

Command Line Interface

R71

Reference Guide

4 December 2011



More Information

The latest version of this document is at:

http://supportcontent.checkpoint.com/documentation_download?ID=10324

For additional technical information about Check Point visit Check Point Support Center (<http://supportcenter.checkpoint.com>).

Feedback

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

Please help us by sending your comments

(mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Command Line Interface R71 Reference Guide).

© 2011 Check Point Software Technologies Ltd.

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

RESTRICTED RIGHTS LEGEND:

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

TRADEMARKS:

Please refer to our Copyright page (<http://www.checkpoint.com/copyright.html>) for a list of our trademarks.

Please refer to our Third Party copyright notices (http://www.checkpoint.com/3rd_party_copyright.html) for a list of relevant copyrights.

Contents

Security Management Server and Firewall Commands	7
comp_init_policy	8
cp_admin_convert	8
cpca_client	8
cpca_client create_cert	8
cpca_client revoke_cert	9
cpca_client lscert	9
cpca_client set_mgmt_tools	9
cp_conf	10
cp_conf sic	10
cp_conf admin	10
cp_conf ca	10
cp_conf finger	11
cp_conf lic	11
cp_conf client	11
cp_conf ha	11
cp_conf snmp	11
cp_conf auto	11
cp_conf sxl	11
cpconfig	12
cpinfo	12
cplic	13
cplic check	13
cplic db_add	14
cplic db_print	14
cplic db_rm	15
cplic del	15
cplic del <object name>	16
cplic get	16
cplic put	17
cplic put <object name>	18
cplic print	18
cplic upgrade	19
cp_merge	20
cp_merge delete_policy	20
cp_merge export_policy	21
cp_merge import_policy and cp_merge restore_policy	22
cp_merge list_policy	23
cppkg	23
cppkg add	23
cppkg delete	24
cppkg get	24
cppkg getroot	25
cppkg print	25
cppkg setroot	25
cpridrestart	26
cpridstart	26
cpridstop	26
cprinstall	26
cprinstall boot	26
cprinstall cpstart	27
cprinstall cpstop	27

cprinstall get	27
cprinstall install	28
cprinstall uninstall	29
cprinstall verify	30
cprinstall snapshot	31
cprinstall show	31
cprinstall revert	31
cprinstall transfer	31
cpstart	32
cpstat	32
cpstop.....	34
cpwd_admin	34
cpwd_admin start.....	34
cpwd_admin stop.....	35
cpwd_admin list	35
cpwd_admin exist	36
cpwd_admin kill	36
cpwd_admin config.....	36
dbedit	37
dbver	40
dbver create.....	40
dbver export.....	40
dbver import.....	41
dbver print.....	41
dbver print_all	42
dynamic_objects.....	42
fw	42
fw -i.....	42
fw ctl	43
fw ctl debug	44
fw ctl affinity	45
fw ctl engine.....	47
fw ctl multik stat	47
fw ctl sdstat.....	47
fw fetch	49
fw fetchlogs.....	49
fw hastat	50
fw isp_link	50
fw kill.....	51
fw lea_notify.....	51
fw lichosts	51
fw log	51
fw logswitch	54
fw mergefiles	55
fw monitor	55
fw lslogs.....	59
fw putkey	60
fw repairlog	60
fw sam	61
fw stat	64
fw tab.....	65
fw ver.....	66
fwm	66
fwm dbimport	66
fwm expdate	68
fwm dbexport	68
fwm dbload	69
fwm ikecrypt.....	70
fwm load	70

fwm lock_admin	70
fwm logexport	71
fwm sic_reset	72
fwm unload <targets>	72
fwm ver	73
fwm verify <policy-name>	73
GeneratorApp	73
inet_alert	73
ldapcmd	75
ldapcompare	76
ldapconvert	76
ldapmodify	79
ldapsearch	79
log_export	80
queryDB_util	83
rs_db_tool	84
sam_alert	85
svr_webupload_config	86
VPN Commands	87
VPN	87
vpn accel	87
vpn compreset	88
vpn compstat	88
vpn crl_zap	89
vpn crlview	89
vpn debug	89
vpn drv	90
vpn export_p12	90
vpn macutil	91
vpn nssm_toplogy	91
vpn overlap_encdom	92
vpn sw_topology	93
vpn tu	93
vpn ver	94
SmartView Monitor Commands	95
RTM	95
rtm debug	95
rtm drv	95
rtm monitor <module_name><interface_name> or rtm monitor <module_name>-filter	96
rtm monitor <module_name>-v<virtual_link_name>	98
rtm rtmd	99
rtm stat	99
rtm ver	99
rtmstart	99
rtmstop	99
SecureClient Commands	100
SCC	100
scc connect	100
scc connectnowait	100
scc disconnect	100
scc erasecreds	101
scc listprofiles	101
scc numprofiles	101
scc restartsc	101
scc passcert	101
scc setmode <mode>	101
scc setpolicy	102
scc sp	102

scc startsc.....	102
scc status.....	102
scc stopsc.....	102
scc suppressdialogs.....	102
scc userpass.....	103
scc ver	103
ClusterXL Commands	104
cphaconf.....	104
cphaprob	105
cphastart	105
cphastop.....	105
Debugging SmartConsole Clients	106
CLI for Other Products	107
CLI Commands in Other Guides.....	107
Index	109

Chapter 1

Security Management Server and Firewall Commands

In This Chapter

comp_init_policy	8
cp_admin_convert	8
cpca_client	8
cp_conf	10
cpconfig	12
cpinfo	12
cplic	13
cp_merge	20
cppkg	23
cpridrestart	26
cpridstart	26
cpridstop	26
cprinstall	26
cpstart	32
cpstat	32
cpstop	34
cpwd_admin	34
dbedit	37
dbver	40
dynamic_objects	42
fw	42
fwm	66
GeneratorApp	73
inet_alert	73
ldapcmd	75
ldapcompare	76
ldapconvert	76
ldapmodify	79
ldapsearch	79
log_export	80
queryDB_util	83
rs_db_tool	84
sam_alert	85
svr_webupload_config	86

comp_init_policy

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

The Initial Policy offers protection to the gateway before the administrator has installed a Policy on the gateway.

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

cp_admin_convert

Description Automatically export administrator definitions that were created in `cpconfig` to SmartDashboard.

Usage `cp_admin_convert`

cpca_client

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

Usage `cpca_client`

cpca_client create_cert

Description Prompt the ICA to issue a SIC certificate for the Security Management server.

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

Syntax

Argument	Description
-d	Debug flag

Argument	Description
-p <ca_port>	Specifies the port 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 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 lscert

Description Show all certificates issued by the ICA.

Usage `cpca_client [-d] lscert [-dn substr] [-stat Pending|Valid|Revoked|Expired|Renewed] [-kind SIC|IKE|User|LDAP] [-ser ser] [-dp dp]`

Syntax

Argument	Description
-d	Debug flag
-dn substring	Filters results to those with a DN that matches this substring
-stat	Filters results to this status
-kind	Filters results for specified kind: SIC, IKE, User, or LDAP
-ser number	Filters results for this serial number
-dp number	Filters results from this CDP

cpca_client set_mgmt_tools

Description 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
<code>-d</code>	Debug flag
<code>set_mgmt_tools on off</code>	<ul style="list-style-type: none"> <code>on</code> - Start ICA Management tool <code>off</code> - Stop ICA Management tool
<code>-p <ca_port></code>	Specifies the port which is used to connect to the CA (if the appropriate service was not run from the default port 18265)
<code>-no_ssl</code>	Configures the server to use clear http rather than https
<code>-a -u"administrator user DN"</code>	Sets the DNs of the administrators or user permitted to use the ICA Management tool

Comments

1. If the command is run 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. After the SSL mode has been modified, the server can be stopped and restarted.

cp_conf

Description Configure/reconfigure a Security Gateway installation. The configuration available options for any machine depend on the installed configuration and products.

Usage `cp_conf`

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 <Security Management server name/IP> <module object name>`
`# Pull certificate (DAIP only)`

cp_conf admin

Description Manage Check Point Administrators.

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 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 Security Management server being accessed by the SmartConsole. This fingerprint is a text string derived from the Security Management server's certificate

Usage cp_conf finger get # Get Certificate's Fingerprint.

cp_conf lic

Description Enables 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 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 Enable or disable High Availability.

