GUIDE

STORMSHIELD NETWORK SECURITY

CLI CONSOLE / SSH COMMANDS REFERENCE GUIDE

Version 4.8.9

Document last update: June 3, 2025

Reference: sns-en-cli_console_ssh_commands_reference_guide-v4

Table of contents

Introduction ... 5
CONTENTS ... 5

CHAPTER1: Category Description ... 6
HARDWARE ... 6
LOW LEVEL CONFIGURATION ... 6
FUNCTIONALITIES ... 6
HIGH LEVEL CONFIGURATION MANAGEMENT ... 7
FACTORY TOOLS ... 8
DAEMON ... 8
MISCELLANEOUS ... 9

CHAPTER2: Commands Description ... 12
ALIVECTL ... 12
ALIVED ... 15
ARKEEPALIVE ... 15
ARPRESET ... 16
ARPSYNC ... 16
ASQD ... 17
ASQSTART ... 17
AUTHD ... 18
AUTOBACKUP.SH ... 18
AUTOUPDATE ... 18
AVCTL ... 19
AVD ... 21
BACKUPINFO ... 21
BACKUPRESTORE ... 22
BIRD4 ... 22
BIRD6 ... 23
BIRD ... 23
BIRDC ... 23
BIRDC4 ... 24
BIRDC6 ... 24
BUILDDHCPD ... 24
BUILDDIALUP ... 25
BUILDDNS ... 25
BUILDEVENT ... 26
BUILDFILTER ... 26
BUILDHA ... 27
BUILDIPSEC ... 27
BUILDLDAPCONF ... 28
BUILDNTP ... 28
BUILDOPENVPN ... 29
BUILDSNMP ... 29
BUILDSSH ... 30
BUILDWIFI ... 30
CAD ... 30
CERTENROL ... 31
CERTINFO ... 32
CERTREQCTL ... 33
CERTREQD ... 34
CHECK_DOWNGRADE_VERSION ... 34
CHECKCRL ... 35
CHECKDB ... 35
CHECKFS ... 37
CHECKFW ... 37
CHECKINTEGRITY ... 38
CHECKINTERNET ... 38
CHECKPIM ... 39
CHECKVERSION ... 39
CHPWD ... 40
CLAMAVD ... 40
CLAMDEFAULT ... 41
CLASSIFYHOST ... 41
CLASSIFYURL ... 42
CLEANFW ... 42
CLEANPATTERN ... 43
CLEARLOG ... 43
CLEARUNWANTEDFILES ... 44
COLLECTORCTL ... 44
COLLECTORD ... 45
CONFBYPASS ... 45
CONFTUNING ... 46
COROSYNC ... 46
CRLINFO ... 47
CURLTOOL ... 47
DATE ... 48
DDNSCLIENT ... 49
DECBACKUP ... 49
DEFAULTCONFIG ... 50
DHCLIENT ... 51
DHCLIENT-SCRIPT ... 52
DHCPD ... 52
DHCPINFO ... 53
DHCRELAY ... 53
DHLEASE-SCRIPT ... 54
DIALUPSTATE ... 54
DKILL ... 54
DMIDECODE ... 55
DNSCACHE ... 55
DSTAT ... 56
DUMPROOT ... 57
ENALIVED ... 57
ENANTIVIRUS ... 58
ENASQ ... 58
ENAUTH ... 59
ENAUTHD ... 59

ENBIRD 60
ENBYPASS 60
ENCBACKUP 61
ENCONSOLE 61
ENDHCP 62
ENDHCRELAY 62
ENDIALUP 63
ENDNS 63
ENEVENT 63
ENFILTER 64
ENGATEMON 65
ENHA 65
ENKEYBOARD 65
ENLDAP 66
ENLOCK 66
ENLOG 67
ENMULTICAST 67
ENNETWORK 68
ENNTP 69
ENOBJECT 69
ENOPENVPN 70
ENPATTERN 70
ENPROXY 71
ENREFRESH 71
ENREPORT 71
ENROLL 72
ENSERVICE 72
ENSL 73
ENSNMP 73
ENSSO 74
ENSWITCH 74
ENTELEMETRY 74
ENTHIND 75
ENTIMEZONE 75
ENTS 76
ENURL 76
ENUSERPREFS 77
ENUSERREQD 77
ENVOUCHER 78
ENVPN 78
ENWEBSERVICES 79
ENWIFI 79
ESTENROLL 80
EVENTD 81
EXPORTCONF 81
FWINIT 82
FWPASSWD 82
FWSHUTDOWN 83
FWSOUND 83
FWUPDATE 84
GATEWAYCTL 84
GATEWAYD 85
GETALARMCONF 86
GETCONF 86
GETLICENCE 87
GETLICENCE_TOKEN 88
GETMODEL 89
GETPCI 90
GETVERSION 91
GLOBALGEN 91
HAACTIVE 92
HADIFF 92
HAINFO 93
HALT 93
HAMODE 94
HAPASSIVE 94
HARDWARECTL 94
HARDWARED 95
HARESET 95
HASCP 96
HASSH 96
HASYNC 97
HASYNCTEST 97
HOSTCHECK 97
IFINFO 98
IOCTLFW 99
KEEPALIVE 99
LAUNCHCTL 100
LAUNCHD 101
LAUNCHER_LOG 101
LDAPCHECK 102
LDAPMANAGER 102
LICENCEDIAG 103
LICENCEUPDATE 103
LOGCTL 104
LOGD 105
LOGDISK 106
MEMCHECK 106
MEMLIMIT 107
MODCHECK 107
MODEMCTL 108
MONITORCTL 108
MPD 109
NDMESG 110
NEWLDAPBASE 110
NGSTAT 111
NHUP 111
NKILL 111
NRAID 112
NRELOAD 112

NRESTART 113
NSBSDSTART 113
NSBSDSTOP 113
NSRPC 114
NSTART 115
NSTOP 115
NTPD 116
NTPQ 117
NVERBOSE 118
NVMCHECK 118
OBJECTSYNC 119
OCSPCHECK 119
OPENVPN 120
OPENVPN_AUTH 120
OPENVPN_CLEAN_USERTABLE 121
OPENVPN_CONNECT 121
OPENVPN_DISCONNECT 122
OPENVPN_PROXYCTL 122
OPENVPN_PROXYD 123
P12IMPORT 123
PAYGPREP 124
PIMCTL 124
PIMD 126
PKICTL 126
POWERSTATUS 127
PPPDOWN 127
PPPDOWN2 128
PPPUP 128
PPPUP2 129
PVMGENCONF 129
REBOOT 130
REMOTE_SHELL 130
ROUTERCTL 131
ROUTERD 132
SECADM 133
SENDALARM 133
SENDFILE 134
SERVERD 135
SERVICE_CLIENT 135
SERVICE_SERVER 136
SETBOOT 136
SETCONF 137
SETKEY 137
SETPERMISSIONS 138
SETSAREPLAYCOUNTER 138
SETURL 139
SFCTL 139
SLAPD 144
SLD 145
SLOTINFO 145
SMARTCK 146
SMARTCTL 147
SMCROUTERD 148
SNMPD 149
SSLINIT 150
SSOD 150
STATECTL 151
STATED 152
STRONGSWAN_AUTH 153
STRONGSWAN_SSO 153
SWANINFO 154
SWITCHCTL 154
SWITCHD 155
SYSDBG 155
SYSINFO 156
SYSUTIL 157
TCPICK 157
TELEMETRYD 158
TESTLDAPBASE 158
THIND 159
TOPIC_MONITOR 159
TOPIC_READER 160
TOPIC_SENDER 160
TPMCTL 161
TPROXYD 162
TSD 163
UDPSYNC 163
URLCTL 164
URLD 164
USERREQD 165
VMREPORT 165
WIZARDINIT 166

Introduction

This documents details all the Stormshield Network commands of the firewall for the release version 4.8.9.

IMPORTANT

- This command list is dedicated to the partners that have been certified by Stormshield and who realize some support to their customers.
- These commands are normally called by "high level" configuration commands to activate parts of the configuration.

No verification are made about coherency when calling directly those commands. A direct call to those commands can put the firewall in an unstable state.

CONTENTS

The command list is an alphabetical order but organized by category. The categories are:

- Hardware
- Low level configuration
- Functionalities
- High level configuration management
- Factory tools
- Daemon
- Miscellaneous

CHAPTER1: Category Description

HARDWARE

Description

This category groups all the commands used to communicate and to manage the hardware.

Index

The alphabetic list of each command of this category is the following:
hardwarectl
powerstatus

LOW LEVEL CONFIGURATION

Description

This category groups all the commands used to manage configuration at low level.

Index

The alphabetic list of each command of this category is the following:
arpreset
arpsync
arpkeepalive
builddhcpd
builddialup
builddns
buildevent
buildfilter
buildipsec
buildha
buildldapconf
buildntp
buildopenvpn
checkpim
buildsnmp
buildssh
buildwifi
ioctlfw

FUNCTIONALITIES

Description

This category groups all the commands which use functionalities of the IPS-Firewall.

Index

The alphabetic list of each command of this category is the following:
alivectl
autoupdate

checkcrl
certenrol
curltool
ddnsclient
dhclient
dhclient-script
dhlease-script
dumproot
estenroll
gatewayctl
hacheckstatus
hastart
keepalive
launchctl
ldapcheck
newldapbase
objectsync
ocspcheck
setkey
setsareplaycounter
sfctl
smartctl
statectl

HIGH LEVEL CONFIGURATION MANAGEMENT

Description

This category groups all the commands used to manage the configuration at high level.

Index

The alphabetic list of each command of this category is the following:

alivectl
avctl
backupinfo
certreqctl
date
defaultconfig
dialupstate
enalived
enantivirus
enasq
enauth
enbird
enbypass
enconsole
endhcp
endhcrelay
endialup
endns
enevent
enfilter
enha
enkeyboard

enldap
enlock
enlog
enmulticast
ennat
ennetwork
enntp
enobject
enopenvpn
enpattern
enproxy
enrouterd
enservice
ensl
ensnmp
ensso
enswitch
entelemetry
enthind
entimezone
enurl
enuserreqd
envpn
enwifi
ifinfo
launcher_log
monitorctl
pimctl
routerctl
setboot
slotinfo
urlctl

FACTORY TOOLS

Description

This category groups all the commands used by the factory. It is not recommended to launch these commands on your IPS-Firewall.

Index

The alphabetic list of each command of this category is the following:
checkintegrity
cleanfw
fwinit
kldbgload.sh
udpsync

DAEMON

Description

This category groups all the daemons of the IPS-Firewall.

Index

The alphabetic list of each command of this category is the following:

alived
asqd
avd
bird
bird4
bird6
cad
clamavd
collectord
dhclient
dhcpd
dhcrelay
dnscache
eventd
gatewayd
hardwared
launchd
logd
mpd
ntpd
openvpn
pimd
routerd
serverd
sld
smcrouterd
snmpd
stated
switchd
telemetryd
thind
tproxyd
urld
userreqd

MISCELLANEOUS

Description

This category groups all the commands that are not in a particular category.

Index

The alphabetic list of each command of this category is the following:

certinfo
checkdb
checkfs
checkintegrity
checkinternet
checkversion
chpwd

clamdefault
cleanunwantedfiles
clearlog
crlinfo
decbackup
dhcpinfo
dkill
dstat
dumpcert
encbackup
enroll
exportconf
formatdisk
fwpasswd
fwshutdown
fwsound
fwupdate
getalarmconf
getconf
getlicence_token
getlicense
getmodel
getpci
getversion
globalgen
haactive
hainfo
halt
hapassive
hareset
hasync
hostcheck
imish
licencediag
licenceupdate
licensemanager
logtools
modemctl
ndmesg
ngstat
nhup
nkill
nreload
nverbose
nrestart
nsbsdstart
nsbsdstop
nsrpc
nstart
nstop
paygprep
ntpq
pppdown
pppdown2
pppup

pppup2 check_downgrade_version
memcheck
certreqctl
certreqd
pvmdbsync
pvmgenconf
reboot
remote_shell
secadm
sendalarm
service_client
service_server
setconf
setpermissions
seturl
strongswan_sso
swaninfo
swapethernet
sysdbg
sysinfo
sysutil
tcpick
testldapbase
topic_collector
topic_emitter
topic_monitor
topic_reader
topic_sender
vmreport

CHAPTER2: Commands Description

Following is the description of every CLI/SSH command:

ALIVECTL

Description

Client application used to access to information provided by the icmp monitoring daemon (alived)

Command

```
alivectl [-h] [-B] [-d] [-o] [-v] -s <hostname> | -l | -r
<arg> | --dump-config

-h, --help : display this message
-B, --background : execute in background (will not print the
results)

-d, --debug : enable debug mode

-o, --libxo <output format> : specify the output format,
<output format> may be "text|html|xml|json[,pretty]" (default
is "text,pretty")

-m, --force-hamode <arg> : Force alived to reload its
configuration with the given ha mode <arg>. Exclusive with
other queries. Arg must be one of: "active","passive"

-l, --list <arg> : list hosts monitored by alived. <arg> can
be used to filter hosts to display

-s, --show-measure <arg> : show measurements published by
alived. <arg> can be used to filter measurements to display
with the following values :
       1. "host" : show the measurements for all links (gateway/ha
link) of the specified router object or a HA peer.
       2. "host":"link" : show measurements for the link of the
specified router object or a HA peer.
       3. :"link" : show measurements for the specified link
regardless of the router object or a HA peer.

-c, --show-measure-counter <arg=5> : Number of measurements
published by alived to show. To use in combination of -s
option. <arg> can't be equal to 0.

-t, --show-measure-timeout <arg=60> : Timeout in seconds to
show the measurements published by alived. To use in
combination of -s option. <arg> can't be equal to 0.

-r, --reload-config <arg> : make alived reload its
configuration. Exclusive with other queries. <arg> must be one
of: "verbose", "objects", "all"
--dump-config : dump alived current configuration. Exclusive
with other queries
```

Results

The statistics, the real-time measure and the list of monitored hosts.

Example

```
VMSNSX00Z0000A0>alivectl -l my_second_router --libxo
json,pretty
{
  "RequestStatus": {
    "Status": "ok"
  },
  "Host": {
    "Status": "ok",
    "Result": {
      "my_second_router": [
        {
          "name": "dmz2_other_in",
          "ipAddress": "192.168.61.102",
          "enableMonitoring": 1,
          "endpoints": [
            {
              "target": "192.168.202.254"
            }
          ]
        },
        {
          "name": "target_network2_out",
          "ipAddress": "192.168.202.254",
          "enableMonitoring": 1,
          "endpoints": [
            {
              "target": "8.8.8.8"
            }
          ]
        },
        {
          "name": "dmz2_other_out",
          "ipAddress": "192.168.202.101",
          "enableMonitoring": 1,
          "endpoints": [
            {
              "target": "204.13.248.112"
            },
            {
              "target": "216.146.43.70"
            },
            {
              "target": "8.8.8.8"
            },
            {
              "target": "8.8.4.4"
            },
            {
              "target": "192.168.202.254"
            }
          ]
```

```
            }
          ]
        }
      }
}
```

ALIVED

Description

ICMP monitoring daemon. Monitor both PBR route and HA links.

