13 March 2017

Command Line Interface

R80.10

Reference Guide

Check Point
SOFTWARE TECHNOLOGIES LTD.

Important Information

Latest Software

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

Check Point R80.10

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

Latest Version of this Document

Download the latest version of this document http://supportcontent.checkpoint.com/documentation_download?ID=.

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

Feedback

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

Please help us by sending your comments mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Command Line Interface R80.10 Reference Guide.

Searching in Multiple PDFs

To search for text in all the R80.10 PDF documents, download and extract the complete R80.10 documentation package http://supportcontent.checkpoint.com/documentation_download?ID=TBD.
Use **Shift-Control-F** in Adobe Reader or Foxit reader.

Revision History

Date	Description
13 March 2017	First release of this document

Contents

CLI Commands for Software Blades

In This Section:

This guide documents CLI (Command Line Interface) commands for Check Point Software Blades and features. For more about CLI commands for Check Point operating systems:

- *R80.10 Gaia Administration Guide*
  http://supportcontent.checkpoint.com/documentation_download?ID=TBD
- *R80.10 Gaia Advanced Routing Administration Guide*
  http://supportcontent.checkpoint.com/documentation_download?ID=24803

CLI Commands in Other Guides

- For CoreXL and Multi-queue commands, see the *R80.10 Performance Tuning Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=TBD.
- For Multi-Domain Security Management commands, see the *R80.10 Multi-Domain Security Management Administration Guide*
  http://supportcontent.checkpoint.com/documentation_download?ID=46532.
- For QoS commands, see the *R80.10 QoS Administration Guide*
  http://supportcontent.checkpoint.com/documentation_download?ID=46533.

Running CLI Commands in Automation Scripts

In This Section:

Introduction to Automation Scripts

Use these CLI commands and tools to create automation scripts:

- `dbedit` - Creates and configures objects and rules in the database for the Security Policy.
- `fwm load` - Installs the specified Security Policy on Security Gateways. The Security Policy is validated, and only valid Policies are installed.
- `send_command` - Runs functions which are not included with standard Check Point CLI commands and tools.

We recommend that you use a separate SmartConsole administrator account for automation scripts. This additional account lets you easily monitor automatic changes and ones made by system administrators.

Working with dbedit

Introduction to dbedit

`dbedit` is a CLI utility that lets you make changes to objects in the Check Point databases. Run `dbedit` in these modes:

- Interactive - For a few changes to the database
- Batch - Import many changes at one time

We recommend that you use batch mode (`dbedit -f`) for automation scripts. You can write the script on the Security Management Server or Multi-Domain Server with standard Linux commands, or import a text file with the script.

Launching the dbedit Utility

When the `dbedit` prompt is showing, you can run `dbedit` commands or scripts. Before you use the `dbedit` utility, make sure that you can log in to Expert mode on the Security Management Server or Multi-Domain Server.

To launch the **`dbedit`** utility:

1. Log in to the CLI of the Security Management Server or Multi-Domain Server.

2. Enter Expert mode, run `expert`

   The Expert prompt is shown.
3. Run `dbedit`
4. Enter the name of the Security Management Server or Multi-Domain Server:
   - For localhost, press **Enter**
   - For a remote connection, enter the hostname or IP address

   The `dbedit` prompt is shown.

```
Please enter a command, -h for help or -q to quit:
dbedit>
```

Using dbedit Commands in a Script

Use these `dbedit` commands to create and configure objects and rules:

- `create` - Creates the object
- `modify` - Changes the applicable object
- `update` - Commits the most recent change to the Security Management Server database
- `update_all` - Commits all the changes to the Security Management Server database

This table shows sample commands and the results.

Example	Result
`create network net-internal`	Creates the object for the network `net-internal`
`modify network_objects gateway-10 ipaddr 192.0.2.100`	Changes the IP address of the `gateway-10` object to `192.0.2.100`
`update network_objects net-internal`	Saves the changes for the `net-internal` objects and updates the Security Management Server database

Locking the Database

We recommend that you use the `-globallock` option when you use `dbedit` to make changes to the Security Management Server database. `dbedit` partially locks the database, if a user configures objects with SmartDashboard, there can be problems in the database. The `-globallock` option does not let SmartDashboard or a `dbedit` user make changes to the database.

When the `-globallock` option is enabled, `dbedit` commands run on a copy of the database. After you change the database and run the `savedb` command, it is saved and committed on the actual database. You can use the `savedb` command multiple times in a `dbedit` script.

At the end of a script, it is a best practice to run these commands:

```
# update_all
# savedb
```

Showing Parameters for a Sample Object

You can create sample objects in SmartDashboard that have the parameters that you are using in a script or `dbedit` command. Export these objects to help make sure that you are using the correct names for the parameters. You can show the parameters in plain or XML format.

To show the parameters for a sample SmartDashboard object:

1. In SmartDashboard, create the object that uses the necessary parameters and settings.
2. From the `dbedit` prompt ("Launching the dbedit Utility" on page 10), run one of these commands:
   - `print network_objects <object name>`
   - `printxml network_objects <object name>`

Using dbedit in Automation Scripts

You can use `dbedit` to configure the initial settings for a Security Gateway and the Security Policy, then update and change the settings when necessary.

Note - Make sure that the script in the text files does not contain blank lines. Otherwise the script will stop with an error.

Initial Configuration

1. Create a text file with an automation script ("Create or Modify Policy Objects (Hosts, Networks)" on page 13). The script can create and configure the necessary objects and rules for the Security Policy.
2. Make a database revision of the management. Use this revision if there is a problem with the script and to identify unauthorized changes to the database.
3. Run `fwm load` and install the policy on one or more Security Gateways ("Pushing the Security Policy to Security Gateways" on page 19).

Updating and Changing the Policy

1. Make sure that the automation administrator changed the database most recently.
   a) Run `send_command -s <domain_server> -u <admin> -p <password> -o db_change_since_last_save`

      The `Last modifier` field shows the administrator name.

   b) If a different administrator changed the database, do not continue to use the automation script. A system administrator must do an analysis of the database.
2. Edit the automation script, create and configure objects and rules for the Security Policy ("Changing a Rule Base" on page 16).
3. Run `fwm load` and install the policy on one or more Security Gateways ("Installing Policy with a Multi-Domain Server" on page 19).

To update and change the commands for a Domain Management Server:

This sample script installs the `Standard` policy from Domain Management Server `Cust_CMA` on the Security Gateway `examplegw`.

```
mdsenv Cust_CMA
send_command -s Cust_CMA -u admin -p admin -o db_change_since_last_save
dbedit -globallock -s Cust_CMA -u admin -p admin -f
dbedit_modifiability_objects.txt
fwm load Standard examplegw
```

Create or Modify Policy Objects (Hosts, Networks)

This section shows sample scripts that create one or more new network or service objects. You can combine one or more of these samples into one script file.

We recommend that you add the `update_all` command to the end of the script file.

Networks

You can use a script to manage database objects that include:

- Networks
- Hosts
- Address Ranges

These are sample scripts that show how to create and configure the database objects.

Creating a Network

Create an object for the database that represents a network. This sample script creates the network `net-internal` with the IP address `190.0.2.0`.

```
Create the object (of type network)
create network net-internal
Configure the network IP address
modify network_objects net-internal ipaddr 192.0.2.0
Configure the netmask (in dotted decimal notation) of the network
modify network_objects net-internal netmask 255.255.255.0
Add a comment to describe what the object is for (optional)
modify network_objects net-internal comments "Created by fwadmin with dbedit"
```

Configuring Automatic NAT

If your network uses NAT (Network Address Translation), you can use `dbedit` to configure an Automatic NAT rule. Add these lines to a script only for a network that uses Automatic NAT rules.

This sample script creates an Automatic NAT rule for the `net-internal` network that starts with the IP address `190.0.2.100`.

```
The next four modify lines are optional and are only needed if you want
to do an automatic NAT rule for this object.
modify network_objects net-internal add_adtr_rule true
modify network_objects net-internal NAT NAT
Set the NAT type, adtr_static or adtr_hide
modify network_objects net-internal NAT:netobj_adtr_method adtr_hide
Set the "valid" IP address for this object.
For a static NAT on a network, the assumption is there is a 1-to-1 ratio
between untranslated and translated addresses and the valid range is
contiguous. This setting is the first IP address in this range.
modify network_objects net-internal NAT:valid_ipaddr 192.0.2.100
```

Creating a Host

This sample script creates the host `host-10` with the IP address `192.0.2.10`.

```
Create the actual object (of type host_plain)
create host_plain host-10
Modify the host IP address
modify network_objects host-10 ipaddr 192.0.2.10
Add a comment to describe what the object is for (optional)
modify network_objects host-10 comments "Created by fwadmin with dbedit"
```

You can also add the lines to this script to configure Automatic NAT for the host ("Configuring Automatic NAT" on page 13). The `modify` commands for this sample rule starts with: `modify network_objects host-10`

Creating an Address Range

This sample script creates the address range object `addr-range` with the IP addresses `192.0.2.150` to `190.0.2.200`.

```
Create the actual object (of type address_range)
create address_range addr-range
Modify the first IP address in the range
modify network_objects addr-range ipaddr_first 192.0.2.150
Modify the last IP address in the range
modify network_objects addr-range ipaddr_last 192.0.2.200
Add a comment to describe what the object is for (optional)
modify network_objects addr-range comments "Created by fwadmin with dbedit"
```

You can also add the lines to this script to configure Automatic NAT for the address range object ("Configuring Automatic NAT" on page 13). The `modify` commands for this sample rule starts with: `modify network_objects addr-range`

Renaming and Deleting Objects

You can change the name of an object or delete it from the database. When you change the name of an object the Security Policy is also updated with the new name.

```
Rename the network object addr-range to IPv4-range
rename network_objects addr-range IPv4-range
```

When you delete an object, the references to it are also deleted from the Rule Base. The `delete` command fails if there is a different object that is dependent on it.

```
Delete the network object addr-range
delete network_objects addr-range
```

Network Groups

You can create and use a group object as a container for network and host objects.

Creating a Network Group

Create a network group that uses networks and hosts. Make sure that these objects are in the management database before you create a network group.

This sample script creates the object `host-group` for the hosts `host-100` and `host-101`.

```
Create a group object
create network_object_group host-group
Add the individual elements to the group
addelement network_objects host-group '' network_objects:host-100
addelement network_objects host-group '' network_objects:host-101
```

Configuring and Deleting a Network Group

You can remove a network or host from a network group. This sample script removes `host-100` from `host-group`.

```
Remove individual elements from the group
rmelement network_objects host-group '' network_objects:host-100
```

You can rename or remove a network group almost the same as objects ("Renaming and Deleting Objects" on page 14).

```
Rename the network object host-group to host-ipaddrs
Rename network_objects host-group host-ipaddrs
Delete the network object host-ipaddrs
delete network_objects host-ipaddrs
```

Services

Services are objects that are used for network protocols.

Creating a Service

This sample script creates these services:

- `tcp_8081` - TCP protocol port 8081
- `udp_8082` - UDP protocol port 8082
- `inspect_svc` - Inspect SVC protocol 6 and with an optional feature that uses the INSPECT expression

```
Create a TCP service
create tcp_service tcp_8081
Set port 8081 for TCP service
modify services tcp_8081 port 8081
Create a UDP service
create udp_service udp_8082
Set port 8082 for UDP service
modify services udp_8082 port 8082
Create a service of type "other." This can be used for random IP protocols
as well as services that require more complex INSPECT code for matching.

Create the service of type other
create other_service inspect_svc
Modify the IP Protocol that matches the service
modify services inspect_svc protocol 6
(Optional) Modify the INSPECT expression that matches this service.
modify services inspect_svc exp "dport=123”
```

Renaming and Deleting a Service

You can rename or remove a service almost the same as objects ("Renaming and Deleting Objects" on page 14).

```
Rename inspect_svc to inspect_tcp123
rename services inspect_svc inspect_tcp123
Delete the network object inspect_tcp123
delete services inspect_tcp123
```

Service Groups

You can create and use a group object as a container for service objects.

Creating a Service Group

Create a service group for more than one service. Make sure that the service objects are in the management database before you create a service group.