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

cp_conf snmp

Description 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 Determine whether or not the Security Gateway/Security Management server 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 Enable or disable SecureXL acceleration.

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

cpconfig

Description Run a command line version of the Check Point Configuration Tool. This tool is used to configure an installed Check Point product. The options shown depend on the installed configuration and products. Amongst others, these options include:

- **Licenses and contracts** - Modify the necessary Check Point licenses and contracts.
- **Administrator** - Modify the administrator authorized to connect to the Security Management server.
- **GUI Clients** - Modify the list of SmartConsole Client machines from which the administrators are authorized to connect to a Security Management server.
- **SNMP Extension** - Configure the SNMP daemon. The SNMP daemon enables SecurePlatform to export its status to external network management tools.
- **PKCS #11 Token** - Register a cryptographic token, for use by SecurePlatform; see details of the token, and test its functionality.
- **Random Pool** - Configure the RSA keys, to be used by SecurePlatform.
- **Certificate Authority** - Install the Certificate Authority on the Security Management server in a first-time installation.
- **Secure Internal Communication** - Set up trust between the gateway on which this command is being run and the Security Management server.
- **Certificate's Fingerprint** - Display the fingerprint which will be used on first-time launch to verify the identity of the Security Management server being accessed by the SmartConsole. This fingerprint is a text string derived from the Security Management server's certificate.
- **Automatic Start of Check Point Products** - Specify whether Check Point Security Gateways will start automatically at boot time.

Usage cpconfig

Further Info. See the *R75.20 Installation and Upgrade Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12269).

cpinfo

Description - CPinfo is a utility that collects data on a machine at the time of execution. The CPinfo output file enables Check Point's support engineers to analyze setups from a remote location. Engineers can open the CPinfo file in demo mode, while viewing real Security Policies and objects. This allows for in-depth analysis of all of configuration options and environment settings.

Usage - cpinfo [-v] [-l] [-n] [-o] [-r | -t [tablename]] [-c Domain Management Server ... | -x vs]

Syntax

Argument	Description
-z	Output gzipped (effective with -o option)
-r	Includes the registry (Windows - very large output)
-v	Prints version information
-l	Embeds log records (very large output)
-n	Does not resolve network addresses (faster)
-o	Output to a file and to the screen
-t	Output consists of tables only (SR only)

Argument	Description
-c	Get information about the specified Domain Management Server (Multi-Domain Security Management)
-x	Get information about the specified VS (VSX)

Further Info. SecureKnowledge solution sk30567
<http://supportcontent.checkpoint.com/solutions?id=sk30567>

cplic

Description This command and all its derivatives relate to Check Point license management.



Note - The SmartUpdate GUI is the recommended way of managing licenses.

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 Security Management server.
- *License repository commands* are executed on the Security Management server.

Usage `cplic`

cplic check

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

Argument	Description
<feature>	<feature> for which license information is requested

cplic db_add

Description Used to add one or more licenses to the license repository on the Security Management 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.

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

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

Syntax

Argument	Description
-l license-file	Adds the license(s) from <code>license-file</code> . The following options are NOT needed: Host Expiration-Date Signature SKU/feature

Comments **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 Displays the details of Check Point licenses stored in the license repository on the Security Management 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 Security 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 Security Management server.

cplic db_rm

Description The `cplic db_rm` command removes a license from the license repository on the Security Management 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 Security Management server.

cplic del

Description Delete a single Check Point license on a host, including unwanted evaluation, expired, and other licenses. 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 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 Security Management server.

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

Syntax

Argument	Description
object name	The name of the Check Point Security Gateway object, as defined in SmartDashboard.
-F outputfile	Divert the output to <code>outputfile</code> rather than to the screen.
-ip dynamic ip	Delete the license on the Check Point Security Gateway with the specified IP address. This parameter is used for deleting a license on a DAIP Check Point Security 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 Security Management server.

cplic get

Description The `cplic get` command retrieves all licenses from a Check Point Security Gateway (or from all Check Point gateways) into the license repository on the Security Management 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 Security Gateway from which licenses are to be retrieved.
hostname	The name of the Check Point Security 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 Security 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 Security Management server.

cplic put

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

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.

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

- `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 Security Gateway object, as defined in SmartDashboard.
-ip dynamic ip	Install the license on the Check Point Security 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 <code>outputfile</code> rather than to the screen.
-l license-file	Installs the license(s) from <code>license-file</code> . 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 Security Management 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.

Argument	Description
-F <outputfile>	Divert the output to <code>outputfile</code> .
-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 <code><inputfile></code>

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

- Upgrade the Security Management server to the latest version.
 - Ensure that there is connectivity between the Security Management server and the remote workstations with the previous version products.
- Import all licenses into the license repository. This can also be done *after* upgrading the products on the remote gateways.
- 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      CFW-FIG-25-41      CK-
49C3A3CC7121  golda
192.168.5.11  26Nov2002 CPSUITE-EVAL-3DES-NG CK-
1234567890    count

```

- In the *User Center* (<http://usercenter.checkpoint.com>), 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, 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 Security Management server.

Further Info. See the *SmartUpdate* chapter of the *R75.20 Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12277).

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 the Security Management server database.

Usage `cp_merge help`

Syntax

Argument	Description
help	Displays the usage for <code>cp_merge</code> .

cp_merge delete_policy

Description 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 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
-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 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
<code>-s <db server></code>	Specify the database server IP address or DNS name.2
<code>-u <user></code>	The administrator's name.1,2
<code>-c <certificate file></code>	The path to the certificate file.1
<code>-p <password></code>	The administrator's password.1,2
<code>-n <package name></code>	Rename the policy package to <package name> when importing.2
<code>-d <input directory></code>	Specify the input directory.2
<code>-f <inputfile></code>	Specify the input file name.
<code>-v</code>	Override an existing policy if found.2

Comments Further considerations

1. Either use certificate file or user and password
2. Optional

The `cp_mergerestore_policy` works only locally on the Security Management server and it will not work from remote machines.

Caution: A Security 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
<code>-s <db server></code>	Specify the database server IP Address or DNS name.2
<code>-u <user></code>	The administrator's name.1,2
<code>-c <certificate file></code>	The path to the certificate file.1,2
<code>-p <password></code>	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 Manage the product repository. It is always executed on the Security Management server.

cppkg add

Description 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 (<http://www.checkpoint.com/techsupport/downloads/downloads.html>). The package file can be added to the Repository directly from the DVD or from a local or network drive.

Usage `cppkg add <package-full-path | CD drive [product]>`

Syntax

Argument	Description
<code>package-full-path</code>	If the package to be added to the repository is on a local disk or network drive, type the full path to the package.
<code>CD drive</code>	If the package to be added to the repository is on a DVD: For Windows machines type the DVD drive letter, e.g. d:\ For UNIX machines, type the DVD root path, e.g. /caruso/image/CPsuite-R75.20 You are 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-R75.20\
Enter package name:
-----
(1) SVNfoundation
(2) firewall
(3) floodgate
(4) rtm

(e) Exit
Enter your choice : 1
Enter package OS :
-----
(1) win32
(2) solaris
(3) linux
(4) ipso

(e) Exit
Enter your choice : 1
You choose to add 'SVNfoundation' for 'win32' OS. Is this
correct? [y/n] : y
```

cppkg delete

Description 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.
version	Package version .
os	Package Operating System. Options are: win32, solaris, ipso, linux.
sp	Package minor version. This parameter is optional.

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

cppkg get

Description Synchronizes the Package Repository database with the content of the actual package repository under `$SUROOT`.

Usage `cppkg get`

cppkg getroot

Description 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 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`

cppkg setroot

Description 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 Security Management 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
<code>repository-root-directory-full-path</code>	The desired location for the product repository.

Comments It is important to reboot the Security Management 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. 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. 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. 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 Security Management server, `cprinstall` commands require licenses for SmartUpdate

On the remote Check Point gateways the following are required:

- Trust must be established between the Security Management server and the Check Point gateway.
- `cpd` must run.
- `cprid` remote installation daemon must run.

cprinstall boot

Description Boot the remote computer.

Usage `cprinstall boot <Object name>`

Syntax

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

Example # cprinstall boot harlin

cprinstall cpstart

Description Enable `cpstart` to be run remotely.