Command

```
alived [-d] [-D] [-h] [-l] [-v]
-D : will daemonize
-d : debug mode
-h : show help message
-l : print the list of hosts to be monitored then exit
-v : verbose mode
```

Results

Example

```
alived -l
host my_router:dmz1_other_in
host my_router:target_network_in
host my_second_router:dmz2_other_in
host my_second_router:target_network2_out
host my_second_router:dmz2_other_out
```

ARKEEPALIVE

Description

Run an ARP request for each entry in the ConfigFiles/arpkeepalive file.

Command

```
arpkeepalive [-v] [-h]
```

Options

-v : verbose mode -h : help

Results

Example

ARPRESET

Description

Sends ARP packets to the interfaces in order to update the ARP tables and to get the MAC adress for macless return routes.

Command

```
arpreset
```

Options

<-a|-A> | <interface> -a -A : all interfaces

Options

-r : send arp request on macless return routes -d : daemonize -c <count> : send reset count times -i <wait> : wait milliseconds between each reset

Results

Example

ARPSYNC

Description

Synchronize the local ARP table.

Command

```
arpsync -a|u|d -[4|6] [-n] [-v] [-h]
-a: setup ARP/NDP table (deprecated)
-d: cleanup ARP/NDP table (deprecated)
-u: update ARP/NDP table
-4: only setup the ARP table
-6: only setup the NDP table
-n: setup/cleanup only NAT entry
```

```
-v: verbose mode
-h: help
Remarks :
By default, both ARP and NDP (if IPv6 is enabled) tables are
setup, unless -4 or -6 option is specified.
The -a and -d option have been deprecated since the
introduction of the -u option.
```

Results

Example

ASQD

Description

Daemon of configuration and supervising ASQ.

Command

```
asqd [-r user] [-D] [-d] [-v]
-r user : Run as the specified user.
-D : Daemon.
-d : Activate debug for the current running asqd (pvm debug).
-v : Display asqd version.
```

Results

Example

ASQSTART

Description

Command

```
asqstart (no argument)
```

Results

Example

AUTHD

Description

Authentication daemon

Command

```
authd [-d] [-v] [-h]

-d, --daemonize: will daemonize

-v, --verbose  : verbose mode

-h, --help     : show this help
```

Results

Example

AUTOBACKUP.SH

Description

Automatic backup the configuration files.

Command

```
autobackup.sh [-d]
-d: debug
```

Results

Example

AUTOUPDATE

Description

Updates data for the modules listed below.

Command

```
autoupdate [-b] [-f] [-s] [-d] [-n] [-v] [-t <module>] | [-?]
-b Build data directories
-f Force a master update
-d Launch autoupdate in the background
-n Accept non-signed updates
-v Enable debug verbose to stdout
-s Show config
-t
(Antispam|URLFiltering|Patterns|CustomPatterns|AdvancedAV|Clam
av|Vaderetro|Pvm|RootCertificates|IPData) module to update
```

Results

Database of the corresponding modules has been updated.

Example

AVCTL

Description

Manages antivirus daemon.

Command

```
avctl [-v] [-o] [-q] [-B] [-r <reload flags>] [-R <reason>] [-
s <filepath>] [-b] [--sbx-profile-file <profile>] [--sbx-ctx-
file <context>] [-d] [-i] [-l]
-h [ --help ] : Display this message
-v [ --verbose ] : Enable verbosity
-q [ --quiet ] : Do not print the results to standard output
-B [ --background ] : Execute in background (will not print
the results)
-s [ --scan-file ] <file_path> : Scan the given file
-b [ --sandboxing ] : Perform a sandboxing analysis (apply
only when action is scan-file)
--sbx-ctx-file <context> : File containing the sandboxing
context parameters
--sbx-ctx-src-addr <ip> : sandboxing context source address
--sbx-ctx-src-port <port> : sandboxing context source port
--sbx-ctx-dst-addr <ip> : sandboxing context destination
address
--sbx-ctx-dst-port <port> : sandboxing context destination
port
--sbx-ctx-dst-name <dtsname> : sandboxing context destination
name ( only used in ftp and pop3 case )
--sbx-ctx-src-mac <mac> : sandboxing context source mac
address
```

```
--sbx-ctx-user <user> : sandboxing context user
--sbx-ctx-is-ssl (0|1) : sandboxing context indicates if the
connection is SSL
--sbx-ctx-blocked-by (av_filtering|sbx_filtering|antispam) :
sandboxing context "blocked by" information
--sbx-ctx-media-type (ex: application/pdf) : the media-type of
the file
--sbx-ctx-submit-file (0|1) : allow sending file to sandboxing
--sbx-ctx-proto (HTTP|FTP|SMTP|POP3) : sandboxing context
protocol
--sbx-ctx-http-method (GET|POST|...) : sandboxing context HTTP
method
--sbx-ctx-http-url-path-query <url> : sandboxing context HTTP
encoded url path and query(ex:
"/download.php%3Fparam1%3Dval1%26param2%3Dval2")
--sbx-ctx-http-dst-name <destname> : sandboxing context HTTP
destination name
--sbx-ctx-http-filename <filename> : sandboxing context HTTP
file name
--sbx-ctx-ftp-command (GET|PUT|...) : sandboxing context FTP
command
--sbx-ctx-ftp-filename <filename> : sandboxing context FTP
file name
--sbx-ctx-ftp-filepath <filepath> : sandboxing context FTP
file path
--sbx-ctx-ftp-is-download (0|1) : sandboxing context FTP
indicates if this is a download
--sbx-ctx-smtp-sender <sender> : sandboxing context SMTP
sender
--sbx-ctx-smtp-recipients <recipients> : sandboxing context
SMTP recipients list (ex: "foo@domain.com,bar@domain.org")
-r [ --reload-config ] (all|verbose|av_engine|av_settings|sbx_
settings) : Make avd reload partially or totaly its
configuration
-d [ --dump-config ] : dump avd current configuration
-i [ --dump-db-info ] : dump information about currently
loaded Database
-l [ --dump-license-info ] : dump information about currently
loaded license
-R [ --reload-reason ] <reason> : Text to explain why the
reload was made
-o [ --libxo ] (text|html|xml|json)[,pretty] : specify the
output format (default is "text,pretty")
--pause : pauses avd scans
--resume : resumes avd scans
--dump-file <filename> : Ask for a response dump, written to
the specified file in json format
--include-dir <directory> : Directory hosting flatbuffer
definitions (optional)
--schema-file <filepath> : Main flatbuffers definition file
```

Results

A command is sent to avd. Execution will hold until a response is recieved from avd, unless a background exection is asked.

Example

AVD

Description

Antivirus daemon for advanced antivirus and Sandboxing analysis.

Command

```
avd [-d] [-D] [-h]
-h Help
-d If an other process is already running, send it a signal to
switch its verbose mode, otherwise start with verbose mode
enabled.
-D Daemonize, run in background.
```

Results

Example

BACKUPINFO

Description

Display some information about the backup partition. Display an information about active partition : main or backup.

Command

```
Backupinfo [-s | -l ]
-s : Print "[BackupInfo]" to the stdout
-l : Internal option.
```

Results

Example

```
SN910A17A1711A7>backupinfo
Active=Main
BackupVersion="4.1.5"
BackupBranch=""
Date="2021-04-20 14:59:59"
Boot=Main
BootPartitionFileMissing=1
```

BACKUPRESTORE

Description

Restore backup from file passed as argument.

Command

```
backuprestore -f <file path> [-P] [-p <password>] [-u] [-v]
-v : verbose mode
-r : refresh after restore
-p : password associated with backup file
-P : prompt for a password
-f : backup file to restore
```

Results

Example

BIRD4

Description

Version 1 of Bird for IPv4, a fully functional dynamic IP routing daemon for IPv4.

Command

```
bird4 [--version] [--help] [-c <config-file>]
```

Results

Example

BIRD6

Description

Fully functional dynamic IP routing daemon for IPv6.

Command

```
bird6 [--version] [--help] [-c <config-file>]
```

Results

Example

BIRD

Description

Fully functional dynamic IP routing daemon for IPv4.

Command

```
bird [--version] [--help] [-c <config-file>]
```

Results

Example

BIRDC

Description

Bird comand-line interface client for IPv4.

Command

```
birdc [-s <control-socket>] [-v] [-r] [-l]
```

Results

Example

BIRDC4

Description

Command-line interface client for Bird version 1 for IPv4.

Command

```
birdc4 [-s <control-socket>] [-v] [-r] [-l]
```

Results

Example

BIRDC6

Description

Bird comand-line interface client for IPv6.

Command

```
birdc6 [-s <control-socket>] [-v] [-r] [-l]
```

Results

Example

BUILDDHCPD

Description

Converts the configuration files of DHCP to the config file for the daemon dhcpd. This binary is called by endhcp.

Command

```
builddhcpd [-4|-6] [-r] [-t] [-o config-file]
-4 : IPv4
-6 : IPv6
-r : Setup dhcp relay configuration and exit
-t : Make dhcpd tests after build
-o config-file : Set configuration file
```

Results

Example

BUILDDIALUP

Description

Converts the configuration files of mpd-netgraph to the config file for the daemon mpd. Dialup access (RTC, RNIS, PPPoE, PPTP). This binary is called by endialup.

Command

```
buildpdialup [-x <if> ]
-x : doesn't modify config files for the interfaces listed in
<if>
```

Results

Example

BUILDDNS

Description

Converts the configuration files of DNS to the config file used by the dnscache. This binary is called by endns.

Command

```
builddns [-c]
-c : update only clients information. This doesn't require a
daemon restart to be effective.
```

Results

Example

BUILDEVENT

Description

Converts the configuration files of the events to the config file for the daemon eventd. This binary is called by enevent.

Command

```
buildevent [-s | -c <eventfile> | -f <output file>] [-v] [-h]
[-?]

-h, -? help
-s show only the valid events but don't write them to disk
-c <event file> strict validation of the content of an event
file
-f <output file> generate the eventd configuration in a
specific file
-v display verbose on stdout
```

Results

Example

BUILDFILTER

Description

Converts the configuration files of filtering slot to the config file. This binary is called by enfilter.

Command

```
buildfilter -h -v -s | -m [-x] | [-i] [-f <Global FilterFile>
<FilterFile>] [-x] [-w] [-e] [-a <ASQ filter rules>] [-p
<Proxy filter rules>]
-f <Global Filterfile> <Local Filterfile> : input
-a <ASQ filter rules> : output
-p <Proxy filter rules> : output
Possible outputs: 'none', 'stdout', 'stderr',
<filename>Default for ASQ filter rules: 'stdout'
```

```
Default for Proxy filter rules: 'none'
-h help
-i implicit filtering rules
-m minimal filtering rules
-v verbose
-s display warning and error messages in a more easy-to-parse
manner
-x XML output
-w suppress warning messages
-e enforce rule checking policy, some warning are now
considered errors
```

Results

Example

BUILDHA

Description

Command

```
buildha:
-o : Check HA config and build Corosync config (default
action)
-b : Do actions that must be done at boot (create cluster or
join cluster)
-c <HA config file> : Create a cluster starting from the given
HA config file
-j <HA config file> : Joins an existing HA cluster
-v : verbose
```

Results

Example

BUILDIPSEC

Description

Converts the configuration files of the VPN IPSEC to the config file for the daemon Charon. This binary is called by envpn.

Command

```
buildipsec <action> --global=<file> --local=<file><action> is
one of the following:
--check : check the configuration
--dumpconf : dump the parsed configuration
--build : build configuration
```

Results

Example

BUILDLDAPCONF

Description

Converts the configuration files of the LDAP to the config file for the daemon ldapd. This binary is called by enldap.

Command

```
buildldapconf [-a][-v][-h]
-a : activate HA
-v : verbose
-h : help
```

Results

Example

BUILDNTP

Description

Converts the configuration files of NTP to the config file for the daemon ntpd. Sanity limit is set to 1 second. This binary is called by enntp.

Command

```
buildntp [-h]
```

Results

Example

BUILDOPENVPN

Description

Converts the configuration files of OpenVPN to the config file for the daemon openvpn. This binary is called by enopenvpn.

Command

```
buildopenvpn [-d <dir>][-h][-v]
-d : set directory to write the config to <dir> (default is
'/var/tmp/Openvpn/')
-v : set verbose level to debug
-h : display this help
```

Results

Example

BUILDSNMP

Description

Converts the configuration files of net-snmp to the config file for the daemon snmpd. This binary is called by ensnmp.

Command

```
Buildsnmp (no argument)
```

Results

Example

BUILDSSH

Description

Converts the configuration files of SSH to the config file for the daemon sshd. This binary is called by enservice.

Command

```
buildssh [-d] [-v]
-d : defaultconfig mode (force ssh key mode!)
-v : activate verbose
```

Results

Example

BUILDWIFI

Description

Converts the configuration files of Wifi and Network to the config file for the daemon hostapd. This binary is called by enwifi. Note: Only available on wifi models.

Command

```
buildwifi [-h] [-t]
-h : display help message
-t : will print 1 on stdout if wifi is activated, regarding
configuration and timeobject, 0 otherwise
```

Results

Example

CAD

Description

SMC agent daemon

Command

```
cad [-v] [-h]
-h : display help message
-v : display cad version
```

Results

Example

```
VMSNSX01A2083A9>cad -v
cad 4.7.0.dev
```

CERTENROL

Description

Perform the SCEP operation for certificate enrolment.

Command

```
certenrol -o
<"viewca"|"addca"|"getcert"|"checkcert"|"compca"|"cleanup"> [-
p <profile>] [-u <URL>] [-m <POST|GET>] [-t <transcation ID>]
[-r <retry_count>] [-f <CA's fingerprint>] [-s
<"none"|"ondisk"]
-o - Operation
       "viewca" view the root CA\'s fingerprint
       "addca" install the CA\'s from the SCEP server if it match
the given fingerprint
       "compca" compare the CA\'s fingerprint with the given one
       "getcert" query for a certificate [renewal]
       "checkcert" check for a previously pending certificate
request
       "cleanup" purge transaction IDs of previously
accepted/rejected requests
-p - Profile: The profile to use for this QUERY
-u - Server URL: SCEP server entry point
-m - Mode: HTTP Request mode (GET|POST)
-t - The transaction ID from a previous pending certificate
request
-r - Number of attempt(s) left for a pending query
-f - Fingerprint: The fingerprint to compare ("compca").
-s - Seal TPM: ("none"|"ondisk").
```

Results

Example

CERTINFO

Description

Display the information related to the certificate defined by the file in the argument.