This sample script creates the object `mysvc-group` for the services SSH and HTTPS.

```
Create a group object
create service_group mysvc-group
Add the individual elements to the group
addelement services mysvc-group '' services:ssh
addelement services mysvc-group '' services:https
```

Configuring and Deleting a Service Group

You can remove a network or host from a network group. This sample script removes the SSH service from `mysvc-group`.

```
Remove individual elements from the group
rmelement services mysvc-group '' services:ssh
```

You can rename or remove a network group almost the same as objects ("Renaming and Deleting Objects" on page 14).

```
Rename the service group mysvc-group to myservices
rename services mysvc-group myservices
Delete the network object my services
delete services myservices
```

Object Naming Restrictions

These are some of the restrictions for object names:

- Objects names can contain only ASCII letters, numbers, and dashes. Other characters such as a plus sign, asterisk, parenthesis, square brackets, and so on, are not supported.
- Object names can have a maximum of 100 characters.
- You cannot use reserved words for objects names and they include words that are policy elements. For example, names of colors, common networks terms (ipv6, nets, routers, servers, and so on).

To see a full list of the naming restrictions, go to sk40179 (http://supportcontent.checkpoint.com/solutions?id=sk40179).

Changing a Rule Base

This section shows sample scripts that change the Policy on a Domain Management Server named `Standard`. We recommend that you write the scripts in a text file and then you import the file to dbedit.

Adding a Rule

When you use dbedit to add a rule, the rule must be added to the bottom of the Rule Base by manually specifying the rule number. If the policy contains no other rules, the rule becomes the policy's first rule.

Note - Rules in SmartDashboard start with rule number 1. Rules in dbedit start with rule number 0.

This sample script creates a new policy called DemoPolicy with a Rule Base that contains this rule:

Source	Destination	Service	Action
Any	Any	Any	Accept

```
create policies_collection ##DemoPolicy
modify policies_collections ##DemoPolicy comments "Demo"
modify policies_collections ##DemoPolicy default 1
update policies_collections ##DemoPolicy
create firewall_policy ##DemoPolicy
modify fw_policies ##DemoPolicy default 0
```

```
modify fw_policies ##DemoPolicy collection policies_collections:##DemoPolicy
addelement fw_policies ##DemoPolicy rule security_rule
modify fw_policies ##DemoPolicy rule:0:name "AcceptAll"
rmbyindex fw_policies ##DemoPolicy rule:0:track 0
addelement fw_policies ##DemoPolicy rule:0:track tracks:None
addelement fw_policies ##DemoPolicy rule:0:time globals:Any
addelement fw_policies ##DemoPolicy rule:0:install:'' globals:Any
addelement fw_policies ##DemoPolicy rule:0:action accept_action:accept
addelement fw_policies ##DemoPolicy rule:0:src:'' globals:Any
modify fw_policies ##DemoPolicy rule:0:src:op ''
addelement fw_policies ##DemoPolicy rule:0:dst:'' globals:Any
modify fw_policies ##DemoPolicy rule:0:dst:op ''
addelement fw_policies ##DemoPolicy rule:0:services:'' globals:Any
modify fw_policies ##DemoPolicy rule:0:services:op ''
update_all
```

Changing a Rule

This sample script changes this rule:

	Source	Destination	Service	Action
Original rule 4	Any	Any	Any	Accept
New rule 4	Any	DMZ	SSH	Accept

```
Modify Rule 4
Previous rule was any any any accept, it will now be any dmz ssh accept

modify fw_policies ##Standard rule:3:comments "Allow SSH to firewall with logging"
modify fw_policies ##Standard rule:3:disabled false
rmbyindex fw_policies ##Standard rule:3:track 0
addelement fw_policies ##Standard rule:3:track tracks:Log
rmbyindex fw_policies ##Standard rule:3:action 0
addelement fw_policies ##Standard rule:3:action accept_action:accept
rmelement fw_policies ##Standard rule:3:src:'' globals:Any
addelement fw_policies ##Standard rule:3:src:'' globals:Any
modify fw_policies ##Standard rule:3:src:op ''
rmelement fw_policies ##Standard rule:3:dst:'' globals:Any
addelement fw_policies ##Standard rule:3:dst:'' network_objects:DMZ
modify fw_policies ##Standard rule:3:dst:op ''
rmelement fw_policies ##Standard rule:3:services:'' globals:Any
addelement fw_policies ##Standard rule:3:services:'' services:ssh
modify fw_policies ##Standard rule:3:services:op ''
```

Adding a Rule - Middle of Rule Base

When it is necessary to add a rule to the middle of a Rule Base, you cannot use `dbedit` to simply insert a rule.

1. Delete all the rules that are after the new rule you are adding.
2. Create one or more new rules.
3. Add again the rules that you deleted in step 1.

This sample script adds a new rule number 2 in a Rule Base that has three rules.

Note - Rules in SmartDashboard start with rule number 1. Rules in dbedit start with rule number 0.

```
Delete rule 2 and 3 (delete in reverse order)

rmbyindex fw_policies ##Standard rule 2
rmbyindex fw_policies ##Standard rule 1

Add new rule 2

addelement fw_policies ##Standard rule security_rule
modify fw_policies ##Standard rule:1:comments "Firewall stealth rule"
modify fw_policies ##Standard rule:1:disabled false
rmbyindex fw_policies ##Standard rule:1:track 0
addelement fw_policies ##Standard rule:1:track tracks:Log
addelement fw_policies ##Standard rule:1:time globals:Any
addelement fw_policies ##Standard rule:1:install:'' globals:Any
rmbyindex fw_policies ##Standard rule:1:action 0
addelement fw_policies ##Standard rule:1:action drop_action:drop
addelement fw_policies ##Standard rule:1:src:'' network_objects:net-internal
modify fw_policies ##Standard rule:1:src:op 'not in'
addelement fw_policies ##Standard rule:1:dst:'' globals:Any
modify fw_policies ##Standard rule:1:dst:op ''
addelement fw_policies ##Standard rule:1:services:'' globals:Any
modify fw_policies ##Standard rule:1:services:op ''

Add New Rule 3 (Old Rule 2)

addelement fw_policies ##Standard rule security_rule
modify fw_policies ##Standard rule:2:comments "Allow selected hosts outbound"
modify fw_policies ##Standard rule:2:disabled false
rmbyindex fw_policies ##Standard rule:2:track 0
addelement fw_policies ##Standard rule:2:track tracks:Log
addelement fw_policies ##Standard rule:2:time globals:Any
addelement fw_policies ##Standard rule:2:install:'' globals:Any
rmbyindex fw_policies ##Standard rule:2:action 0
addelement fw_policies ##Standard rule:2:action accept_action:accept
addelement fw_policies ##Standard rule:2:src:'' network_objects:flamer-100
addelement fw_policies ##Standard rule:2:src:'' network_objects:flamer-101
modify fw_policies ##Standard rule:2:src:op ''
addelement fw_policies ##Standard rule:2:dst:'' network_objects:net-internal
modify fw_policies ##Standard rule:2:dst:op 'not in'
addelement fw_policies ##Standard rule:2:services:'' globals:Any
modify fw_policies ##Standard rule:2:services:op ''

Add New Rule 4 (Old Rule 3)

addelement fw_policies ##MyPolicy rule security_rule
modify fw_policies ##MyPolicy rule:3:comments "Drop all"
modify fw_policies ##MyPolicy rule:3:disabled false
rmbyindex fw_policies ##MyPolicy rule:3:track 0
addelement fw_policies ##MyPolicy rule:3:track tracks:Log
addelement fw_policies ##MyPolicy rule:3:time globals:Any
addelement fw_policies ##MyPolicy rule:3:install:'' globals:Any
rmbyindex fw_policies ##MyPolicy rule:3:action 0
addelement fw_policies ##MyPolicy rule:3:action drop_action:drop
addelement fw_policies ##MyPolicy rule:3:src:'' globals:Any
modify fw_policies ##MyPolicy rule:3:src:op ''
addelement fw_policies ##MyPolicy rule:3:dst:'' globals:Any
modify fw_policies ##MyPolicy rule:3:dst:op ''
addelement fw_policies ##MyPolicy rule:3:services:'' globals:Any
modify fw_policies ##MyPolicy rule:3:services:op ''
```

Pushing the Security Policy to Security Gateways

After you change or update the Security policy, you can use `fwm load` command to push the configuration to the Security Gateways. This command validates the policy and makes sure that rules agree with each other.

In this example, the `fwm load` command successfully pushes the policy (`Standard`) to the Security Gateway (`samplegw`).

```
# fwm load Standard samplegw
Installing policy on R80.10 compatible targets:
Standard.W: Security Policy Script generated into CustomerPolicy.pf
Standard:
Compiled OK.
Installing Security Gateway policy on: examplegw ...
Security Gateway policy installed successfully on examplegw...
Security Gateway policy installation complete
Security Gateway policy installation succeeded for:
examplegw
```

If the policy did not install successfully, the output of the `fwm load` command shows an error message. The Security Gateway continues to enforce the policy that was installed before you ran the script.

Installing Policy with a Multi-Domain Server

To install the policy for a Domain Management Server, run the necessary Multi-Domain Server CLI commands. You can run them individually or as part of a script.

This sample script installs the `Standard` policy from Domain Management Server `Cust_CMA` on the Security Gateway `examplegw`.

```
mdsenv Cust_CMA
dbedit -globallock -s Cust_CMA -u admin -p admin -f dbedit_createpolicy_objects.txt
fwm load Standard examplegw
```

Error Codes in dbedit

- If there is a syntax error in the `dbedit` script, this error is shown:
  `"syntax error in line 1 Aborting."`
  The script stops running at the error.
- When a script uses tables or objects that are not in the database, `dbedit` stops the script and shows this message:
  `"Object Not Found"`
  `"Error in line: 2"`
- You can use the parameter `ignore_script_failure` to continue running the script and ignore errors
- You can use the parameter `continue_updating` to ignore errors and run the `update_all` command at the end of the script

Using XML to Export Settings for a Domain Management Server

You can export the settings for a Domain Management Server to an XML file that you can use with external automation systems. You can include the `printxml` commands in a script or run them individually from the CLI.

This sample script exports these settings to XML:

- Security policy Rule Base
- Network objects
- Services

```
printxml fw_policies ##Standard
printxml network_objects
printxml services
```

Security Management Server and Firewall Commands

In This Section:

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.

Syntax

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

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

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

cp_admin_convert

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

Syntax

```
> cp_admin_convert
```

cpca_client

Description These commands execute operations on the ICA (Internal Certificate Authority).

Syntax

```
> cpca_client
```

cpca_client create_cert

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

Syntax

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

Parameter	Description
-d	Runs the command in debug mode
-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 to <common name>
-f <PKCS12>	Specifies the file name, <PKCS12>, that stores the certificate and keys.

cpca_client revoke_cert

Description Revoke a certificate issued by the ICA.

Syntax

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

Parameter	Description
-d	Runs the command in debug mode
-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 to <common name>

cpca_client lscert

Description Show all certificates issued by the ICA.

Syntax

```
> cpca_client [-d] lscert [-dn <substring>] [-stat
{Pending|Valid|Revoked|Expired|Renewed}] [-kind SIC|IKE|User|LDAP] [-ser
<ser>] [-dp <dp>]
```

Parameter	Description
-d	Runs the command in debug mode
-dn substring	Filters results to those with a DN that matches this <substring>
-stat	Filters results to the specified certificate status: Pending, Valid, Revoke, Expire, or Renewed
-kind	Filters results for specified kind: SIC, IKE, User, or LDAP
-ser <serial>	Filters results for this serial number
-dp <dp>	Filters results from this CDP (certificate distribution point)

cpca_client init_certs

Description Imports a list of DNs for users and creates a file with registration keys for each user.

Syntax

```
> cpca_client init certs [-p <ca_port>] -i <input_file> -o <output_file>
```

Parameter	Description
`-p <ca_port>`	Specifies the port which is used to connect to the CA. The default port is `18265`.
`-i <input_file>`	Imports the specified file. Make sure to use the full path. Make sure that there is an empty line between each DN in the file: CN=test1,OU=users <empty line> CN=test2,OU=users
`-o <output_file>`	Saves the registration keys to the specified file.

cpca_client set_mgmt_tool

Description Starts or stops the ICA Management Tool.