All products on the Check Point Security Gateway must be of the same version.

Usage `cprinstall cpstart <object name>`

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

cprinstall cpstop

Description Enables `cpstop` to be run remotely.

All products on the Check Point Security Gateway must be of the same version.

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

Syntax

Argument	Description
Object name	Object name of the Check Point Security 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 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 Security Gateway object defined in SmartDashboard.

Example

```

cprinstall get gw1
Checking cprid connection...
Verified
Operation completed successfully
Updating machine information...
Update successfully completed
'Get Gateway Data' completed successfully
Operating system   Major Version      Minor Version
-----
SecurePlatform    R75.20             R75.20

Vendor             Product            Major Version      Minor Version
-----
Check Point        VPN-1 Power/UTM   R75.20             R75.20
Check Point        SecurePlatform    R75.20             R75.20
Check Point        SmartPortal       R75.20             R75.20

```

cprinstall install

Description 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
<code>-boot</code>	Boot the remote computer after installing the package. Only boot after ALL products have the same version. Boot will be cancelled in certain scenarios. See the Release Notes for details.
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
vendor	Package vendor (e.g. <code>checkpoint</code>)
product	Package name
version	Package version
sp	Package minor version

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
R70

Installing firewall R75.20 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 uninstall

Description 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. Boot will be cancelled in certain scenarios. See the Release Notes for details.
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
vendor	Package vendor (e.g. checkpoint)
product	Package name
version	Package version
sp	Package minor version.

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 Security Gateway data by running `cprinstall get`.

Example

```
# cprinstall uninstall fred checkpoint firewall R75.20

Uninstalling firewall R75.20 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 verify

Description 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 Security Gateway defined in SmartDashboard.
vendor	Package vendor (e.g. checkpoint).
product	Package name Options are: SVNfoundation, firewall, floodgate.
version	Package version .
sp	Package minor version. This parameter is optional.

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

Verify succeeds:

```
cprinstall verify harlin checkpoint SVNfoundation R75.20

Verifying installation of SVNfoundation R75.20 on jimmy...
Info : Testing Check Point Gateway.
Info : Test completed successfully.
Info : Installation Verified, The product can be installed.
```

Verify fails:

```
cprinstall verify harlin checkpoint SVNfoundation R75.20

Verifying installation of SVNfoundation R75.20 on jimmy...
Info : Testing Check Point Gateway
Info : SVN Foundation R70 is already installed on
192.168.5.134
Operation Success. Product cannot be installed, did not pass
dependency check.
```

cprinstall snapshot

Description Creates a snapshot <filename> on the Check Point Security Gateway.

Usage `cprinstall snapshot <object name> <filename>`

Syntax

Argument	Description
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
filename	Name of the snapshot file.

Comments Supported on SecurePlatform only.

cprinstall show

Description Displays all snapshot (backup) files on the Check Point Security Gateway.

Usage `cprinstall show <object name>`

Syntax

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

Comments Supported on SecurePlatform only.

Example

```
# cprinstall show GW1
SU_backup.tzg
```

cprinstall revert

Description Restores the Check Point Security Gateway from a snapshot.

Usage `cprinstall revert <object name> <filename>`

Syntax

Argument	Description
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
filename	Name of the snapshot file.

Comments Supported on SecurePlatform only.

cprinstall transfer

Description Transfers a package from the repository to a Check Point Security Gateway without installing the package.

Usage `cprinstall transfer <object name> <vendor> <product> <version> <sp>`

Syntax

Argument	Description
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
vendor	Package vendor (e.g. checkpoint).
product	Package name
version	Package version .
sp	Package minor version. This parameter is optional.

cpstart

Description 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] [-s SICname] [-f flavor] [-o polling] [-c count] [-e period] [-d] application_flag`

Syntax

Argument	Description
-h host	A resolvable hostname, a dot-notation address (for example:192.168.33.23), or a DAIP object name. The default is localhost.
-p port	Port number of the AMON server. The default is the standard AMON port (18192).
-s	Secure Internal Communication (SIC) name of the AMON server.
-f flavor	The flavor of the output (as it appears in the configuration file). The default is the first flavor found in the configuration file.
-o	Polling interval (seconds) specifies the pace of the results. The default is 0, meaning the results are shown only once.
-c	Specifies how many times the results are shown. The default is 0, meaning the results are repeatedly shown.
-e	Specifies the interval (seconds) over which 'statistical' olds are computed. Ignored for regular olds.

Argument	Description
-d	Debug mode.
application_flag	<p>One of the following:</p> <ul style="list-style-type: none"> • fw — Firewall component of the Security Gateway • vpn — VPN component of the Security Gateway • fg — QoS (formerly FloodGate-1) • ha — ClusterXL (High Availability) • os — OS Status • mg — for the Security Management server • persistency - for historical status values • polsrv • uas • svr • cpsemd • cpsead • asm • ls • ca

The following flavors can be added to the application flags:

- fw — "default", "interfaces", "all", "policy", "perf", "hmem", "kmem", "inspect", "cookies", "chains", "fragments", "totals", "ufp", "http", "ftp", "telnet", "rlogin", "smtp", "pop3", "sync"
- vpn — "default", "product", "IKE", "ipsec", "traffic", "compression", "accelerator", "nic", "statistics", "watermarks", "all"
- fg — "all"
- ha — "default", "all"
- os — "default", "ifconfig", "routing", "memory", "old_memory", "cpu", "disk", "perf", "multi_cpu", "multi_disk", "all", "average_cpu", "average_memory", "statistics"
- mg — "default"
- persistency — "product", "Tableconfig", "SourceConfig"
- polsrv — "default", "all"
- uas — "default"
- svr — "default"
- cpsemd — "default"
- cpsead — "default"
- asm — "default", "WS"
- ls — "default"
- ca — "default", "crl", "cert", "user", "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 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 `cpnid`. `cpnid` 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`.

`fwd` does not work in a Security Management Only machine. To work with `fwd` in a Security 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`

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

Example To start and monitor the `fw` process.

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

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

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

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

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

cpwd_admin list

Description 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 Check whether cpwd is alive.

Usage cpwd_admin exist

cpwd_admin kill

Description Kill cpwd.

Usage cpwd_admin kill

cpwd_admin config

Description 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.

Argument	Description
sleep_mode	<ul style="list-style-type: none"> 1 - wait timeout 0 - ignore timeout. Rerun the process immediately
dbg_mode	<ul style="list-style-type: none"> 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).
rerun_mode	<ul style="list-style-type: none"> 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_adminconfig -d` have no effect if `cpwd` is running. They will affect `cpwd` the next time it is run.

dbedit

Description Edit the objects file on the Security Management server. Editing the `objects.C` file on the gateway is not 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
<code>-s server</code>	The Security Management server on which the <code>objects_5_0.C</code> 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.

Argument	Description
<code>-u user </code> <code>-c certificate</code>	The user's name (the name used for the SmartConsole) or the full path to the certificate file.
<code>-p password</code>	The user's password (the password used for the SmartConsole).
<code>-f filename</code>	The name of the file containing the commands. If <i>filename</i> is not given, then the user will be prompted for commands.
<code>-r db-open-reason</code>	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.
<code>-help</code>	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 <code>update</code> or <code>quit</code> command.
modify [table_name] [object_name] [field_name] [value]	Modify fields of an object which is: <ul style="list-style-type: none"> stored in the database (the command will lock the object in such case). newly created by <code>dbedit</code> Extended Formats for owned objects can be used: For example, <code>[field_name] = Field_A:Field_B</code>
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.
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. <code>rename network_objects london chicago</code>
quit	Quit <code>dbedit</code> 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
<code>version_name</code>	the name of the revision
<code>version_comment</code>	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	<ul style="list-style-type: none"> • <code>delete</code> removes the revision from the revisions repository. • <code>keep</code> 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 2009
Version Name: save
Created by Administrator: jbrown
Major Version: R75.20
Minor Version: R75.20
```

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

-s Description `dynamic_objects` specifies an IP address to which the dynamic object will be resolved on this machine. First, define the dynamic object in the SmartDashboard. Then create the same object with the CLI (`-n` argument). After the new object is created on the gateway with the CLI, you can use the `dynamic_objects` command to specify an IP address for the object.

This command cannot be executed when the Check Point gateway is running.