Command

```
certinfo <certfile><certfile> : Certificate file located in
/usr/Firewall/System/
```

Results

This command displays the same information about the certificate as the serverd command PKI CERT SHOW

Example

```
U2504C099999999999>certinfo ConfigFiles/Certificates/C=FR\
O=Stormshield\ OU=QA\ team\ CN=OCSP\ Authority/C=FR\
O=Stormshield\ OU=QA\ team\
CN=OCSP.expired.Responder1.test.cert.pem
[Certificate]
IssuerHash="d8e46c44"
SubjectHash="04767abd"
Issuer="/C=FR/O=Stormshield/OU=QA team/CN=OCSP Authority"
Subject="/C=FR/O=Stormshield/OU=QA
team/CN=OCSP.expired.Responder1.test"
Version="3"
SerialNumber="09"
NotBefore="Nov 25 08:24:50 2010 GMT"
NotAfter="Aug 29 08:24:50 2018 GMT"
PublicKeyAlgorithm="rsaEncryption"
SignatureAlgorithm="sha256WithRSAEncryption"
[Subject]
countryName="FR"
organizationName="Stormshield"
organizationalUnitName="QA team"
commonName="OCSP.expired.Responder1.test"
[Issuer]
countryName="FR"
organizationName="Stormshield"
organizationalUnitName="QA team"
commonName="OCSP Authority"
[config]
OCSP="http://www.ocspserver.com/,http://www.ocspserver2.com/"
```

```
CRLDP="http://www.crldp.com/ca.crl,http://www.crldp2.com/ca.cr
l"
U2504C099999999999>
```

CERTREQCTL

Description

Requests a brief analysis of a certificate from certreqd.

Command

```
certreqctl [-h|--help] [-B|--background] [-v|--verbose] [-o|--
libxo <arg>] [-c|--get-certificate] [-i|--ip-address <arg>] [-
p|--port <arg>] [-s|--sni <arg>] [-a|--alpn <arg>] [-r|--
reload-config <arg>] [-D|--dump-config  [-t|--timeout <arg>]]
-h: Display this message
-h: Display this message.
-B: Execute in background (will not print the results).
-v: Enable verbose mode.
-o: Specify the output format, arg may be "text|html|xml|json
[,pretty]" (default is "text,pretty").
-c: Get the certificate.
-i: Give the IP address for get-certificate.
-p: Give the port for get-certificate.
-s: Give the sni for get-certificate.
-a: Give the ALPN for get-certificate.
-t: Give the timeout (in s) for get-certificate.

-r: Make certreqd reload its configuration. Exclusive with
other queries. Arg must be one of: "verbose", "roots", "all"
-D: Dump certreqd current configuration. Exclusive with other
queries.
```

Results

For a -c command, the SubjectName and AltNames of the certificate with a simple analysis of problems it has.

Example

```
U2504C099999999999>certreqctl -c -i 157.240.21.35 -p 443 -s
facebook.com -a h2,http/1.1,http/0.9
[RequestStatus]
Status="ok"
[Certificate]
Status="ok"
Certificate="/C=US/ST=California/L=Menlo Park/O=Facebook,
Inc./CN=*.facebook.com"
AltName="*.facebook.com"
```

```
AltName1="*.facebook.net"
AltName2="*.fbcdn.net"
AltName3="*.fbsbx.com"
AltName4="*.m.facebook.com"
AltName5="*.messenger.com"
AltName6="*.xx.fbcdn.net"
AltName7="*.xy.fbcdn.net"
AltName8="*.xz.fbcdn.net"
AltName9="facebook.com"
AltName10="messenger.com"
Diagnostic="OK"
ALPN="h2"
```

CERTREQD

Description

Certificate Request Daemon used (via userreqd) by ASQ to retrieve certificates from TLS 1.3 servers

Command

```
userreqd [-d] [-D] [-h]
-h: Display this message.<br />-D: Daemonize, run in
background.<br />-d: If an other process is already running,
send it a signal to switch its verbose mode, otherwise start
with verbose mode enabled.
```

Results

Example

CHECK_DOWNGRADE_VERSION

Description

Check if we can downgrade to the target version from the current version Print an error message with the pivot version to use if the downgrade is not possible

Command

```
check_downgrade_version <target_version>RESULT
return 0 if the downgrade can be done
return 1 and print an error message with the current pivot
version to use before doing the downgrade
```

Example

```
check_downgrade_version 3.2.0
Can't downgrade to target version, downgrade to the following
pivot version before downgrading to the desired version:
4.3.10
```

CHECKCRL

Description

Check the validity of CRL. Return minor or major alarm (via alarmd) if CRL has expired or will expire in 3 days or less

Command

```
checkcrl [-h] [-?] [-d] [-i] [-v] [-s] [-w <days>] [-t
<timeout>] [-g <authority name> -p <password>] [-b <bindaddr>]
[-f <minutes>] [-c <scope>]
-d toggle debug mode
-i show information of the currently running checkcrl
-s do not use dns name resolution
-w [1-30] number of days to warn the expiration. default : 3
-t [0-3600] second before timeout, 0 is for unlimited. default
: 300
-g <authority name> Disable check and generate the CRL for the
given authority
-p <password> Give the passphrase of the authority in CRL
generation mode
-f <minutes> number of minutes before the expiration of the
current CRL to fetch a new CRL
-c <scope> Allow to specify the scope of the CRLs we want to
check. Can be 'local' (default) or 'global'
-b <bindaddr> Set the bind address for CRL download
-h -? this help
-v version
During the run can use [CTRL]-t to show current taskset
```

Results

Example

CHECKDB

Description

Perform an integrity check on the given database.

Command

```
Usage: checkdb [-BRv] [-C] DBPATH
       checkdb [-BRv] -c   DBPATH
       checkdb [-BRv] -r   DBPATH
       checkdb -h

Actions:
  -c   Check the database integrity and update its backup if
not corrupted.
  -C   Check the database integrity, attempt to repair it if
corrupted and update its backup if not corrupted.
  -r   Restore the database from its backup. DBPATH must not
exist.

Default action is -C.

Options:
  -B : Don't create a backup of the database even if it pass
the integrity check.
  -R : Don't restart daemons once done.
  -v : Be verbose.

Exit Status:
  64 (USAGE)       Bad usage. Use -h to get some help.
  65 (DATAERR)     The database is corrupted and/or cannot be
repaired.
  69 (UNAVAILABLE) Unable to restart some daemons.
  70 (SOFTWARE)    An internal error occured.
  74 (IOERR)       Unable to empty the backup.
  75 (TEMPFAIL)    Lock prevent operating on the live
database.
  78 (CONFIG)      Missing live database file. Or unable to
create the backup directory.
```

Results

Example

```
foo>checkdb -C /var/db/reports/reports.db
Now running: 57360
reports: Checking integrity of the live database...
Executing: enlock -s reporting -c trylock -p 57360
Executing: enlock -s reporting -c unlock -p 57360
reports: Integrity check passed.
reports: Updating backup database...
Executing: mkdir -p /data/Main/Reports//var/db/reports
Executing: mv /data/Main/Reports//var/db/reports/reports.db
/data/Main/Reports//var/db/reports/reports.db.orig
Executing: enlock -s reporting -c trylock -p 57360
Executing: enlock -s reporting -c unlock -p 57360
```

```
Executing: rm
/data/Main/Reports//var/db/reports/reports.db.orig
reports: Backup database written to
/data/Main/Reports//var/db/reports/reports.db.
foo>
```

CHECKFS

Description

Checks if the file system is clean or not. Must be used ONLY on UNMOUNTED filesystems !

Command

```
checkfs [-v] [-d] -[r] [-h]|<device>-v : Verbose mode
-d : Dump mode
-r : Root check
-h : Help
```

Results

Example

CHECKFW

Description

Check firewall configuration

Command

```
checkfw [-a] [-v | --verbose] [-n | --nocolor] [-i | --
fileintegrity] [-s | --section <section_name>] [-h | --help]

-a, to all checks (all sections and integrity)

-v, --verbose

-n, --nocolor

-i, --fileintegrity

-s, --section <section_name>\t section_name: all (default),
firmware, hard, health, asq, ipsec, cert, ips, verbose, proto,
licence, proxy, filter, network, log, conf, ha, remotesrv,
nvm, cryptotest
```

```
-h, --help
```

Results

Example

CHECKINTEGRITY

Description

Check integrity of programs and files, based on MD5 file hashing

Command

```
checkintegrity :
-h : this help
-q : quiet mode
```

Results

Example

```
U250XA0A0803770>checkintegrity < toto
All checked files are correct
U250XA0A0803770>
```

CHECKINTERNET

Description

Checks if the firewall has an Internet access.

Command

```
checkinternet (no argument)
```

Results

Nothing if OK. Error message if KO.

Example

CHECKPIM

Description

Check the configuration files of PIM.

Command

```
checkpim [-h] [-v] [-i <dynamic_cfg_fn,interfaces_cfg_fn>]
-h [ --help ] : Display this message.
-v [ --verbose ] : Enable verbosity
-i [ --input-files ] [=arg
(=/usr/Firewall/ConfigFiles/Multicast/dynamic,/usr/Firewall/Co
nfigFiles/Multicast/interfaces)]
                : Set the pim SNS configuration files to read.
```

Results

Example

CHECKVERSION

Description

Checks if a new firmware version is available on the Stormshield servers. If so, an alarm is sent.

Command

```
checkversion [-c][-l][-h]
-c : launch checkversion in command mode
-l : Show LTSB versions only
-h : display this help
```

Results

Nothing.

Example

CHPWD

Description

Mount the root device in rw access (if error perform a filesystem check and try to mount it again). Run script "enkeyboard" in order to set the language. Run "fwpasswd" program which change the SRP/SSH password for admin. Then finally reboot the firewall.

Command

```
Chpwd (no argument)
```

Results

New password is set for admin. 8 characters min. The firewall will reboot after password confirmation.

Example

```
U2504C099999999999>chpwd
You are now with the keyboard langage configured on Firewall
########################################
## Change SRP/SSH password for admin ##
########################################
setting password for admin
enter password:
verify:
Modify SRP/SSH password of user 'admin' successful
Firewall Rebooting !
Shutdown NOW!
shutdown: [pid 738]
*** FINAL System shutdown message from
admin@U2504C099999999999
***
System going down IMMEDIATELY
```

CLAMAVD

Description

Daemon of the antivirus clamav.

Command

```
clamavd [-gdnvxh?]
-d : debug
-h -? : help
-n <timeout in ms> : noscan
-v : version
```

```
-g : full verbose for debug
-x : unpack cvd
```

Results

Example

CLAMDEFAULT

Description

Restore the clamav default configuration.

Command

```
clamdefault
```

Results

Example

CLASSIFYHOST

Description

Classifies an host based on his IP address.

Command

```
classifyhost [-vht] <host_address>-v : verbose mode
-h : show this help message
-t : types of information to look for (geo, iprep, hostrep or
all)
```

Results

Properties attached to this host

Example

```
Fw > classifyhost 8.8.8.8
GEOLOC: na:us
HOSTREP: 0
```

```
IPREP:
Fw > classifyhost -t geo 8.8.4.4
GEOLOC: na:us
```

CLASSIFYURL

Description

Classifies an url.

Command

```
classifyurl [-v] <URL>-v:verbose mode
```

Results

Categories where url is classified

Example

```
Fw > classifyurl www.google.fr
oemgroup=Search Engines & Portals
```

CLEANFW

Description

Clean some files in the firewall.

Command

```
cleanfw [-cls]
-c : Clean the firewall after the script fwtest :
       Kill all test processes in progress : burnP6, bonnie++,
netserver
       Restore default configuration, clear History
-l : Remove all log in /log
-s : Remove exlusives secrets of the firewall : CA, SSH keys,
SMC information, SSL keys
```

Results

If -c option is used, the firewall must be rebooted.

Example

```
U2504C099999999999>cleanfw -c
Kill all test process
Remove all log
Restore default configuration
Restoration done, reboot recommended
Clear History
U2504C099999999999>
```

CLEANPATTERN

Description

Remove obsolete files or directories related to the patterns.

Command

```
cleanpattern [-v][-h]
-v : Verbose mode
-h : Help
```

Results

Example

CLEARLOG

Description

Clear log files.

Command

```
clearlog -a|<logname> [date]
-a : clear all logs
<logname> : clear <logname> file
[date] : delete logs before this date
Date format is "YYYY-mm-dd HH:MM:SS"
```

Results

Example

CLEARUNWANTEDFILES

Description

Removes files from the Firewall, only applies to Kaspersky and Bitdefender library files for the moment. A warning is displayed if High Availablity is enabled for this Firewall.

Command

```
clearunwantedfiles:
-f: skips all usage controls of the Kaspersky or Bitdefender
libraries and forces the removal.
-h: displays a help message with examples
Kaspersky: Name for the Kaspersky files to remove.
Bitdefender: Name for the Bitdefender files to remove.
```

Results

Kaspersky or Bitdefender library files are removed from the Firewall and a flag is set in the configuration files to prevent any reccurence (e.g. after an update).

Example

```
U2504C099999999999>clearunwantedfiles -f Kaspersky
Warning: HA is enabled, this action should be done on the
passive UTM too.
```

COLLECTORCTL

Description

collectorctl can communicate with collectord to change its configuration.

Command

```
collectorctl
Options:
  -h [ --help ]                Display this message.
  -B [ --background ]          Execute in background (will not
print the
                               results).
  -v [ --verbose ]             Enable client verbose mode.
  -o [ --libxo ] arg           Specify the output format, arg
may be
                               "text|html|xml|json[,pretty]"
(default is
                               "text,pretty").
  -r [ --reload-config ] arg Reload collectord configuration
and verbose
```

Results

Result of the commands.

Example

```
$> collectorctl -r all
[RequestStatus]
Status="ok"
```

COLLECTORD

Description

Collect all kind of informations on the firmware and send them to telemetryd.

Command

```
collectord [-D] [-h]
-D: will daemonize
-h: show help message
```

Results

Example

```
$> collectord -d
collectord (pid 2444) is already running
Signal SIGINFO was sent to current process
Verbose status is modified
```

CONFBYPASS

Description

Add the Bypass section in ConfigFiles/system if a bypass extension module has been detected. Add the token Bypass and set its value to 1 on the interfaces that can do Bypass on the file ConfigFiles/network.

Command

```
confbypass
```

Results

Always returns 0

Example

```
confbypass
```

CONFTUNING

Description

Configuration tuning with CSV file.

Command

```
conftuning file.csv directory_path
List of supported operations:
setconf : set new configuration value to token
delconf : remove token or section
setglobal : set new global value
createHA : create HA cluster
joinHA : join HA cluster
initTPM : initialize TPM
p12import : import PKCS#12 file
sethostname : set UTM system name and system node name
```

Results

Example

COROSYNC

Description

Corosync cluster engine.

Command

```
corosync:
-f : Start application in foreground.
-p : Do not set process priority.
-v : Display version and SVN revision of Corosync and exit.
```

Results

Example

CRLINFO

Description

Display the information related to the CRL defined by the file in the argument.

Command

```
crlinfo <crlfile>
```

Results

This command display the result of the Hash function, the CRL version, the algorithm for signature and revoked certificates. (SignatureAlgorithm, RevokedCertificates...)

Example

```
U2504C099999999999>crlinfo stormshield_network_crl.pem
[Global]
Hash=99b2031a
Version=02
Issuer="/C=FR/ST=NORD/O=Stormshield/OU=NPI/L=VDA"
LastUpdate="Feb 18 15:08:45 2004 GMT"
NextUpdate="Mar 20 15:08:45 2004 GMT"
SignatureAlgorithm=md5WithRSAEncryption
[RevokedCertificates]
U2504C099999999999>
```

CURLTOOL

Description

Simple wrapper for the libfwcurl.