Syntax

```
> cpca_client [-d] set_mgmt_tool {on|off|add|remove|clean|print} [-p
<ca_port>] [-no_ssl] {-a <administrator DN>, -u <user DN>, -c <custom user
DN>, ...}
```

Parameter	Description
`-d`	Runs the command in debug mode.
`set_mgmt_tool {on|off|add|remove|clean|print}`	• `on` - Starts ICA Management Tool • `off` - Stops ICA Management Tool • `add` - Adds an administrator, user, or custom user • `remove` - Removes an administrator, user, or custom user • `clean` - Removes all the administrators, users, or custom users • `print` - Shows the administrators, users, or custom users
`-p <ca_port>`	Specifies the port which is used to connect to the CA. The default port is `18265`.
`-no_ssl`	Configures the server to use HTTP instead of HTTPS.
`-a <administrator DN>`	Sets the DNs of the administrators that are permitted to use the ICA Management Tool.

Parameter	Description
`-u <user DN>`	Sets the DNs of the users that are permitted to use the ICA Management Tool.
`-c <custom user DN>`	Sets the DN for custom users that can 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.

cpca_client set_sign_hash

Description Sets the hash algorithm that the CA uses to sign the file has. The default algorithm is `sha1`.

Syntax

```
> cpca_client set_sign_hash {sha1|sha256|sha384|sha512}
```

cpca_client search

Description Searches for certificates in the ICA (Internal Certificate Authority).

Syntax

```
> cpca_client search <string> [-where {dn|comment|serial}] [-kind
[SIC|IKE|User|LDAP]] [-stat [Pending|Valid|Revoked|Expired|Renewed]] [-max
<max results>] [-showfp {y|n}]
```

Parameter	Description
`-where {dn|comment|serial}`	Where to search for the string, in the **dn**, **serial number**, or **comment** field. The default is all locations.
`-kind [SIC|IKE|User|LDAP]`	The type of certificate. You can enter multiple values in this format: `-kind value1 value2 value3`. The default is all values.
`-stat [Pending|Valid|Revoked|Expired|Renewed]`	Filters according to the status of the certificate. You can enter multiple values in this format: `-stat value1 value2 value3`. The default is all values.
`-max <max results>`	Enter the maximum number of results to show. The default setting is `200`.
`-showfp {y|n}`	Show the certificate's fingerprint: yes or no. The default is yes.

Example `> cpca_client search samplecompany -where comment -kind SIC LDAP -stat Pending Valid Renewed`

cpca_client get_crldp

Description Shows the name that the computer or server uses to initialize with the CA.

Syntax

```
> cpca_client get_crldp [-p <ca_port>]
```

Parameter	Description
`-p <ca_port>`	Specifies the port which is used to connect to the CA. The default port is `18265`.

cpca_client get_pubkey

Description Saves the encoding of the public key for the ICA to a file.

Syntax

```
> cpca_client [-p <ca_port>] get_pubkey <output>
```

Parameter	Description
`-p <ca_port>`	Specifies the port which is used to connect to the CA. The default port is `18265`.
`<output>`	Name of the file where the public key is saved

cpca_client double_sign

Description Creates a second signature for a certificate.

Syntax

```
> cpca_client [-p <ca_port>] -i <cert file> [-o <output file>]
```

Parameter	Description
`-p <ca_port>`	Specifies the port which is used to connect to the CA. The default port is `18265`.
`-i <cert file>`	Imports the specified certificate only in PEM format.
`[-o <output file>]`	Saves the certificate to the specified file.

cp_conf

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

Syntax

```
> cp_conf
```

cp_conf sic

Description Use the `cp_conf sic` commands to manage SIC on the Security Management Server.

Syntax

```
> cp_conf sic state
> cp_conf sic init <key> [norestart]
> cp_conf sic cert_pull <management> <object>
```

Parameter	Description
`state`	Shows the SIC trust state.
`init <key>`	Restarts SIC with the Activation Key <key>.
`[no restart]`	By default, the Security Gateway runs `cpstop` and `cpstart` when you restart SIC. Use the `norestart` parameter to restart SIC and to not run `cpstop` and `cpstart`.
`cert_pull`	For DAIP Security Gateways, pulls a certificate from the Security Management Server for the `<object>`
`<management>`	Name or IP address of the Security Management Server

cp_conf admin

Description Manage Check Point system administrators for the Security Management Server

Syntax

```
> cp_conf admin get # Get the list of administrators.
> cp_conf admin add <user> <pass> {a|w|r}
> cp_conf admin del <admin1> <admin2>...
```

Parameter	Description
`get`	Shows a list of the administrators
`add <user> <pass>`	Adds a new administrator <user> with password <pass>
`{a|w|r}`	Sets the permissions for the new administrator: a - Read, write and manage administrators w - Read and write r - Read only
`del <admin1>`	Deletes one or more administrators <admin1>, <admin2>, and so on

cp_conf ca

Description Initialize the Certificate Authority

Syntax

```
> cp_conf ca init
> cp_conf ca fqdn <name>
```

Parameter	Description
`init`	Initializes the internal CA
`fqdn <name>`	Sets the FQDN of the internal CA to `<name>`

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

Syntax

```
> cp_conf finger get
```

cp_conf lic

Description Shows the installed licenses and lets you manually add new ones.

Syntax

```
> cp_conf lic get
> cp_conf lic add -f <file>
> cp_conf lic add -m <Host> <Date> <Key> <SKU>
> cp_conf lic del <Signature Key>
```

Parameter	Description
`get`	Shows the installed licenses
`add -f <file>`	Adds the license from `<file>`
`add -m`	Manually adds a license with these parameters: `<host>` - name of the Security Management Server `<Date>` - Date of the license `<Key>` - License key `<SKU>` - License SKU
`del <Key>`	Deletes license `<key>`

cp_conf client

Description Manage the GUI clients that can use SmartConsoles to connect to the Security Management Server.

Syntax

```
> 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.
```

Parameter	Description
`get`	Shows the IP addresses of the allowed GUI clients.
`add <GUI client>`	Adds the `<GUI client>` IP address to the list of allowed GUI clients.
`del <GUI client1> <GUI client 2>`	Deletes one or more IP addresses from the list of allowed GUI clients.
`createlist <GUI client1> <GUI client 2>`	Deletes allowed GUI clients and creates a new list. The new list allows `<GUI client 1>`, `<GUI client 2>`, and so on.

cp_conf ha

Description Enable or disable High Availability.

Syntax

```
> cp_conf ha {enable|disable} [norestart]
```

cp_conf snmp

Description Activate or deactivate SNMP.

Syntax

```
> cp_conf snmp get # Get SNMP Extension status.
> cp_conf snmp {activate|deactivate} [norestart] # Deactivate SNMP Extension.
```

Parameter	Description
`get`	Shows the SNMP status.
`{activate|deactivate}`	Enables or disables SNMP.
`[no restart]`	By default, the Security Gateway runs `cpstop` and `cpstart` when you enable or disable SNMP. Use the `norestart` parameter to configure SNMP and to not run `cpstop` and `cpstart`.

cp_conf auto

Description Configure the Security Gateway and Security Management Server products that start automatically when the appliance or server reboots.

Syntax

```
> cp_conf auto get [fw1] [fg1] [rm] [all]
> cp_conf auto {enable|disable} <product1> <product2>...
```

Parameter	Description
`get`	Shows which products start automatically

Parameter	Description
`{enable|disable} <product1> <product2>`	Enables or disables the one or more products that start automatically

cp_conf sxl

Description Enable or disable SecureXL acceleration.

Syntax

```
> cp_conf sxl {enable|disable}
```

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.

Syntax `

```
> cpconfig
```

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.

Syntax

```
> cpinfo [-v] [-l] [-n] [-o ] [-r | -t [tablename]] [-c <domain> ... | -x <vs>]
```

Parameter	Description
`-z`	Output gzipped (effective with -o option)
`-r`	Includes the registry (for Windows servers - shows a 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)
`-c <domain>`	Get information about the specified `<domain>` Domain Management Server (Multi-Domain Security Management)
`-x <vs>`	Get information about the specified `<vs>` Virtual System (VSX)

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

cplic

The `cplic` command and all its derivatives relate to Check Point license management.

Note - 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.

cplic check

Description Makes sure that the license includes the feature on the local gateway or Security Management Server.

Syntax

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

Parameter	Description
`-p <product>`	Product for which license information is requested. For example `fw1, netso`
`-v <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 `feature` option is not needed
`-S|-SRusers`	Check how many SecuRemote users are allowed.
`<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, and can only be executed on the Security Management server.

Syntax

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

Parameter	Description
`-l <license-file>`	Name of the file that contains the license
`<host>`	Security Management Server hostname or IP address
`<expiration-date>`	The license expiration date
`<signature>`	The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (The string is case sensitive and 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 `192.0.2.11.lic` contains one or more licenses, the command: `cplic db_add -l 192.0.2.11.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.

Syntax

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

Parameter	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, and 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.

Syntax

```
> cplic db_rm <signature>
```

Parameter	Description
`Signature`	The signature string within the license.

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

Comments This command is a license repository command, and 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

Syntax

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

Parameter	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 Security 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.

Syntax

```
> cplic del <object name> [-F <outputfile>] [-ip <dynamic ip>] <signature>
```

Parameter	Description
`<object name>`	The name of the Check Point Security Gateway object, as defined in SmartDashboard.
`-F <outputfile>`	Divert the output to `outputfile` 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 Security Gateway (or from all Security Gateways) into the license repository on the Security Management Server. This command helps you to synchronize the repository with the Check Point Security Gateways. When the command is run, all local changes are updated.

Syntax

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

Parameter	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.

Syntax

```
> 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>]
```

Parameter	Description
-o|-overwrite	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.
-c|-check-only	Verify the license. Checks if the IP of the license matches the machine, and if the signature is valid
-s|-select	Select only the Local licenses whose IP address matches the IP address of the machine.
-F <outputfile>	Outputs the result of the command to the designated file rather than to the screen.
-P|-Pre-boot	Use this option after upgrading and before rebooting the machine. Use of this option will prevent certain error messages.
-K|-kernel-only	Push the current valid licenses to the kernel. For Support use only.

Parameter	Description
`-l <license-file>`	Name of the file that contains the license
`<host>`	Security Management Server hostname or IP address
`<expiration-date>`	The license expiration date
`<signature>`	The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (The string is case sensitive and the hyphens are optional)
`<SKU/features>`	The SKU of the license summarizes the features included in the license. For example: `CPSUITE-EVAL-3DES-vNG`

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.

Syntax

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

Parameter	Description
`object name`	The name of the Check Point Security Gateway object, as defined in SmartDashboard.

Parameter	Description
`-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 `<outputfile>` rather than to the screen.
`-l <license-file>`	Installs the license(s) from `<license-file>`.
`-l <license-file>`	Name of the file that contains the license
`<host>`	Security Management Server hostname or IP address
`<expiration-date>`	The license expiration date
`<signature>`	The License signature string. For example: `aa6uwknDc-CE6CRtjhv-zipoVWSnm-z98N7Ck3m` (The string is case sensitive and the hyphens are optional)
`<SKU/features >`	The SKU of the license summarizes the features included in the license. For example: `CPSUITE-EVAL-3DES-vNG`

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

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.

Syntax

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

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

Syntax

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

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

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

- Upgrade the Security Management Server to the latest version.

  Ensure that there is connectivity between the Security Management Server and the Security Gateways 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
192.0.2.11   Never      CPFW-FIG-25-53        CK-49C3A3CC7121 golda
192.0.2.11   26Nov2012  CPSUITE-EVAL-3DES-NGX CK-1234567890   count
```

- In the *User Center* http://usercenter.checkpoint.com, view the licenses for the products that were upgraded from version NGX to a Software Blades license and create new upgraded licenses.
- Download a file containing the upgraded licenses. Only download licenses for the products that were upgraded from version NGX to Software Blades.
- If you did not import the version NGX licenses into the repository, import the version NGX licenses now using the command `cplic get -all`
- 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 Security Gateways are updated with the new licenses.

  - A report of the results of the license upgrade is printed.
- In the example, there are two Software Blades licenses in the file. One does not match any license on a remote Security Gateway, the other matches a version NGX license on a Security Gateway that should be upgraded:

Comments This is a *Remote Licensing Command* which affects remote Security Gateways, that is executed on the Security Management Server.

Further Info. For more about managing licenses, see the *R80.10 Installation and Upgrade Guide* http://downloads.checkpoint.com/dc/download.htm?ID=TBD.

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.

Syntax

```
> cp_merge help
```

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

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.

Syntax

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

Parameter	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.

Syntax

```
> 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]
```

Parameter	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

Parameter	Description
-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.

Syntax

```
> 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]
```

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

Comments Further considerations

1. Either use certificate file or user and password

2. Optional

The cp_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

Syntax

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

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

Comments Further considerations:

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

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

```
> cp_merge list_policy -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. The package file can be added to the Repository directly from the DVD or from a local or network drive.