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

Syntax

Argument	Description
<code>-o <object_name></code>	The name of the object, as defined in SmartDashboard and the <code>dynamic_objects -n <name></code> command.
<code>-r [fromIP toIP] ...</code>	address ranges — one or more "from IP address to IP address" pairs
<code>-a [fromIP toIP] ...</code>	add ranges to object
<code>-d [fromIP toIP] ...</code>	delete range from object
<code>-l</code>	list dynamic objects
<code>-n <object_name></code>	create new object (if Security Gateway is not running)
<code>-c</code>	compare the objects in the dynamic objects file and in <code>objects.C</code> .
<code>-do object_name</code>	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 the firewall. All `fw` commands are executed on the Check Point Security gateway.

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

Usage `fw`

fw -i

Description Generally, when Check Point Security gateway commands are executed on a Security gateway they will relate to the gateway as a whole, rather than to an individual kernel instance. For example, the `fw tab` command will enable viewing or editing of a single table of information aggregated for all kernel instances.

This command specifies that certain commands apply to an individual kernel instance. By adding `-i <kern>` after `fw` in the command, where `<kern>` is the kernel instance's number.

Usage `fw -i` applies to the following commands:

```
fw ctl debug (when used without the -buf parameter)
fw ctl get
fw ctl set
fw ctl leak
fw ctl pstat
fw monitor
fw tab
```

For details and additional parameters for any of these commands, refer to the command's entry.

Example To view the connections table for kernel instance #1 use the following command:

```
fw -i 1 tab -t connections
```

fw ctl

Description The `fw ctl` command controls the Firewall kernel module.

Usage

```
fw ctl <install|uninstall>
fw ctl debug [-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>	<ul style="list-style-type: none"> Uninstall — tells the operating system to stop passing packets to the Security Gateway, and unloads the Security Policy. The networks behind it become unprotected. Install — tells the operating system to start passing packets to the Security Gateway. The command <code>fw ctl install</code> runs automatically when <code>cpstart</code> is performed. <p>Note - If you run <code>fw ctl uninstall</code> followed by <code>fw ctl install</code>, the Security Policy is not restored.</p>
debug	Generate debug messages to a buffer. See <code>fw ctl debug</code> (on page 44).
kdebug	<p>Reads the debug buffer and obtains the debug messages. If there is no debug buffer, the command will fail.</p> <ul style="list-style-type: none"> <code>[-f]</code> read the buffer every second and print the messages, until <code>Ctrl-C</code> is pressed. Otherwise, read the current buffer contents and end. <code>[-t/-T]</code> print the time field (seconds/microseconds) <code>[-p]</code> to print specific fields all proc pid date mid type freq topic time ticks tid text err host vsid cpu <code>[-m]</code> - number of cyclic files, <code>[-s]</code> - size of each

Argument	Description
<code>pstat [-h] [-k] [-s] [-n] [-l]</code>	Displays Security Gateway 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 Security Gateway statistics.
<code>iflist</code>	Displays the IP interfaces known to the kernel, by name and internal number.
<code>arp [-n]</code>	Displays ARP proxy table. -n — Do not perform name resolution.
<code>block <on off></code>	on — Blocks all traffic. off — Restores traffic and the Security Policy.
<code>chain</code>	Prints the names of internal Security Gateways that deal with packets. Use to ensure that a gateway is loaded. The names of these gateways can be used in the <code>fw monitor -p</code> command.
<code>conn</code>	Prints the names of the connection modules.

fw ctl debug

Description Generate debug messages to a buffer.

Usage A number of debug options are available:

```
fw ctl debug -buf [buffer size]
fw ctl debug [-m module] [+ | -] <options| all|0>
fw ctl debug 0
fw ctl debug [-d <comma separated list of strings>]
fw ctl debug [-d <comma separated list of ^strings>]
fw ctl debug [-s <string>]
fw ctl debug -h
fw ctl debug -x
```

Syntax

Argument	Description
<code>-buf [buffer size]</code>	Allocates a buffer of size kilobytes (default 128) and starts collecting messages there. If the <code>-buf</code> argument is not set, the debug messages are printed to the console.
<code>-m <module></code>	Specify the Security Gateway module you wish to debug. The default module is <code>fw</code> . For example: <code>fw ctl debug -m VPN all</code>

Argument	Description
[+ -] <options all 0>	Sets or resets debug flags for the requested gateway). <ul style="list-style-type: none"> • 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.
-h	Print a list of debug modules and flags.
0	Returns all flags in all gateways to their default values, releases the debug buffer (if there was one).
-d <comma separated list of strings>	Only lines containing these strings are included in the output. (Available in R70 or higher)
-d <comma separated list of ^strings>	Lines containing these strings are omitted from the output (Available in R70 or higher) For example: <pre>fw ctl debug -d error,failed,^packet</pre> Output shows only lines containing the words "error" or "failed" and not the word "packet"
-s <string>	Stop debug messages when a certain string is issues (Available in R70 or higher) For example: <pre>fw ctl debug -s error</pre>
-x	Shuts down the debug.

fw ctl affinity

fw ctl affinity -s

Description Sets CoreXL affinities when using multiple processors. For an explanation of kernel, daemon and interface affinities, see the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267).

`fw ctl affinity -s` settings are not persistent through a restart of the Security Gateway. If you want the settings to be persistent, either use:

- `sim affinity` (a Performance Pack command) - for details, see the *R75.20 Performance Pack Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12274).
- Or edit the `fwaffinity.conf` configuration file - for details, see the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267).

To set interface affinities, you should use `fw ctl affinity` only if Performance Pack is not running. If Performance Pack is running, you should set affinities by using the Performance Pack `sim affinity` command. These settings will be persistent. If Performance Pack's `sim affinity` is set to Automatic mode (even if Performance Pack was subsequently disabled), you will not be able to set interface affinities by using `fw ctl affinity -s`.

Usage `fw ctl affinity -s <proc_selection> <cpuid>`

Syntax `<proc_selection>` is one of the following parameters:

Argument	Description
-p <pid>	Sets affinity for a particular process, where <pid> is the process ID#.
-n <cpdname>	Sets affinity for a Check Point daemon, where <cpdname> is the Check Point daemon name (for example: fwd).
-k <instance>	Sets affinity for a kernel instance, where <instance> is the instance's number.
-i <interfacename>	Sets affinity for an interface, where <interfacename> is the interface name (for example: eth0).

<cpuid> should be a processing core number or a list of processing core numbers. To have no affinity to any specific processing core, <cpuid> should be: all.



Note - Setting an Interface Affinity will set the affinities of all interfaces sharing the same IRQ to the same processing core. To view the IRQs of all interfaces, run: `fw ctl affinity -l -v -a`.

Example To set kernel instance #3 to run on processing core #5, run:

```
fw ctl affinity -s -k 3 5
```

fw ctl affinity -l

Description Lists existing CoreXL affinities when using multiple processors. For an explanation of kernel, daemon and interface affinities, see the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267).

Usage `fw ctl affinity -l [<proc_selection>] [<listtype>]`

Syntax If <proc_selection> is omitted, `fw ctl affinity -l` lists affinities of all Check Point daemons, kernel instances and interfaces. Otherwise, <proc_selection> is one of the following parameters:

Argument	Description
-p <pid>	Displays the affinity of a particular process, where <pid> is the process ID#.
-n <cpdname>	Displays the affinity of a Check Point daemon, where <cpdname> is the Check Point daemon name (for example: fwd).
-k <instance>	Displays the affinity of a kernel instance, where <instance> is the instance's number.
-i <interfacename>	Displays the affinity of an interface, where <interfacename> is the interface name (for example: eth0).

If <listtype> is omitted, `fw ctl affinity -l` lists items with specific affinities, and their affinities. Otherwise, <listtype> is one or more of the following parameters:

Argument	Description
-a	All: includes items without specific affinities.
-r	Reverse: lists each processing core and the items that have it as their affinity.
-v	Verbose: list includes additional information.

Example To list complete affinity information for all Check Point daemons, kernel instances and interfaces, including items without specific affinities, and with additional information, run:

```
fw ctl affinity -l -a -v
```

fw ctl engine

Description Enables the INSPECT2C engine, which dynamically converts INSPECT code to C code.

Run the command on the Check Point Security Gateway.

Usage `fw ctl engine {on | off | stat | setdefault}`

Syntax

Argument	Description
on	<p>Compile the engine if necessary, and activate it.</p> <p>Because the engine may not have been previously compiled, turning the engine ON may not activate it immediately. Instead, the engine is activated in the background after the compilation.</p> <p>After turning the engine ON, the engine recompiles and reactivates itself every policy installation regardless of the values of <code>inspect2c_compile</code> and <code>inspect2c_activate</code>.</p>
off	<p>Deactivates the engine if active. Subsequent policy installation on the gateway do NOT auto-activate the engine unless the command is used again.</p>
stat	<p>Print the status of the engine. For example: "During compilation", "Before auto-activation", "Deactivated".</p>
setdefault	<p>Restore control to database settings. Security Management server settings are ignored.</p> <p>At the next policy installation, return the control of the engine to the values of the following gateway database attributes:</p> <ul style="list-style-type: none"> <code>inspect2c_compile</code> (true/false) - controls whether or not the engine is compiled on the gateway during policy installation. Compilation is performed in the background and may take a few minutes. <code>inspect2c_activate</code> (true/false) - controls whether the engine is automatically activated after it is compiled. When set to true, the engine is compiled regardless of the value of <code>inspect2c_compile</code>. <p>Use GuiDBEdit to change the values of the attributes.</p>

fw ctl multik stat

Description Displays multi-kernel statistics for each kernel instance. The state and processing core number of each instance is displayed, along with:

- The number of connections currently being handled.
- The peak number of concurrent connections the instance has handled since its inception.

fw ctl sdstat

Description The IPS performance counters measure the percentage of CPU consumed by each IPS protection. The measurement itself is divided according to the type of protection: Pattern based protections

or INSPECT based protections. In addition, the IPS counters measure the percentage of CPU used by each section ("context") of the protocol, and each protocol parser.

Usage `fw ctl zdebug >& outputfile`
`fw ctl sdstat start`
`fw ctl sdstat stop`

Syntax

Argument	Description
<code>fw ctl zdebug >& outputfile</code>	Turn on debug mode and specify an output file.
<code>fw ctl sdstat start</code>	Activate the IPS counters
<code>fw ctl sdstat stop</code>	Print a report and stop the counters.

Example The workflow is as follows:

Run the following commands on the Check Point Security Gateway (version R70 or higher):

On the Check Point Security Gateway:

- Run `fw ctl zdebug >& outputfile`
- Run `fw ctl sdstat start`

Let the counters run. However- do not leave the counters on for more than 10 minutes.

- Run `fw ctl sdstat stop`

It is important to stop the counters explicitly, otherwise there may be performance penalty

This generates the output file `outputfile` that must be processed on the (SecurePlatform only) Security Management Server.

On the Security Management Server:

- From `$FWDIR/script`, run the script
`./sdstat_analyse.csh outputfile`

The output of the script is a report in csv format that can be viewed in Microsoft Excel.

If there is a problem in the report, or if more details are needed, a debug flag is available which prints extra information to `outputfile`.

- Run `fw ctl zdebug + spii >& outputfile`

Example Debug Message	Explanation
<code>sdstat_get_stats_all_instances : Smart Defense report objects are not initialized, hence no report can be done.</code>	User tried to create a report without initializing the counters, or an error occurred during initialization and the user then tried to print a report.
<code>FW-1 - sdstats_print_report: Failed to calculate Smart Defense (total_smart_defense is 0)</code>	The measurement process failed and the total time units for IPS is zero.

Comments

1. A value in the report of "< 1" means that the percentage of CPU used by a protection is less than 1%.
2. The report generated by the `sdstat_analyse` script may contain a number instead of a protection name. This is because the original output contains a signature id, but the id is missing from the Security Policy on the Gateway.

fw fetch

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

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 <code>*.log</code> . 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 `l` 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 *R75.20 Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12277).
http://supportcontent.checkpoint.com/documentation_download?ID=8745

fw hastat

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

Usage `fw hastat [<target>]`

Syntax

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

fw isp_link

Description Takes down (or up) a redundant ISP link.

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

Syntax

Argument	Description
<code>target</code>	The name of the Check Point gateway.
<code>link-name</code>	The name of the ISP link as defined in the ISP-redundancy tab.

Comments This command can be executed locally on the Check Point Security Gateway or remotely from the Security Management server. In the latter case, the `target` argument must be supplied. For this command to work, the Check Point Security Gateway should be using the ISP redundancy feature.

fw kill

Description Prompts the kernel to shut down all firewall daemon processes. The command is located in the `$FWDIR/bin` directory on the Security Management server or gateway machine.

The firewall 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 snmp daemon is: `$FWDIR/tmp/snmpd.pid`.

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

Syntax

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

This command should be run from the Security Management server.

Usage `fw lea_notify`

fw lichosts

Description Print a list of hosts protected by Security Gateway products. The list of hosts is in the file `$fwdir/database/fwd.h`

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

Syntax

Argument	Description
<code>-x</code>	Use hexadecimal format.
<code>-l</code>	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]	<p>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.</p> <p>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.</p> <p>-t must come with a -f flag. These flags are relevant only for active files.</p>
-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).
-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). <i>starttime</i> 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). <i>endtime</i> 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. <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> : 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 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: Security Gateway
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: Security Gateway;
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: Security Gateway;
```

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 Security Management server can use `fw logswitch` to change a Log File on a remote machine and transfer the Log File to the Security Management server. This same operation can be performed for a remote machine using `fw lslogs` (on page 59) and `fw fetchlogs` (on page 49).

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

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

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

Syntax

Argument	Description
<code>-audit</code>	Does logswitch for the Security Management server audit file. This is relevant for local activation.
<code>filename</code>	The name of the file to which the log is saved. If no name is specified, a default name is provided.
<code>-h hostage</code>	The resolvable name or IP address of the remote machine (running either a Security Gateway or a Security Management server) on which the Log File is located. The Security Management server (on which the <code>fw logswitch</code> command is executed) must be defined as one of <code>host</code> 's Security Management servers. In addition, you must initialize SIC between the Security Management server and the <code>host</code> .
<code>+</code>	Change a remote log and copy it to the local machine.
<code>-</code>	Change 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 Security Management server when the `+` or `-` parameters are specified. Note that if `-` is specified, the Log File on the `host` is deleted rather than renamed.

`hostage` specified:

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

On the Security Management server, the copied file will have the same name, but prefixed by `hostage_`. For example, `target_2003-03-26_041200.log`.

`hostage` not specified:

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

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 Merge 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
<code>-s</code>	Sort merged file by log records time field.
<code>-t</code> <code>time_conversion_file</code>	Fix different GMT zone log records time in the event that the log files originated from Log Servers in different time zone. The <code>time_conversion_file</code> format is as follows: <code>ip-address signed_date_time_in_seconds</code> <code>ip-address signed_date_time_in_seconds</code>
<code>log_file_name_n</code>	<i>Full pathnames</i> of the Log File(s).
<code>output_file</code>	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 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 Security Gateway 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 kernel module chain* — `fw monitor` allows you to capture packets at multiple capture positions within the firewall 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
<code>-u s</code>	Printing the UUID or the SUUID: The option <code>-u</code> or <code>-s</code> 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.
<code>-i</code>	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 <code>fw monitor</code> process and want to be sure that all data is written to a file.
<code>[-d] [-D]</code>	Debugging fw monitor: The <code>-d</code> option is used to start <code>fw monitor</code> in debug mode. This will give you an insight into <code>fw monitor</code> 's inner workings. This option is only rarely used outside Check Point. It is also possible to use <code>-D</code> to create an even more verbose output.
<code><{-e expr>+ -f <filter-file ->></code>	Filtering fw monitor packets: <code>fw monitor</code> has the ability to capture only packets in which you are interested. <code>fw monitor</code> filters use a subset of INSPECT to specify the packets to be captured. Set the filter expression: <ul style="list-style-type: none"> • on the command line using the <code>-e</code> switch. • by reading it from a file using the <code>-f</code> switch. • by reading it from standard input using the <code>-f -</code> switch.