Command

```
curltool: -r <GET|POST> -u <URI(http://XXXXXXX> [-a <User
Agent>] [-p <POST parameters>] [-o (output filename)] -h
-r Request : Send a GET or POST request
-u URI : Uniform Resource Identifier (protocole + server +
param)
http://www.stormshield.eu/mapage.html?param1=value1¶m2=value
2...)
-a User Agent : User Agent useed for this request. Default
agent is:<model>-<serial> : curltool (1.0)
```

```
-p The POST parameters : post_param1=post_value1&post_
param2=post_value2...
-o Output File : Path to file for storing the output (!!! file
is overwrite !!!)
-h Help : Display this help
```

Results

Example

DATE

Description

Get or set the current date and time of the Firewall. The date cannot be changed if the NTP is running.

Command

```
date [-u] | [-d] | [-e] | [-n] | [-b] "YYYY-MM-DD hh:mm:ss"
date : display system date in Stormshield format
date "YYYY-MM-DD hh:mm:ss" : set new date in Stormshield
Network format
       Remark : ntp daemon must be off
-b : (for boot) do not send signal of date change to daemons
-u : display date in UNIX format
-d : display date in Stormshield Network format without
timezone
-e : display date in seconds since Epoch
-n : display date in nanoseconds since Epoch
```

Results

Example

```
U2504C099999999999>date
"2004-01-15 15:37:29" zone=GMT tz=+0000 ntp=Off
U2504C099999999999>date -u
Thu Jan 15 15:37:32 GMT 2004
U2504C099999999999>date -d
2004-01-15 15:37:34
U2504C099999999999>date "2004-01-16"
"2004-01-16 15:37:47" zone=GMT tz=+0000 ntp=Off
U2504C099999999999>date -n && date -e && date -n
1676022185398334475
1676022185
1676022185423613131
```

DDNSCLIENT

Description

Updates the input of the dynamic DNS.

Command

```
ddnsclient: [-t -vvv] {-i <interface>|-r} -a <ipaddress>-h :
print this usage message and exits
-i : interface name to check
-o : set offline
-r : parse every configuration to do renew and retry
operations
-a : IP address
-f : run as a background daemon
-t : test mode : do not send request
-v : verbose level 1: print basic update steps
-vv : verbose level 2: more verbose, add steps and request
-vvv : verbose level 3: most verbose, add structure dump and
different codes
```

Results

Example

DECBACKUP

Description

Decypher a .na file (which is the save format of the configurations) to a .tgz file.

Command

```
decbackup -i <backup> -o <output archive> [-p <password>] [-d
]
-i <backup> : name of encrypted backup input file
-o <output archive> : name of decrypted backup output file
-p <password> : password used for backup encryption
-d : Dump backup header
```

Results

Example

DEFAULTCONFIG

Description

Reset the configuration with the default one. The current configuration is saved in the file "ConfigFiles.old"

Command

```
defaultconfig [options]
-f: Force
-r: Reboot after defaultconfig
-D: Only Restore the data partition
-u: Check usb token boot restoration
-d: Dump root partition after defaultconfig
-k: Keep autoupdate data (Pattern, Pvm, Clamav, AdvancedAV,
URLFiltering), default SSL proxy authority, default sslvpn
full authority and ssh host keys
-l: Keep network configuration file
-c: No backup files (.old)
-L: Remove logs
-t: Reset TPM (TPM password is required)
```

Results

"Replacing current configuration with the default configuration": The default configuration has been restored, the firewall must be rebooted to activate the modifications. The admin password is not modified. "Previous defaultconfig found... remove it manually": enter the following command :"rm -R /Firewall/ConfigFiles.old" and restart the procedure.

Example

```
VMSNSX01B2085A9>defaultconfig -f -r -p
replacing current configuration with the default
configuration...
deleting Pattern database...
deleting Custom Patterns database...
deleting IP databases...
deleting Protocols Templates...
deleting Pvm database...
deleting RootCertificates...
deleting antivirus database...
deleting URLFiltering URL group database...
cleaning /usr/Firewall/var...
deleting ssh host keys...
deleting ssh ha key...
deleting ssh authorized_keys...
Reinitializing secret file...
[2021-04-29 13:54:33] [INFO] Creating links to /data
directories
[2021-04-29 13:54:33] [INFO] Creating real directories
[2021-04-29 13:54:33] [INFO] Creating data specific
```

```
directories
[2021-04-29 13:54:33] [INFO] Changing owner of specific
directories
[2021-04-29 13:54:33] [INFO] Setting pattern version for Main
partition to 6.1.amd64
Reboot the VM to launch install wizard
restoring default password...
################################################
## Restore default SRP/SSH password for admin ##
################################################
Modify SRP/SSH password of user 'admin' successful
reset urlgroup versions...
deploy plugin.def and create default configuration for dynamic
plugins (based on plugin.def)
recompiling pattern database...

Generate the default SSL proxy authority.
| Key generation in progress. Please wait...
SSL proxy default authority done.

Generate the default sslvpn full authority.

Generate the server certificate of default sslvpn full
authority.
\ Key generation in progress. Please wait...
Generate the user certificate of default sslvpn full
authority.
/ Key generation in progress. Please wait...
sslvpn-full-default-authority done.
- Key generation in progress. Please wait...
```

DHCLIENT

Description

The client DHCP.

Command

```
dhclient [-4|-6] [-SNTPRI1dvrxi] [-nw] [-p <port>] [-D LL|LLT]
[--dad-wait-time seconds]  [-s server-addr] [-cf config-file]
[-df duid-file] [-lf lease-file] [-pf pid-file] [--no-pid] [-e
VAR=val] [-sf script-file] [interface]*
```

Results

Example

DHCLIENT-SCRIPT

Description

Called to modify the configuration DHCP client with the new IP address.

Command

```
dhclient-script (no argument)
```

Results

Example

DHCPD

Description

DHCP server.

Command

```
dhcpd [-p <UDP port#>] [-f] [-d] [-q] [-t|-T] [-4|-6] [-cf
config-file] [-lf lease-file] [-tf trace-output-file] [-play
trace-input-file] [-pf pid-file] [--no-pid] [-s server] [if0
[...ifN]]
```

Results

Example

```
IPAddress="192.168.3.101" State="free" Start="2021-04-28
10:33:37" End="2021-04-29 10:33:37"
MacAddress="00:90:f5:c0:d6:e8"
IPAddress="192.168.3.102" State="active" Start="2021-04-29
07:40:24" End="2021-04-30 07:40:24"
MacAddress="34:48:ed:34:78:e8" Hostname="mypc"
[Stat_Lease]
NBTotal=2
NBActive=1
```

DHCPINFO

Description

Dump dhcp leases and return a section list.

Command

```
dhcpinfo [-v] [-h]
-h : help
-v : verbose
```

Results

Example

```
U30SXA02L2173A7>dhcpinfo
```

DHCRELAY

Description

DHCP relay.

Command

```
dhcrelay [-4] [-d] [-q] [-a] [-D] [-A <length>] [-c <hops>] [-
p <port>] [-b <BindAddr>] [-pf <pid-file>] [--no-pid] [-m
append|replace|forward|discard] [-i interface0 [ ... -i
interfaceN] [-iu interface0 [ ... -iu interfaceN] [-id
interface0 [ ... -id interfaceN] [-U interface] server0 [ ...
serverN]
dhcrelay [-6] [-d] [-q] [-I] [-c <hops>] [-p <port>] [-pf
<pid-file>] [--no-pid] [-s <subscriber-id>] -l lower0 [... -l
lowerN] -u upper0 [... -u upperN] lower (client link):
[address%]interface[#index] upper (server link):
[address%]interface
```

Results

Example

DHLEASE-SCRIPT

Description

This script is executed in synchronous mode by DHCP server.

Command

```
dhlease-script (commit|release|expiry) <lease address>
[<ethernet address> [<client hostname option>]]
```

Results

Example

DIALUPSTATE

Description

Display current state of dialups. Short delay exists between dialup state and link effective state. Called during dialup boot and stop processes.

Command

```
dialupstate [-h]
-h : Help
```

Results

Example

DKILL

Description

Kill all daemons present in /var/supervise/ except the sshd daemon.

Command

```
dkill (no argument)
```

Results

Warning ! Calling this command will set the firewall in an unstable state because no more daemon are running. Launching this command is not recommended.

Example

```
U2504C099999999999>dkill
No matching processes were found
U2504C099999999999>
```

DMIDECODE

Description

Reports information about FW system's hardware.

Command

```
dmidecode [
```

Options

]

Options

-d, --dev-mem FILE Read memory from device FILE (default: /dev/mem)
-h, --help Display this help text and exit
-q, --quiet Less verbose output
-s, --string KEYWORD Only display the value of the given DMI string
-t, --type TYPE Only display the entries of given type
-u, --dump Do not decode the entries --dump-bin FILE Dump the DMI data to a binary file --from-dump FILE Read the DMI data from a binary file -V, --version Display the version and exit

Results

Example

DNSCACHE

Description

Cache DNS daemon.

Command

```
dnscache (no argument)
```

Results

Example

DSTAT

Description

Display the list of each daemon, with information of state (up or down) and with time duration from last change of the state.

Command

```
dstat [up|down|<daemon>]
```

Results

"asqd" : daemon name. "/var/supervise/asqd" : path of the daemon. "up / down" : daemon state. "pid xxx" : service number affected to the daemon. "xxx seconds" : time duration since the latest change of the state.

Example

```
V50XXA3E0000000>dstat
asqd : /var/supervise/asqd: up (pid 913) 4992 seconds
bird : /var/supervise/bird: down 4993 seconds
clamavd : /var/supervise/clamavd: down 4993 seconds
corosync : /var/supervise/corosync: down 4993 seconds
dhclient : /var/supervise/dhclient: down 4993 seconds
dhcpd : /var/supervise/dhcpd: down 4993 seconds
dhcrelay : /var/supervise/dhcrelay: down 4993 seconds
dns : /var/supervise/dns: down 4993 seconds
eventd : /var/supervise/eventd: up (pid 1012) 4989 seconds
hardwared : /var/supervise/hardwared: up (pid 911) 4992
seconds
ldap : /var/supervise/ldap: down 4993 seconds
logd : /var/supervise/logd: up (pid 906) 4993 seconds
mpd : /var/supervise/mpd: down 4993 seconds
ntp : /var/supervise/ntp: down 4993 seconds
rtadvd : /var/supervise/rtadvd: down 4993 seconds
serverd : /var/supervise/serverd: up (pid 916) 4992 seconds
sld : /var/supervise/sld: up (pid 1214) 4987 seconds
snmpd : /var/supervise/snmpd: down 4993 seconds
```

```
sshd : /var/supervise/sshd: up (pid 930) 4991 seconds
stated : /var/supervise/stated: up (pid 1126) 4987 seconds
switchd : /var/supervise/switchd: down 4993 seconds
tproxyd : /var/supervise/tproxyd: down 4993 seconds
```

DUMPROOT

Description

Do a backup of the file system to the backup partition.

Command

```
dumproot [-b] [-f] [-v]
-b : Executes dumproot at the next reboot
-f : Executes dumproot regardless of an ongoing autoupdate
-v : Enables verbose
```

Results

Return nothing if OK Return error message related to the error type.

Example

ENALIVED

Description

Active/Reload the alived daemon. If HA is not active, launch alived if there is an object to monitor If HA is active, always launch alived

Command

```
enalived [-m <ha_mode>]
-m : Reloads the daemon by forcing it into the given <ha_
mode>. Accept only "active" and "passive" parameters.
```

Results

Example

ENANTIVIRUS

Description

Active the antivirus configuration.

Command

```
enantivirus [-a] [-v] [-e] [-s] [-u] [-t [clamav]
[,advancedav]] [-R reason] [-h?]
-a : Launch autoupdate if base is missing
-v : Verbose mode activated
-e : reload engine of selected antivirus
-s : reload scan settings of selected antivirus
-u : Force a complete reload of antivirus
-R : arg arg is the reason explaining why enantivirus was
executed
-t : By default all antivirus are selected
-t clamav : Select Clamav
-t advancedav : Select advancedav
-t clamav,advancedav : In order to cumulate antivirus
```

Results

Example

```
U2504C099999999999> enantivirus -d -t clamav,advancedav
enantivirus: clamav init successful
enantivirus: advancedav init successful
U2504C099999999999>
```

ENASQ

Description

Activates ASQ configuration.

Command

```
enasq [-b] [-f] [--no-pvm] [--no-icmp] [--no-userreq] [--no-
pattern] [--no-stealth]
-b : boot mode (asqd will reload object db)
-f : force asqd to reload (asqd will restart)
--no-pvm : Don't reload pvm db
--no-userreq : Don't launch/reload userreqd daemon
--no-pattern : Don't reload asq pattern
--no-icmp : Don't update net.inet.ip.redirect(6) sysctl's
flags
```

```
--no-stealth : Don't update net.inet.ip.stealth/icmpreply
sysctl's flags
```

Results

Example

ENAUTH

Description

Activates authentication daemon according to it's configuration. enauth is an alias to "ensl"

Command

```
See ensl command
```

Results

Example

ENAUTHD

Description

Activates authd daemon according to its configuration.

Command

```
enauthd [-u]
-u: reload the daemon configuration
```

Results

Example

ENBIRD

Description

Starts or stops bird according to its state.

Command

```
enbird [-f] [-v] [--no-asq]
-f: restarts BIRD instead of sending SIGHUP
-v : verbose mode
--no-asq : do not reload asqd
```

Results

Example

ENBYPASS

Description

Activates/deactivates the hardware bypass or get its configuration.

Command

```
enbypass [-r][-i][-v][-h]
-r  : rearm Run-time Bypass watchdog
-i  : return Bypass status (from Bypass hardware registers)
-v  : set verbose level to info
-h  : print this help message
without option, activate/deactivate Bypass according to
configuration file.
```

Results

Example

```
SNI40A18A1607A5>enbypass -i
FW major version: 1
FW minor version: 6
Module capability:
System-Off bypass supported
Just-On bypass supported
Run-Time bypass supported
Run-Time Watchdog1 timer supported
Run-Time watchdog1 timer capability: 1~255 seconds
```

```
System-Off Bypass setting: Enable
Just-On Bypass setting: Enable
Run-Time Bypass setting: Disable
Run-Time watchdog1 timer status: Timer Running
Run-Time watchdog1 pair setting:
bypass will Enable while timeout
Run-Time watchdog1 timer count: 60 seconds
I2C Address: 55
SNI40A18A1607A5>
```

ENCBACKUP

Description

Encrypt backup file.

Command

```
encbackup -i <archive to protect> -o <backup> -t <backup
content> [-c comment] [-p password]
-i : input file
-o : output file
-t : backup content list
-c : backup comment
-p : encryption password
```

Results

Example

```
encbackup -i backup.network.tgz -o backup.network.na -t
network
```

ENCONSOLE

Description

Activates the console configuration. Sends SIGHUP to init and reloads tty configuration.