Syntax

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

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

Parameter	Description
CD drive	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-R80.10 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-R80.10\
Enter package name:
----------------------
(1) SVNfoundation
(2) firewall
(3) floodgate
(4) rtm

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

Syntax

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

Parameter	Description
vendor	Package vendor (for example, checkpoint)
product	Package name
version	Package version
os	Package Operating System. Options are: win32, solaris, ipso, linux

Parameter	Description
sp	Package minor version

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`.

Syntax

```
> 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`.

Syntax

```
> 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.

Syntax

```
> 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 content 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.

Syntax

```
> cppkg setroot <repository>
```

Parameter	Description
<repository>	The full path for 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.

Syntax

```
> 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.

Syntax

```
> 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.

Syntax

```
> cprinstall boot <object name>
```

Parameter	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.

Syntax

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

Parameter	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.

Syntax

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

Parameter	Description
`-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.

Parameter	Description
-nopolicy	
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.

cprinstall get

Description Obtain details of the products and the operating system installed on the specified Check Point Security Gateway, and to update the database.

Syntax

```
> cprinstall get <object name>
```

Parameter	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 Security Gateways. To install a product package you must specify a number of options. Use the `cppkg print` command and copy the required options.

Syntax

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

Parameter	Description
-boot	Boot the remote computer after installing the package. Only boot after ALL products have the same version. Boot will be canceled in certain scenarios.
Object name	Object name of the Check Point Security Gateway defined in SmartDashboard.
vendor	Package vendor (e.g. `checkpoint`)

Parameter	Description
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 Security Gateways. To uninstall a product package you must specify a number of options. Use the `cppkg print` command and copy the required options.

Syntax

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

Parameter	Description
-boot	Boot the remote computer after installing the package. Only boot after ALL products have the same version. Boot will be canceled 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 Makes sure these operations were successful:

- If a specific product can be installed on the remote Check Point Security 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

Syntax

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

Parameter	Description
`Object name`	Object name of the Check Point Security Gateway defined in SmartDashboard.
`vendor`	Package vendor (for example `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 192.0.2.134
Operation Success. Product cannot be installed, did not pass dependency check.
```

cprinstall snapshot

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

Syntax

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

Parameter	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.

Syntax

```
> cprinstall show <object name>
```

Parameter	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.

Syntax

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

Parameter	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.

Syntax

```
> cprinstall transfer <object name> <vendor> <product> <version> [sp]
```

Parameter	Description
`Object name`	Object name of the Check Point Security Gateway defined in SmartDashboard.
`vendor`	Package vendor (for example, `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 an appliance or server.

Syntax

```
> 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 or on another appliance or server, in various formats.

Syntax

```
> cpstat [-h <host>][-p <port>][-s <SICname>][-f <flavor>][-o <polling>][-c
<count>][-e <period>][-d] <application_flag>
```

Parameter	Description
`-h <host>`	A resolvable hostname, a dot-notation address (for example: 192.0.2.23), 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 <SICname>`	Secure Internal Communication (SIC) name of the AMON server.

Parameter	Description
`-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>`	Polling interval (seconds) specifies the pace of the results. The default is 0, meaning the results are shown only once.
`-c <count>`	Specifies how many times the results are shown. The default is 0, meaning the results are repeatedly shown.
`-e <period>`	Specifies the interval (seconds) over which 'statistical' olds are computed. Ignored for regular olds.
`-d`	Debug mode.
`<application_flag>`	One of the following: • `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 parameters 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 an appliance or server.

Syntax

```
> cpstop
> cpstop -fwflag {-proc|-default}
```

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

Comments This command cannot be used to terminate `cprid`. `cprid` is invoked when the appliance or server 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`.

Syntax

```
> cpwd_admin
```

cpwd_admin start

Description Start a new process by `cpwd`.

Syntax

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

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

Example To start and monitor the `fwm` process.

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

cpwd_admin stop

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

Syntax

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

Parameter	Description
`-name <process name>`	A name for the process to be watched by WatchDog.

Parameter	Description
-path <"full path">	The full path to the executable (including the executable name) that is used to stop the process.
-command <"executable name">	The name of the executable file mentioned in -path

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

Example Stops the FWM process using fw kill

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

cpwd_admin list

Description Print a status of the selected processes being monitored by cpwd.

Syntax

```
> 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.

Syntax

```
> cpwd_admin exist
```

cpwd_admin kill

Description Terminate cpwd.

Syntax

```
> cpwd_admin kill
```

cpwd_admin config

Description Set `cpwd` configuration parameters. When parameters are changed, these changes do not take effect until `cpwd` has been stopped and restarted.

Syntax

```
> cpwd_admin config {-p|-a <value=data value=data...>|-d <value
value...>|-r}
```

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

These are the descriptions of the `<value>` parameters:

Value	Description
`timeout` (any value in seconds)	If `rerun_mode=1`, how much time passes from process failure to rerun. The default is 60 seconds.
`no_limit` (any value in seconds)	Maximum number of times that `cpwd` will try to restart a process. The default is 5.
`zero_timeout` (any value in seconds)	After failing `no_limit` times to restart a process, `cpwd` will wait `zero_timeout` seconds before retrying. The default is 7200 seconds. Should be greater than `timeout`.
`sleep_mode`	• 1 - wait timeout • 0 - ignore timeout. Rerun the process immediately
`dbg_mode`	• 1 - Accept pop-up error messages (with exit-code#0) displayed when a process terminates abruptly (Windows NT only). • 0 -Do not receive pop-up error messages. This is useful if pop-up error messages freeze the machine. This is the default (Windows NT only).
`rerun_mode`	• 1 - Rerun a failed process. This is the default. • 0 - Do not rerun a failed process. Perform only monitoring.
`stop_timeout`	The time in seconds that the `cpwd` will wait for a stop command to be completed. Default is 60 seconds.
`reset_startups`	Indicates the time in seconds that the `cpwd` waits after the process begins before it resets the `startup_counter`. Default value is 1 hour, meaning that an hour after the process begins its startup counter is reset to 0.

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

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

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

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

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

disconnect_client

SmartDashboard can connect to a Security Management Server using one of these modes:

- **Read/Write** - Administrators have full permissions to create or change all objects, settings and policies.
- **Read Only** - Administrators can see all objects, settings and policies, but cannot add, change or delete them.

Only one administrator can use SmartDashboard to connect to a Security Management Server in the read/write mode at one time. When an administrator connects in the Read/Write mode, this prevents other administrators from doing these actions:

- Connecting to the same management in the read/write mode
- Creating or changing objects, settings and policies
- Backing up the management server database
- Installing a Security Policy

You can use a special command line utility to disconnect a different SmartDashboard client that is open in the Read/Write mode.

To remove the database lock, run `disconnect_client` from the Security Management Server command line.

For more information, see sk65146 http://supportcontent.checkpoint.com/solutions?id=sk65146

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.

Syntax

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

Parameter	Description
`-s server`	The Security Management server on which the `objects_5_0.C` file to be edited is located. If this is not specified in the command line, then the user will be prompted for it. If the server is not localhost, the user will be required to authenticate.
`-u user |` `-c certificate`	The user's name (the name used for the SmartConsole) or the full path to the certificate file.
`-p password`	The user's password (the password used for the SmartConsole).
`-f filename`	The name of the file containing the commands. If *filename* is not given, then the user will be prompted for commands.
`-r db-open-reason`	A non-mandatory flag used to open the database with a string that states the reason. This reason will be attached to audit logs on database operations.
`-help`	Print usage and short explanation.

dbedit commands:

Parameter	Description

`create`
`[object_type]`
`[object_name]`
	Create an object with its default values.
	The create command may use an extended (or "owned") object.
	Changes are committed to the database only by an `update` or `quit` command.

`modify`
`[table_name]`
`[object_name]`
`[field_name] [value]`
	Modify fields of an object which is:
	• stored in the database (the command will lock the object in such case).
	• newly created by `dbedit`
	Extended Formats for owned objects can be used:
	For example, `[field_name] = Field_A:Field_B`

`update`
`[table_name]`
`[object_name]`
	Update the database with the object. This command will check the object validity and will issue an error message if appropriate.

`delete`
`[table_name]`
`[object_name]`
	Delete an object from the database and from the client implicit database.

`addelement`
`[table_name]`
`[object_name]`
`[field_name] [value]`
	Add an element (of type string) to a multiple field.

`rmelement`
`[table_name]`
`[object_name]`
`[field_name] [value]`
	Remove an element (of type string) from a multiple field.

Parameter	Description
`rename [table_name][object_name] [new_object_name]`	Assign a new name for a given object. The operation also performs an update. Example: Rename network object London to Chicago. `rename network_objects london chicago`
`quit`	Quit `dbedit` and update the database with modified objects not yet committed.

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

```
modify network_objects my_obj firewall_setting NULL
```

Example **Extended Format**

`firewall_properties` owns the object `floodgate_preferences`.

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

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

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

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

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

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

dbver

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

Syntax

```
dbver> export <version_numbers> <delete|keep>
dbver> import <exported_version_in_server>
dbver> create <version_name> <version_comment>
dbver> delete <version_numbers>
dbver> print <version_file_path>
dbver> print_all
```

dbver create

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

Syntax

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

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

dbver export

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

Syntax

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

Parameter	Description
`<version_numbers>`	The file name of the exported version.
`<delete|keep>`	• `delete` removes the revision from the revisions repository • `keep` maintains the revision in the revisions repository

dbver import

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

Syntax

```
dbver> import <exported_version_in_server>
```

Parameter	Description
`<exported_version_in_server>`	The file name of the exported version.

dbver print

Description Print the properties of the revision.

Syntax

```
dbver> print <version_file_path>
```

Parameter	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`

Syntax

```
dbver> print_all
```

dynamic_objects

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` parameter). 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.

Syntax

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

Parameter	Description
`-o <object_name>`	The name of the object, as defined in SmartDashboard and the `dynamic_objects -n <name>` command.
`-r <fromIP> <toIP> ...`	Address ranges — one or more "from IP address to IP address" pairs
`-a <fromIP> <toIP> ...`	Add ranges to object
`-d <fromIP> <toIP> ...`	Delete range from object
`-l`	List dynamic objects
`-n <object_name>`	Create new object (if Security Gateway is not running)
`-c`	Compare the objects in the dynamic objects file and in `objects.C`.
`-do object_name`	Delete object

Example Create a new dynamic object named "bigserver" and add to it the IP address range 192.0.2.1-192.0.2.40: `dynamic_objects -n bigserver -r 192.0.2.1 192.0.2.40 -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.

Syntax

```
> 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.

Syntax

`> 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.

Syntax

```
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
```

Parameter	Description
{Install|Uninstall}	• 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 fw ctl install runs automatically when cpstart is performed. **Note -** If you run fw ctl uninstall followed by fw ctl install, the Security Policy is not restored.
debug	Generate debug messages to a buffer. See fw ctl debug (on page 64).
kdebug	Reads the debug buffer and obtains the debug messages. If there is no debug buffer, the command will fail. • [-f] read the buffer every second and print the messages, until Ctrl-C is pressed. Otherwise, read the current buffer contents and end. • [-t/-T] print the time field (seconds/microseconds) • [-p] to print specific fields all|proc|pid|date|mid|type|freq|topic|time|ticks|tid|text|err|host|vsid|cpu • [-m] - number of cyclic files, [-s] - size of each
pstat [-h] [-k] [-s] [-n] [-l]	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.
iflist	Displays the IP interfaces known to the kernel, by name and internal number.
arp [-n]	Displays ARP proxy table. -n — Do not perform name resolution.
block {on|off}	on — Blocks all traffic. off — Restores traffic and the Security Policy.
chain	Prints the names of internal Security Gateways that deal with packets. Use to ensure that a gateway is loaded. The names of these gateways can be used in the fw monitor -p command.
conn	Prints the names of the connection modules.

fw ctl debug

Description Generate debug messages to a buffer.

Syntax 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
```

Parameter	Description
`-buf [buffer size]`	Allocates a buffer of size kilobytes (default 128) and starts collecting messages there. If the -buf argument is not set, the debug messages are printed to the console.
`-m <module>`	Specify the Security Gateway module you wish to debug. The default module is fw. For example: `fw ctl debug -m VPN all`
`[+ | -] <options|all|0>`	Sets or resets debug flags for the requested gateway). • 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: `fw ctl debug -d error,failed,^packet` 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: `fw ctl debug -s error`

Parameter	Description
-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 *R80.10 Performance Tuning Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=TBD.

