no_ssl] [-k num]
[-n <myname>] [-p <pswd>] host...
```

Syntax

Argument	Description
`-opsec`	Only VPN-1 & FireWall-1 control connections are enabled.
`-no_opsec`	Only OPSEC control connections are enabled.
`-ssl`	The key is used for an SSL connection.
`-no_ssl`	The key is not used for an SSL connection.
`-k num`	The length of the first S/Key password chain for fwa1 authentication (Check Point's proprietary authentication protocol). The default is 7. When fewer than 5 passwords remain, the hosts renegotiate a chain of length 100, based on a long random secret key. The relatively small default value ensures that the first chain, based on a short password entered by the user, is quickly exhausted.

Argument	Description
`-n <myname>`	The IP address (in dot notation) to be used by VPN-1 & FireWall-1 when identifying this host to all other hosts, instead of, for example, the resolution of the `hostname` command.
`-p <psw>`	The key (password). If you do not enter the password on the command line, you will be prompted for it.
`host`	The IP address(es) or the resolvable name(s) of the other host(s) on which you are installing the key (password). This should be the IP address of the interface "closest" to the host on which the command is run. If it is not, you will get error messages such as the following: "`./fwd: Authentication with hostname for command sync failed`"

Comments This command is never used in a script.

fw repairlog

Description `fw repairlog` rebuilds a Log file's pointer files. The three files *name*`.logptr`, *name*`.loginitial_ptr` and *name*`.logaccount_ptr` are recreated from data in the specified Log file. The Log file itself is modified only if the `-u` flag is specified.

Usage

```
fw repairlog [-u] logfile
```

Syntax

Argument	Description
`-u`	Indicates that the unification chains in the Log file should be rebuilt.
`logfile`	The name of the Log file to repair.

fw sam

Description

This command is used to manage the Suspicious Activity Monitoring (SAM) server. Use the SAM server to block connections to and from IP addresses without the need to change the Security Policy.

SAM commands are logged. Use this command to (also) monitor active SAM requests (see `-M` option).

To configure the SAM server on the SmartCenter server or FireWall-1 gateway machine, use SmartDashboard to edit the **Advanced > SAM** page of the Check Point gateway object.

Usage

```
Add/Cancel SAM rule according to criteria:

fw sam [-v][-s <sam server>][-S <server sic name>][-f <fw
host>][-t timeout][-l log][-C] -<n|i|I|j|J> <Criteria>
```

Delete all SAM rules:

```
fw sam [-v][-s <sam server>][-S <server sic name>][-f <fw
host>] -D
```

Monitor all SAM rules:

```
fw sam [-v][-s <sam server>][-S <server sic name>][-f <fw
host>] -M -ijn all
```

Monitor SAM rules according to criteria:

```
fw sam [-v][-s <sam server>][-S <server sic name>][-f <fw
host>] -M -ijn <Criteria>
```

Syntax

Parameter	Meaning
`-v`	Verbose mode. Writes one message (describing whether the command was successful or not) to `stderr` for each VPN-1 Power gateway machine on which the command is enforced.
`-s sam_server`	The IP address (in dot format) or the resolvable name of the FireWalled host that will enforce the command. The default is `localhost`.
`-S server_sic_name`	The SIC name for the SAM server to be contacted. It is expected that the SAM server will have this SIC name, otherwise the connection will fail. If no server SIC name is supplied the connection will proceed without SIC names comparison. For more information about enabling SIC refer to the OPSEC API Specification.
`-f <fw host>`	Specify the `host`, the VPN-1 Power gateway machine on which to enforce the action. `host` can be one of the following (default is `All`): • `localhost`—Specify the computer running the SAM server to enforce the action on it. • The name of the object or group—the action is enforced on this object; if this object is a group, on every object in the group. • `Gateways`—Action enforced on FireWalls defined as gateways and managed by SmartCenter server where the SAM server runs. • `All`—Enforced on FireWalls managed by SmartCenter server where SAM server runs.
`-D`	Cancel all inhibit (`-i`, -j,`-I`,`-J`) and notify (`-n`) commands. To “uninhibit” inhibited connections, execute `fw sam` with the `-C` or `-D` parameters. It is also possible to use this command for active SAM requests.

Parameter	Meaning
`-C`	Cancel the command to inhibit connections with the specified parameters. These connections will no longer be inhibited (rejected or dropped). The command parameters must match the ones in the original command, except for the `-t` (timeout) parameter.
`-t timeout`	The time period (in seconds) for which the action will be enforced. The default is forever or until cancelled.
`-l log`	The type of the log for enforced actions can be one of the following: `nolog`, `long_noalert`, `long_alert`. The default is `long_alert`.
`-n`	Notify, or generate, a long-format log entry. Generates an alert when connections that match the specified services or IP addresses pass through the FireWall. This action does not inhibit or close connections.
`-i`	Inhibit (do not allow) new connections with the specified parameters. Each inhibited connection is logged according to log type. Matching connections will be *rejected*.
`-I`	Inhibit new connections with the specified parameters, and close all existing connections with the specified parameters. Each inhibited connection is logged according to the log type. Matching connections will be *rejected*.
`-j`	Inhibit new connections with the specified parameters. Each inhibited connection is logged according to the log type. Connections will be *dropped*.
`-J`	Inhibit new connections with the specified parameters, and close all existing connections with the specified parameters. Each inhibited connection is logged according to the log type. Connections will be *dropped*.
`-M`	Monitor the active SAM requests with the specified actions and criteria.
`all`	Get all active requests. For monitoring purposes only.

Usage

Criteria are used to match connections, and are composed of various combinations of the following parameters:

```
<source ip><source netmask><destination ip><destination
netmask> <service><protocol>
```

Possible combinations are:

```
src <ip>

dst <ip>

any <<ip>

subsrc <ip><netmask>

subdst <ip><netmask>

subany <ip><netmask>

srv <src ip><dest ip><service><protocol>

subsrv <src ip><src netmask><dest ip><dest netmask><service>
<protocol>

subsrvs <src ip><src netmask><dest ip><service><protocol>

subsrvd <src ip><dest ip><dest netmask><service><protocol>

dstsrv <dest ip><service><protocol>

subdstsrv <dest ip><dest netmask><service><protocol>

srcpr <ip><protocol>

dstpr <ip><protocol>

subsrcpr <ip><netmask><protocol>

subdstpr <ip><netmask><protocol>
```

Syntax

Criteria Parameters	Description
`src <ip>`	Match the source IP address of the connection.
`dst <ip>`	Match the destination IP address of the connection.
`any <ip>`	Match either the source IP address or the destination IP address of the connection.

Criteria Parameters	Description
`subsrc <ip> <netmask>`	Match the source IP address of the connections according to the netmask.
`subdst <ip> <netmask>`	Match the destination IP address of the connections according to the netmask.
`subany <ip> <netmask>`	Match either the source IP address or destination IP address of connections according to the netmask.
`srv <src ip> <dst ip> <service> <protocol>`	Match the specific source IP address, destination IP address, service and protocol.
`subsrv <src ip> <netmask> <dst ip> <netmask> <service> <protocol>`	Match the specific source IP address, destination IP address, service and protocol. Source and destination IP addresses are assigned according to the netmask.
`subsrvs <src ip> <src netmask> <dest ip> <service> <protocol>`	Match the specific source IP address, source netmask, destination netmask, service and protocol.
`subsrvd <src ip> <dest ip> <dest netmask> <service> <protocol>`	Match specific source IP address, destination IP, destination netmask, service and protocol.
`dstsrv <dst ip> <service> <protocol>`	Match specific destination IP address, service and protocol.
`subdstsrv <dst ip> <netmask> <service> <protocol>`	Match specific destination IP address, service and protocol. Destination IP address is assigned according to the netmask.
`srcpr <ip> <protocol>`	Match the source IP address and protocol.

Criteria Parameters	Description
dstpr <ip> <protocol>	Match the destination IP address and protocol.
subsrcpr <ip> <netmask> <protocol>	Match the source IP address and protocol of connections. Source IP address is assigned according to the netmask.
subdstpr <ip> <netmask> <protocol>	Match the destination IP address and protocol of connections. Destination IP address is assigned according to the netmask.

Example

This command inhibits all connections originating on louvre for 10 minutes. Connections made during this time will be rejected:

```
fw sam -t 600 -i src louvre
```

This command inhibits all FTP connections from the louvre subnet to the eifel subnet. All existing open connections will be closed. New connection will be dropped, a log is kept and an alert is sent:

```
fw sam -l long_alert -J subsrvs louvre 255.255.255.0 eifel
21 6
```

The previous command will be enforced forever - or until canceled by the following command:

```
fw sam -C -l long_alert -J subsrvs louvre 255.255.255.0
eifel 21 6
```

This command monitors all active “inhibit” or “notify SAM” requests for which lourve is the source or destination address:

```
fw sam -M -nij any lourve
```

This command cancels the command in the first example:

```
fw sam -C -i src louvre
```

fw stat

Description

State tables are used to keep state information which the FireWall-1 virtual machine, and other components of FireWall-1 need in order to correctly inspect the packet. The tables are actually the ‘memory’ of the virtual machine in the kernel, and are the key component of Check Point Stateful Inspection technology. State tables are implemented as dynamic hash tables in kernel memory. All field values are in hexadecimal, apart from the time-out value at the end of the entry, when present.

The `fw tab` command displays the content of state tables on the target hosts in various formats. For each host, the default format displays the host name and a list of all tables with their elements.

Usage

```
fw tab [-all |-conf conffile] [-s][-m number][-u][-t
tname][-x tname][-d] <targets>
```

Syntax

Argument	Description
-all	The command is to be executed on all targets specified in the default system configuration file ($FWDIR/conf/sys.conf).
-conf conffile	The command is to be executed on the targets specified in conffile.
-s	Summary of the number of entries in each table: host name, table name, table ID, and its number of entries
-m number	For each table, display only its first number of elements (default is 16 entries at most).
-u	Do not limit the number of entries displayed for each table.
-t tname	Display only tname table.
-x tname	Delete all entries in all tables
-d	Debug mode
targets	The command is executed on the designated targets.

A table has a list of associated attributes.

Example

To display only the `arp_table` table,

Comments