Command

```
enconsole [ modem | nomodem ]
modem :
nomodem :
modem and nomodem parameters are set by builddialup
```

Results

Example

ENDHCP

Description

Activates DHCP daemon according to its configuration.

Command

```
endhcp [-4|-6] [-b] [--no-asq]
-4 activates dhcpd configuration for IPv4 only.
-6 activates dhcpd configuration for IPv6 only.
--no-asq: do not reload asqd
When no IP version is specified, both IPv4 and IPv6 dhcpd
configurations are activated.
-b for boot process
```

Results

Example

ENDHCRELAY

Description

Activates DHCP relay according to its configuration.

Command

```
endhcrelay [-4|-6]
-4 enable only dhcrelay on IPv4.
-6 enable only dhcrelay on IPv6.
When no IP version is specified, both IPv4 and IPv6 dhcrelays
are configured.
```

Results

Example

ENDIALUP

Description

Activates the dialups configuration.

Command

```
Endialup [-u]
-u : reload only if conf files did change
```

Results

All the dialup connections are re-negociated. Warning, the internet connection, the NAT filtering and the VPN tunnels in progress are re-initialized.

Example

ENDNS

Description

Activates DNS daemon according to its configuration. Reload NAT and Filter slot if configuration has been modified. Flush nated DNS connections if authorized clients list have changed.

Command

```
endns [-b] [-u]
-b : Boot process
-u : Update clients list. Don't restart dnscache : cache isn't
flushed.
```

Results

Example

ENEVENT

Description

Activates events daemon according to its configuration.

Command

```
enevent [-u] [-h]
-h : display help
-u : do not send a SIGHUP to eventd if the configuration did
not change
```

Results

Example

ENFILTER

Description

Activates or re-activates a filtering slot after having modified it.

Command

```
enfilter [on | off] [-b] [-f] [-s] [-w] [--no-routing] [--no-
asq] <-u | FilterSlot [-g GfilterSlot]>on : activate the last
active slot.
off : deactivate filter, pass from any to any without
modifying the active slot configuration.
-b : no filter rules at boot.
-f : force the activation of the slot.
-c : force commit of the slot even if equal to previous one.
-s : display warning and error messages in a more easy-to-
parse manner (buildfilter option)
-u : re-activate the current slot
-w : do not display warnings (buildfilter option)
--no-routing : do not update routing information
--no-asq: do not reload asqd
FilterSlot : activate the filtering slot. FilterSlot = 00 to
10
-g GfilterSlot : activate the global filtering slot.
GfilterSlot = 00 to 10
```

Results

Example

```
U2504C099999999999>enfilter 10
No QoS rules, QoS disabled
U2504C099999999999>
```

ENGATEMON

Description

Activates the configuration of the advanced routing. Removes host memory. Call enevent to build hostcheck rules. Call endialup to update dialup configuration. Call ennetwork to update routing.

Command

```
engatemon (no argument)
```

Results

Example

ENHA

Description

Rebuilds corosync. If configuration differs, stops stated then restarts corosync, then starts stated. Else simply restarts stated.

Command

```
enha [-w] [-u] [-v] [-f]
-w : don't wait for the HA cluster to be ready
-u : soft reload (won't rebuild Corosync configuration)
-v : verbose
-f : force Corosync and Gatewayd restart
```

Results

"ha is disabled!": This message indicates that the "high availability" is not available on your IPS-Firewall.

Example

ENKEYBOARD

Description

Activates the configuration parameters for the keyboard language from file /usr/Firewall/ConfigFiles/language.

Command

```
enkeyboard (no argument)
```

Results

Example

ENLDAP

Description

Activates LDAP daemon according to its configuration.

Command

```
enldap [-h] [-n] [-f][-v]
-h: prints this help and exit
-n: generates a new internal base
-f: forces refresh
-v : verbose
```

Results

Example

ENLOCK

Description

Lock or unlock a script for a duration time.

Command

```
enlock -s <scriptname> [-c (lock|unlock|trylock)] [-d
<timeout>] [-p <pid>]
-s <scriptname> : used to deduce the name of the lock
-c <action> :
       -c lock : wait for the lock to be available and take it
       -c unlock : release the lock
       -c trylock : try to take the lock, but abort immediatly if
it's held by another process
-c : Default action = lock
-d <timeout> : maximum time to wait to get the lock
```

```
-v : verbose
Only valid for '-c lock' and between 0 and 300
-1 = forever (default)
-p <caller pid> : pid written in the lock file (by default,
getppid())
```

Results

Example

ENLOG

Description

Restart logd.

Command

```
enlog (no argument)
```

Results

Example

ENMULTICAST

Description

Activates multicast daemon (static/dynamic) according to its configuration.

Command

```
enmulticast [-r|-f] [-v] [-h]"
-r : quick restart of the active multicast daemon
(static/dynamic)
-f : force restart of the active multicast daemon
(static/dynamic)
-v : verbose mode
-h : help
```

Results

Example

ENNETWORK

Description

Reload the configuration parameters from the file /usr/Firewall/ConfigFiles/network: - generate new object in case of option "-b" is not set: - synchronize tty status - update stateful structure - load ARP entries - update filter rules because dynamic rule have not been updated with the new IP address - update NAT because dynamic rule have not been updated with the new IP address - update VPN because dynamic rule have not been updated with the new IP address - update events because dynamic dns might have been changed - update authentification because interfaces might have been changed - update snmp because interfaces speed might have been changed - try to reset arp entry of hosts for Firewall IP addresses - notify switch of configuration change in case of option "-b" is set : - notify switch of configuration change

Command

```
ennetwork
[-b]
[-c <old_network_file> [<old_hacluster_file>] [<old_ha_conf_
file>]]
[-C <new_network_file> [<new_hacluster_file>] [<new_ha_conf_
file>]]
[-d] [-f] [-v [<ERROR|WARN|INFO|DEBUG>]] [-r] [-h] [-z] [-i]
[-H]
-b boot
-c <old_network_file> [<old_hacluster_file>]
[<old_ha_conf_file>] : old network configuration file Defaults
are :
    /var/tmp/network
    /var/tmp/hacluster
    /var/tmp/highavailability
-C <new_network_file> [<new_hacluster_file>]
[<new_ha_conf_file>] : new network configuration file Defaults
are :
    /usr/Firewall/ConfigFiles/network
    /usr/Firewall/ConfigFiles/HA/hacluster
    /usr/Firewall/ConfigFiles/HA/highavailability
-d dry-run mode (display the operations that would be executed
but
do not execute them, imply -v)
-f force : refresh all interfaces even if configuration has
not
changed
-H no HA
-h dhcp
-r route
-s check static routes
-v verbose
```

```
-z dad
-i only updates interfaces configuration
-w check if new network file requires a reboot, imply -d and -
v
```

Results

Example

ENNTP

Description

Activates NTP daemon according to its configuration.

Command

```
enntp [-u | off][-h]
-h : help
-u : starts ntpd
off : stops ntpd
```

Results

Example

ENOBJECT

Description

Synchronize the object base (protocols, hosts, network, services).

Command

```
enobject [-a] [-h] [--no-asq] [--no-log]
-a: Do NOT synchronize ARP table (do not call 'arpsync -a')
--no-asq: Do not reload asqd
--no-log: Do not reload logd
-h: Help
```

Results

Example

ENOPENVPN

Description

Generate OpenVPN configuration from configuration files.

Command

```
enopenvpn [-v]
-v : activate verbose
```

Results

Example

ENPATTERN

Description

Compiles the signatures files of the ASQ.

Command

```
enpattern [options]
```

Options

-h : print this help message -r : generate resource language file and ASQ template -c <ctx> : process only the specified context <ctx>-a : same as -r + compile context -p : generate dynamic plugin configuration based on plugin.def -l : list all available ASQ pattern contexts -n : display the version of the downloaded files and the version of generated .match separated by a dot (<download version>.<.match version>) -f : force mode -v : verbose mode -t <filename> : test Patterns input file, results will be produced into "/usr/Firewall/Data/CustomPatterns/Download/" directory. -z : generate an active-update archive for Custom Patterns

Results

Example

ENPROXY

Description

Activates the proxy daemon according to its configuration for HTTP, POP3, SNMP and FTP.
Warning: 'enproxy' (without -u) is obsolete, use 'enfilter -u' instead.

Command

```
enproxy [-u] [-c] | [-r]
-u refresh tproxyd
-c clear ssl fake certificates
```

Results

Example

ENREFRESH

Description

Refresh all modules.

Command

```
enrefresh
```

Results

Example

ENREPORT

Description

Reporting module management:

Command

```
Usage: enreport [-v] [-r]
       enreport [-v] -H
       enreport [-v] -m
       enreport [-v] -u
```

```
Actions:
  -H: Synchronize the reports on the HA cluster and exit.
  -m: Mount the memory disk and exit.
  -r: Reload the daemons and exit.
  -u: Umount the memory disk and exit.

Default action is -r.

Options:
  -v : Be verbose.
```

Results

Example

ENROLL

Description

PAYG virtual machine enrollment utility.

Command

```
enroll [-h] [-q] [-v] -e
enroll [-h] [-q] [-v] [-f] -r

-h, --help    : show this help

-e, --enroll  : enroll PAYG Virtual Machine on the online
service

-r, --renew   : renew the PAYG licence (if needed)

-f, --force   : force the renew

-q, --quiet   : disable output

-v, --verbose : verbose in console
```

ENSERVICE

Description

Activates serverd daemon according to its configuration.

Command

```
enservice [-h] [-b] [-s]
-h: print this help and exits
-b: don't reload filter slot
-s: secure mode
```

Results

Example

ENSL

Description

Activates sld daemon according to its configuration.

Command

```
ensl [-u] | [-b]
-u : soft update
-b : boot
```

Results

Example

ENSNMP

Description

Activates snmpd daemon according to its configuration.

Command

```
ensnmp [-u]
-u : Only send a SIGHUP to net-snmp
```

Results

Example

ENSSO

Description

Activates sso daemon according to its configuration.

Command

```
ensso [-u]
-u : soft update
```

Results

Example

ENSWITCH

Description

Reload the configuration and active the daemon which manages the ports of the switch.

Command

```
enswitch [-v]
-v : verbose
```

Results

Example

ENTELEMETRY

Description

Activates the telemetryd and collectord daemons.

Command

```
entelemetry
```

Results

Example

ENTHIND

Description

Activates the thind daemon.

Command

```
enthind
```

Results

Example

ENTIMEZONE

Description

Updates timezone information. Must be done during upgrade process with no service running. Firewall has to be rebooted after changing timezone.

Command

```
entimezone [-F] [-u] [-d] [-r <1|2>] [-f] [-l] [-b] [-s <zone_
name>]
-F : Force (used with -u and -r options to prevent mistakes)
-u : update timezone
-r <1|2> : (disabled) configuration handled by ha if -r 1
-l : list timezones
-s <zone_name> : set timezone to <zone_name> (format given by
entimezone -l)
-f : force reloading of the current timezone
-b : check/restore timezone configuration regarding
configuration flag : currentZone. (used at boot time only).
-d : update timezone configuration file to "localtime"
```

Results

Example

```
U2504C099999999999>entimezone -l
Africa/
```

```
Africa/Algiers
Africa/Luanda
Africa/Porto-Novo
Africa/Gaborone
Africa/Ouagadougou
Africa/Bujumbura
...
Pacific/Midway
Pacific/Wake
Pacific/Efate
Pacific/Wallis
Pacific/Honolulu
Pacific/Easter
Pacific/Galapagos
WET
U2504C099999999999>entimezone -s Europe/Paris
timezone change : GMT -> Europe/Paris. Needs reboot. If HA is
enabled, needs HA synchronisation
U2504C099999999999>
```

ENTS

Description

Activates tsd daemon according to its configuration.

Command

```
ents [-u]
-u : soft update
```

Results

Example

ENURL

Description

Activate specified URL filtering. Special slot 00 desactivates URL filtering configuration.

Command

```
enurl [--copyonly]
--copyonly : allow bypassing call enproxy -u
```

Results

Example

ENUSERPREFS

Description

Save and load the user preferences

Command

```
enuserprefs [-s] [-r]
-s: Save the userprefs to ConfigFiles
-l: Load the userprefs from ConfigFiles
```

Results

Example

ENUSERREQD

Description

Activates the userreqd daemon.

Command

```
enuserreqd
```

Results

Example

ENVOUCHER

Description

Activates voucher LDAP daemon according to its configuration.

Command

```
envoucher [-h] [-n] [-f]
-h: prints this help and exit
-n: generates a new internal base
-f: forces refresh
```

Results

Example

ENVPN

Description

Activate specified VPN configuration. Special slot 00 desactivates VPN configuration. Note: envpn -u without changes in slot does NOTHING.

Command

```
envpn [-u | on | off | -h | slotnumber | -g globalslotnumber]
[--dry-run]
-h : Help
-u|on : re-activate the current slot
off : deactivate the current slot
slotnumber : activate the local filtering slot
(00<=slot<=10)
-g globalslotnumber: activate the global filtering slot
(00<=slot<=10)
--dry-run: perform a trial run with no changes made (checks
are run)
```

Results

Example

```
U2504C099999999999>envpn 01
Activating new VPN tunnel...
Done.
current global slot =
```

```
current slot = IPsec 01
No QoS rules, QoS disabled
U2504C099999999999>
```

ENWEBSERVICES

Description

Generates a custom webservices base from a compressed CSV file (.gz)

Command

```
enwebservices [options]
```

Options

-c, --check : check storage space and number of lines of the input file (by default the backup.gz CSV in /data is used)
-i, --input : input compressed CSV file (by default the backup.gz CSV in /data is used)
-o, --output : output directory (by default the CustomIPData directory in /data)
-d, --delete : delete existing base in output directory if the input does not exist
-h, --help : show this help

Results

Example

ENWIFI

Description

Build and refresh configuration for wifi. Will Start or Stop hostapd if needed. Note: Only available on wifi models.

Command

```
enwifi [-h]
enwifi -s
-h : display help message
-s : turn on/off wifi, if configuration allows it. It will
rebuild hostapd config (only if hostapd is not in the state it
must be) but not eventd's one.
```

Results

Example

ESTENROLL

Description

Perform EST operations for certificate enrolment.