`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)
- Or edit the `fwaffinity.conf` configuration file

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`.

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

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

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

Syntax

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

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

Parameter	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 *R80.10 Performance Tuning Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=TBD.

Syntax

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

`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:

Parameter	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:

Parameter	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.

Syntax

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

Parameter	Description
`on`	Compile the engine if necessary, and activate it. 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. After turning the engine ON, the engine recompiles and reactivates itself every policy installation regardless of the values of `inspect2c_compile` and `inspect2c_activate`.
`off`	Deactivates the engine if active. Subsequent policy installation on the gateway does NOT auto-activate the engine unless the command is used again.
`stat`	Print the status of the engine. For example: "During compilation", "Before auto-activation", "Deactivated".
`setdefault`	Restore control to database settings. Security Management server settings are ignored. At the next policy installation, return the control of the engine to the values of the following gateway database attributes: • `inspect2c_compile` (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. • `inspect2c_activate` (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 `inspect2c_compile`. Use GuiDBEdit to change the values of the attributes.

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.

Syntax

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

Parameter	Description
`fw ctl zdebug >& outputfile`	Turn on debug mode and specify an output file.
`fw ctl sdstat start`	Activate the IPS counters
`fw ctl sdstat stop`	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
`sdstat_get_stats_all_instances : Smart Defense report objects are not initalized, hence no report can be done.`	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.
`FW-1 - sdstats_print_report: Failed to calculate Smart Defense (total_smart_defense is 0)`	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.

Syntax

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

Parameter	Description
`-n`	Fetch the Security Policy from the Security Management server to the local `state` directory, and install the Policy only if the fetched Policy is different from the Policy already installed.
`-f <filename>`	Fetch the Security Policy from the Security Management server listed in <filename>. If filename is not specified, the list in `conf/masters` is used.
`-c`	Cluster mode, get policy from one of the cluster members, from the Check Point High Availability (CPHA) kernel list.
`-i`	Ignore SIC information (for example, SIC name) in the database and use the information in `conf/masters`. This option is used when a Security Policy is fetched for the first time by a DAIP gateway from a 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 `targets` is not specified, or if `targets` is inaccessible, the Policy is fetched from `localhost`.

fw fetchlogs

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

Syntax

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

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

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

Fetching Current Log Data

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

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

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

Example The following command:

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

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

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

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

fw hastat

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

Syntax

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

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

fw isp_link

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

Syntax

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

Parameter	Description
target	The name of the Check Point Security Gateway.
link-name	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.

Syntax

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

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

Syntax

```
> 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

Syntax

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

Parameter	Description
-x	Use hexadecimal format
-l	Use long format

fw log

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

Syntax

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

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

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

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

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

Example

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

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

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

Example Output

```
14:56:39 reject jam.checkpoint.com >daemon alert src: veredr.checkpoint.com;
dst: jam.checkpoint.com; user: a; rule: 0; reason: Client Encryption: Access
denied - wrong user name or password  ; scheme: IKE; reject_category:
Authentication error; product: 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 a 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 75) and fw fetchlogs (on page 69).

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

Syntax

```
> fw logswitch [-audit] [<filename>]
> fw logswitch -h <hostage> [+|-][<filename>]
```

Parameter	Description
`-audit`	Does logswitch for the Security Management server audit file. This is relevant for local activation.
`<filename>`	The name of the file to which the log is saved. If no name is specified, a default name is provided.
`-h <hostage>`	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 `fw logswitch` command is executed) must be defined as one of `host`'s Security Management servers. In addition, you must initialize SIC between the Security Management server and the `host`.
`+`	Change a remote log and copy it to the local machine.

Parameter	Description
-	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 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.

Syntax

```
> fw lslogs [[-f <filename>] ...] [-e] [-s {<name>|<size>|<stime>|<etime>}]
[-r] [<machine>]
```

Parameter	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`.

Parameter	Description
-e	Display an extended file list. It includes the following data: • Size - The size of the file and its related pointer files together. • Creation Time - The time the Log File was created. • Closing Time - The time the Log File was closed. • Log File Name - The file name.
-s	Specify the sort order of the Log Files using one of the following sort options: • name - The file name. • size - The file size. • stime - The time the Log File was created. • etime - The time the Log File was closed. The default is stime.
-r	Reverse the sort order (descending order).
<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 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.

Syntax

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

Parameter	Description
-s	Sort merged file by log records time field.

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

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

fw monitor

Description Inspecting network traffic is an essential part of troubleshooting network deployments. `fw monitor` is a powerful built-in tool to simplify the task of capturing network packets at multiple capture points within the firewall 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 firewall 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.

Syntax

```
> 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>] [-h] -T
```

Parameter	Description
-u|s	**Printing the UUID or the SUUID:** The option -u or -s is used to print UUIDs or SUUIDs for every packet. Please note that it is only possible to print the UUID or the SUUID – not both.
-i	**Flushing the standard output:** Use to make sure that captured data for each packet is at once written to standard output. This is especially useful if you want to kill a running fw monitor process and want to be sure that all data is written to a file.
[-d] [-D]	**Debugging fw monitor:** The -d option is used to start fw monitor in debug mode. This will give you an insight into fw monitor's inner workings. This option is only rarely used outside Check Point. It is also possible to use -D to create an even more verbose output.
{-e <expr>|{-f <filter-file>|-}}	**Filtering fw monitor packets:** fw monitor has the ability to capture only packets in which you are interested. fw monitor filters use a subset of INSPECT to specify the packets to be captured. Set the filter expression: • on the command line using the -e switch. • by reading it from a file using the -f switch. • by reading it from standard input using the -f - switch.
-l <len>	**Limiting the packet length:** fw monitor lets you 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 lets you to capture only the headers of a packet (e.g. IP and TCP header) while omitting the actual payload. Therefore you can debug the communication without seeing the actual data transmitted. Another possibility is to keep the amount of data low. If you don't need the actual payload for debugging you can decrease the file site by omitting the payload. It's also very useful to reduce packet loss on high-loaded machines. fw monitor uses a buffer to transfer the packets from kernel to user space. If you reduce the size of a single packet this buffer won't fill up so fast.
-m <mask>	**Setting capture masks:** By default fw monitor captures packets before and after the virtual machine in both directions. These positions can be changed. This option allows you to specify in which of the four positions you are interested.

Parameter	Description
`-x <offset>[,<len>]`	**Printing packet/payload data:** In addition to the IP and Transport header `fw monitor` can also print the packets' raw data using the `-x` option. Optionally it is also possible to send all data that is written only to the screen the data written.
`-o <file>`	**Write output to file:** Save the raw packet data to a file in a standard (RFC 1761) format. The file can be examined using by tools like snoop, tcpdump or Ethereal. **Note -** The snoop file format is normally used to store Layer 2 frames. For "normal" capture files this means that the frame includes data like a source and a destination MAC address. `fw monitor` operates in the firewall kernel and therefore has no access to Layer 2 information like MAC addresses. Instead of writing random MAC addresses, fw monitor includes information like interface name, direction and chain position as "MAC addresses".
`-T`	Print time stamp in microseconds. `-T` is needed only when -o is not used. When -o is used the exact time is written to the snoop file by default as of Corsica.
`[[-pi <pos>] [-pI <pos>] [-po <pos>] [-pO <pos>] | -p all]]`	**Insert fw monitor chain module at a specific position:** In addition to capture masks (which give the ability to look at packets in a specific position) `fw monitor` has the ability to define where exactly in the firewall chain the packets should be captured. This can be defined using these options.
`-a`	**Use absolute chain positions:** If you use `fw monitor` to output the capture into a file (option `-o`), one of the fields written down to the capture file is the chain position of the `fw monitor` chain module. Together with a simultaneous execution of `fw ctl` chain you can determine where the packet was captured. Especially when using `-p` all you will find the same packet captured multiples times at different chain positions. The option `-a` changes the chain ID from a 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.
`[-ci <count>] [-co <count>]`	**Capture a specific number of packets:** `fw monitor` enables you to limit the number of packets being captured. This is especially useful in situations where the firewall is filtering high amounts of traffic. In such situations `fw monitor` may bind so many resources (for writing to the console or to a file) that recognizing the break sequence (Control-C) might take very long.
`-h`	Displays the usage.

Example The easiest way to use `fw monitor` is to invoke it without any parameter. This will output every packet from every interface that passes (or at least reaches) the Check Point Security

Gateway. The same packet appears 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]: 192.0.2.133 -> 192.0.2.2 (TCP) len=285 id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:I[285]: 192.0.2.133 -> 192.0.2.2 (TCP) len=285 id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:o[197]: 192.0.2.2 -> 192.0.2.133 (TCP) len=197 id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:O[197]: 192.0.2.2 -> 192.0.2.133 (TCP) len=197 id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:o[1500]: 192.0.2.2 -> 192.0.2.133 (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]: 192.0.2.133 -> 192.0.2.2 (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 192.0.2.133 to 192.0.2.2 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 a 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 cannot 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 monitor Filters

Description Use these expressions to help when you are filtering `fw monitor`.

Syntax `> fw monitor -e "accept <expression>;"`

Expressions for Protocols

Expression	Description
tcp	TCP
udp	UDP
icmp4	ICMP v4
icmp6	ICMP v6
esp	ESP

Expressions for Services

Expression	Description
http	HTTP (TCP port 80)
https	HTTPS (TCP port 443)
ftp	FTP (TCP port 20 or 21)
ssh	SSH (TCP port 22)
telnet	TELNET (TCP port 23)
smtp	SMTP (TCP Port 25)
pop3	POP3 (TCP port 110)
dns	DNS (TCP / UDP port 53
proxy	HTTP (TCP port 8080)

Expressions for VPN

For more information, see sk52421 http://supportcontent.checkpoint.com/solutions?id=sk52421.

Expression	Description	Check Point Description
ike	IKE (UDP port 500)	
natt	NAT-T (UDP port 4500)	
uenc	UDP encapsulation (UDP port 2746)	Check Point SecuRemote IPsec Transport Encapsulation Protocol
rdp	Check Point RDP (UDP port 259)	Proprietary Check Point "Reliable Data Protocol" (does not comply with RDP as specified in RFC 908/RFC 1151)
topo	Check Point Security Gateway SecuRemote Topology Requests (TCP port 264)	Topology Download from Security Gateway (by FWD daemon) to SecuRemote (build 4100 and higher) and SecureClient

Expression	Description	Check Point Description
`l2tp`	L2TP (TCP port 1701)	
`test`	Check Point Tunnel Testing (UDP port 18234)	Check Point tunnel testing application - Testing ICA through VPN by SecuRemote / SecureClient

Expressions for ICA (Internal Certificate Authority)

For more information, see sk52421 http://supportcontent.checkpoint.com/solutions?id=sk52421.

Expression	Description	Check Point Description
`pull`	Check Point Internal CA Pull Certificate Service (TCP port 18210)	Pulling certificates by Security Gateway from Security Management Server (by CPCA daemon)
`push`	Check Point Internal CA Push Certificate Service (TCP port 18211)	Pushing certificates from the Internal Certificate Authority (ICA) on Security Management Server (by CPD daemon) to Security Gateway
`crl`	Check Point Internal CA Fetch CRL and User Registration Services (TCP port 18264)	Protocol for Certificate Revocation Lists and registering users when using the Policy Server (needed when, e.g., Security Gateway is starting). See sk35292 http://supportcontent.checkpoint.com/solutions?id=sk35292.
`ica`	Check Point Internal CA Management Tools (TCP port 18265)	• Managing the ICA and central administration of Internal Certificate Authority (ICA) on the Security Management Server • Needs to be started separately with the Security Management Server and *cpca_client*

Expressions for Security Management Server

Expression	Description
`smc`	Port 18190 (SmartConsole)
`policy`	Port 18191 (Install policy)
`amon`	Port 18192 (AMON server)
`pslog`	Port 18231
`scv`	Port 18233 (Client SCV)

Expressions for Common Tasks

Expression	Description
`cpca`	Uses these expressions: `camgmt`, `pull`, `crl`, and `ica`

Expression	Description
`sic`	Uses these expressions: `cpca`, `push`, and `policy`
`vpnd`	Uses these expressions: `ike`, `natt`, `uenc`, `rdp`, `topo`, `l2tp`, `test`, `pslog` and `scv`
`vpn`	Uses expressions for standard site to site: `esp`, and `ike`
`vvpn`	Uses expressions for remote access: `natt`, and `https`
`multi`	Uses expressions for multi-portal: `https`, and port 444
`vpnall`	Uses expressions for all VPN services: `esp`, `vpnd`, `crl`, and `multi`
`vpn1`	Uses expressions for VPN and common test services: `vpn`, `vvpn`, `ftp`, and `ping`

Expressions to Exclude Background Traffic

Expression	Description
`no_term`	Uses expressions to exclude remote terminal: `not ssh` and `not telnet`
`no_mgmt`	Uses expressions to exclude Check Point management services: `not smc` and `not policy` and `not amon`

Example: `> fwmonitor -e "accept https;"`

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.