```
fw tab -t arp_table
```

```
fw sam -C -i src louvre
```

fw tab

Description The fw tab command enables you to view kernel table contents and change them (that is, only dynamic tables since the content of a static table is indeed static).

Usage

```
fw tab [-t <table>] [-s] -c] [-f] [-o <filename>] [-r] [-u
| -m <maxvals>] [[-x | -a} -e entry] [-y] [hostname]”
```

Syntax

Argument	Description
`- t <table>`	Specifies a table for the command.
`-s`	Displays a short summary of the table (s) information.
`-y`	Specifies to not prompt a user before executing any commands.
`-f`	Displays a formatted version of the table content. Every table may have its own specific format style.
`-o <filename>`	Dumps CL formatted output to filename, which can later be read by fw log or any other entity that can read FW log formats.
`-c`	Displays formatted table informatin in common format.
`-r`	Resolves IP addresses in formatted output.
`-x, -a, -e`	It is possible to add or remove an entry from an existing dynamic table by using the -a or the -x flags, respectively. These flags must be followed by the -e flag and an entry description (<entry>).
`[hostname]`	A list of one or more targets. When not used, the local machine is used as the default target.

Example

fw tab -t <table-name> -a -e “1,2;3,4,5” or
fw tab -t <table-name> -a -e “<1,2;3,4,5>”
Adds an entry:
<00000001,00000002,00000003,00000004,00000005,>to<table-name>

fw tab -t <table-name> -a -e “1,2,” or
fw tab -t <table-name> -a -e “<1,2>”
Adds an entry with only a key field: <00000001,00000002>

If table<table-name> contains the following entry:
<0000000,00000001,00000002>
fw tab -t <table-name> -x -e “0,1” or
fw tab -t <table-name> -x -e “0,1;2”

Removes the entry from the specified table.

Comments If table has the ‘expire’ attribute, entries added using the -a flag will receive the default table timeout.
This feature only works on local machine kernel tables and does not work on a remote machine’s tables like additional fw tab commands.
The -x flag can be used independantly of the -e flag in which case the entire table content is deleted.
This feature should only be used for debug purposes. It is not advisable to arbitrarily change the content of any kernel table since doing so may have unexpected results including unexpected security and connectivity impacts.

fw ver

Description This command displays the VPN-1 Power major and minor version number and build number.

Usage

```
fw ver [-k][-f <filename>]
```

Syntax

Argument	Description
-k	Print the version name and build number of the Kernel module.
-f <filename>	Print the version name and build number to the specified file.

fwm

Description This command is used to perform management operations on VPN-1. It controls *fwd* and all Check Point daemons.

Usage

```
fwm
```

In This Section

fwm dbimport

Description `fwm dbimport` imports users into the Check Point User Database from an external file. You can create this file yourself, or use a file generated by `fwm dbexport`.

Usage `fwm dbimport [-m] [-s] [-v] [-r] [-k` *errors*`] [-f` *file*`] [-d` *delim*`]`

Syntax

Argument	Description
`-m`	If an existing user is encountered in the import file, the user's default values will be replaced by the values in the template (the default template or the one given in the attribute list for that user in the import file), and the original values will be ignored.
`-s`	Suppress the warning messages issued when an existing user's values are changed by values in the import file.
`-v`	verbose mode

Argument	Description
`-r`	`fwm dbimport` will delete all existing users in the database.
`-k errors`	Continue processing until `nerror` errors are encountered. The line count in the error messages starts from 1 including the attributes line and counting empty or commented out lines.
`-f file`	The name of the import file. The default import file is `$FWDIR/conf/user_def_file`. Also see the requirements listed under "File Format" on page 72.
`-d delim`	Specifies a delimiter different from the default value (`;`).

Comments The IKE pre shared secret does not work when exporting from one machine and importing to another.

To ensure that there is no dependency on the previous database values, use the `-r` flag together with the `-m` flag.

File Format

The import file must conform to the following Usage:

- The first line in the file is an attribute list.

  The attribute list can be any partial set of the following attribute set, as long as `name` is included:

```
{name; groups; destinations; sources; auth_method;
fromhour; tohour; expiration_date; color; days;
internal_password; SKEY_seed; SKEY_passwd; SKEY_gateway;
template; comments; userc}
```

- The attributes must be separated by a delimiter character.

  The default delimiter is the `;` character. However, you can use a different character by specifying the `-d` option in the command line.
- The rest of the file contains lines specifying the values of the attributes per user. The values are separated by the same delimiter character used for the attribute list. An empty value for an attribute means use the default value.

- For attributes that contain a list of values (for example, days), enclose the values in curly braces, that is,{ }. Values in a list must be separated by commas. If there is only one value in a list, the braces may be omitted. A + or - character appended to a value list means to add or delete the values in the list from the current default user values. Otherwise the default action is to replace the existing values.
- Legal values for the days attribute are: `MON`, `TUE`, `WED`, `THU`, `FRI`, `SAT`, `SUN`.
- Legal values for the authentication method are: `Undefined`, `S/Key`, `SecurID`, `Unix Password`, `VPN-1 & FireWall-1 Password`, `RADIUS`, `Defender`.
- Time format is `hh:mm`.
- Date format is `dd-mmm-yy`, where `mmm` is one of {`Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Oct`, `Nov`, `Dec`}.
- If the S/Key authentication method is used, all the other attributes regarding this method must be provided.
- If the VPN-1 & FireWall-1 password authentication method is used, a valid VPN-1 & FireWall-1 password should be given as well. The password should be encrypted with the C language `encrypt` function.
- Values regarding authentication methods other than the one specified are ignored.
- The `userc` field specifies the parameters of the user's SecuRemote connections, and has three parameters, as follows:

  key encryption method - DES, CLEAR, Any

  data encryption method - DES, CLEAR, Any

  integrity method - MD5,[blank] = no data integrity

  "Any" means the best method available for the connection. This depends on the encryption methods available to both sides of the connection. For example,

  {`DES,CLEAR,`} means: key encryption method is DES; no data encryption; no data integrity
- A line beginning with the `!` character is considered a comment.

fwm dbexport

Description `fwm dbexport` exports the Check Point User Database to a file. The file may be in one of the following formats:

- the same Usage as the import file for `fwm dbimport`
- LDIF format, which can be imported into an LDAP server using `ldapmodify`

Usage

To export the User Database to a file that can be used with `fwm dbimport`:

```
fwm dbexport [ [-g group | -u user] [-d delim]
[-a {attrib1, attrib2, ...} ] [-f file] ]
```

To export the User Database as an LDIF file:

```
fwm dbexport -l -p [-d] -s subtree [-f file]  [-k
IKE-shared-secret]
```

Syntax

Argument	Description
`-g group`	Specifies a group (`group`) to be exported. The users in the group are not exported.
`-u user`	Specifies that only one user (`user`) is to be exported.
`-d`	Debug flag
`-a {attrib1, attrib2, ...}`	Specifies the attributes to export, in the form of a comma-separated list, between {} characters, for example, `-a {name,days}`. If there is only one attribute, the {} may be omitted.
`-f file`	`file` specifies the name of the output file. The default output file is `$FWDIR/conf/user_def_file`.
`-l`	Create an LDIF format file for importation by an LDAP server.
`-p`	The profile name.
`-s`	The branch under which the users are to be added.
`-k`	This is the Account Unit’s IKE shared secret (**IKE Key** in the **Encryption** tab of the **Account Unit Properties** window

Comments

Note:

- The IKE pre shared secret does not work when exporting from one machine and importing to another.
- If you use the `-a` parameter to specify a list of attributes, and then import the created file using `fwm dbimport`, the attributes not exported will be deleted from the user database.
- `fwm dbexport` and `fwm dbimport` (non-LDIF Usage) cannot export and import user groups. To export and import a user database, including groups, proceed as follows:

  * Run `fwm dbexport` on the source SmartCenter server.

  * On the destination SmartCenter server, create the groups manually.

  * Run `fwm dbimport` on the destination SmartCenter server.

The users will be added to the groups to which they belonged on the source SmartCenter server.

- If you wish to import different groups of users into different branches, run `fwm dbexport` once for each subtree, for example:

```
fwm dbexport -f f1 -l -s ou=marketing,o=WidgetCorp,c=us
fwm dbexport -f f2 -l -s ou=rnd,o=WidgetCorp,c=uk
```

  Next, import the individual files into the LDAP server one after the other. For information on how to do this, refer to the documentation for your LDAP server.

- The LDIF file is a text file which you may wish to edit before importing it into an LDAP server. For example, in the Check Point user database, user names may be what are in effect login names (such as “maryj”) while in the LDAP server, the DN should be the user’s full name (“Mary Jones”) and “maryj” should be the login name.

Example

Suppose the User Database contains two users, “maryj” and “ben”.