Command

```
estenroll --operation <cacerts|simpleenroll|simplereenroll> --
url <URL> --httpsca <caname> [--alias
<alias>] [--bindaddr <addr/host/interface>] [--bindport
<port>] [--httpslogin <login>]
[--httpspassword <password>] [--promptpassword] [--keytype
<RSA|SECP|Brainpool>] [--keysize <size>] [--reqtype
<user|server|smartcard>] [--subj <X509 name>] [--upn <upn>] [-
-altnames <altnames>] [--caname <caname>] [--name <certname>]
[--tpm <none|ondisk>]
--operation :
       cacerts Retrieve and import the EST CA
       simpleenroll Enroll a certificate
       simplereenroll Renew a certificate
--url - Server URL: EST server base URL
(https://<host>:<port>/)
--alias - EST server alias (when server provides multiple CAs)
--bindaddr - addr/host/interface to bind the connection to
--bindport - port to bind the connection to
--httpsca - TLS Server CA certificate
--httpslogin - HTTPS basic auth login
--httpspassword - HTTPS basic auth password
--promptpassword - prompt for the HTTPS password
--keytype - Requested keytype ("RSA"|"SECP"|"Brainpool")
--keysize - Requested keysize
--reqtype - CSR type ("server"|"user"|"smartcard")
--subj - Requested X509 name ("/C=value0/ST=value1/S=...")
--upn - Requested X509v3 UPN (for smartcard requests)
--altnames - Requested X509v3 altnames (semi-colon separated
IP Address/DNS list)
--caname - CA for the requested certificate (for
simpleenroll/simplereenroll)
--name - Desired import name (for simpleenroll) or certificate
to be renewed (for simplereenroll)
--tpm - TPM seal: (none|ondisk) (for simpleenroll)
--dr_force - force import in DR mode on non-compliant cert
--help - This help
```

Results

Example

EVENTD

Description

Events scheduler.

Command

```
eventd (no argument)
```

Results

Example

EXPORTCONF

Description

This program exports type of configuration to a file stored in /tmp by default.

Command

```
exportconf -t filter -s index_number -g index_number [-o
output_file_format] [-d directory_name ] [-v] [-h]
This program exports type of configuration to a file stored in
/tmp by default.
-t|--type    filter              : type of configuration to
export
-s|--slot    index_number        : export rules of the slot
index of the local policy (default is slot index equal to 0)
-g|--global  index_number        : export rules of the slot
index of the global policy (default is slot index equal to 0)
-o|--output  output_file_format  : output format of the
created file (default is : csv)
-d|--directory directory_name    : indicate a directory to
store the created file
-v|--verbose                     : enable verbose
-h|--help                        : print this help message
```

Results

Example

```
SNI40A16B0743A8>exportconf -t filter
Creating file: /tmp/SNI40A16B0743A8_policy0_filter_nat_rules_
local_2017-04-18_1200.csv
SNI40A16B0743A8>SNI40A16B0743A8>exportconf -t filter -g 10 -d
/data/tmp
Creating file: /data/tmp/SNI40A16B0743A8_policy10_filter_nat_
rules_global_2017-04-18_1100.csv
SNI40A16B0743A8>
```

FWINIT

Description

Generate firewall key

Command

```
fwinit -f file
```

Results

Example

FWPASSWD

Description

Change SRP and SSH password for admin.

Command

```
fwpasswd [-d] [-u] [-h] [-p newpassword -c currentpassword]
By default : changes only SRP/SSH password for admin
-d : Restore default SRP/SSH password for admin
-u : Change UNIX password for admin
-p newpassword : Set new password non-interactively (requires
-c option)
-c currentpassword : Set current password (has no effect
without -p option)
-h : Print help
```

Results

Example

```
U2504C099999999999>fwpasswd
#######################################
## Change SRP/SSH password for admin ##
#######################################
setting password for admin
enter password:
verify:
Modify SRP/SSH password of user 'admin' successful
U2504C099999999999>
```

FWSHUTDOWN

Description

This command does a virtual shutdown of the Firewall. The following commands are launched : enfilter 00 enservice -s

Command

```
fwshutdown (no argument)
```

Results

Example

FWSOUND

Description

Play sound on the Firewall speaker.

Command

```
fwsound [1 | 2 | 3 | 4]
1 : Start sound
2 : Stop sound
3 : Play predefined sound 1
4 : Play predefined sound 2
```

Results

Example

FWUPDATE

Description

Install or update the Firewall.

Command

```
fwupdate [-r] [-F] (-f <file path> | -s)
-r : reboot at the end, if no error
-F : Force install (same version)
-f : install one maj given by <file path>-s : install one maj
given from stdin
```

Results

Example

GATEWAYCTL

Description

Gatewayctl can communicate with gatewayd to change its configuration.

Command

```
gatewayctl
-h [ --help ] Display this message
-v [ --verbose ] Enable verbosity
--update_peer <peer_uid>:<peer_ip>     Update a member in the
cluster with a serial number and the new --remove_peer <peer_
uid> IPv4. If it didn't exist in the cluster already, it will
be added automatically.
--remove_peer <peer_uid>       Remove a member in the cluster with a
serial number.
--refresh_peers Refreshes connections to peers in the cluster
--list_peers
       List members in the cluster.
--update_channel <channel_name>:<channel_type>:<channel_prio>
Update replication of a channel. It needs the channel name,
```

```
its type ('topic' or 'service') and a priority ('high' or
'low'). If the replication of the channel didn't exist, it
will be added.
--remove_channel <channel_name>:<channel_type> Remove a
replication of a channel. It  need the channel name, its type
('topic' or 'service')
--list_channels
       List replication of channels.
```

Results

Result of the commands.

Example

```
$> gatewayctl --list_channels
[test/topic-low_prio]
type=topic
priority=low
[test/topic-high_prio]
type=topic
priority=high

$> gatewayctl --remove_channel test/topic-high_prio:topic
[Result]
OK

$> gatewayctl --list_channels
[test/topic-low_prio]
type=topic
priority=low
```

GATEWAYD

Description

Gatewayd replicates messages from internal messaging to members of an HA cluster.

Command

```
gatewayd [-h] [-D] [-d]
-h [ --help ]: Display this message.
-D [ --daemonize ]: Daemonize, run in background.
-d [ --debug ]: If another process is already running, send it
a signal to switch its verbose mode, otherwise start with
verbose mode enabled.
```

Results

Example

GETALARMCONF

Description

Display alarm configuration.

Command

```
getalarmconf -i <config_index> [-p <protocol>] [-c
"protocol|<ASQ context>"] [-a <alarm id>] [-v]
```

Results

Example

```
U250XA0A0803770>getalarmconf -i 1
protocol=dns context=protocol id=32 action=block level=major
dump=0
new=0 origin=profile_template msg="RÃ©cursion de label
DNS" modify=0 sensible=0 category=""
protocol=dns context=protocol id=38 action=block level=major
dump=0
new=0 origin=profile_template msg="DNS id spoofing" modify=0
sensible=0 category=""
U250XA0A0803770>
```

GETCONF

Description

Return the field value of the specified "file + section + item".

Command

```
getconf [-i <index>] <file> <section> [<item>] [<default>]
-i <index> :
<file>: Path+name of the configuration file
<section>: Section name inside the conf file
<item>: Item inside the section
<default>: Default value

getconf -l <section> <item> [<default>]
-l :
<section>: Section name inside the conf file
```

```
<item>: Item inside the section
<default>: Default value

getconf -d <licencedateitem><licencedateitem> : One item of
the following list :
       Update
       Pattern
       VulnBase
       URLFiltering
       URLVendor
       AntiVirus
       VirusVendor
       AntiSPAM
       SPAMVendor
       NotBefore
       NotAfter
       Warranty
       ExpressWarranty

getconf -y <section> <item> [<default>]
-y :
<section> : Section name inside the payg licence
<item> : Item inside the section
<default> : Default value

getconf -p
REMARKS
getconf -i <index> <file> <section>    returns the index-th
"token=value" or only "token" (if no value)
getconf -i <index> <file> <section> <item>     returns the index-
th value for <item>, values must be coma separated
getconf -y <section> <item> [<default>]
       returns the PAYG licence item value
getconf -p
       checks if the PAYG licence is valid
```

Results

Example

```
U2504C099999999999>getconf /usr/Firewall/ConfigFiles/network
ethernet1 address 10.X.X.X
U2504C099999999999>
```

GETLICENCE

Description

Display licence information.

Command

```
getlicence
```

Results

List of all information and dates related to the licenses.

Example

```
V50XXA3E0000000>getlicence
[Global]
Version=9
Temporary=0
Comment=
[Flags]
PKI=1
...
ExpressWarranty=2037-12-31
NotBefore=2002-05-14
NotAfter=2037-12-31
V50XXA3E0000000>
```

GETLICENCE_TOKEN

Description

Return the field value of the specified "section + item" in the licence.

Command

```
getlicence_token -l <section> <item> [<default>]
-l:
<section>: Section name inside the conf file
<item>: Item inside the section
<default>: Default value

getlicence_token -d <licencedateitem><licencedateitem>: One
item of the following list:
       Update
       Pattern
       VulnBase
       URLFiltering
       URLVendor
       AntiVirus
       VirusVendor
       AntiSPAM
       SPAMVendor
       NotBefore
       NotAfter
```

```
        Warranty
        ExpressWarranty

getlicence_token -y <section> <item> [<default>]
-y:
<section>: Section name inside the payg licence
<item>: Item inside the section
<default>: Default value

getlicence_token -p

getlicence_token -h
REMARKS
getlicence_token -y <section> <item> [<default>]
        returns the PAYG licence item value
getlicence_token -p
        checks if the PAYG licence is valid
getlicence_token -h
        print help
```

Results

Example

```
VMSNSX00Z0000A0>getlicence_token Proxy HTTPProxy
```

GETMODEL

Description

Display information about type and version number of the Firewall.

Command

```
getmodel [-a | -b | -t | -m | -p | -A | -B | -H | -S | -s | -n
| --libxo]
-a : Display all version numbers and type of the Firewall.
-b : Display Build model.
-t : Display type value.
-m : Display main model value.
-p: Display equivalent running model for VM.
-A: Display the generic model used.
-B : Display branch name.
-H : Display hardware type.
-S : Display product serial number.
-s : Display manufacturer serial.
-n : Display hardware type name.
--libxo : Pass parameters to libxo (see libxo doc)
```

Results

Example

```
SN910A17A1711A7>getmodel
SN910
```

GETPCI

Description

Display the list of PCI devices.

Command

```
getpci [-h] [-v/-e] [-c <PCI class>] [-s <PCI subclass>] [-C
<chip>] [-d]
-h: help and display PCI classes and subclasses
-v: verbose
-e: enumerate (ignore -v option)
-c: get PCI class (format: -c "a class")
-s: get PCI subclass (format: -s "a subclass")
-C: get chip (format: -C 0x1234abcd)
-d: get attached driver (format: -d "attached driver")
```

Results

Example

```
U2504C099999999999>getpci
hostb0@pci0:0:0: class=0x060000 card=0x00000000
chip=0x06011106 rev=0x05 hdr=0x00
pcib1@pci0:1:0: class=0x060400 card=0x00000000 chip=0x86011106
rev=0x00 hdr=0x01
isab0@pci0:7:0: class=0x060100 card=0x00000000 chip=0x06861106
rev=0x40 hdr=0x00
atapci0@pci0:7:1: class=0x01018a card=0x00000000
chip=0x05711106 rev=0x06 hdr=0x00
uhci0@pci0:7:2: class=0x0c0300 card=0x12340925 chip=0x30381106
rev=0x1a hdr=0x00
uhci1@pci0:7:3: class=0x0c0300 card=0x12340925 chip=0x30381106
rev=0x1a hdr=0x00
none0@pci0:7:4: class=0x000000 card=0x00000000 chip=0x30571106
rev=0x40 hdr=0x00
fxp0@pci0:8:0: class=0x020000 card=0x020011d6 chip=0x12098086
rev=0x10 hdr=0x00
fxp1@pci0:9:0: class=0x020000 card=0x020011d6 chip=0x12098086
```

```
rev=0x10 hdr=0x00
fxp2@pci0:10:0: class=0x020000 card=0x020011d6 chip=0x12098086
rev=0x10 hdr=0x00
fxp3@pci0:11:0: class=0x020000 card=0x020011d6 chip=0x12098086
rev=0x10 hdr=0x00
none1@pci1:0:0: class=0x030000 card=0x85001023 chip=0x85001023
rev=0x6a hdr=0x00
U2504C099999999999>
```

GETVERSION

Description

Display Firewall software version.

Command

```
getversion [-a|-b|-v|-d|--libxo]
By default, displays Firewall software name version
-a : Display ASQ name version
-b : Display build version
-d : Display devel branch, git SHA and the timestamp of the
build
-v : Display revision number
--libxo : pass parameters to libxo (see libxo doc)
```

Results

Example

```
SN910A17A1711A7>getversion
Firewall software version 4.1.5
```

GLOBALGEN

Description

Generate mapping between real network interface name and internal name and compute model limits.

Command

```
globalgen [-m <model> -o <file>]
-m: model name
-o: output file
```

```
-m and -o options are used together to launch a globalgen dry
run.
```

Results

Example

```
VMSNSX08K0011A9>globalgen
globalgen: 4 ethernet interfaces detected
globalgen: 0 WIFI interfaces detected
VMSNSX08K0011A9>
```

HAACTIVE

Description

Force the local firewall to become the active member of the cluster, overriding any previous forced state.

Command

```
haactive
```

Results

Example

HADIFF

Description

Compare local and peer configuration files.

Command

```
hadiff <filter to diff>
```

Results

Example

HAINFO

Description

Display the status of all nodes in the HA cluster.

Command

```
hainfo
```

Results

Example

HALT

Description

Stops the IPS-Firewall. Warning ! No confirmation is required. This action stops the HA monitoring.

Command

```
When HA is enabled :
Halt [-f] [-v] [-r]
-f : Force
-v : Verbose
-r : Reboot
```

Results

Example

```
1003D011690200701>halt
Shutdown NOW!
shutdown: [pid 829]
*** FINAL System shutdown message from
admin@U2504C099999999999
***
System going down IMMEDIATELY
```

HAMODE

Description

Display ha mode (active or passive fw).

Command

```
hamode
```

Results

Example

```
V50XXA3E0000000>hamode
HA Mode : Active
```

HAPASSIVE

Description

Force the local firewall to become the passive member of the cluster, overriding any previous forced state.

Command

```
hapassive
```

Results

Example

HARDWARECTL

Description

Send command to hardwared, like setting the front panel lights or setting the watchdog timer.

Command

```
hardwarectl -c <command> [-a <command_arg>]
command_arg must be an integer between 0 and 255
Commands list :
```

```
HWD_STATE_WARNING
HWD_STATE_NORMAL
HWD_STATE_READY
HWD_STATE_HA_READY
HWD_STATE_SHUTTING_DOWN
HWD_STATE_SYSTEM_OFF
HWD_STATE_AMNESIAC
HWD_CMD_STOPWATCHDOG
HWD_CMD_SETWATCHDOG (argument needed)
HWD_CMD_KEEPWATCHDOG
HWD_CMD_STOPREFRESHBYPASSHW
HWD_CMD_STARTBYPASSHW
HWD_CMD_TOGGLEREFRESHBYPASSHW
```

Results

Example

HARDWARED

Description

Single point of communication with hardware addon. Wait for button state change and react accordingly. Animate minor/major LED. Restore default configuration when button is pressed.

Command

```
hardwared [-s] [-S on|off|blink] [-o on|off|blink] [-v]
-s: print status
-S: on|off|blink: status led test mode
-o: on|off|blink: online led test mode
-v: print hardware version
```

Results

Example

```
SN910A17A1711A7>hardwared -v
hardwared 4.1.5
```

HARESET

Description

Cancel any previous forced state for all members of the cluster.