Syntax

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

Parameter	Description
`-opsec`	Only control connections are enabled.
`-no_opsec`	Only OPSEC control connections are enabled.
`-ssl`	The key is used for an SSL connection.
`-no_ssl`	The key is not used for an SSL connection.

Parameter	Description
`-k <num>`	The length of the first S/Key password chain for fwa1 authentication (Check Point's proprietary authentication protocol). The default is 7. When fewer than 5 passwords remain, the hosts renegotiate a chain of length 100, based on a long random secret key. The relatively small default value ensures that the first chain, based on a short password entered by the user, is quickly exhausted.
`-n <myname>`	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 `hostname` command.
`-p <psw>`	The key (password). If you do not enter the password on the command line, you will be prompted for it.
`<host>`	The IP address(es) or the resolvable name(s) of the other host(s) on which you are installing the key (password). This should be the IP address of the interface "closest" to the host on which the command is run. If it is not, you will get error messages such as the following: "`./fwd: Authentication with hostname for command sync failed`"

Comments This command is never used in a script.

fw repairlog

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

Syntax

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

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

fw sam

Description 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.

Syntax

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
-{i|j|n} all
```

Monitor SAM rules according to criteria:

```
> fw sam [-v][-s <sam server>][-S <server sic name>][-f <fw host>] -M
-{i|j|n} <Criteria>
```

Syntax

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

Parameter	Description
`-t <timeout>`	The time period (in seconds) for which the action will be enforced. The default is forever or until cancelled.
`-l <log>`	The type of the log for enforced actions can be one of the following: `nolog`, `long_noalert`, `long_alert`. The default is `long_alert`.
`-n`	Notify, or generate, a long-format log entry. Generates an alert when connections that match the specified services or IP addresses pass through the FireWall. This action does not inhibit or close connections.
`-i`	Inhibit (do not allow) new connections with the specified parameters. Each inhibited connection is logged according to log type. Matching connections will be *rejected*.
`-I`	Inhibit new connections with the specified parameters, and close all existing connections with the specified parameters. Each inhibited connection is logged according to the log type. Matching connections will be *rejected*.
`-j`	Inhibit new connections with the specified parameters. Each inhibited connection is logged according to the log type. Connections will be *dropped*.
`-J`	Inhibit new connections with the specified parameters, and close all existing connections with the specified parameters. Each inhibited connection is logged according to the log type. Connections will be *dropped*.
`-M`	Monitor the active SAM requests with the specified actions and criteria.
`all`	Get all active requests. For monitoring purposes only.

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

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

Possible combinations are:

```
src <ip>
dst <ip>
any <<ip>
subsrc <ip><netmask>
subdst <ip><netmask>
subany <ip><netmask>
srv <src ip><dest ip><service><protocol>
subsrv <src ip><src netmask><dest ip><dest netmask><service> <protocol>
subsrvs <src ip><src netmask><dest ip><service><protocol>
subsrvd <src ip><dest ip><dest netmask><service><protocol>
dstsrv <dest ip><service><protocol>
subdstsrv <dest ip><dest netmask><service><protocol>
srcpr <ip><protocol>
dstpr <ip><protocol>
subsrcpr <ip><netmask><protocol>
subdstpr <ip><netmask><protocol>
```

Syntax

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

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

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

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

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

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

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

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

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

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

This command cancels the command in the first example:

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

fw stat

Description Use `fw stat` to view the policy installed on the gateway, and which interfaces are being protected.

Note - The `cpstat` command is an enhanced version of `fw stat`

Syntax

```
> fw stat -l
> fw stat -s
```

Parameter	Description
`-l`	Show a long, detailed listing of the installed policies.
`-s`	Shows a short summary of the installed policies.

Examples

```
> fw stat
HOST      POLICY        DATE
localhost Standard      18Apr2012 15:01:51 :  [>eth0] [<eth0]
```

Two interfaces are being protected. The arrows show the direction of the packets.

After the policy is uninstalled, the output becomes:

```
> fw stat
HOST      POLICY     DATE
localhost -          -                :   >eth0   <eth0
```

This shows that there is no policy installed, and the interfaces are not protected.

fw tab

Description The `fw tab` command shows data from the kernel tables, and lets you change the content of dynamic kernel tables. You cannot change the content of static kernel tables.

Kernel tables (also known as State tables) store data that the Firewall and other modules in the Security Gateway use to inspect packets. These kernel tables are the "memory" of the virtual computer in the kernel and are a critical component of Stateful Inspection. The kernel tables are dynamic hash tables in the kernel memories.

Syntax

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

Parameter	Description
`- t <table>`	Specifies a table for the command.
`-s`	Shows a short summary of the table (s) data.
`-c`	Shows formatted table information in common format.
`-f`	Shows a formatted version of the table data. Each table can use a different style.
`-o <filename>`	Outputs CL formatted file called `<filename>`.You can open the file with `fw log` and other commands or processes that can read FW log formats.
`-r`	Resolves IP addresses in formatted output.
`-u`	Show unlimited table entries.
`-m <maxval>`	Sets the maximum table entries that are shown to `<maxval>`.

Parameter	Description
-a|-x	Adds (-a) or removes (-x) an entry from the specified table. Include the -t <table> parameter when you run the fw tab command with the -a and -x parameters. You cannot run these parameters on remote appliances or servers. **Caution** - If you use the -a and -x parameters incorrectly, you can cause the appliance or server to become unstable.
-e <entry>	One or more entries that you add or remove from the table.
-y	Do not show a prompt to users before they run commands.
[<hostname>]	One or more target appliances or servers for the fw tab command. If you do not use this parameter, the default setting is localhost.

Example `> fw tab -t arp_table -a -e "1,2,3,4,5"`
Adds an entry: `<00000001,00000002,00000003,00000004,00000005,>` to `arp_table`

`fw tab - m 100 -r sample-gw`

Comments If a table has the `expire` attribute, when you use the -a parameter to add entries, the default table timeout is added.

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.

Syntax

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

Parameter	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.

Syntax

```
> 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`.

Syntax

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

Parameter	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
`-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`.
`-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.

Syntax

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

Syntax

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]
```

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

Comments Note:

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

Syntax

```
gw> fwm dbload [-a|-c <conffile>] [<targets>]
```

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

Syntax

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

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

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

fwm getpcap

Description `fwm getpcap` command line fetches the packet capture.

Syntax `> fwm getpcap -g <gw> -u <cap id> [-p <path>] [-c <domain>]`

Parameter	Description
`-g <gw>`	Host name of the gateway
`-u <cap id>`	Capture UID
`-p <path>`	Output pathname
`-c <domain>`	Host name of the Domain Management Server

Note - This command only works with IPS packet captures stored on the Gateway in $FWDIR//opt/CPsuite-R77/fw1/log/captures_repository. It does not work with other blades such as Anti-Bot and Anti-Virus that store packet captures in `$FWDIR/log/blob`.

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.

Syntax `> fwm load [-p <plug-in>] [-S] <rulebase> <targets>`

Parameter	Description
`-S`	The targets are UTM-1 Edge gateways.

Parameter	Description
`-p <plug-in>`	Specifies the product name `<plug-in>` if applicable.
`rulebase`	A Rule Base created by the GUI. Specify the name of the rulebase, such as `Standard` (case sensitive).
`<targets>`	Execute command on the designated target.

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

```
fwm load Standard johnny
```

fwm lock_admin

Description View and unlock locked administrators.

Syntax >`fwm lock_admin [-v][-u <administrator>][-ua]`

Parameter	Description
`-v`	View the names of all locked administrators.
`-u <administrator>`	Unlock a single administrator.
`-ua`	Unlock all locked administrators.

fwm logexport

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

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

Parameter	Description
`-d <delimiter>`	Set the output delimiter. The default is a semicolon (`;`).
`-i <filename>`	The name of the input Log file. The default is the active Log file, `fw.log`
`-o <outputfile>`	The name of the output file. The default is printing to the screen.
`-n`	Do not perform DNS resolution of the IP addresses in the Log file (this option significantly speeds the processing).
`-p`	Do not perform service resolution. A service port number is displayed.
`-f`	If this is the active Log file (`fw.log`), wait for new records and export them to the ASCII output file as they occur.

Parameter	Description
-m {initial|semi|raw}	This flag specifies the unification mode. • initial - the default mode. Complete the unification of log records; that is, output one unified record for each id. • semi - step-by-step unification, that is, for each log record, output a record that unifies this record with all previously-encountered records with the same id. • raw - output all records, with no unification.
-a	Show account records only (the default is to show all records).

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

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

```
[Fields_Info]
included_fields = field1,field2,field3,<REST_OF_FIELDS>,field100
excluded_fields = field10,field11
```

note that:

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

Format:

The `fwm logexport` output appears in tabular format. The first row lists the names of all fields included in the subsequent records. Each of the subsequent rows consists of a single log record, whose fields are sorted in the same order as the first row. If a record 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_messa
ge:;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).

Syntax `> fwm sic_reset`

fwm unload <targets>

Description Uninstall the currently loaded Inspection Code from selected targets.

Syntax `> fwm unload <targets> [-all|-c <conffile>]`

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

fwm ver

Description `fwm ver` shows the build number.

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

Parameter	Description
`-f <filename>`	Exports the build number data to a file

fwm verify

Description The `fwm verify` command verifies the specified policy package without installing it.

Syntax `> fwm verify <policy>`

Parameter	Description
`<policy>`	The name of an available policy package.

GeneratorApp

Description Generate a report for SmartReporter. Both command line parameters are required. Run this command from Expert mode.

Syntax `# GeneratorApp <Directory> <ReportID>`

Parameter	Description
`<Directory>`	The result directory (that is, the location at which the result is placed).

Parameter	Description
<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 "Predefined Reports" in the *R80.10 SmartReporter Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=24813.

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.

Syntax

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

Parameter	Description
-s <ipaddr>	The IP address (in dot format) of the ELA Proxy to be contacted.
-o	Print the alert log received by inet_alert to stdout. Use this option when inet_alert is part of a pipe.
-a <auth_type>	The type of connection to the ELA Proxy. One of the following values: • **ssl_opsec**. Means the connection is authenticated and encrypted, (Default) • **auth_opsec**. Means the connection is authenticated. • **clear**. Means the connection is neither authenticated nor encrypted.
-p <port>	The ELA proxy's port number. Default is 18187.

Parameter	Description
`-f <token value>`	A field to be added to the log, represented by a `token-value` pair as follows: • `token` is the name of the field to be added to the log. `token` may not contain spaces. • `value` is the field's value. `value` may not contain spaces. This option may be used multiple times to add multiple `token-value` pairs to the log. If `token` is a reserved log field name, the specified field's value will appear in the corresponding column in SmartView Tracker. Otherwise, the `token-value` pair will be displayed in the **Info.** column in SmartView Tracker.
`-m <alerttype>`	The alert to be triggered at the ISP site. This alert overrides the alert specified in the log message generated by the alert daemon. The response to the alert is handled according to the actions specified in the ISP's Security Policy: The following alerts execute the OS commands defined in the corresponding fields of the **Log and Alert** tab of the **Properties Setup** window in **Global Properties**: • **alert**. `Popup alert command.` • **mail**. `Mail alert command.` • **snmptrap**. `SNMP trap alert command.` • **spoofalert**. `Anti-spoof alert command.` The following NetQuota and ServerQuota alerts execute the OS commands specified in: `$FWDIR/conf/objects.C:` `value=clientquotaalert. Parameter=clientquotaalertcmd`

Return Value

Exit status	Description
`0`	Execution was successful.
`102`	Undetermined error.
`103`	Unable to allocate memory.
`104`	Unable to obtain log information from `stdin`.
`106`	Invalid command line arguments.
`107`	Failed to invoke the OPSEC API.

Example

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

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

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

Syntax