```
fwm dbexport -l -s o=WidgetCorp,c=us
```

creates a LDIF file consisting of two entries with the following DNs:

```
cn=ben,o=WidgetCorp,c=us
cn=maryj,o=WidgetCorp,c=us
```

fwm dbload

Description This command downloads the user database and network objects information to selected targets. If no target is specified, then the database is downloaded to localhost.

Usage

```
fwm dbload [-all | -conf conffile] [targets]
```

Syntax

Argument	Description
-all	Execute command on all targets specified in the default system configuration file (`$FWDIR/conf/sys.conf`). This file must be manually created.
-conf *conffile*	Only OPSEC control connections are enabled.
targets	Execute command on the designated targets.

fw hastat

Description The `fwm hastat` command displays information about High Availability machines and their states.

Usage

```
fw hastat [<target>]
```

Syntax

Argument	Description
<target>	A list of machines whose status will be displayed. If `target` is not specified, the status of the local machine will be displayed.

fwm ikecrypt

Description `fwm ikecrypt` command line encrypts the password of a SecuRemote user using IKE. The resulting string must then be stored in the LDAP database.

Usage

```
fwm ikecrypt shared-secret user-password
```

Syntax

Argument	Description
`shared-secret`	The IKE Key defined in the **Encryption** tab of the **LDAP Account Unit Properties** window.
`user-password`	The SecuRemote user's password.

Comments An internal CA must be created before implementing IKE encryption. An Internal CA is created during the initial configuration of the SmartCenter server, following installation.

fwm load

Description This command compiles and installs a Security Policy or a specific version of the Security Policy on the target's VPN-1 Power gateways. This is done in one of two ways:

- `fwm load` compiles and installs an Inspection Script (`*.pf`) file on the designated VPN-1 Power gateways.
- `fwm load` converts a Rule Base (`*.W`) file created by the GUI into an Inspection Script (`*.pf`) file then installs it to the designated VPN-1 Power gateways.

Versions of the Security Policy and databases are maintained in a version repository on the SmartCenter server. Using this command specific versions of the Security Policy can be installed on a gateway (local or remote) without changing the definition of the current active database version on the SmartCenter server.

To protect a target, you must load a Policy that contains rules whose scope matches the target. If none of the rules are enforced on the target, then all traffic through the target is blocked.

Usage

```
fwm load [-p <plug-in product name>] [-S] <rulebase version
name> <targets>
```

Syntax

Argument	Description
-S	The targets are VPN-1 UTM Edge gateways.
-P	Specifies the plug-in product name if applicable.
rulebase	A Rule Base file (`*.W`) created by the GUI. The file's full pathname must be given.
-v version number	Retrieve the Security Policy from the version repository. The version number represents the number of the Security Policy as it is saved in the version repository.
targets	Execute command on the designated target.

Comments Currently, targets can be gateways from version NG FP3 and higher.

Example The following command installs the Security Policy `standard` in the target gateway `johnny`.

```
fwm load -v18 standard johnny
```

fwm lock_admin

Description This commands enables you to view and unlock locked administrators.

Usage

```
fwm lock_admin [-v][-u administrator][-ua]
```

Syntax

Argument	Description
-v	View the names of all locked administrators
-u administrator	Unlock a single administrator
-ua	Unlock all locked administrators

fwm logexport

Description `fwm logexport` exports the Log file to an ASCII file.

Usage

```
fwm logexport [-d delimiter] [-i filename] [-o outputfile]
[-n] [-p]
[-f] [-m <initial | semi | raw>] [-a]
```

Syntax

Argument	Description
`-d delimiter`	Set the output delimiter. The default is a semicolon (`;`)
`-i filename`	The name of the input Log file. The default is the active Log file, `fw.log`
`-o outputfile`	The name of the output file. The default is printing to the screen.
-n	Do not perform DNS resolution of the IP addresses in the Log file (this option significantly speeds the processing).
-p	Do not perform service resolution. A service port number is displayed.

Argument	Description
-f	If this is the active Log file (`fw.log`), wait for new records and export them to the ASCII output file as they occur.
-m	This flag specifies the unification mode. • `initial` - the default mode. Complete the unification of log records; that is, output one unified record for each id. . • `semi` - step-by-step unification, that is, for each log record, output a record that unifies this record with all previously-encountered records with the same id. • `raw` - output all records, with no unification.
-a	Show account records only (the default is to show all records)

Comments

Controlling the Output of `fwm logexport` using `logexport.ini`

The output of `fwm logexport` can be controlled by creating a file called `logexport.ini` and placing it in the `conf` directory: `$FWDIR/conf`.The `logexport.ini` file should be in the following format:

```
[Fields_Info]

included_fields =
field1,field2,field3,<REST_OF_FIELDS>,field100

excluded_fields = field10,field11
```

note that:

- the `num` field will always appear first, and cannot be manipulated using `logexport.ini`
- `<REST_OF_FIELDS>` is a reserved token that refers to a list of fields. It is optional. If `-f` option is set, `<REST_OF_FIELDS>` is based on a list of fields taken from the file `logexport_default.C`.

- If `-f` is not set, `<REST_OF_FIELDS>` will be based on the given input log file.
- It is not mandatory to specify *both* `included_fields` and `excluded_fields`.

Format:

The `fwm logexport` output appears in tabular format. The first row lists the names of all fields included in the subsequent records. Each of the subsequent rows consists of a single log record, whose fields are sorted in the same order as the first row. If a records has no information on a specific field, this field remains empty (as indicated by two successive semi-colons).

Example

```
num;date;time;orig;type;action;alert;i/f_name;i/f_dir;product;sys_m
essage:;service;s_port;src;dst;
0; 5Dec2002;9:08:44;jam.checkpoint.com;control;
;;daemon;inbound;VPN-1 & FireWall-1;The hme0 interface is
not protected by the anti-spoofing feature. Your network may
be at risk;;;;;

1; 5Dec2002;9:08:44;jam.checkpoint.com;control;
;;daemon;inbound;VPN-1 &
FireWall-1;;ftp;23456;1.2.3.4;3.4.5.6;
```

fwm sic_reset

Description This command is used to reset the Internal CA and delete all the certificates from the Internal CA and the Internal CA itself. After running `sic_reset`, the ICA should be initialized through the `cpconfig` command. If this command is run all the certified IKE from the Internal CA should be removed (using the SmartConsole).

Usage

```
fwm sic_rest
```

Syntax

Argument	Description
sic_reset	Resets the internal CA SIC certificates and deletes the Internal CA.

fwm unload <targets>

Description This command uninstalls the currently loaded Inspection Code from selected targets.

Usage `fwm unload <targets>[-all | -conf conffile]`

Syntax

Argument	Description
targets	Execute command on the designated targets.
-all	Execute command on all targets specified in the default system configuration file (`$FWDIR/conf/sys.conf`). This file must be manually created.
conf *conffile*	Execute command on targets specified in the `conffile`.

fwm ver

Description `fwm ver` displays the build number.

Usage `fwm ver [-f <filename>]`

Syntax

Parameter	Meaning
-f <filename>	Print the version name and build number to the specified file.
`-p port`	The ELA Proxy's port number. Default is 18187.
`-f token value`	A field to be added to the log, represented by a `token-value` pair as follows: • `token` is the name of the field to be added to the log. `token` may not contain spaces. • `value` is the field's value. `value` may not contain spaces. This option may be used multiple times to add multiple `token-value` pairs to the log. If `token` is a reserved log field name, the specified field's value will appear in the corresponding column in the Log Viewer. Otherwise, the `token-value` pair will be displayed in the **Info.** column in the Log Viewer.

`-m alertty`	The alert to be triggered at the ISP site. This alert overrides the alert specified in the log message generated by the alert daemon.
	The response to the alert is handled according to the actions specified in the ISP's Security Policy:
	The following alerts execute the OS commands defined in the corresponding fields of the **Log and Alert** tab of the **Properties Setup** window in **Global Properties**:
	• **alert**. `Popup alert command.`
	• **mail**. `Mail alert command.`
	• **snmptrap**. `SNMP trap alert command.`
	• **spoofalert**. `Anti-spoof alert command.`
	The following NetQuota and ServerQuota alerts execute the OS commands specified in:
	`$FWDIR/conf/objects.C:`
	`value=clientquotaalert.`
	`Parameter=clientquotaalertcmd`

GeneratorApp

Description This command generates a report for the SmartView Reporter. Both command line parameters are required.

Usage

```
GeneratorApp [Directory/""] {ReportID}
```

Syntax

Argument	Description
Directory	The result directory (that is, the location at which the result is placed).
ReportID	The report ID required for command line generations. The Report ID must be enclosed within curly braces. For a list of all Report IDs see Appendix B "Predefined Reports" in the SmartView Reporter Administration Guide.

Example

For automatic directory computation use "". In such a case, the directory should be as follows:

<Result location>/<Report Name>/<Generation Date and Time>

inet_alert

Description

This command notifies a company's Internet Service Provider (ISP) when the company's corporate network is under attack. The `inet_alert` utility forwards log messages generated by the alert daemon to an external Management Station, typically located at the ISP site. The ISP can then analyze the alert and decide how to react.

`inet_alert` uses the ELA Protocol to send the alert. The Management Station receiving the alert must be running the ELA Proxy.

If communication with the ELA Proxy is to be authenticated or encrypted, a key exchange must be performed between the Management Station running the ELA Proxy and the VPN-1 Power gateway generating the alert.

To use this utility, enter it into a script. From **Global Properties > Logs and alert > alert commands > early versions compatibility > run 4.x alert script**, and enter the name of the script.

Usage

```
inet_alert -s ipaddr [-o] [-a auth_type] [-p port] [-f token
value] [-m alerttype]
```

Syntax

Parameter	Meaning
`-s ipaddr`	The IP address (in dot format) of the ELA Proxy to be contacted.
`-o`	Print the alert log received by `inet_alert` to `stdout`. Use this option when `inet_alert` is part of a pipe.
`-a auth_type`	The type of connection to the ELA Proxy. One of the following values: • **ssl_opsec**. `Means the connection is authenticated and encrypted, (Default)` • **auth_opsec**. `Means the connection is authenticated.` • **clear**. `Means the connection is neither authenticated nor encrypted.`

Parameter	Meaning
`-p port`	The ELA Proxy's port number. Default is 18187.
`-f token value`	A field to be added to the log, represented by a `token-value` pair as follows: • `token` is the name of the field to be added to the log. `token` may not contain spaces. • `value` is the field's value. `value` may not contain spaces. This option may be used multiple times to add multiple `token-value` pairs to the log. If `token` is a reserved log field name, the specified field's value will appear in the corresponding column in the Log Viewer. Otherwise, the `token-value` pair will be displayed in the **Info.** column in the Log Viewer.

`-m alertty`	The alert to be triggered at the ISP site. This alert overrides the alert specified in the log message generated by the alert daemon.
	The response to the alert is handled according to the actions specified in the ISP's Security Policy:
	The following alerts execute the OS commands defined in the corresponding fields of the **Log and Alert** tab of the **Properties Setup** window in **Global Properties**:
	• **alert**. `Popup alert command.`
	• **mail**. `Mail alert command.`
	• **snmptrap**. `SNMP trap alert command.`
	• **spoofalert**. `Anti-spoof alert command.`
	The following NetQuota and ServerQuota alerts execute the OS commands specified in:
	`$FWDIR/conf/objects.C:`
	`value=clientquotaalert.`
	`Parameter=clientquotaalertcmd`

Return Value

exit status	meaning
0	Execution was successful.
102	Undetermined error.
103	Unable to allocate memory.

exit status	meaning
104	Unable to obtain log information from stdin.
106	Invalid command line arguments.
107	Failed to invoke the OPSEC API.

Example