Command

```
hareset
```

Results

Example

HASCP

Description

Scp to ha peer.

Command

```
hascp
```

Results

Example

HASSH

Description

Ssh ha peer.

Command

```
hassh
```

Results

Example

HASYNC

Description

Synchronizes all configuration files between the local firewall and the HA cluster members.

Command

```
hasync
```

Results

Example

HASYNCTEST

Description

Tests rsync of hasync in dry mode.

Command

```
hasynctest
```

Results

Example

HOSTCHECK

Description

Test the availability of a specified host. This binary is deprecated.

Command

```
Hostcheck [-h|i|o] [-v] [-c <CheckHost>] [-t <Type>] <Host>
<MaxWait> <MaxTries>-h: The host address must be resolved
using hosts file
-i: The host address is an IP address
-o: The host address must be resolved using the object
database
-v: Force Verbosity to stdout
```

```
-c: Check <CheckHost> through <Host> instead of <Host>-t: set
a type of check (string used in the state file name, must not
contain '/')
-q: Do not raise a system alarm
<Host>: The host to check. Can be an IP address, a resolvable
host or an object depending on the configuration parameter
Resolve in ConfigFiles/route at section [Config]
<MaxWait>: maximum time to wait for the response to the "ping"
test before considering it a failure. Must be >=1 and <=10
(expressed in seconds)
<MaxTries>: maximum number of "ping" tries before returning
that the host is considered DOWN or inactive. Must be >=1 and
<=10
```

Results

Returns 0|1|2|3 0 : if there has been NO change in the state of the checked host 1 : if there HAS been a change in the state of the checked host and it is UP 2 : if there HAS been a change in the state of the checked host and it is DOWN 3 : for invalid argument

Example

IFINFO

Description

Gives the information of the network interfaces configurations.

Command

```
ifinfo <name> <command> [<index>]
<name> :
in
out
dialup
pptp
ethernet
vlan
ipsec
gretun
gretap
loopback
<command> :
mac_name : get the name of the network interface
mac_address : get the MAC address of the network interface
mac_throughput : get the maximum media throughput
ip_address : get the configured IP address
ip_netmask : get the network address
ip_broadcast : get the broadcast address
ip_network : get the network address
```

```
count : get the count of interface type ( <name> = dialup,
pptp, ethernet, vlan, ipsec, gretun, gretap, loopback)
ip_config : get the configured IP address/mask
bridge_name : if bridged, return bridgename
peer_address : get the peer address of P2P interface
[<index>] : optional.
```

Results

Example

```
U2504C099999999999>ifinfo
interface list:
bridge0
10.2.32.254/255.255.0.0
out (fxp1)
in (protected,fxp0)
dmz1 (protected,fxp2)
dmz2 (protected,fxp3)
ipsec (enc0)
U2504C099999999999>
```

IOCTLFW

Description

Get and set the ASQ ioctl privileges

Command

```
ioctlfw [-h] [-v] [-l] [-d]<br/>-h: display help<br/>-v:
enable debug verbose<br/>-l: load the configuration into
ASQ</br>-d: dump the current ASQ configuration
```

Results

Example

KEEPALIVE

Description

Sends IPSec keepalive packets

Command

```
Keepalive [time_value]
time_value : 30, 60, 120, 300, 600, 0
```

Results

Example

LAUNCHCTL

Description

launchd interface for daemons management.

Command

```
launchctl <subcommand>help This help output.
load Load configuration files and/or directories.
unload Unload configuration files and/or directories.
remove Remove/stop specified job.
list List jobs and information about jobs.
sig Send a signal to a specified job.
-u Start the specified job (will be restarted on exit).
-o Start the specified job (will not be restarted on exit).
-d Stop specified job.
-p Send a STOP signal to the service.
-c Send a CONT signal to the service.
-h Send a HUP signal to the service.
-a Send a ALRM signal to the service.
-i Send a INT signal to the service.
-t Send a TERM signal to the service.
-k Send a KILL signal to the service.
-1 Send a USR1 signal to the service.
-2 Send a USR2 signal to the service.
-x Prepare for launchd shutdown.
wd Svwaitdown -k.
wu Svwaitup.
```

Results

Example

LAUNCHD

Description

Daemon which manages other daemons.

Command

```
launchd [-d | -f | -h ]
-d: Daemonize.
-h: This usage statement.
-f: Force.
```

Results

Example

LAUNCHER_LOG

Description

Log in verbose file which processes have called the process with PID passed as argument.

Command

```
launcher_log [-b | --begin] [-e | --end] [-p | --pid ] [-s | -
-section] [-c | --category ] [-h | --help]

-b, --begin : log that the process with the PID passed as the
argument has begun

-e, --end : log that the process with the PID passed as
argument has terminated

-p, --pid : PID

-s, --section : section to load

-c, --category : category to use when logging

-h, --help : print this help and exit
```

Results

Example

```
VMSNSX08K0013A9>launcher_log -b -s "enlaunchers" -c
"ENANTIVIRUS" -p "1664"
VMSNSX08K0013A9>
```

LDAPCHECK

Description

Command line program to check information in a ldap

Command

```
ldapcheck --user <userid>[ --domain <domain>][ --group
<group>] --check <command>--user   : id of the user to be
checked
--domain : domain used for the check, default one if not
specified
--group  : group used for the check
--check  : the kind of check you want like 'belongs-to-group'
* 'belongs-to-domain': check if the user belongs to the domain
passed in parameters
* 'belongs-to-group': check if the user belong to the group
passed in parameters
```

Results

[ldapcheck] Result=ko|ok

Example

```
ldapcheck --user "test" --group "testgroup" --check "belongs-
to-group"
```

LDAPMANAGER

Description

Manage an internal LDAP base.

Command

```
ldapmanager
ldapmanager -m export -f <LDIF output file path>ldapmanager -m
import -f <LDIF input file path>ldapmanager -m adduser -u
<uid> -n <name> [-g <gname>]
```

```
ldapmanager -m remuser -u <uid>ldapmanager -m listuser
ldapmanager -m raz
Remark :default action is equivalent to "objecttest -d all"
ldapmanager -m export : Export the LOCAL LDAP base to LDIF
file
ldapmanager -m import : Import a LDIF file to the LOCAL LDAP
ldapmanager -m adduser : Add an user to the LOCAL LDAP
ldapmanager -m remuser : Remove an user from the LOCAL LDAP
ldapmanager -m listuser : List the user(s) in the LOCAL LDAP
ldapmanager -m raz : Remove ALL UER(S) from the LOCAL LDAP
```

Results

Example

```
ldapmanager -m export -f ~/Configfiles/data/base.ldif
ldapmanager -m import -f ~/Configfiles/data/base.ldif
ldapmanager -m adduser -u user_uid -n user_name -g user_gname
ldapmanager -m remuser -u user_uid
ldapmanager -m listuser
ldapmanager -m raz
```

LICENCEDIAG

Description

Command line program to troubleshoot licence issues

Command

```
licencediag [--help] [--file <licence>] [--cert <certificate>
--key <key>] [--text] [--force_display]
```

Results

Example

LICENCEUPDATE

Description

Command line program to download and activate the firewall license

Command

```
licenceupdate [-d|-D] [-a|-A] [-v] [-t <n>]
-d: download new licence
-D: force download new licence
-a: activate licence
-A: force activate licence
-c: check if a new licence has been downloaded
-v: activate the verbose
-t: number of retries per licence
<no arg> : use configuration file
```

Results

Example

```
VMSNSX08K0011A9>licenceupdate -D -A
VMSNSX08K0011A9>cat /log/verbose.licenceupdate
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Prepare
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Download
/usr/Firewall/Data/Licence/VMSNSX08K0011A9.licence from
licence1-sns.stormshieldcs.eu (try 1)
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Download
complete
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] No new licence
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Checking
licence /usr/Firewall/Data/Licence/VMSNSX08K0011A9.licence
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Finalize
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Activate
licence (forced)
[2020-03-26 10:19:06] [INFO ] [LICENCEUPDATE ] Activated
licence diff:
```

LOGCTL

Description

Display information logs and reports.

Command

```
logctl [-c [-ri]] [-h] [-t <log_id>] [-T <log_id>] [-p <log_
id>] [-q] [-v]
-h: this help.
-c [-ri]: print information about SHM and failure counters.
-r: reset information after printing them
-i: print information on one line
-t <log_id>: Test reports regex. Read fake log lines from
stdin
```

```
-T <log_id>: Send log lines to Logd. Read log lines from stdin
-p <log_id>: Write log disk properties on stdout
+ Valid values for log_id are:
l_alarm, l_connection, l_filter, l_web, l_smtp, l_date, l_ftp,
l_system, l_plugin, l_vpn, l_auth, l_server, l_pop3, l_xvpn,
l_monitor, l_pvm, l_count, l_filterstat, l_ssl
-o <report> <period> : Get the requested report.
Unable to load reports configuration: Nothing to do (State=0?)
+ Possible periods are:
lasthour, day-0, day-1, day-2, day-3, day-4, day-5, day-6,
day-7, last7days, last30days, all
-q: Quiet, don't insert info in log files
-v: Verbose (-vv enables debug)
```

Results

Example

LOGD

Description

Log daemon.

Command

```
logd [-t] [-d] [-D] [-h?] [-v]
-t: check if logd is ready
-d: activate verbose mode
-D: daemonize
-h -?: help
-v: version
```

Results

U2504C099999999999>logd -d LOGD starts in verbose mode. 2011-04-11 16:26:34 | logd_config_deb | LOGD verbose ON 2011-04-11 16:26:34 | logd_config_deb | Verbose=0, no verbose activated. Please put the wanted debug level into this token (between 1 and 3) 2011-04-11 16:26:34 | logd_config_deb | LOGD verbose OFF

Example

LOGDISK

Description

Manage partition logs.

Command

```
logdisk ( -s | -l | -f [<disk/partition> [-w]] | -m
[<partition>] | -u | -c | -b | -h ) [-v]
-s : Display log partition status
-l : List all available disks/partitions.
-f [<disk/partition>] : Format current/specified log
disk/partition.
For current partition, unmount, format and mount it
automatically.
-w option forces the add of a swap partition even if model
does not require it
-m [<partition>] : Mount current/specified partition. Unmount
last partition if necessary.
-u : Unmount current partition.
-c : Do sanity checks on log partition. Try to mount back
partition in case of problem.
-b : Used during boot to mount log partition if necessary.
Skip daemons interaction.
-h : Display this usage.
-v : Verbose mode
```

Results

Example

MEMCHECK

Description

Restart a daemon if it takes more than 1Gb of memory

Command

```
memcheck <daemon_name> <memory_limit> [verbose_path]
```

Example

```
memcheck serverd 1000000 /log/dbg
```

MEMLIMIT

Description

execute a process with both its memory limits set and malloc config set

Command

```
memlimit path [args...]
path: path of the process to be executed
args: args passed to the executed process
```

Results

the executed process result

Example

MODCHECK

Description

Modcheck command Update configuration related to objects given as parameter

Command

```
modcheck -t <OBJ_TYPE_NAME> -o <OBJ_NAME> [-g] [-f] [-v]
modcheck -i <OBJ_TYPE_ID> -o <OBJ_NAME> [-g] [-f] [-v]
-t : mandatory (or -i), object type name
-i : mandatory (or -t), object type id
-o : mandatory, object name
-g : object type id
-f : set OBJCHK_FIND_FLAGS to OBJCHK_FIND_FLAGS_IGNORE_
GENERATED_GROUP_MEMBERSHIP
-v : verbose
```

Results

Example

```
modcheck -t certificate -o full_renew
modcheck -t host -o hello
```

MODEMCTL

Description

Configuration helper for usb modem.

Command

```
modemctl ( devinfos [<device>] | eject <device> | reset
<device> ) [-v]
A device is referenced by its unit address with the
ugen<unit>.<addr> form (ugen4.2)
devinfos      : Display information about all plugged USB
devices.
eject         : Power off <device> to eject safely.
reset         : Restart <device>. Useful to trigger probing by
the kernel.
-v --verbose  : Verbose mode
-h --help     : This help
```

Results

Example

```
./modemctl devinfos
ugen4.2: <Mass Storage Generic> at usbus4, cfg=255 md=HOST
spd=HIGH (480Mbps) pwr=OFF (200mA)
VendorId=058f
ProductId=6387

ugen4.3: <USB Modem USB Modem> at usbus4, cfg=0 md=HOST
spd=HIGH (480Mbps) pwr=ON (500mA)
VendorId=1c9e
ProductId=9603

ugen4.4: <HUAWEIMOBILE HUAWEIMOBILE> at usbus4, cfg=0 md=HOST
spd=HIGH (480Mbps) pwr=ON (2mA)
VendorId=12d1
ProductId=15cf
./modemctl eject ugen4.4
ugen4.4 has been powered off and can be ejected safely
```

MONITORCTL

Description

Monitord client

Command

```
monitorctl [options]
Options:
-h [ --help ]           Display this message.
-o [ --libxo ] arg      Specify the output format, arg may be
                       "text|html|xml|json[,pretty]" (default is
                       "text,pretty").
-r [ --reload-config ]  request monitord to reload its
configuration
-f [ --force-refresh ]  request monitord to refresh its
monitoring information
```

Results

Example

MPD

Description

Multi network protocol daemon.

Command

```
mpd [options] [system]
Options:

-b, --background : Run as a background daemon

-d, --directory config-dir : Set config file directory

-k, --kill : Kill running mpd process before start

-f, --file config-file : Set configuration file

-o, --one-shot : Terminate daemon after last link shutdown
-p,
--pidfile filename : Set PID filename

-s, --syslog-ident ident : Identifier to use for syslog

-m, --pam-service service : PAM service name

-v, --version : Show version information

-h, --help : Show usage information
```

Results

Example

NDMESG

Description

Print the kernel ring buffer with date

Command

```
ndmesg (no argument)
```

Results

Example

NEWLDAPBASE

Description

Generate an LDAP base. Called by enldap.

Command

```
Usage: newldapbase [ -o Orgname -d DC [-p tmppass][-P][-v]
-o Orgname : organization name
-d DC : domain component
-p tmppassword : temporary password
-P : prompt for the temporary password
-v : verbose
-h : displays help
```

Results

Example

NGSTAT

Description

Gives information on the interfaces generated by mpd daemon.

Command

```
ngstat [name] [protocol]
name : netgraph interface name listed in /var/run/mpd.pid
protocol :
<PPTP | pptp><PPPOE | PPPoE | pppoe>
```

Results

Example

NHUP

Description

Sends SIGHUP signal to specified daemon (must be a daemon from /var/supervise).

Command

```
nhup <daemon name>With <daemon_name> a daemon listed by dstat
command
```

Results

Example

NKILL

Description

Kill the specified daemon (must be a daemon listed in /var/supervise).

Command

```
nkill <daemon name>With <daemon_name> a daemon listed by dstat
command
```

Results

Example

NRAID

Description

Creates and rebuilds raid.