Argument	Description
-l len	<p>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 size 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.</p>
-m mask	<p>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.</p>
-x offset[,len]	<p>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.</p>
-o <file>	<p>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.</p> <p>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 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".</p>
-T	<p>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.</p>
<[-pi pos] [-pI pos] [-po pos] [-pO pos] -p all >	<p>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 chain the packets should be captured. This can be defined using these options.</p>
-a	<p>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.</p>

Argument	Description
<code>[-ci count] [-co count]</code>	Capture a specific number of packets: <code>fw monitor</code> 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 <code>fw monitor</code> 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.
<code>[-vs vsid or vsname]</code>	Capture on a specific Virtual Router or Virtual Machine: VPN-1 Power VSX enables you to run multiple Virtual Routers and FireWalls on one physical machine. Using the option <code>-vs</code> you can specify on which virtual component the packets should be captured. This option is only available on a VPN-1 Power VSX module. Please refer to <code>fw monitor</code> on FireWall-1 VSX for more information.
<code>-h</code>	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 Check Point gateway. 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 SecureKnowledge solution sk30583 (<http://supportcontent.checkpoint.com/solutions?id=sk30583>).

fw lslogs

Description Display a list of Log Files residing on a remote or local machine. You must initialize SIC between the Security Management server and the remote machine.

Usage `fw lslogs [[-f <filename>] ...] [-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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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).
<machine>	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 Install a Check Point authentication password on a host. This password is used to authenticate internal communications between Security Gateways and between a Check Point Security Gateway and its Security Management 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
<code>-opsec</code>	Only control connections are enabled.
<code>-no_opsec</code>	Only OPSEC control connections are enabled.
<code>-ssl</code>	The key is used for an SSL connection.
<code>-no_ssl</code>	The key is not used for an SSL connection.
<code>-k num</code>	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.
<code>-n <myname></code>	The IP address (in dot notation) to be used by the Check Point Security Gateway when identifying this host to all other hosts, instead of, for example, the resolution of the <code>hostname</code> command.
<code>-p <psw></code>	The key (password). If you do not enter the password on the command line, you will be prompted for it.
<code>host</code>	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: <pre>"/fwd: Authentication with hostname for command sync failed"</pre>

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
<code>-u</code>	Indicates that the unification chains in the Log file should be rebuilt.
<code>logfile</code>	The name of the Log file to repair.

fw sam

Description 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 Security Management server or Security Gateway, use SmartDashboard to edit the **Advanced > SAM** page of the Check Point Security 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
<code>-v</code>	Verbose mode. Writes one message (describing whether the command was successful or not) to <code>stderr</code> for each Security Gateway machine on which the command is enforced.
<code>-s sam_server</code>	The IP address (in dot format) or the resolvable name of the FireWalled host that will enforce the command. The default is <code>localhost</code> .
<code>-S server_sic_name</code>	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.
<code>-f <fw host></code>	Specify the <code>host</code> , the Security Gateway machine on which to enforce the action. <code>host</code> can be one of the following (default is <code>All</code>): <ul style="list-style-type: none"> <code>localhost</code>—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. <code>Gateways</code>—Action enforced on FireWalls defined as gateways and managed by Security Management server where the SAM server runs. <code>All</code>—Enforced on FireWalls managed by Smart- Center server where SAM server runs.
<code>-D</code>	Cancel all inhibit (<code>-i</code> , <code>-j</code> , <code>-I</code> , <code>-J</code>) and notify (<code>-n</code>) commands. To "uninhibit" inhibited connections, execute <code>fw sam</code> with the <code>-C</code> or <code>-D</code> 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 <code>-t</code> (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: <code>nolog</code> , <code>long_noalert</code> , <code>long_alert</code> . The default is <code>long_alert</code> .
-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 <i>rejected</i> .
-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 <i>rejected</i> .
-j	Inhibit new connections with the specified parameters. Each inhibited connection is logged according to the log type. Connections will be <i>dropped</i> .
-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 <i>dropped</i> .
-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.
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.
dstpr <ip> <protocol>	Match the destination IP address and protocol.

Criteria Parameters	Description
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 `louvre` is the source or destination address:

```
fw sam -M -nij any louvre
```

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 virtual machine, and other components of the Security Gateway 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 (<code>\$FWDIR/conf/sys.conf</code>).
-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.

Argument	Description
-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 <code>fw log</code> or any other entity that can read FW log formats.
-c	Displays formatted table information 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 <code>-a</code> or the <code>-x</code> flags, respectively. These flags must be followed by the <code>-e</code> flag and an entry description (<entry>).
	 Caution - Improper use of the <code>-a</code> and <code>-x</code> flags may cause system instability.
[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:

`<00000000,00000001,00000002>`

```
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 independently 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 Display the Security Gateway 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 Perform management operations on the Security Gateway. It controls *fwd* and all Check Point daemons.

Usage fwm

fwm dbimport

Description 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 Check Point password authentication method is used, a valid Check Point 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 expdate

Description 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`

fwm dbexport

Description Export 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
<code>-g group</code>	Specifies a group (<code>group</code>) to be exported. The users in the group are not exported.
<code>-u user</code>	Specifies that only one user (<code>user</code>) is to be exported.
<code>-d</code>	Debug flag
<code>-a {attrib1, attrib2, ...}</code>	Specifies the attributes to export, in the form of a comma-separated list, between {} characters, for example, <code>-a {name, days}</code> . If there is only one attribute, the {} may be omitted.
<code>-f file</code>	file specifies the name of the output file. The default output file is <code>\$FWDIR/conf/user_def_file</code> .
<code>-l</code>	Create an LDIF format file for importation by an LDAP server.
<code>-p</code>	The profile name.
<code>-s</code>	The branch under which the users are to be added.
<code>-k</code>	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 Security Management server.
 - * On the destination Security Management server, create the groups manually.
 - * Run `fwm dbimport` on the destination Security Management server.

The users will be added to the groups to which they belonged on the source Security Management 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 Download 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
<code>-all</code>	Execute command on all targets specified in the default system configuration file (<code>\$FWDIR/conf/sys.conf</code>). This file must be manually created.
<code>-conf <conffile></code>	Only OPSEC control connections are enabled.
<code><targets></code>	Execute command on the designated targets.

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
<code>shared-secret</code>	The IKE Key defined in the Encryption tab of the LDAP Account Unit Properties window.
<code>user-password</code>	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 Security Management server, following installation.

fwm load

Description Compile and install a Security Policy or a specific version of the Security Policy on the target's Security Gateways. This is done in one of two ways:

- `fwm load` compiles and installs an Inspection Script (*.pf) file on the designated Security 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 Security Gateways.

Versions of the Security Policy and databases are maintained in a version repository on the Security Management 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 Security Management 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
<code>-S</code>	The targets are UTM-1 Edge gateways.
<code>-p</code>	Specifies the product name if applicable.
<code>rulebase</code>	A Rule Base created by the GUI. Specify the name of the rulebase, such as <code>Standard</code> (case sensitive).
<code>-v version number</code>	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.
<code>targets</code>	Execute command on the designated target.

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