```
inet_alert -s 10.0.2.4 -a clear -f product cads -m alert
```

This command specifies that in the event of an attack, inet_alert should take the following actions:

- Establish a clear connection with the ELA Proxy located at IP address 10.0.2.4.
- Send a log message to the specified ELA Proxy. The product field of this log message should be set to "cads". This means that "cads" will be displayed in the **product** column of the Log Viewer.

Trigger the OS command specified in the **Popup Alert Command** field of the **Log and Alert** tab of the **Properties** Setup window in the Policy Editor.

ldapcmd

Description

ldapcmd is used to manage processes running on the FireWall-1 collectively or individually. It includes:

Cache

cache operations, such as emptying the cache, as well as providing debug information.

Statistics

lookup statistics such as, all user search, pending lookups (when two or more lookups are identical) and total lookup time (the total search time for a specific lookup)

cache statistics such as hits and misses

Logging

view the alert and warning log regarding debug

Usage

```
ldapcmd -p process_name | all command [-d debug_level]
 [command_arg]
```

where command is:

- `cacheclear` (either `all` or `UserCacheObject` or `TemplateCacheObject` or `TemplateExtGrpCacheObject`)
- `cachetrace` (either `all` or `UserCacheObject` or `TemplateCacheObject` or `TemplateExtGrpCacheObject`)
- `stat` [either `print_interval (reset interval time in secs)` or `0 (stop statistics)`]
- `log` (either `on` or `off`)

Syntax

Argument	Description
-p	run a specified process or run all processes
command	specify a command
log	specify whether or not to create LDAP logs

ldapcompare

Description

`ldapcompare` is used to perform compare queries that prints a message whether the result returned a match or not. `ldapcompare` opens a connection to an LDAP directory server, binds, and performs the comparison specified on the command line or from a specified file.

Usage

```
ldapcompare -d [options] dn attribute value
```

Syntax

Argument	Description
-d	Debug flag
options	See below.
dn	The DN object.
attribute	The attribute of the DN object.
value	The value of the attribute of the DN object.

The `ldapcompare` options are as follows:

- `-u` -Include user-friendly entry names in the output.
- `-d <level>` -Set LDAP debugging level to “level”.
- `-F sep` -Print “sep” instead of “=” between attribute names and values.

- `-f <file>` -Perform sequence of compares listed in “file”.
- `-D <binddn>` -Bind DN.
- `-w <passwd>` -Bind password (for simple authentication).
- `-h <host>` -LDAP server.
- `-p <port>` -Port on the LDAP server.
- `-T <timeout>` -Client side timeout for all operations (in milliseconds).
- `-l <time limit>` -Server Side time limit (in seconds) for compare.
- `-z <size limit>` -Server Side size limit (in entries) for compare.

ldapconvert

Description

`ldapconvert` is a utility program to port from `Member` mode to `MemberOf` mode. This is done by searching all specified group/template entries and fetching their Member attribute values.

Each value is the DN of a member entry. The entry identified by this DN will be added the `MemberOf` attribute value of the group/template DN at hand. In addition, those Member attribute values will be deleted from the group/template unless `Both` mode is specified.

While running the program, a log file, named `ldapconvert.log`, is generated in the current directory, logging all modifications done and errors encountered.

Usage

```
ldapconvert -d -h <host> -p <port> -D user_DN -w <secret> [-g
group_DN | -f <file>] -m mem_attr -o memberof_attr -c
memberobjectclass[extra options]
```

Syntax

Argument	Description
-d	Debug flag
`-h <host>`	LDAP server IP address.
`-p <port>`	LDAP server port number.
`-D user_DN`	LDAP bind DN.
`-w <secret>`	LDAP bind password.
`-g group_DN`	Group or template DN to perform the conversion on. May appear multiple times for multiple entries.

Argument	Description
`-f <file>`	File containing a list of group DNs each separated by a new line.
`-m mem_attr`	LDAP attribute name when fetching and (possibly) deleting a Member attribute value.
`-o memberof_attr`	LDAP attribute name when adding a "MemberOf" attribute value.
`-c memberobjectclass`	LDAP objectclass attribute value that filters which type of member entries to modify. May appear multiple times creating a compound filter.
`extra options`	See below

The `ldapcomvert extra options` are as follows:

- `-M` -Maximum number of member LDAP updated simultaneously (default is 20).
- `-B` -Convert to Both mode
- `-p <port>` -LDAP port (default is 389).
- `-T <timeout>` -Client side timeout for LDAP operations, in milliseconds: default is "never".
- `-l <time limit>` -Server side time limit for LDAP operations, in seconds: default is "never".
- -s -Server side size limit for LDAP operations (in entries) (default is "none").
- `-z` -Use SSL.

Comments

It is recommended to backup the LDAP server before running the conversion program in case unrecoverable errors are encountered.

There are two `GroupMembership` modes: template-to-groups and user-to-groups. It is imperative to keep these modes consistent. For instance, if you apply conversion on LDAP users to include '`MemberOf`' attributes for their groups, then this conversion should also be applied on LDAP defined templates for their groups.

Why does a command run with the option `-M` fail?

The program terminates with an error message stating the connection terminated unexpectedly.

This means that the LDAP server could not handle so many LDAP requests simultaneously and closed the connection. The solution is to run the program again with a lower value for the `–M` option (the default value should be adequate but could also cause a connection failure in extreme situation). Continue to reduce the value until the program exits normally. Each time you run the program with the same set of groups the program will pick up where it left off.

Example

A group is defined with the DN: `cn=cpGroup,ou=groups, ou=cp, c=il` and the following attributes:

```
...
cn=cpGroup
uniquemember=”cn=member1,ou=people, ou=cp,c=il”
uniquemember=” cn=member2, ou=people, ou=cp,c=il”
...
```

For the 2 member entries:

```
...
cn=member1
objectclass=fw1Person
...
```

and:

```
...
cn=member2
objectclass=fw1Person
...
```

Run `ldapconvert` with the following arguments:

```
ldapconvert -g cn=cpGroup,ou=groups, ou=cp, c=il -h myhost
-d cn=admin -w secret  \ –m uniquemember -o memberof -c
fw1Person
```

The result for the group DN will be as follows:

```
...
cn=cpGroup
...
```

The result for the 2 member entries will be as follows:

```
...
cn=member1
objectclass=fw1Person
memberof=”cn=cpGroup,ou=groups, ou=cp, c=il”
...
```

and

```
...
cn=member2
objectclass=fw1Person
memberof=" cn=cpGroup,ou=groups, ou=cp, c=il"
...
```

Running the same command with the -B options, will produce the same result but the group entry will not be modified.

If there exists another member attribute value for the same group entry:

```
uniquemember="cn=template1,ou=people, ou=cp,c=il"
```

and the template is:

```
cn=member1
objectclass=fw1Template
```

after running the same command line the template entry will stay intact because the command line specified the option `-c fw1Person` but the object class of template1 is `fw1Template`.

ldapmodify

Description

`ldapmodify` imports users to an LDAP server. The input file must be in the LDIF format.

Usage

```
ldapmodify -a -c -d -h <host> -p <port> -D <LDAPadminDN> -p
<LDAPadminPassword> -f <exportfilename>.ldif -d
```

Syntax

Argument	Description
-a	Add users.
-c	Continue on errors.
-h <host>	LDAP server IP address.
-d	Debug flag
-p <port>	LDAP server port number.
-D <LDAPadminDN>	LDAP Administrator DN.
-p <LDAPadminPassword>	LDAP Administrator password.
-f <exportfilename>.ldif	Specifies the name of the input file. This file must be in the LDIF format.

Comments You can import the VPN-1 Power User Database to an LDAP server by first generating an LDIF file using `fwm dbexport`, and then using `ldapmodify`.

Before importing, prepare the LDAP directory as follows:

- Make sure the root branch is defined as an allowed branch on your LDAP server.
- Restart the LDAP server.
- Create the branch into which the users will be imported, either by using **Create Tree Object** in the Account Management Client or with the `ldapmodify` command:

```
ldapmodify -a -h <host> -p <port> -D <LDAPadminDN> -w
<LDAPadminPassword>
dn: o=myOrg,c=US
objectclass: organization
o:myOrg
```

Example Importing Users using `ldapmodify`:

1. Export the users using `fwm dbexport` using `hello1234` as the pre-shared secret..