```
# ldapcmd -p {<process_name>|all} <command> [-d debug_level] [command_arg]
```

Parameter	Description
`-p`	Run a specified process or run all processes.
`<command>`	These are the valid values for the `command` parameter: • `cacheclear {all|UserCacheObject|TemplateCacheObject|TemplateExtGrpCacheObject}` • `cachetrace {all|UserCacheObject|TemplateCacheObject|TemplateExtGrpCacheObject}` • `stat {print_interval {<reset interval time in secs>|0} [stop statistics]}` • `log {on|off}`
`log`	Specify whether or not to create LDAP logs.

ldapcompare

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

Syntax

```
# ldapcompare -d [<options>] dn <attribute> <value>
```

Parameter	Description
`-d`	Debug flag
`<options>`	See below
`dn`	The DN object
`attribute`	The attribute of the DN object
`value`	The value of the attribute of the DN object

The `ldapcompare` options are as follows:

- `-u` – Include user-friendly entry names in the output.
- `-d <level>` – Set LDAP debugging level to "level".
- `-F sep` –Print "sep" instead of "=" between attribute names and values.
- `-f <file>` – Perform sequence of compares listed in "file".
- `-D <binddn>` – Bind DN.
- `-w <passwd>` – Bind password (for simple authentication).
- `-h <host>` – LDAP server.
- `-p <port>` – Port on the LDAP server.
- `-T <timeout>` – Client side timeout for all operations (in milliseconds).
- `-l <time limit>` – Server Side time limit (in seconds) for compare.
- `-z <size limit>` – Server Side size limit (in entries) for compare.

ldapconvert

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

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

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

Syntax

```
> 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>]
```

Parameter	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 `ldapcomvert extra options` are as follows:

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

Comments It is recommended to make a backup of 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 is 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.

Syntax

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

Parameter	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.

Syntax

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

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

The following are the attributes for options:

- -A –Retrieve attribute names only (without values).
- -B –Do not suppress printing of non-ASCII values.
- -D bindDN –The DN to be used for binding to the LDAP server.
- -F separator –Print separator between attribute name and value instead of "=".
- -h host –The LDAP server identified by IP address or resolvable name.
- -l timelimit –The server side time limit for search, in seconds.
- -p portnum –The port number. The default is standard LDAP port 389.
- -S attribute –Sort the results by the values of attribute.
- -s scope –One of the following: "base", "one", "sub".
- -b –Base distinguished name (DN) for search.
- -t –Write values to files in /tmp. Each attribute-value pair is written to a separate file, named: /tmp/ldapsearch-<attribute>-<value>.
- For example, for the fw1color attribute, the file written is named.
- /tmp/ldapsearch-fw1color-a00188.
- -T timeout – Client-side timeout (in milliseconds) for all operations.

- `-u` – Show "user friendly" entry names in the output. For example, show "`cn=Babs Jensen, users, omi`" instead of "`cn=Babs Jensen, cn=users,cn=omi`"
- `-w password` – The password.
- `-Z` – Encrypt using SSL.
- `-z sizelimit` –Server-side size limit for search, in entries.

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

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

log_export

Description `log_export` is a utility that allows you to transfer Log data to an external database. This utility behaves as a LEA client. LEA (Log Export API) enables 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

Syntax

```
# 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]
```

Parameter	Description
`-f <conf_file>`	The Configuration File from which `log_export` reads the Log file parameters. If `conf_file` is not specified, the default Configuration File `log_export.conf`, located in the current working directory.
`-l <lea_server_ip_address>`	The IP address of the LEA server.
`-g <log_file_name>, <log_file_name>,...`	A comma separated list of log file names from where the logs will be taken.
`-t <database_table_name>`	The name of the table in the database to which the logs will be added.
`p <database_password>`	The database login password. If you do not want to specify the password in the Configuration File for security reasons, you can enter the password using the command line where it will not be saved anywhere.
`-h`	Display `log_export` help.

Parameter	Description
-d	Display debugging information.

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

Comments Only Oracle database is currently supported.

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

- the ORACLE_HOME environment variable is set correctly.
- $ORACLE_HOME/lib is located in the PATH environment variable on the 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.

Syntax

```
# 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]
```

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

Syntax

```
# 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> >
```

Parameter	Description
`-d`	debug file.
`-operation add`	Add entry to database.
`<-name object_name>`	Enter the name of the gateway object.
`<-ip module_ip>`	Enter the IP Address of the gateway
`<-TTL Time-To-Live>`	The relative time interval (in seconds) during which the entry is valid. A value of zero specifies "unlimited".
`- operation fetch`	Get entry from database.
`- operation delete`	Delete entry from database.
`- operation list`	List all the database entries.
`- operation sync`	Synchronize the database.

sam_alert

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

Syntax

```
sam_alert [-o] [-v] [-s <sam_server>] [-t <timeout>] [-f <fw_host1>
<fw_host2>...]
[-C] [-n|-i|-I -src|-dst|-any|-srv]
```

Parameter	Description
`-o`	Prints the input of this tool to the standard output (for pipes).
`-v`	Turns on verbose mode (of the `fw sam` command).
`-s <sam_server>`	The sam server to be contacted. Localhost is the default.

Parameter	Description
-t <timeout>	The time period, in seconds, for which the action will be enforced. The default is forever.
-f <fw_host>	Identifies the FireWalls to run the operation on. Default is "all FireWalls."
-C	Cancels the specified operation.
-n	Notify every time a connection that matches the specified criteria passes the Firewall.
-i	Inhibit connections that match the specified criteria.
-I	Inhibit connections that match the specified criteria and close all existing connections that match the criteria.
-src	Match the source address of connections.
-dst	Match the destination address of connections.
-any	Match either the source or destination address of the connection.
-srv	Match specific source, destination, protocol and service.

svr_webupload_config

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 *R80.10 SmartReporter Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=24813.

Syntax

```
# svr_webupload_config [-i <perl_int_loc>]
[-p <rep_dir_root>]
```

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

VPN Commands

In This Section:

Overview

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 crl_zap

Description Erase all Certificate Revocation Lists (CRLs) from the cache.

Syntax

```
> 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 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.

Syntax

```
> vpn crlview -obj <object name> -cert <certificate name>
> vpn crlview -f <filename>
> vpn crlview -view
```

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

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

vpn debug

Description 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`

Syntax

```
> 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

Parameter	Description
`on`	Turns on high level VPN debugging.

Parameter	Description
`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.

Syntax

```
> vpn drv on|off
> vpn drv stat
```

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

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.

Syntax

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

Parameter	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 `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.

Syntax

```
> vpn macutil <username>
```

Example `vpn macutil John`

Output

```
20-0C-EB-26-80-7D, "John"
```

vpn nssm_toplogy

Description Generate and upload a topology (in NSSM format) to NSSM server for use by clients.

Syntax

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

Parameter	Description
-url	URL of the 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

Parameter	Description
-action	Specifies the action the Symbian client should take if the packet is not destined for an IP address in the VPN domain. Legal options are **Bypass** (default) or **Drop**
-print_xml	The topology is in XML format. This flag writes that topology to a file in XML format.

vpn overlap_encdom

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

Syntax

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

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

Example `vpn overlap_encdom communities`

Output

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

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

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

vpn sw_topology

Description Download the topology for a Safe@ or Edge gateway.

Syntax

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

Parameter	Description
-d	Debug flag
-dir	Output directory for file
-name	Nickname of site which appears in remote client
-profile	Name of the Safe@ or Edge 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.

Syntax

```
> 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 (GW) or user (Client)
(4)              List all IPsec SAs for a given peer (GW) or user (Client)
(5)              Delete all IPsec SAs for a given peer (GW)
(6)              Delete all IPsec SAs for a given User (Client)
(7)              Delete all IPsec+IKE SAs for a given peer (GW)
(8)              Delete all IPsec+IKE SAs for a given User (Client)
(9)              Delete all IPsec SAs for ALL peers and users
(0)              Delete all IPsec+IKE SAs for ALL peers and users

(Q)              Quit

*****************************************
```

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.

Syntax

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

Parameter	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.

ClusterXL Commands

In This Section:

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 <computer id>] [-p <policy id>] [-b <db id>] [-n <ClusterXL num>]
[-c <ClusterXL 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
cphaconf stop_all_vs
```

Syntax

Parameter	Description
`set_ccp <broadcast/multicast>`	Sets whether ClusterXL Control Protocol (CCP) packets should be sent with a broadcast or multicast destination MAC address. The default behavior is multicast. The setting created using this command will survive reboot. **Note:** The same value (either broadcast or multicast) should be set on all ClusterXL members.
`stop_all_vs`	Stops the ClusterXL product on all Virtual Systems on a VSX Gateway.

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

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

cphastart

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

cphastop

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

Identity Awareness Commands

In This Section:

Introduction

These terms are used in the CLI commands:

- **PDP** - The process on the Security Gateway responsible for collecting and sharing identities.
- **PEP** - The process on the Security Gateway responsible for enforcing network access restrictions. Decisions are made according to identity data collected from the PDP.
- **AD Query** - AD Query is the module responsible for acquiring identities of entities (users or computers) from the AD (Active Directory). AD Query was called Identity Logging in previous versions and in some cases is also referenced as AD Log. The adlog is the command line process used to control and monitor the AD Query feature.

The PEP and PDP processes are key components of the system. Through them, administrators control user access and network protection.

AD Query can run either on a Security Gateway that has been enabled with Identity Awareness or on a Log Server. When it runs on a Security Gateway, AD Query serves the Identity Awareness feature, and gives logging and policy enforcement. When it runs on a Log Server, AD Query gives identity logging. The command line tool helps control users' statuses as well as troubleshoot and monitor the system.

pdp

Description These commands control and monitor the PDP process.

Syntax `# pdp [command]... <parameter>`

Parameter	Description
`<none>`	Display available options for this command and exit
`debug`	Control debug messages
`tracker`	Tracker options
`connections`	pdp connections information
`network`	pdp network information
`status`	pdp status information

Parameter	Description
`control`	pdp control commands
`monitor`	Display monitoring data
`update`	Recalculate users and computers group membership (deleted accounts will not be updated)
`ad`	Operations related to AD Query
`timers`	Show pdp timers information
`nested_groups`	Nested groups configuration
`auth`	Authentication or authorization options
`ifmap`	Monitor or control IFMAP
`vpn`	Display connected vpn gateways that send vpn client identity data
`radius`	RADIUS accounting options
`idc`	Operations related to Identity Collector
`tasks_manager`	The task manager menu

pdp monitor

Description Lets you monitor the status of connected sessions. You may perform varied queries according to the usage below to get the output you are interested in.

Syntax `# pdp monitor <parameter> <option>`

Parameter	Description
`all`	Display information for all connected sessions
`user <user name>`	Display session information for the given user name
`ip <IP address>`	Display session information for the given IP address
`machine <computer name>`	Display session information for the given computer name
`mad`	Display all sessions that relate to a managed asset (i.e. all sessions that successfully performed computer authentication)

Parameter	Description
`client_type [unknown|portal|"Identity Agent"|"AD Query"]`	Display all sessions connecting via the given client type Possible client types are: • Unknown - User was identified by an unknown source • Portal - User was identified by the Captive Portal • Identity Agent - User/computer was identified by an Identity Awareness Agent • AD Query - User was identified by AD Query
`groups <group name>`	Display all sessions of users / computers that are members of the given group name
`cv_ge <version>`	Display all sessions that are connected with a client version that is higher than (or equal to) the given version
`cv_le <version>`	Display all sessions that are connected via a client version that is lower than (or equal to) the given version.
`s_port`	Print sessions filtered by assigned source port (MUH sessions only)
`network`	Print sessions filtered by a network wild card (example: 192.168.72.*)
`user_exact`	Print sessions filtered by the exact user
`machine_exact`	Print sessions filtered by the exact machine name

Example

```
pdp monitor ip 192.0.2.1
```

Shows the connected user behind the given IP address (192.0.2.1).

Note - The last field "Published " indicates whether the session information was already published to the Gateway PEPs whose IP addresses are listed.

pdp connections

Description These commands assist in monitoring and synchronizing the communication between the PDP and the PEP.