```
fwm load -v18 Standard johnny
```

fwm lock_admin

Description View and unlock locked administrators.

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

Syntax

Argument	Description
<code>-v</code>	View the names of all locked administrators.
<code>-u administrator</code>	Unlock a single administrator.
<code>-ua</code>	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
<code>-d delimiter</code>	Set the output delimiter. The default is a semicolon (;).
<code>-i filename</code>	The name of the input Log file. The default is the active Log file, <code>fw.log</code>
<code>-o outputfile</code>	The name of the output file. The default is printing to the screen.
<code>-n</code>	Do not perform DNS resolution of the IP addresses in the Log file (this option significantly speeds the processing).
<code>-p</code>	Do not perform service resolution. A service port number is displayed.
<code>-f</code>	If this is the active Log file (<code>fw.log</code>), wait for new records and export them to the ASCII output file as they occur.
<code>-m</code>	This flag specifies the unification mode. <ul style="list-style-type: none"> <code>initial</code> - the default mode. Complete the unification of log records; that is, output one unified record for each id. <code>semi</code> - 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. <code>raw</code> - output all records, with no unification.
<code>-a</code>	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_message;;
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 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_reset`

Syntax

Argument	Description
<code>sic_reset</code>	Resets the internal CA SIC certificates and deletes the Internal CA.

fwm unload <targets>

Description Uninstall the currently loaded Inspection Code from selected targets.

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

Syntax

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

fwm ver

Description `fwm ver` displays the build number.

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

fwm verify <policy-name>

Description The `fwm verify <policy-name>` command verifies the specified policy package without installing it.

Usage `fwm verify <policy-name>`

Syntax

Argument	Description
<policy-name>	The name of an available policy package.

GeneratorApp

Description Generate a report for SmartReporter. 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 <i>R75.20 SmartReporter Administration Guide</i> (http://supportcontent.checkpoint.com/documentation_download?ID=12281).

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 Notify 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 Security 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
<code>-s ipaddr</code>	The IP address (in dot format) of the ELA Proxy to be contacted.
<code>-o</code>	Print the alert log received by <code>inet_alert</code> to <code>stdout</code> . Use this option when <code>inet_alert</code> is part of a pipe.
<code>-a auth_type</code>	The type of connection to the ELA Proxy. One of the following values: <ul style="list-style-type: none"> • 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.
<code>-p port</code>	The ELA Proxy's port number. Default is 18187.
<code>-f token value</code>	A field to be added to the log, represented by a <code>token-value</code> pair as follows: <ul style="list-style-type: none"> • <code>token</code> is the name of the field to be added to the log. <code>token</code> may not contain spaces. • <code>value</code> is the field's value. <code>value</code> may not contain spaces. <p>This option may be used multiple times to add multiple <code>token-value</code> pairs to the log.</p> <p>If <code>token</code> is a reserved log field name, the specified field's value will appear in the corresponding column in SmartView Tracker. Otherwise, the <code>token-value</code> pair will be displayed in the Info column in SmartView Tracker.</p>
<code>-m alertty</code>	The alert to be triggered at the ISP site. This alert overrides the alert specified in the log message generated by the alert daemon. <p>The response to the alert is handled according to the actions specified in the ISP's Security Policy:</p> <p>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:</p> <ul style="list-style-type: none"> • alert. <code>Popup alert</code> command. • mail. <code>Mail alert</code> command. • snmptrap. <code>SNMP trap alert</code> command. • spoofalert. <code>Anti-spoof alert</code> command. <p>The following NetQuota and ServerQuota alerts execute the OS commands specified in: <code>\$FWDIR/conf/objects.C</code>:</p> <p><code>value=clientquotaalert.</code> <code>Parameter=clientquotaalertcmd</code></p>

Return Value

exit status	meaning
0	Execution was successful.
102	Undetermined error.
103	Unable to allocate memory.
104	Unable to obtain log information from <code>stdin</code> .
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 SmartView Tracker.
- Trigger the OS command specified in the **Popup Alert Command** field of the **Log and Alert** tab of the **Properties** Setup window in the SmartDashboard.

ldapcmd

Description `ldapcmd` is used to manage processes running on the Security Gateway 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).
- 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:

- `cachedclear` (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.

Idapcompare

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.

Idapconvert

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.
-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 `ldapconvert` 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 Security Management 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 <code>options</code> attributes below.
filter	RFC-1558 compliant LDAP search filter. For example, <code>objectclass=fwlhost</code> .
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 `fwlcolor` attribute, the file written is named.
- `/tmp/ldapsearch-fwlcolor-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=fwlhost objectclass`

This means that the LDAP directory will be queried for `fwlhost` 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 Security Gateway Log data to be exported to third-party applications. `log_export` receives the Logs from the Security Management server via LEA so it can be run from any host that has a SIC connection with the Security Management 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
<code>-f conf_file</code>	The Configuration File from which <code>log_export</code> reads the Log file parameters. If <code>conf_file</code> is not specified, the default Configuration File <code>log_export.conf</code> , located in the current working directory.
<code>-l <lea_server_ip_address></code>	The IP address of the LEA server.
<code>-g log_file_name,log_file_name,...</code>	A comma separated list of log file names from where the logs will be taken.
<code>-t <database_table_name></code>	The name of the table in the database to which the logs will be added.
<code>p <database_password></code>	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.
<code>-h</code>	Display <code>log_export</code> usage.
<code>-d</code>	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 Windows 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 SmartReporter.

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.

Argument	Description
-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 ddmmmyyyy 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 ddmmmyyyy defaults to today's date on the client.
-p m u h t f	Short print options: <ul style="list-style-type: none"> • 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 <i>object_name</i> >	Enter the name of the gateway object.
<-ip <i>module_ip</i> >	Enter the IP Address of the gateway
<-TTL <i>Time-To-Live</i> >	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 <code>fw sam</code> command).
-s <i>sam_server</i>	The sam server to be contacted. Localhost is the default.
-t <i>timeout</i>	The time period, in seconds, for which the action will be enforced. The default is forever.
-f <i>fw_host</i>	Identifies the FireWalls to run the operation on. Default is "all FireWalls."
-C	Cancel the specified operation.
-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.

Argument	Description
-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

This utility is used to configure the SmartReporter 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 *R75.20 SmartReporter Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12281).

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 2

VPN Commands

In This Chapter

VPN

87

VPN

Description VPN commands generate status information regarding VPN processes, or are used to stop and start specific VPN services. All VPN commands are executed on the Security Gateway. 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.

`vpn accel`

Description Perform operations on VPN accelerator cards (encryption only cards, not the full SecureXL cards). 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
<code>autostart on off</code>	Automatically starts/stops the vpnx accelerator software
<code>on/off</code>	Enable/disable accelerator card or vpnx accelerator module
<code>stat [-l]</code>	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 Reset 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 Display compression/decompression statistics.

Usage `vpn compstat`

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

vpn crl_zap

Description 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 Retrieve 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
<code>-obj -cert</code>	<ul style="list-style-type: none"> • <code>-obj</code> refers to the name of the CA network object • <code>-cert</code> refers to the name of the certificate
<code>-f</code>	Refers to the filename of the certificate
<code>-view</code>	Views the CRL
<code>-d</code>	Debug option

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

vpn debug

Description Instruct 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 Install the VPN kernel (vpnk) and connects to the firewall kernel (fwk), attaching the VPN driver to the Firewall driver.

Usage `vpn drv on|off`

`vpn drv stat`

Syntax

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

vpn export_p12

Description Export 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_p12 -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_topology

Description Generate and upload 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

Argument	Description
<code>-print_xml</code>	The topology is in XLM format. This flag writes that topology to a file in XLM format.

vpn overlap_enddom

Description Display 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_enddom [communities | traditional]`

Syntax

Argument	Description
<code>Communities</code>	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.
<code>Traditional</code>	Default flag. All pairs of objects with overlapping VPN domains are displayed.