```
fwm dbexport -l -f ./o_file.ldif -s "o=bigcorp,c=uk" -k
hello1234
```

2. Create the `"o=bigcorp,c=uk"` branch.
3. Import the users:

```
ldapmodify -a -c -h <host> -p <port> -D bindDN -w bindPas -f
./o_file.ldif
```

4. Define an Account Unit with these parameters.

ldapsearch

Description `ldapsearch` queries an LDAP directory and returns the results.

Usage `ldapsearch [`*`options`*`]` *`filter`* `[`*`attributes`*`] -d`

Syntax

Argument	**Description**
options	See the `options` attributes below.
`filter`	RFC-1558 compliant LDAP search filter. For example, `objectclass=fw1host`.
`attributes`	The list of attributes to be retrieved. If no attributes are given, all attributes are retrieved.
-d	Debug flag

The following are the attributes for `options`:

- `-A` -Retrieve attribute names only (without values).
- `-B` -Do not suppress printing of non-ASCII values.
- `-D bindDN` -The DN to be used for binding to the LDAP server.
- `-F separator` -Print `separator` between attribute name and value instead of “=”.
- `-h host` -The LDAP server identified by IP address or resolvable name.
- `-l timelimit` -The server side time limit for search, in seconds.
- `-p portnum` -The port number. The default is standard LDAP port 389.
- `-S attribute` -Sort the results by the values of `attribute`.
- `-s scope` -One of the following: “`base`”, “`one`”, “`sub`”.
- `-b` -Base distinguished name (DN) for search.
- `-t` -Write values to files in `/tmp`. Each attribute-value pair is written to a separate file, named: `/tmp/ldapsearch-<attribute>-<value>`. For example, for the `fw1color` attribute, the file written is named `/tmp/ldapsearch-fw1color-a00188`.
- `-T timeout -` Client-side timeout (in milliseconds) for all operations.
- `-u -` Show “user friendly” entry names in the output. For example, show “`cn=Babs Jensen, users, omi`” instead of “`cn=Babs Jensen, cn=users,cn=omi`”
- `-w password -` The password.
- `-Z -` Encrypt using SSL.
- `-z sizelimit` -Server-side size limit for search, in entries.

Example

```
ldapsearch -p 18185 -b cn=omi objectclass=fw1host objectclass
```

This means that the LDAP directory will be queried for `fw1host` objects using port number 18185 with DN common name “`omi`”. For each object found, the value of its `objectclass` attribute will be printed.

log_export

Description

`log_export` is a utility that allows you to transfer Log data to an external database. This utility behaves as a LEA client. LEA (Log Export API) enables VPN-1 Power Log data to be exported to third-party applications. `log_export` receives the Logs from the SmartCenter server via LEA so it can be run from any host that has a SIC connection with the SmartCenter server and is defined as an OPSEC host. To run `log_export`, you need a basic understanding and a working knowledge of:

- Oracle database administration
- LEA

Usage

```
log_export [-f conf_file] [-l <lea_server_ip_address>] [-g
log_file_name,log_file_name,...] [-t <database_table_name>]
 [-p <database_password>][-h] [-d].
```

Syntax

Argument	Description
`-f conf_file`	The Configuration File from which `log_export` reads the Log file parameters. If `conf_file` is not specified, the default Configuration File `log_export.conf`, located in the current working directory.
-l <lea_server_ip_address>	The IP address of the LEA server.
-g log_file_name,log_file_name,...	A comma separated list of log file names from where the logs will be taken.
-t <database_table_name>	The name of the table in the database to which the logs will be added.

Argument	Description
p <database_password>	The database login password. If you do not want to specify the password in the Configuration File for security reasons, you can enter the password using the command line where it will not be saved anywhere.
`-h`	Display `log_export` usage.
`-d`	Display debugging information.

Further Info. For more information about LEA, see *Check Point LEA (Log Export API) Specification*

Comments Only Oracle database is currently supported.

Before you can run `log_export`, the Oracle client must be installed and configured. Make sure that:

- the `ORACLE_HOME` environment variable is set correctly.
- `$ORACLE_HOME/lib` is located in the PATH environment variable on the NT platform or `LD_LIBRARY_PATH` on Solaris and Linux platforms.
- If `log_export` is running from another machine, you must install and configure at least SVN Foundation and Reporting gateway.

The log_export Configuration File

`log_export` has a Configuration File. The Configuration File is a Check Point Set file and should be configured according to Set file conventions. The Configuration File contains the default parameters for `log_export`. `log_export` reads all parameters from the Configuration File that is specified in the command line.

Modifying the Configuration File

`log_export` parameters are defined in the Configuration File. To change the parameters, you can either modify the Configuration File or use the command line. Any parameter entered using the command line will override the parameters in the Configuration File.

Modify the Configuration File according to the following parameters:

- `db_connection_string` - The string that defines the Oracle database server. For example, the name of the server.
- `db_table_name` - The name of the table in the database to which the logs will be added.
- `create_db_table` - Following are the available options:

  - 1 - create a new table in the database
  - 0 - use the existing table.

  If there is an existing table, the logs will be added to that table. This requires that the existing table have the same format as the logs you are adding. If you enter 0 and there is no existing table, you will get an error message. The default is 1.
- db_user_name - The database login user name.
- db_password - The database login password.
- log_server_ip_address - The IP address of the LEA server.
- log_server_port - Port number of the LEA server. The default LEA port is 18184.
- log_file_name - A list of log file names from where the logs will be taken.
- log_fields - The name of the Log file as known by LEA.
- db_field_name - The Log field name as represented in the database table.
- db_field_type - The Log field type in the database table. This parameter can be one of the following:
  - STRING
  - NUMBER
  - DATE
- db_field_size - The size of the field in the database table. This parameter is required only if the db_field_type is either STRING or NUMBER.

Example Configuration File Example

```
:db_table_name (fw_log)
 :db_connection_string (database_service_name)
 :db_user_name (scott)
 :db_password (tiger)
 :log_server_ip_address (127.0.0.1)
 :log_server_port (18184)
 :create_db_table (1)
 :log_file_name (fw.log)
 :log_fields (
  : (time
    :db_field_name (log_time)
    :db_field_type (DATE)
  )
  : (product
    :db_field_name (product)
    :db_field_type (STRING)
    :db_field_size (25)
  )
  : (i/f_name
    :db_field_name (interface)
    :db_field_type (STRING)
    :db_field_size (100)
  )
  : (orig
    :db_field_name (origin)
    :db_field_type (STRING)
    :db_field_size (16)
  )
  : (action
    :db_field_name (action)
    :db_field_type (STRING)
    :db_field_size (16)
  )
  : (service
    :db_field_name (service)
    :db_field_type (STRING)
    :db_field_size (40)
  )
```

queryDB_util

Description queryDB_util enables searching the object database according to search parameters.

Usage

```
queryDB_util [-t <table_name>] [-o <object_name>] [-a]
 [-mu <modified_by>] [-mh <modified_from>]
 [-ma <modified_after>] [-mb <modified_before>] [-p|m|u|h|t|f]
 [-f filename} [-h] [-q]
```

Syntax

Argument	**Description**
-t <table_name>	The name of the table.
-o <object_name>	The name of the object.
[-a]	All objects.
-mu <modified_by>	The name of the administrator who last modified the object.
-mh <modified_from>	The host from which the object was last modified.
-ma <modified_after>	The date after which the object was modified <[hh:mm:ss][ddmmmyyyy]>. Either or both options may be used. Omitting hh:mm:ss defaults to today at midnight, omitting ddmmmyyyyy defaults to today's date on the client.
-mb <modified_before>	The date before which the object was modified <[hh:mm:ss][ddmmmyyyy]>. Either or both options may be used. Omitting hh:mm:ss defaults to today at midnight, omitting ddmmmyyyyy defaults to today's date on the client.
-plmlulhltlf	Short print options: • c -creation details • m -last_modification details • u - administrator name (create/modify) • h -host name (create/modify) • t -time (create/modify) • f -field details
-f filename	The name of the output file.
-h	Display command usage information.
-q	Quit.

Example Print modification details of all objects modified by administrator “aa”

```
query> -a -mu Bob -pm
Object Name:my_object
Last Modified by:Bob
Last Modified from:london
Last Modification time:Mon Jun 19 11:44:27 2000

Object Name:internal_ca
Last Modified by:Bob
Last Modified from:london
Last Modification time:Tue Jun 20 11:32:58 2000

A total of 2 objects match the query.
```

rs_db_tool

Description rs_db_tool is used to manage DAIP gateways in a DAIP database.

Usage rs_db_tool [-d] <-operation <add <-name *object_name*> <-ip *module_ip*> <-TTL *Time-To-Live*> >

rs_db_tool [-d] <-operation fetch <-name *object_name*> >

rs_db_tool [-d] <-operation <delete <-name *object_name*> >

rs_db_tool [-d] <-operation <list> >

rs_db_tool [-d] <-operation <sync> >

Syntax

Argument	**Description**
-d	debug file
-operation add	Add entry to database.
<-name *object_name*>	Enter the name of the gateway object.
<-ip *module_ip*>	Enter the IP Address of the gateway

Argument	Description
<-TTL *Time-To-Live*>	The relative time interval (in seconds) during which the entry is valid. A value of zero specifies "unlimited".
`- operation fetch`	Get entry from database.
`- operation delete`	Delete entry from database.
`- operation list`	List all the database entries.
`- operation sync`	Synchronize the database.

sam_alert

Description This tool executes FW-1 SAM (Suspicious Activity Monitoring) actions according to information received through Standard input. This tool is for executing FW-1 SAM actions with the FW-1 User Defined alerts mechanism.

Usage

```
sam_alert [-o] [-v] [-s sam_server] [-t timeout] [-f
fw_host]... [-C] -n|-i|-I -src|-dst|-any|-srv
```

Syntax

Argument	Description
`-o`	Prints the input of this tool to the standard output (for pipes).
`-v`	Turns on verbose mode (of the `fw sam` command).
`-s sam_server`	The sam server to be contacted. Localhost is the default.
`-t timeout`	The time period, in seconds, for which the action will be enforced. The default is forever.
`-f fw_host`	Identifies the FireWalls to run the operation on. Default is "all FireWalls."
`-C`	Cancels the specified operation.

Argument	Description
`-n`	Notify every time a connection that matches the specified criteria passes the FireWall.
`-i`	Inhibit connections that match the specified criteria.
`-I`	Inhibit connections that match the specified criteria and close all existing connections that match the criteria.
`-src`	Match the source address of connections.
`-dst`	Match the destination address of connections.
`-any`	Match either the source or destination address of the connection.
`-srv`	Match specific source, destination, protocol and service.

svr_webupload_config

Description

This utility is used to configure the SmartView Reporter web upload script. For the complete upload procedure and additional information refer to the section "How to upload reports to a web server" in the SmartView Reporter Administration Guide.

Usage