Syntax `pdp connections <argument>`

Argument	Description
`pep`	Shows the connection status of all the PEPs that should be updated by the current PDP
`ts`	Shows a list of terminal servers that are connected
`ifmap`	Shows a list of the active IFMAP sessions

pdp control

Description Provides commands to control the PDP process.

Syntax `# pdp control <parameter> <option>`

Parameter	Description
`revoke_ip <IP address>`	Log out the session that is related to the given IP.
`revoke_pt_key <session id.>`	Revoke the packet tagging key if one exists.
`sync`	Force an initiated synchronization operation between the PDPs and the PEPs. When running this command, the PDP will inform its related PEPs the up-to-date information of all connected sessions. At the end of this operation, the PDP and the PEPs will contain the same and latest session information.

pdp network

Description Shows information about network related features.

Syntax `# pdp network <parameter>`

Parameter	Description
`info`	Display a list of networks known by the PDP.
`registered`	Display the mapping of a network address to registered gateways (PEP module).

pdp debug

Description Activates and deactivates the debug logs of the PDP daemon.

Syntax `# pdp debug <parameter> <option>`

Parameter	Description
`on`	Turn on the debug logs (should be followed by the command "set" to determine the required filter).
`off`	Turn off the debug logs.
`set <topic name> [critical|surprise| important|events| all]…`	Filter the debug logs that would be written to the debug file according to the given topic and severity **Best Practice -** For debug it is recommended to run: `pdp debug set all all` Note that you can place a number of topics and severity pairs. For example: `topicA severityA topicB severityB ...`
`unset <topic name>…`	Unset a specific topic or topics.

Parameter	Description
`stat`	Show the status of the debug option.
`reset`	Reset the debug options of severity and topic. The debug is still activated after running this command.
`rotate`	Rotate the log files (increase the index of each log file) so that the current log file that will be written is the PDP log. For example, pdpd.elg becomes pdpd.elg.0 and so on.
`ccc [on|off]`	Allows enabling or disabling writing of the CCC debug logs into the PDP log file.

Important - Activating the debug logs affects the performance of the daemon. Make sure to turn off the debug after you complete troubleshooting.

pdp tracker

Description Adds the TRACKER topic to the PDP logs (on by default). This is very useful when monitoring the PDP-PEP identity sharing and other communication on distributed environments. This can be set manually by adding the TRACKER topic to the debug logs.

Syntax `# pdp tracker <parameter>`

Parameter	Description
`on`	Turns on logging of TRACKER events in the PDP log.
`off`	Turns off the logging of TRACKER events in the PDP log.

pdp status

Description Displays PDP status information such as start time or configuration time.

Syntax `# pdp status <parameter>`

Parameter	Description
`show`	Display PDP information.

pdp update

Description Initiates a recalculation of group membership for all users and computers. Note that deleted accounts will not be updated.

Syntax `# pdp update <parameter>`

Parameter	Description
`all`	Recalculate group membership for all users and computers.

pdp ad associate

Description For AD Query, adds an identity to the Identity Awareness database on the Security Gateway. The group data must be in the AD.

Syntax `# pdp ad associate ip <ip> u <username> d <domain> [m <machine>] [t <timeout>] [s]`

Parameter	Description
`ip <ip>`	IP address for the identity.
`u <username>`	Username for the identity.
`m <machine>`	Computer that is defined for the identity.
`d <domain>`	Domain of the ID server.
`t <timeout>`	Timeout setting for the AD Query (default is 5 hours).
`s`	Associates `u <username>` and `m <machine>` parameters sequentially. First the `<machine>` is added to the database and then the `<username>`.

pdp ad disassociate

Description Removes the identity from the Identity Awareness database on the Security Gateway. Identity Awareness does not authenticate a user that is removed.

Syntax `# pdp ad disassociate ip <ip> {u <username>|m <machine>} [r {probed|override|timeout}]`

Parameter	Description
`ip <ip>`	IP address for the identity
`u <username>`	Username for the identity
`m <machine>`	Computer that is defined for the identity
`t <timeout>`	Timeout setting for the AD Query (default is 5 hours)
`r {probed|override|timeout}`	Reason that is shown in the **Logs & Monitor** > **Logs** tab

pep

Description Provides commands to control and monitor the PEP process.

Syntax `# pep [command]... <parameter>`

Parameter	Description
`tracker`	Tracker options.
`show`	Display PEP information.

Parameter	Description
debug	Control debug messages.
control	Control and set PEP parameters.

pep show

Description Displays information regarding pep status.

Syntax `# pep show <parameter> <option>`

pep show user

Description Enables monitoring the status of sessions that are known to the PEP. You can perform varied queries according to the usage below to get the output you are interested in.

Syntax `# pep show user all`

Parameter	Description
all	Display all sessions with information summary.

Query Syntax `# pep show user query <parameter>`

Parameter	Description
usr <username>	Display session information for the given user name.
mchn <computer name>	Display session information for the given computer name.
cid <IP>	Display session information for the given IP.
uid <uidString>	Display session information for the given session ID.
pdp <IP>	Display all session information that was published from the given PDP IP.
ugrp <group>	Display all sessions of users that are members of the given user group name.
mgrp <group>	Display all sessions of computers that are members of the given computer group name.

Note - You can use multiple query tokens (parameters) at once to create a logical "AND" correlation between them. For example, to display all users that have a sub string of "jo" AND are part of the user group "Employees" then you can use:

```
# pep show user query usr jo ugrp Employees
```

pep show pdp

Description Enables monitoring the communication channel between the PEP and the PDP. The output displays the connect time and the number of users that were shared through the connection.

Syntax `# pep show pdp <parameter>`

Parameter	Description
`all`	List all the PDPs that are connected to the current PEP with the relevant information.
`id <IP>`	Display connection information of the given PDP IP.

pep show stat

Description Shows the last time the daemon was started and the last time a policy was received.

Important - Each time the daemon starts, it loads the policy and the two timers (Daemon start time and Policy fetched at) will be very close.

Syntax `# pep show stat`

pep show network

Description Shows network related information.

Syntax `# pep show network <parameter>`

Parameter	Description
`pdp`	Shows information about mapping between the network and PDPs.
`registration`	Shows which networks this PEP is registered to.

pep debug

Description See pdp debug (on page 126).

adlog

Description Provides commands to control and monitor the AD Query process.

When AD Query runs on a Security Gateway, AD Query serves the Identity Awareness feature that gives logging and policy-enforcement. In this case the command line is: `adlog a <argument>` (see below for options)

When it runs on a Log Server, AD Query gives identity logging. In this case, the command line is: `adlog l <argument>`. Note: the `l` in `adlog l` is a lowercase L.

Options for `adlog a` and `adlog l` are identical.

Syntax `# adlog {a|l} <command>... <argument>`

Parameter	Description
`<none>`	Display available options for this command and exit.

Parameter	Description
`{a|l}`	Set the working mode: `adlog a` - if you are using AD Query for Identity Awareness. `adlog l` - if you are using a Log Server (identity logging)
`query`	See sections below.
`debug`	
`dc`	
`statistics`	
`control`	
`control muh`	
`control srv_account`	

adlog query

Description Shows the database of identities acquired by AD Query, according to the given filter.

Usage `adlog [a|l] query <argument>`

Syntax

Parameter	Description
`ip <IP address>`	Filters identities relating to the given IP.
`string <string>`	Filters identity mappings according to the given string.
`user <user name>`	Filters identity mappings according to a specific user.
`machine <computer name>`	Filters identity mappings according to a specific computer.
`all`	No filtering, shows the entire identity database.

Example

```
adlog a query user jo
```

Shows the entry that contains the string "jo" in the user name.

adlog dc

Description Shows status of connection to the AD domain controller.

Usage `adlog [a|l] dc`

Syntax None

adlog statistics

Description Displays statistics regarding NT Event Logs received by adlog, per IP and by total. It also shows the number of identified IPs.

Usage `adlog [a|l] statistics`

Syntax None

adlog debug

Description Turns on/off debug flags for controlling the debug file. The debug file is located at $FWDIR/log/pdpd.elg (for Identity Awareness on a Security Gateway) or $FWDIR/log/fwd.elg (for identity logging on a log server).

Usage `adlog [a|l] debug <parameter>`

Syntax

Parameter	Description
`on`	Turn on debug.
`off`	Turn off debug.
`mode`	Show debug status (on/off).
`extended`	Turn on debug and add extended debug topics.

adlog control

Description Sends control commands to AD Query.

Usage `adlog {a|l} control <parameter>`

Syntax

Parameter	Description
`stop`	Stop AD Query. New identities are not acquired via AD Query.
`reconf`	Send a reconfiguration command to AD Query, which means it resets to policy configuration as was set in SmartDashboard.

adlog service_accounts

Description Shows accounts that are suspected to be "service accounts". Service accounts are accounts that don't belong to actual users, rather they belong to services running on a computer. They are suspected as such if they are logged in more than a certain number of times.

Usage `adlog [a|l] service_accounts`

Syntax None

IPS Commands

In This Section:

Overview

Description - IPS commands let you configure and show the IPS on the Security Gateway without installing a new policy.

Comments - Changes in the IPS configuration are not persistent. If you install a policy or restart the computer, the changes are deleted.

ips bypass stat

Description - Shows the status of the bypass mode.

Usage- `ips bypass stat`

Comments - Shows this information:

- IPS bypass mode - on or off
- CPU thresholds
- Memory thresholds

ips bypass on|off

Description - Manages IPS bypass. When IPS bypass is enabled:

- If the CPU or memory goes above the `high` threshold, IPS enters bypass mode and is automatically disabled.
- When the CPU or memory goes below the `low` threshold, IPS exits bypass mode and is automatically enabled.

Usage - `ips bypass {on|off}`

Syntax

Parameter	Description
`on`	IPS bypass is enabled.
`off`	IPS bypass is disabled.

Example- `ips bypass on`

ips bypass set

Description - Configures the thresholds for the `ips bypass` command.

Usage - `ips bypass set {cpu|mem} {low|high} <th>`

Syntax

Parameter	Description
`cpu`	Configure the CPU threshold
`mem`	Configure the memory threshold.
`low`	Configure the lower threshold to exit bypass mode.
`high`	Configure the higher threshold to enter bypass mode.
`<th>`	The CPU or memory threshold value.

Example - `ips bypass set cpu low 80`

ips debug

Description - Shows the IPS debug information.

Usage - `ips debug [-e <filter>] -o <outfile>`

Syntax

Parameter	Description
`-e`	Filters which packets are captured.
`<filter>`	Uses a subset of INSPECT to specify which packets are captured.
`-o <outfile>`	Outputs the debug information to the file `<outfile>`.

Example - `ips debug -o sampledebug`

ips pmstats

Description - Shows statistics about the pattern matcher. These statistics are shown for each

pattern:

- Memory
- CPU usage
- Compilation time

Usage - `ips pmstats -o <outfile>`

Syntax

Parameter	Description
`-o <outfile>`	Outputs the debug information to the file `<outfile>`.

Example - `ips pmstats -o samplefile`

ips pmstats reset

Description - Resets the data that is collected to calculate the `pmstat` statistics.

Usage - `ips pmstats reset`

ips refreshcap

After installing a new policy, IPS captures the first packet for each protection and saves it in the packet capture repository.

Description - Refreshes the packet capture repository. IPS designates the next packet of each protection as the first packet. The new first packet replaces the previous one in the packet capture repository.

Usage - `ips refreshcap`

ips stat

Description - Shows the IPS status of these items:

- IPS enabled or disabled
- Active profile
- Update version
- Global detect mode - on or off
- Bypass mode - on or off

Syntax - `ips stat`

ips stats

Description - Print IPS and Pattern Matcher performance statistics. Without arguments, runs on current gateway for 20 seconds. This is a resource intensive command and should not be run on a system experiencing a high load.

Usage - `ips stats [<ip_address> -m] [-g <seconds>] [<ip_address> <seconds>]`

Syntax

Parameter	Description
`-m`	Analyzes input statistics file from gateway. Give IP address of the gateway. Run from the Security Management Server.
`-g`	Collect statistics for current gateway.
`seconds`	Period in which statistics are gathered

Examples

```
ips_stats 192.0.2.14 40
```

Run statistics on gateway with address 192.0.2.14 for 40 seconds

```
ips_stats -g 30
```

Run the statistics on the current gateway for 30 seconds

```
ips_stats 192.0.2.14 -m
```

Analyze the statistics taken from the gateway with address 192.0.2.14