Example `vpn overlap_enddom 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 Download the topology for a SofaWare gateway.

Usage `vpn [-d] sw_topology -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 Launch 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-
27cb65clafd28bc6

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 Display the VPN 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 3

SmartView Monitor Commands

In This Chapter

RTM

95

RTM

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

rtm debug

Description Send 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 Start, stop or check 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 Starts the monitoring process and specify 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	<aggregate individual>
-w	<bandwidth loss rtt>
-t	<wire application>
-i	<number of seconds>
@@	specifies subrule (for example, 'rule@@subrule')
default values	'-y bytes -a aggregate -w bandwidth -i2
grouping types	svc src dst ip fgrule topsvc topsrc topdst topip topfw topfgrule
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
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 default is eitherbound.

Argument	Description
-g	Specifies one of the following grouping options for monitored traffic: <ul style="list-style-type: none"> - 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.
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 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 Concurrent-Connections for the top 50 sources passed on interface eth0, inbound (that is, not telnet or 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 entities 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 Start 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 value 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.
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.

Argument	Description
application	Shows the data as the application sees it (that is, not compressed and not encrypted).

rtm rtmd

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

Usage `rtm rtmd`

rtm stat

Description Display 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
v1	Current virtual links
view	Current views

rtm ver

Description Display the SmartView Monitor version.

Usage `rtm ver [-k]`

Syntax

Argument	Description
-k	Displays the SmartView Monitor kernel version.

rtmstart

Description Load the SmartView Monitor kernel module and starts the SmartView Monitor daemon.

Usage `rtmstart`

rtmstop

Description Kill the SmartView Monitor daemon and unloads the SmartView Monitor kernel module.

Usage `rtmstop`

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.

scc connect

Description Connect 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 Connect 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 Disconnect 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 Unset authorization credentials

Usage `scc erasecreds`

Comments Shortcut: `scc ep`

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

scc listprofiles

Description List all profiles.

Usage `scc listprofiles`

Comments Shortcut: `scc lp`

You must be in CLI mode to run this command.

scc numprofiles

Description Display 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 Restart SecureClient services.

Usage `scc restartsc`

Comments You need administrator privileges to run this command.

scc passcert

Description Set 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 Switch 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 Enable or disable 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 Display the current default security policy.

Usage `scc sp`

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

scc startsc

Description Start SecureClient services.

Usage `scc startsc`

Comments You need administrator privileges to run this command.

scc status

Description Display the connection status.

Usage `scc status`

Comments Shortcut: `scc s`

scc stopsc

Description Stop SecureClient services.

Usage `scc stopsc`

Comments You need administrator privileges to run this command.

scc suppressdialogs

Description Enable or suppress 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 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 Displays the current SecureClient version.

Usage `scc ver`

Chapter 4

ClusterXL Commands

In This Chapter

cphaconf	104
cphaprob	105
cphastart	105
cphastop	105

cphaconf

Description The cphaconf command configures ClusterXL.



Important - Running this command is not recommended. It should be run automatically, only by the Security Gateway 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
<code>cphaconf set_ccp <broadcast/multicast></code>	<p>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.</p> <p>Note, the same value (either broadcast or multicast) should be set on all cluster members.</p>

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
<code>cphaprob -d <device> -t <timeout(sec)> -s <ok init problem> [-p] register</code>	Register <code><device></code> as a critical process, and add it to the list of devices that must be running for the cluster member to be considered active.
<code>cphaprob -f <file> register</code>	Register all the user defined critical devices listed in <code><file></code> .
<code>cphaprob -d <device> [-p] unregister</code>	Unregister a user defined <code><device></code> as a critical process. This means that this device is no longer considered critical.
<code>cphaprob -a unregister</code>	Unregister all the user defined <code><device></code> .
<code>cphaprob -d <device> -s <ok init problem> report</code>	Report the status of a user defined critical device to ClusterXL.
<code>cphaprob [-i[a]] [-e] list</code>	View the list of critical devices on a cluster member, and of all the other machines in the cluster.
<code>cphaprob state</code>	View the status of a cluster member, and of all the other members of the cluster..
<code>cphaprob [-a] if</code>	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.

Chapter 5

Debugging SmartConsole Clients

You can run any SmartConsole client in debug mode and save the debug information in a text file.

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

Syntax:

Argument	Description
<client.exe>	The SmartConsole client executable file. For example, fwpolicy.exe.
-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 6

CLI for Other Products

This guide documents Command Line Interface (CLI) commands for Check Point Products and features. The commands are documented by product.

In This Chapter

[CLI Commands in Other Guides](#)

107

CLI Commands in Other Guides

- For CoreXL commands, see the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267).
- For SmartProvisioning and SmartLSM Security Gateway commands, see the *R75.20 SmartProvisioning Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12280).
- For Multi-Domain Security Management commands, see the *R75.20 Multi-Domain Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12273).
- For QoS commands, see the *R75.20 QoS Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12275).

Index

C

- CLI Commands in Other Guides • 107
- CLI for Other Products • 107
- ClusterXL Commands • 104
- comp_init_policy • 8
- cp_admin_convert • 8
- cp_conf • 10
- cp_conf admin • 10
- cp_conf auto • 11
- cp_conf ca • 10
- cp_conf client • 11
- cp_conf finger • 11
- cp_conf ha • 11
- cp_conf lic • 11
- cp_conf sic • 10
- cp_conf snmp • 11
- cp_conf sxl • 11
- cp_merge • 20
- cp_merge delete_policy • 20
- cp_merge export_policy • 21
- cp_merge import_policy and cp_merge restore_policy • 22
- cp_merge list_policy • 23
- cpca_client • 8
- cpca_client create_cert • 8
- cpca_client lscert • 9
- cpca_client revoke_cert • 9
- cpca_client set_mgmt_tools • 9
- cpconfig • 12
- cphaconf • 104
- cphaprob • 105
- cphastart • 105
- cphastop • 105
- cpinfo • 12
- cplic • 13
- cplic check • 13
- cplic db_add • 14
- cplic db_print • 14
- cplic db_rm • 15
- cplic del • 15
- cplic del <object name> • 16
- cplic get • 16
- cplic print • 18
- cplic put • 17
- cplic put <object name> ... • 18
- cplic upgrade • 19
- cppkg • 23
- cppkg add • 23
- cppkg delete • 24
- cppkg get • 24
- cppkg getroot • 25
- cppkg print • 25
- cppkg setroot • 25
- cpidrestart • 26
- cpidstart • 26
- cpidstop • 26
- cprinstall • 26
- cprinstall boot • 26
- cprinstall cpstart • 27

- cprinstall cpstop • 27
- cprinstall get • 27
- cprinstall install • 28
- cprinstall revert • 31
- cprinstall show • 31
- cprinstall snapshot • 30
- cprinstall transfer • 31
- cprinstall uninstall • 29
- cprinstall verify • 30
- cpstart • 32
- cpstat • 32
- cpstop • 34
- cpwd_admin • 34
- cpwd_admin config • 36
- cpwd_admin exist • 36
- cpwd_admin kill • 36
- cpwd_admin list • 35
- cpwd_admin start • 34
- cpwd_admin stop • 35

D

- dbedit • 37
- dbver • 40
- dbver create • 40
- dbver export • 40
- dbver import • 41
- dbver print • 41
- dbver print_all • 42
- Debugging SmartConsole Clients • 106
- dynamic_objects • 42

F

- fw • 42
- fw ctl • 43
- fw ctl affinity • 45
- fw ctl affinity -l • 46
- fw ctl affinity -s • 45
- fw ctl debug • 44
- fw ctl engine • 47
- fw ctl multik stat • 47
- fw ctl sdstat • 47
- fw fetch • 49
- fw fetchlogs • 49
- fw hastat • 50
- fw -i • 42
- fw isp_link • 50
- fw kill • 51
- fw lea_notify • 51
- fw lichosts • 51
- fw log • 51
- fw logswitch • 54
- fw lslogs • 59
- fw mergefiles • 55
- fw monitor • 55
- fw putkey • 60
- fw repairlog • 60
- fw sam • 61
- fw stat • 64
- fw tab • 65
- fw ver • 66
- fwm • 66
- fwm dbexport • 68
- fwm dbimport • 66
- fwm dbload • 69
- fwm expdate • 68

fwm ikecrypt • 70
fwm load • 70
fwm lock_admin • 70
fwm logexport • 71
fwm sic_reset • 72
fwm unload <targets> • 72
fwm ver • 73
fwm verify <policy-name> • 73

G

GeneratorApp • 73

I

inet_alert • 73

L

ldapcmd • 75
ldapcompare • 76
ldapconvert • 76
ldapmodify • 79
ldapsearch • 79
log_export • 80

Q

queryDB_util • 83

R

rs_db_tool • 84
RTM • 95
rtm debug • 95
rtm drv • 95
rtm monitor <module_name><interface_name>
or rtm monitor <module_name>-filter • 96
rtm monitor <module_name>-
v<virtual_link_name> • 98
rtm rtmd • 99
rtm stat • 99
rtm ver • 99
rtmstart • 99
rtmstop • 99

S

sam_alert • 85
SCC • 100
scc connect • 100
scc connectnowait • 100
scc disconnect • 100
scc erasecreds • 101
scc listprofiles • 101
scc numprofiles • 101
scc passcert • 101
scc restartsc • 101
scc setmode <mode> • 101
scc setpolicy • 102
scc sp • 102
scc startsc • 102
scc status • 102
scc stopsc • 102
scc suppressdialogs • 102
scc userpass • 103
scc ver • 103
SecureClient Commands • 100
Security Management Server and Firewall
Commands • 7

SmartView Monitor Commands • 95
svr_webupload_config • 86

V

VPN • 87
vpn accel • 87
VPN Commands • 87
vpn compreset • 88
vpn compstat • 88
vpn crl_zap • 89
vpn crlview • 89
vpn debug • 89
vpn drv • 90
vpn export_p12 • 90
vpn macutil • 91
vpn nssm_topology • 91
vpn overlap_encdom • 92
vpn sw_topology • 93
vpn tu • 93
vpn ver • 94