```
svr_webupload_config [-i perl_int_loc]
[-p rep_dir_root]
```

Syntax

Argument	Description
-i	Specifies the Perl interpreter location.
-p	Specifies the path for the reports virtual directory.

Chapter 3

VPN-1 Commands

VPN

Description This command and subcommands are used for working with various aspects of VPN-1. VPN commands executed on the command line generate status information regarding VPN processes, or are used to stop and start specific VPN services. All VPN commands are executed on the VPN-1 Power module. The vpn command sends to the standard output a list of available commands.

Usage `vpn`

Comments Sends to the standard output a list of available commands.

In This Section

vpn accel

Description This command performs operations on VPN accelerator cards (encryption only cards, not the full SecureXL cards) and VPNx. VPNx is a software module that takes advantage of multiple CPUs to accelerate VPN operations. The command comes in three flavours -- for turning the accelerator card on and off, for collecting statistics, and enabling or disabling the accelerator card or acceleration software.

Usage

```
vpn accel [-d vpnx] on|off

vpn accel [-d vpnx] stat[-l]

vpn accel -d vpnx autostart on|off
```

Syntax

Argument	Description
autostart onloff	Automatically starts/stops the vpnx accelerator software
on/off	Enable/disable accelerator card or vpnx accelerator module
stat [-l]	Reports the status of the accelerator card in long format

Example

```
vpn accel -d vpnx stat
```

Output

```
VPN-1: VPNx started
  Number of initialization errors: 0
  Number of processing errors: 0

vpn accel -d vpnx stat -l
VPN-1: VPNx started
  Number of initialization errors: 0
  Number of processing errors: 0
  Number of ESP valid contexts: 0
  Number of packets queued to the accelerator: 0
  High water mark of number of packets in queue: 1
```

Example

```
vpn accel -d vpnx stat -l
```

Output

```
VPN-1: VPNx started
  Number of initialization errors: 0
  Number of processing errors: 0

vpn accel -d vpnx stat -l
VPN-1: VPNx started
  Number of initialization errors: 0
  Number of processing errors: 0
  Number of ESP valid contexts: 0
  Number of packets queued to the accelerator: 0
  High water mark of number of packets in queue: 1



  Number of packets and bytes since last activation

-------------------------------------------------------------
-
                                Packets                 Bytes

-------------------------------------------------------------
-
    ESP decrypted                     52                 7072
    ESP encrypted                     52                 7072
    ESP total                        104                14144
    Total                            104                14144



  Average rates for the last 42.343 seconds

-------------------------------------------------------------
-
                             Packets/sec             Kbit/sec

-------------------------------------------------------------
-
    ESP decrypted                      0                 0.00
    ESP encrypted                      0                 0.00
    ESP total                          0                 0.00
    Total                              0                 0.00
```

vpn compreset

Description This command resets the compression/decompression statistics to zero.

Usage `vpn compreset`

Comments Run this command before running `vpn compstat`. This command is mostly obsolete. More compression/decompression information is available via `cpstat`.

vpn compstat

Description This command displays compression/decompression statistics

Usage

```
vpn compstat
```

Comments This command is mostly obsolete. More compression/decompression information is available via `cpstat`.

vpn crl_zap

Description This command is used to erase all Certificate Revocation Lists (CRLs) from the cache.

Usage

```
vpn crl_zap
```

Return Value 0 for success; any other value equals failure.

vpn crlview

Description This command retrieves the Certificate Revocation List (CRL) from various distribution points and displays it for the user. The command comes in three flavors:

`vpn crlview -obj <MyCA> -cert <MyCert>`. The VPN daemon contacts the Certificate Authority called **MyCA** and locates the certificate called **MyCert**. The VPN daemon extracts the certificate distribution point from the certificate then goes to the distribution point, which might be an LDAP or HTTP server. From the distribution point, the VPN daemon retrieves the CRL and displays it to the standard output.

`vpn crlview -f d:\temp\MyCert`. The VPN daemon goes to the specified directory, extracts the certificate distribution point from the certificate, goes to the distribution point, retrieves the CRL, and displays the CRL to the standard output.

`vpn crlview -view <lastest_CRL>`. If the CRL has already been retrieved, this command instructs the VPN daemon to display the contents to the standard output.

Usage

```
vpn crlview -obj <object name> -cert <certificate name>

vpn crlview -f <filename>

vpn crlview -view
```

Syntax

Argument	**Description**
-obj -cert	• -obj refers to the name of the CA network object • -cert refers to the name of the certificate
-f	Refers to the filename of the certificate
-view	Views the CRL
-d	Debug option

Return Value 0 for success; any other value equals failure.

vpn debug

Description

This command instructs the VPN daemon to write debug messages to the VPN log file: in `$FWDIR/log/vpnd.elg`. Debugging of the VPN daemon takes place according to topics and levels. A topic is a specific area on which to perform debugging, for example if the topic is LDAP, all traffic between the VPN daemon and the LDAP server are written to the log file. Levels range from 1-5, where 5 means “write all debug messages”.

This command makes use of **TdError**, a Check Point infrastructure for reporting messages and debug information. There is no legal list of topics. It depends on the application or module being debugged.

To debug all available topics, use: ALL for the debug topic.

IKE traffic can also be logged. IKE traffic is logged to `$FWDIR/log/IKE.elg`

Usage

```
Usage: vpn debug < on [ DEBUG_TOPIC=level ] | off | ikeon
| ikeoff | trunc | timeon <SECONDS>| timeoff
```

```
vpn debug on DEBUG_TOPIC=level |off timeon<SECONDS>]|timeoff

vpn debug ikeon | ikeoff timeon|timeoff

vpn debug trunc
```

Syntax

Argument	Description
on	Turns on high level vpn debugging.
on topic=level	Turns on the specified debug topic on the specified level. Log messages associated with this topic at the specified level (or higher) are sent to `$FWDIR/log/vpnd.elg`
off	Turns off all vpn debugging.
timeon/timeoff	Number of seconds to run the debug command
ikeon	Turns on IKE packet logging to: `$FWDIR/log/IKE.elg`
ikeoff	Turns of IKE logging
trunc	Truncates the `$FWDIR/log/IKE.elg` file, switches the cyclic vpnd.elg (changes the current `vpnd.elg` file to `vpnd0.elg` and creates a new `vpnd.elg`),enables vpnd and ike debugging and adds a timestamp to the `vpnd.elg` file.

Return Value 0= success, failure is some other value, typically -1 or 1.

Example `vpn debug on all=5 timeon 5.`

This writes all debugging information for all topics to the vpnd.elg file for five seconds.

Comments IKE logs are analyzed using the support utility `IKEView.exe`.

vpn drv

Description This command installs the VPN-1 kernel (vpnk) and connects to the FireWall-1 kernel (fwk), attaching the VPN-1 driver to the FireWall-1 driver.

Usage

```
vpn drv on|off

vpn drv stat
```

Syntax

Argument	Description
on/off	Starts/stops the VPN-1 kernel
stat	Returns the status of the VPN-1 kernel, whether the kernel is on or off

vpn export_p12

Description This command exports information contained in the network objects database and writes it in the PKCS#12 format to a file with the p12 extension.

Usage

```
vpn export_12 -obj <network object> -cert <certificate
object> -file <filename> -passwd <password>
```

Syntax

Argument	Description
-obj	Name of the gateway network object
-cert	Name of the certificate
-file	What the file with the p12 should be called
-passwd	Password required to open the encrypted p12 file

Return Value 0 for success; any other value equals failure.

Example

```
vpn export_p12 -obj Gateway1 -cert MyCert -file mycert.p12
-passwd kdd432
```

vpn macutil

This command is related to Remote Access VPN, specifically Office mode, generating a MAC address per remote user. This command is relevant only when allocating IP addresses via DHCP.

Remote access users in Office mode receive an IP address which is mapped to a hardware or MAC address. This command displays a generated hardware or MAC address for each name you enter.

Usage

```
vpn macutil <username>
```

Example vpn macutil John

Output

```
20-0C-EB-26-80-7D, “John”
```

vpn nssm_toplogy

Description This command generates and uploads a topology (in NSSM format) to a Nokia NSSM server for use by Nokia clients.

Usage

```
vpn nssm_topology -url <“url”> -dn <“dn”> -name <“name”>
-pass <“password”> [-action <bypass|drop>][-print_xml]
```

Syntax

Argument	**Description**
`-url`	URL of the Nokia NSSM server
`-dn`	Distinguished name of the NSSM server needed to establish an SSL connection
`-name`	Valid Login name for NSSM server
`-pass`	Valid password for NSSM server
`-action`	Specifies the action the symbian client should take if the packet is not destined for an IP address in the VPN domain. Legal options are **Bypass** (default) or **Drop**
`-print_xml`	The topology is in XLM format. This flag writes that topology to a file in XLM format.

vpn overlap_encdom

Description

This command displays all overlapping VPN domains. Some IP addresses might belong to two or more VPN domains. The command alerts for overlapping encryption domains if one or both of the following conditions exist:

- The same VPN domain is defined for both gateway
- If the gateway has multiple interfaces, and one or more of the interfaces has the same IP address and netmask.

If the gateway has multiple interfaces, and one or more of the interfaces have the same IP address and netmask

Usage

```
vpn overlap_encdom [communities | traditional]
```

Syntax

Argument	Description
Communities	With this flag, all pairs of objects with overlapping VPN domains are displayed -- but only if the objects (that represent VPN sites) are included in the same VPN community. This flag is also used if the same destination IP can be reached via more than one community.
Traditional	Default flag. All pairs of objects with overlapping VPN domains are displayed.

Example

```
vpn overlap_encdom communities
```

Output

```
c:\> vpn overlap_encdom communitie
The objects Paris and London have overlapping encryption
domains.
The overlapping domain is:
10.8.8.1 - 10.8.8.1
10.10.8.0 - 10.10.9.255
- This overlapping encryption domain generates a multiple entry
points configuration in MyIntranet and RemoteAccess
communities.
- Same destination address can be reached in more than one
community (Meshed, Star). This configuration is not supported.

The objects Paris and Chicago have overlapping encryption
domains. The overlapping domain is:
10.8.8.1 - 10.8.8.1
- Same destination address can be reached in more than one
community (MyIntranet, NewStar). This configuration is not
supported.

The objects Washington and Tokyo have overlapping encryption
domains.
The overlapping domain is:
10.12.10.68 - 10.12.10.68
10.12.12.0 - 10.12.12.127
10.12.14.0 - 10.12.14.255
- This overlapping encryption domain generates a multiple entry
points configuration in Meshed, Star and NewStar communities.
```

vpn sw_topology

Description This command downloads the topology for a SofaWare gateway.

Usage

```
vpn [-d] sw_toplogy -dir <directory> -name <name> -profile
<profile> [-filename <filename>]
```

Syntax

Argument	Description
-d	Debug flag
-dir	Output directory for file
-name	Nickname of site which appears in remote client
-profile	Name of the sofaware profile for which the topology is created
-filename	Name of the output file

vpn tu

Description This command launches the TunnelUtil tool which is used to control VPN tunnels.

Usage

```
vpn tu

vpn tunnelutil
```

Example vpn tu

Output

```
**********     Select Option     **********

(1)             List all IKE SAs
(2)             List all IPsec SAs
(3)             List all IKE SAs for a given peer
(4)             List all IPsec SAs for a given peer
(5)             Delete all IPsec SAs for a given peer
(6)             Delete all IPsec+IKE SAs for a given peer
(7)             Delete all IPsec SAs for ALL peers
(8)             Delete all IPsec+IKE SAs for ALL peers

(A)             Abort

*******************************************     vpn debug
1
In Progress ...

ALL IKE SA
----------



Peer: 194.29.40.225     Cookies
ebc5cf1c68c2925b-27cb65c1afd28bc6

Peer: 194.29.40.225     Cookies
8670f30aa0a04a30-4672a6998758071d
Hit <Enter> key to continue ...
```

Further Info. When viewing Security Associations for a specific peer, the IP address must be given in dotted decimal notation.

vpn ver

Description This command displays the VPN-1 major version number and build number.

Usage

```
vpn ver [-k] -f <filename>
```

Syntax

Argument	Description
`ver`	Displays the version name and version build number
`-k`	Displays the version name and build number and the kernel build number
`-f`	Prints the version number and build number to a text file.

Chapter **4**

SmartView Monitor Commands

RTM

Description This command and all its derivatives are used to execute SmartView Monitor operations.

In This Section

rtm debug

Description This command sends debug printouts to the $FWDIR/log/rtmd.elg file.

Usage

```
rtm debug <on | off> [OPSEC_DEBUG_LEVEL |
TDERROR_<AppName>_<Topic>=<ErrLevel>]
```

Syntax

Argument	Description
on	Start debug mode
off	Stop debug mode
OPSEC_DEBUG_LEVEL	Turn on OPSEC debug printouts
TDERROR_RTM_ALL	Turn on SmartView Monitor debug printouts

rtm drv

Description This command starts, stops or checks the status of the SmartView Monitor kernel driver.

Usage

```
rtm drv <on | off | stat>
```

Syntax

Argument	Description
on	Start the SmartView Monitor kernel driver
off	Stop the SmartView Monitor kernel driver
stat	SmartView Monitor kernel driver status

rtm monitor <module_name><interface_name> or rtm monitor <module_name>-filter

Description This command starts the monitoring process and specifies parameters for monitoring an interface.

Usage

```
rtm monitor
<module_name><interface_name>[options]-g<grouping>[entity-1.
..entity-n]
or
rtm monitor <module_name>-filter["complex
filter"][options]-g<grouping>[entity-1...entity-n]
```

Syntax

Argument	Description
-a	<aggregatelindividual>
-w	<bandwidthllosslrtt>
-t	<wirelapplication>
-i	<number of seconds>
@@	specifies subrule (for example, ‘rule@@subrule’)
default values	‘-y bytes -a aggregate -w bandwidth -i2
grouping types	svclsrcldstliplfgruleltopsvcltopsrcltopdstltopipltopfwltopfgrule
module-name	The name of the SmartView Monitor module.
interface-name	The name of the monitored interface.
-d	Specifies one of the following monitor directions: - inbound - outbound - eitherbound
inbound	Monitors the inbound direction.
outbound	Monitors the outbound direction.
eitherbound	Monitors both directions.
-y	Specifies one of the following measurement units: - bytes - pkts - line

Argument	Description
c	Indicates the number of new connections opened per second.
C	Average concurrent connections
-a	Aggregate - displays a specific type of connections as an aggregate. Individual - displays a specific type of connections as an individual. The defualt is eitherbound.
-g	Specifies one of the following grouping options for monitored traffic: - svc - src - dst - ip - fgrule - topsvc - topsrc - topdst - topdst - topfwm - topfgrule
svc	Monitors according to a service.
src	Monitors according to a network object (source only).
dst	Monitors according to a network object (destination only).
ip	Monitors according to a network object (source and destination).
fgrule	Monitors according to a QoS Policy rule.
topsvc	Monitors the traffic of the top 50 services.
topsrc	Monitors the traffic of the top 50 sources.

Argument	Description
topdst	Monitors the traffic of the top 50 destinations.
topdst	Monitors traffic to and from the top 50 IP addresses (source of destination).
topfwn	Monitors according to the top 50 FireWall-1 rules.
topfgrule	Monitors according to the top 50 QoS Policy rules.
-p	Specifies whether or not thousands will be separated by commas.
-filter	[“<complex filter>”] Only monitors traffic that matches the complex -filter boolean expression.

Example The following command line displays monitoring data in bytes-per-sec for the top 50 services passed on any interface in both directions:

```
rtm monitor localhost -filter -g topsvc
```

The following command will display monitoring data in Conncurrent-Connections for the top 50 sources passed on interface eth0, inbound (that is, not telnet of http).

```
rtm monitor localhost -filter “[and[[interface 0 [[eth0in]]][svc 1
[telnet http]]]” -y C -g topsrc
```

The default monitors all traffic on any interface in both directions.

Comments The specified entitites should correspond to the specified grouping option. For example, if the monitoring process works according to a service (svc), all of the monitored services should be listed and separated by single spaces.

When monitoring occurs according to the QoS Policy rule (fgrule), ‘rule@@subrule” should be used to specify a subrule entity.

There is no need to specify the top grouping options since they automatically monitor the top 50 entities according to the specified group.

Example The following command displays monitoring data in bytes-per-sec for the top 50 services passed on interface hme1.

```
rtm monitor localhost hme1 -g topsvc -y b
```

rtm monitor <module_name>-v<virtual_link_name>

Description This command starts the monitoring process and specifies parameters for monitoring a Virtual Link.

Usage

```
rtm monitor
<module_name>-v<virtual_link_name>[options]entity-1...entity
-n
```

Syntax

Argument	Description
module-name	The name of the SmartView Monitor module.
-virtual-link-name	The name of the monitored Virtual Link.
-d	Specifies one of the following monitoring directions: - a2b - b2a - a2b_b2a
a2b	Monitors End Point A to End Point B.
b2a	Monitors End Point B to End Point A.
a2b_b2a	Monitors both directions.
-y	Specifies one of the following measurement units. It is only required when the -w balue is bandwidth. - bytes - pkts
-w	Specifies the displayed data type.
bandwidth	Displays the effective bandwidth.
loss	Displays the difference between the transmission rate and the receiving rate.

Argument	Description
rtt	Displays the time required to make the round trip between the two End Points.
-t	Specifies the data type. It is only required when the -w value is bandwidth.
wire	Shows the data on the wire after compression or encryption.
application	Shows the data as the application sees it (that is, not compressed and not encrypted).

rtm rtmd

Description This command starts the SmartView Monitor daemon manually. This also occurs manually when rtmstart is run.

Usage

```
rtm rtmd
```

rtm stat

Description This command displays the general SmartView Monitor status. In addition, it displays the status of the daemon, driver, opened views and active virtual links.

Usage

```
rtm stat [flavor(s)] [-h] [-v[v][v]]
```

Syntax

Argument	Description
-h	Help
-v	Verbose
vl	Current virtual links
view	Current views

rtm ver

Description This command displays the SmartView Monitor version.

Usage

```
rtm ver [-k]
```

Syntax

Argument	Description
-k	Displays the SmartView Monitor kernel version.

rtmstart

Description This command loads the SmartView Monitor kernel module and starts the SmartvView Monitor daemon.

Usage

```
rtmstart
```

rtmstop

Description This command kills the SmartView Monitor daemon and unloads the SmartView Monitor kernel module.

Usage

```
rtmstop
```

Chapter 5

SecureClient Commands

SCC

Description

VPN commands executed on SecureClient are used to generate status information, stop and start services, or connect to defined sites using specific user profiles. Typically, a SecureClient does not need to open a command prompt and use these commands, but the site administrator may wish to include them in a script which is then transferred to remote users. In this way, the SecureClient CLI exposes SecureClient operations (such as Connect/Disconnect) to external third party applications via scripting.

The general format for SecureClient commands is:

```
C:\> scc <command> [optional arguments]
```

Some of the commands have keyboard shortcuts. Some of the commands require you to be in command line mode. Use the `setmode` command for switching to command line mode. Once in CLI mode, the system tray SecureClient icon is disabled.

Return Value

All the scc commands return 0 on success and (-1) on error. Any textual output goes to stdout on success (for example:'scc numprofiles'), and any error string goes to stderr.

In This Section

scc connect

Description This command connects to the site using the specified profile, and waits for the connection to be established. In other words, the OS does not put this command into the background and executes the next command in the queue.

Usage

```
connect [-p] <profilename>
```

Syntax

Argument	**Description**
-p	Displays connection progress

Comments Shortcut: `scc c`

You must be in CLI mode to run this command.

scc connectnowait

Description This command connects asynchronously to the site using the specified profile. This means, the OS moves onto the next command in the queue and this command is run in the background.

Usage `connectnowait <profilename>`

Comments Shortcut: `scc cn`

You must be in CLI mode to run this command.

scc disconnect

Description This command disconnects from the site using a specific profile.

Usage `scc disconnect -p <profilename>`

Syntax

Argument	Description
-p	Displays disconnect progress

Comments Shortcut: `scc d`

You must be in CLI mode to run this command.

scc erasecreds

Description This command unsets authorization credentials

Usage `scc ersecreds`

Comments Shortcut: `scc ep`

You need to be in CLI mode to run this command.

scc listprofiles

Description This command lists all profiles

Usage `scc listprofiles`

Comments Shortcut: `scc lp`

You must be in CLI mode to run this command.

scc numprofiles

Description This command displays the number of profiles.

Usage

```
scc numprofiles
```

Comments Shortcut: `scc np`

You need to be in CLI mode to run this command.

scc restartsc

Description This command restarts SecureClient services.

Usage

```
scc restartsc
```

Comments You need administrator privileges to run this command.

scc passcert

Description This command sets the user's authentication credentials when authentication is performed using certificates.

Usage

```
scc passcert <certificate> <password>
```

Comments Shortcut: `scc pc`

You need to be in CLI mode to run this command.

scc setmode <mode>

Description This command switches the SecuRemote/SecureClient mode

Usage

```
scc setmode [-cli | -con]
```

Syntax

Argument	Description
-cli	command line interface mode
-con	connect mode

Comments You need administrator privileges to run this command.

scc setpolicy

Description This command enables or disables the current default security policy.

Usage

```
scc setpolicy [on|off]
```

Comments Shortcut: `scc sp`

You need administrator privileges to run this command.

scc sp

Description This command displays the current default security policy.

Usage

```
scc sp
```

Comments You need to be in CLI mode to run this command.

scc startsc

Description This command starts SecureClient services.

Usage

```
scc startsc
```

Comments You need administrator privileges to run this command.

scc status

Description This is command displays the connection status.

Usage

```
scc status
```

Comments Shortcut: `scc s`

scc stopsc

Description This command stops SecureClient services.

Usage

```
scc stopsc
```

Comments You need administrator privileges to run this command.

scc suppressdialogs

Description This command enables or suppresses dialog popups. By default, suppressdialogs is off.

Usage

```
scc suppressdialogs [on|off]
```

Comments When using `suppressdialogs on`, only popups requesting authentication credentials appear.

Shortcut: `scc sd`

You need to be in CLI mode to run this command.

scc userpass

Description This commands sets the user's authentication credentials -- username, and password.

Usage

```
scc userpass <username> <password>
```

Comments Shortcut `scc up`

You need to be in CLI mode to run this command.

scc ver

Description This command displays the current SecureClient version

Usage

```
scc ver
```

Chapter 6

ClusterXL Commands

In This Chapter

cphaconf

Description The cphaconf command configures ClusterXL.

Warning - Running this command is not recommended. It should be run automatically, only by Firewall-1 or by Check Point support. The only exception to this rule is running this command with `set_cpp` option, as described below.

Usage

```
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
```

Syntax

Argument	Description
cphaconf set_ccp <broadcast/multicast>	Sets whether Cluster Control Protocol (CCP) packets should be sent with a broadcast or multicast destination MAC address. The default behavior is multicast. The setting created using this command will survive reboot. Note, the same value (either broadcast or multicast) should be set on all cluster members.

cphaprob

Description The `cphaprob` command verifies that the cluster and the cluster members are working properly.

Usage

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

Syntax

Argument	Description
cphaprob -d <device> -t <timeout(sec)> -s <oklinitlproblem> [-p] register	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.
cphaprob -f <file> register	Register all the user defined critical devices listed in <file>.
cphaprob -d <device> [-p] unregister	Unregister a user defined <device> as a critical process. This means that this device is no longer considered critical.

Argument	Description
cphaprob -a unregister	Unregister all the user defined <device>.
cphaprob -d <device> -s <ok|init|problem> report	Report the status of a user defined critical device to ClusterXL.
cphaprob [-i[a]] [-e] list	View the list of critical devices on a cluster member, and of all the other machines in the cluster.
cphaprob state	View the status of a cluster member, and of all the other members of the cluster..
cphaprob [-a] if	View the state of the cluster member interfaces and the virtual cluster interfaces.

cphastart

Description

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.

cphastop

Description

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.

THIRD PARTY TRADEMARKS AND COPYRIGHTS

Entrust is a registered trademark of Entrust Technologies, Inc. in the United States and other countries. Entrust's logos and Entrust product and service names are also trademarks of Entrust Technologies, Inc. Entrust Technologies Limited is a wholly owned subsidiary of Entrust Technologies, Inc. FireWall-1 and SecuRemote incorporate certificate management technology from Entrust.

Verisign is a trademark of Verisign Inc.

The following statements refer to those portions of the software copyrighted by University of Michigan. Portions of the software copyright © 1992-1996 Regents of the University of Michigan. All rights reserved. Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "as is" without express or implied warranty. Copyright © Sax Software (terminal emulation only).

The following statements refer to those portions of the software copyrighted by Carnegie Mellon University.

Copyright 1997 by Carnegie Mellon University. All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The following statements refer to those portions of the software copyrighted by The Open Group.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following statements refer to those portions of the software copyrighted by The OpenSSL Project. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following statements refer to those portions of the software copyrighted by Eric Young. THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Copyright © 1998 The Open Group.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

The PHP License, version 3.0

Copyright (c) 1999 - 2004 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.

4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"

5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes PHP, freely available from <http://www.php.net/>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the PHP Group. The PHP Group can be contacted via Email at group@php.net.

For more information on the PHP Group and the PHP project, please see <http://www.php.net>. This product includes the Zend Engine, freely available at <http://www.zend.com>.

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Copyright (c) 2003, Itai Tzur <itzur@actcom.co.il>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Neither the name of Itai Tzur nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright © 2003, 2004 NextHop Technologies, Inc. All rights reserved.

Confidential Copyright Notice

Except as stated herein, none of the material provided as a part of this document may be copied, reproduced, distrib-uted, republished, downloaded, displayed, posted or transmitted in any form or by any means, including, but not lim-ited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of NextHop Technologies, Inc. Permission is granted to display, copy, distribute and download the materials in this doc-ument for personal, non-commercial use only, provided you do not modify the materials and that you retain all copy-right and other proprietary notices contained in the materials unless otherwise stated. No material contained in this document may be "mirrored" on any server without written permission of NextHop. Any unauthorized use of any material contained in this document may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes. Permission terminates automatically if any of these terms or condi-tions are breached. Upon termination, any downloaded and printed materials must be immediately destroyed.

Trademark Notice

The trademarks, service marks, and logos (the "Trademarks") used and displayed in this document are registered and unregistered Trademarks of NextHop in the US and/or other countries. The names of actual companies and products mentioned herein may be Trademarks of their respective owners. Nothing in this document should be construed as granting, by implication, estoppel, or otherwise, any license or right to use any Trademark displayed in the document. The owners aggressively enforce their intellectual property rights to the fullest extent of the law. The Trademarks may not be used in any way, including in advertising or publicity pertaining to distribution of, or access to, materials in

this document, including use, without prior, written permission. Use of Trademarks as a "hot" link to any website is prohibited unless establishment of such a link is approved in advance in writing. Any questions concerning the use of these Trademarks should be referred to NextHop at U.S. +1 734 222 1600.

U.S. Government Restricted Rights

The material in document is provided with "RESTRICTED RIGHTS." Software and accompanying documentation are provided to the U.S. government ("Government") in a transaction subject to the Federal Acquisition Regulations with Restricted Rights. The Government's rights to use, modify, reproduce, release, perform, display or disclose are

restricted by paragraph (b)(3) of the Rights in Noncommercial Computer Software and Noncommercial Computer Soft-ware Documentation clause at DFAR 252.227-7014 (Jun 1995), and the other restrictions and terms in paragraph (g)(3)(i) of Rights in Data-General clause at FAR 52.227-14, Alternative III (Jun 87) and paragraph (c)(2) of the Commer-cial

Computer Software-Restricted Rights clause at FAR 52.227-19 (Jun 1987).

Use of the material in this document by the Government constitutes acknowledgment of NextHop's proprietary rights in them, or that of the original creator. The Contractor/Licensor is NextHop located at 1911 Landings Drive, Mountain View, California 94043. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in applicable laws and regulations.

Disclaimer Warranty Disclaimer Warranty Disclaimer Warranty Disclaimer Warranty

THE MATERIAL IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND EITHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT POSSIBLE PURSUANT TO THE APPLICABLE LAW, NEXTHOP DISCLAIMS ALL WARRANTIES,

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON INFRINGEMENT OR OTHER VIOLATION OF RIGHTS. NEITHER NEXTHOP NOR ANY OTHER PROVIDER OR DEVELOPER OF MATERIAL CONTAINED IN THIS DOCUMENT WARRANTS OR MAKES ANY REPRESEN-TATIONS REGARDING THE USE, VALIDITY, ACCURACY, OR RELIABILITY OF, OR THE RESULTS OF THE USE OF, OR OTHERWISE RESPECTING, THE MATERIAL IN THIS DOCUMENT.

Limitation of Liability

UNDER NO CIRCUMSTANCES SHALL NEXTHOP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, BUT NOT LIMITED TO, LOSS OF DATA OR PROFIT, ARISING OUT OF THE USE, OR THE INABILITY TO USE, THE MATERIAL IN THIS DOCUMENT, EVEN IF NEXTHOP OR A NEXTHOP AUTHORIZED REPRESENTATIVE HAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IF YOUR USE OF MATERIAL FROM THIS DOCUMENT RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ANY COSTS THEREOF. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT FULLY APPLY TO YOU.

PCRE LICENCE

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language. Release 5 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

Written by: Philip Hazel <ph10@cam.ac.uk>

University of Cambridge Computing Service, Cambridge, England. Phone:

+44 1223 334714.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Eventia Reporter includes software whose copyright is owned by, or licensed from, MySQL AB.