Command

```
nraid -h | -c | -s | -z | -a | -w <disk> | -r
-h : print this help and exit
-c : create the RAID array
-s: show current disks status
-z: reset raid ata port and probe new plugged disk
-w: wipe disk info and make it blank
-r : rebuild raid if one disk has failed
-a: try to create automaticaly RAID silently
```

Results

Example

NRELOAD

Description

Reload the specified daemon into launchd, allowing to switch to another version of the startup script.

Command

```
nreload [-f ] <daemon> [<version>]
-f : force to reload the daemon, even if the version is
unchanged.
```

Results

Example

NRESTART

Description

Restart the specified daemon (must be a daemon listed in /var/supervise).

Command

```
nrestart <daemon name>With <daemon_name> a daemon listed by
dstat command
```

Results

Example

NSBSDSTART

Description

Called during boot to set up some system values.

Command

```
nsbsdstart (no argument)
```

Results

Example

NSBSDSTOP

Description

Updates /boot/loader.conf according to the configuration. Called during shutdown.

Command

```
nsbsdstop [-d]
-d : Activate debugging
```

Results

Information written in file /boot/loader.conf

Example

NSRPC

Description

This command is used to have access to the serverd commands. The -f option is used to force the "admin" connection. The -r option is used to specify the access rights of the user. The list of access rights is written as a string with each right separated by a comma. The rights that can be specified are the following : modify, base, other, log, filter, vpn, url, pki, object, user, admin. Encoding depend on the locale LC_ALL

Command

```
nsrpc [-a|-d|-f] [-C connection timeout] [-R reading timeout]
[(-4|-6)] [-c command file] [-l log file] [-r rights] user
[:password]@server[:port]
nsrpc [-d|-f] [-C connection timeout] [-R reading timeout] [(-
4|-6)] -t targets file -c command file [-l log file] [-r
rights]
-a: automatically connect with default password
-c: set file with firewall commands
-C: set connection timeout (min: 5 ; max: 600 ; default: 600)
-d: activate debug
-f: force login
-l: set file to output commands and firewall results
-r: set rights
-R: set reading timeout (min: 5 ; max: 600 ; default: 600)
-t: set file with target firewalls ("IP[;port];login;password"
on each line)
-h: this usage
-4: connect using IPv4 (default)
-6: connect using IPv6
WARNING : stormshield_network.ca file must be in the same path
as nsrpc
```

Results

Example

```
U2504C099999999999>nsrpc admin@127.0.0.1
Welcome to Cipher/SRP client
Enter password:
Connecting to 127.0.0.1...
Using SRP authentication only.
User=admin
Level="modify,mon_
write,base,other,log,filter,vpn,url,pki,object,user,admin,netw
```

```
ork,route,maintenance,asq,pvm,globalobject,globalfilter,global
other"
SessionLevel="modify,mon_
write,base,other,log,filter,vpn,url,pki,object,user,admin,netw
ork,route,maintenance,asq,pvm,globalobject,globalfilter,global
other"
Srpclient>
```

NSTART

Description

Start the specified daemon (must be a daemon listed in /var/supervise).

Command

```
nstart <daemon name>With <daemon_name> a daemon listed by
dstat command
```

Results

Example

NSTOP

Description

Stop the specified daemon (must be a daemon listed in /var/supervise).

Command

```
nstop <daemon name>With <daemon_name> a daemon listed by dstat
command
```

Results

Example

NTPD

Description

NTP daemon program.

Command

```
ntpd [ -<flag> [<val>] | --<name>[{=| }<val>] ]..[<server1>
... <serverN>] novirtualips
Do not listen to virtual interfaces
Flag Arg Name Description
-4 no ipv4 Force IPv4 DNS name resolution - prohibits the
option 'ipv6'
-6 no ipv6 Force IPv6 DNS name resolution - prohibits the
option 'ipv4'
-a no authreq Require crypto authentication - prohibits the
option 'authnoreq'
-A no authnoreq Do not require crypto authentication -
prohibits the option 'authreq'
-b no bcastsync Allow to sync to broadcast servers
-c Str configfile Configuration file name
-d no debug-level Increase output debug message level - may
appear multiple times
-D Str set-debug-level Set the output debug message level -
may appear multiple times
-f Str driftfile Frequency drift file name
-g no panicgate Allow the first adjustment to be Big - may
appear multiple times
-G no force-step-once Step any initial offset correction.
-i no jaildir Built without --enable-clockctl or --enable-
linuxcaps or --enable-solarisprivs
-I Str interface Listen to an interface name or address - may
appear multiple times
-k Str keyfile Path to symmetric keys
-l Str logfile Path to log file
-L no
-n no nofork Do not fork - prohibits the option 'wait-sync'
-N no nice Run at high priority
-p Str pidfile Path to PID file
-P Num priority priority Process priority
-q no quit Set the time and quit - prohibits these options:
saveconfigquit wait-sync
-r Str Str propagationdelay saveconfigquit
Broadcast/propagation delay Save parsed configuration and quit
- prohibits these options: quit wait-sync
-s Str statsdir Statistics file location
-t Str trustedkey Trusted key number
-u --- user built without --enable-clockctl or --enable-
linuxcaps or --enable-solarisprivs
-U Num Str Str updateinterval var dvar interval in seconds
between scans for new or dropped interfaces make ARG an ntp
```

```
variable (RW). May appear multiple times. make ARG an ntp
variable (RW|DEF). May appear multiple times.
-w Num wait-sync Seconds to wait for first clock sync -
prohibits these options: nofork quit saveconfigquit
-x no slew Slew up to 600 seconds opt version Output version
information and exit
-? no help Display extended usage information and exit
-! no more-help Extended usage information passed thru pager
Options are specified by doubled hyphens and their name or by
a single hyphen and the flag character.
The following option preset mechanisms are supported:
- examining environment variables named NTPD_*
```

Results

Example

NTPQ

Description

Standard NTP query program

Command

```
ntpq [ -<flag> [<val>] | --<name>[{=|}<val>] ]... [ host ...]
-4 no ipv4 Force IPv4 DNS name resolution - prohibits the
option 'ipv6'
-6 no ipv6 Force IPv6 DNS name resolution - prohibits the
option 'ipv4'
-c Str command run a command and exit - may appear multiple
times
-d no debug-level Increase output debug message level - may
appear multiple times
-D Str set-debug-level Set the output debug message level -
may appear multiple times
-i no interactive
-i no interactive Force ntpq to operate in interactive mode -
prohibits these options: command peers
-n no no opt numeric old-rv version numeric host addresses
Always output status line with readvar Output version
information and exit
-p no peers Print a list of the peers -prohibits the option
'interactive'
-w no opt wide version Display the full 'remote' value output
version information and exit
-? no help Display extended usage information and exit
-! no more-help Extended usage information passed thru pager
```

```
-> opt save-opts Save the option state to a config file
-< Str load-opts Load options from a config file
```

Results

Example

NVERBOSE

Description

Activate/deactivate verbose on the specified daemon. If the given daemon isn't started or doesn't support verbose, an error will be reported.

Command

```
nverbose <daemon>
```

Results

Example

NVMCHECK

Description

Check NVM version of Intel XL card

Command

```
nvmcheck [-h] [-v]
-h : display help
-i : display information about update
-f : do not check last update status
-u : exec nvmupdate if some NVM should be updated
-v : verbose mode
```

Results

0 if all cards NVM are up to date 1 if some cards NVM should be updated 2 reboot is required by nvmupdate exec

Example

OBJECTSYNC

Description

Synchronize the dynamic objects.

Command

```
objectsync [-v] [-c] [-t <host> | -4 <host> | -6 <host>]
-h: this help
-v: turn verbose on
-c: use the cached value of the dynamic object, if it doesn't
exist, then perform a DNS query
-t <host>: resolve the IPv4 and IPv6 address of host <host>-4
<host>: resolve the IPv4 address of host <host>-6 <host>:
resolve the IPv6 address of host <host>
```

Results

Example

OCSPCHECK

Description

Check OCSP server connectivity for a given certificate

Command

```
ocspcheck --name <certname> --caname <caname> [--bindaddr
<addr/host/interface>] [--bindport <port>] [--dgst
<algorithm>] [--method <GET|POST>] [--uri <URI>] [--no-verify]
[--no-conf] [--dump-files] [--no-nonce] [--verbose] [--quiet]
--name              : certificate to be checked (must be
present in local PKI dir)
--caname            : CA for the certificate to be checked
(must be present in local or global PKI dir)
--bindaddr <addr>   : address/host/interface to bind the
connection to
--bindport <port>   : port to bind the connection to
--dgst <algorithm>  : hash algorithm to be used for
certificate IDs in OCSP request (default SHA1)
--method <GET|POST> : HTTP method to use (default POST)
--uri <URI>         : URI to use (overrides URIs found in cert
```

```
and conf)
--no-verify          : bypass response verification step
--no-conf            : do not retrieve CA and ocsp signer certs
from VPN configuration
--dump-files         : dump OCSP request and response in ocsp_
req.der and ocsp_resp.der (existing files will be overwritten)
--no-nonce           : do not send a nonce in the OCSP request
--verbose            : redirect verbose to stdout
--quiet              : print result only and no extra info
--help               : this help
```

Results

Example

```
ocspcheck --caname "C=FR ST=STST O=OO OU=OUOU CN=SubCA" --name
"C=FR ST=STST O=OO OU=OUOU CN=VPNClient.com" --quiet
```

OPENVPN

Description

OpenVPN Daemon

Command

Results

Example

OPENVPN_AUTH

Description

Authenticate user and control his access.

Command

```
openvpn_auth tcp|udp
openvpn_auth tcp : Authenticate TCP user
openvpn_auth udp : Authenticate UDP user
```

Results

Example

OPENVPN_CLEAN_USERTABLE

Description

Called by launchd on OpenVPN daemon shutdown and ensures to clean ASQ users table entries flagged with OPENVPN method.

Command

```
openvpn_clean tcp|udp
openvpn_clean tcp : Clean ASQ TCP users table entries flagged
with OPENVPN method
openvpn_clean udp : Clean ASQ UDP users table entries flagged
with OPENVPN method
openvpn_clean all : Clean ASQ TCP and UDP users table entries
flagged with OPENVPN method
```

Results

Example

OPENVPN_CONNECT

Description

Register user in ASQ users table.

Command

```
openvpn_connect tcp|udp
openvpn_connect tcp : Register TCP user in ASQ users table
openvpn_connect udp : Register UDP user in ASQ users table
```

Results

Example

OPENVPN_DISCONNECT

Description

Remove user in ASQ users table.

Command

```
openvpn_disconnect tcp|udp
openvpn_disconnect tcp
openvpn_disconnect udp
```

Results

Example

OPENVPN_PROXYCTL

Description

OpenVPN proxy daemon controller

Command

```
openvpn_proxyctl [-d] [-v] [-h] command

-h, --help    : show this help

-v, --verbose: enable verbose

-d, --daemon : run in background
commands:
mgmt_raw <tcp|udp> command: send the given command to openvpn
list_users [tcp|udp|both]: list users of given openvpn server
```

Results

Example

OPENVPN_PROXYD

Description

OpenVPN proxy daemon

Command

```
openvpn_proxyd [-d] [-v] [-h]

-d, --daemonize: will daemonize

-v, --verbose  : verbose mode

-h, --help     : show this help
```

Results

Example

P12IMPORT

Description

Import PKCS#12 file.

Command

```
p12import -f <file path> [-p <password>] [-P] [-v]
-v : verbose mode
-t : if specified, TPM seal is forced to ONDISK, NONE
otherwise
-p : password associated with PKCS#12 file
-P : prompt for a password
-f : import PKCS#12 file given by <file path>
```

Results

Example

PAYGPREP

Description

PAYG template provisioning utility

Command

```
paygprep
This wizard provisions the virtual machine to a PAYG template.
```

Results

Example

PIMCTL

Description

Interrogates pimd for status information.

Command

```
Usage:
  pimctl [
```

Options

```
][
```

Command

```
]

Options:

-m, --monitor                Run '
```

Command

```
' every two seconds, like watch(1)

-p, --plain                  Use plain table headings, no ctrl
chars

-t, --no-heading             Skip table headings
```

```
-h, --help                  This help text

-x, --xml                   Set output format to XML (only for
cli commands)

Commands:
  help                        This help text
  kill                        Kill running daemon, like SIGTERM
  restart                     Restart and reload .conf file,
like SIGHUP
  graceful                    Graceful restart of all protocol
layers
  show status                 Show router status
  show igmp                   Show interfaces and group
memberships
  show interface              Show router interface table
  show mrt [detail]           Show multicast routing table
  show neighbor               Show router neighbor table
  show rp                     Show Rendezvous-Point (RP) set
  show crp                    Show candidate Rendezvous-Point
(CRP) set
  show pim [detail]           Show interfaces, neighbors and
routes (default)
  show compat [detail]        Show router status, compat mode
  cli igmp                    Execute command cli for IGMP
table
  cli interface               Execute command cli for INTERFACE
table
  cli mrt                     Execute command cli for MRT table
  cli rp                      Execute command cli for RP table
  cli bsr                     Execute command cli for BSR table
  config check                Check if the configuration files
match what is currently loaded in pimd.
  config reload               Reload configuration
```

Results

The information requested.

Example

```
VMSNSX00Z0000A0>pimctl show rp

PIM Rendez-Vous Point Set Table
Group Address     RP Address       Prio  Holdtime  Type
232/8             169.254.0.1         1   Forever  Static
239/8             10.221.39.221       1   Forever  Static
```

PIMD

Description

Daemon that manages multicast routing using the PIM (Protocol Independent Multicast) protocol.

Command

```
Usage:
  pimd [-hnrs] [-w SEC]

Options:

-n, --foreground          Run in foreground do not detach from
calling terminal
  -r                        Retry (forever) if not all
configured interfaces are
                            available when starting up, e.g.
wait for DHCP lease
      --disable-vifs        Disable all virtual interfaces
(phyint) by default

-h, --help                Show this help text

-w, --startup-delay=SEC   Initial startup delay before probing
interfaces
```

Results

Launches the daemon. Normally launched only by launchd.

Example

PKICTL

Description

Manage PKI content.

Command

```
pkictl <operation> [--help] [--verbose] [--pkidir <path>|--
global] [options...]
<operation> can be one of the following :
         create             : create a new object into PKI
         import             : import object(s) into PKI
         stage              : stage PKI objects in a restricted folder
```

```
        verify            : verify the PKI object against the
specified parent
  --help              : display help for the selected operation
  --verbose           : verbose mode
  --pkidir <path>     : PKI dir to use
  --global            : shortcut to use the global PKI dir
```

Results

Example

POWERSTATUS

Description

Display status of power slots

Command

```
powerstatus [-s <0|1>]
-s <0|1>: slot to display (if missing, display all slots)
```

Results

Example

```
SN6KXA04F0015A8>powerstatus
POWER0: OK
POWER1: OK
```

PPPDOWN

Description

Called when a PPP link is down.

Command

```
pppdown <dialup-interface>dialup-interface : interface name to
check
```

Results

Example

PPPDOWN2

Description

Called in background when a PPP link is down.

Command

```
pppdown <dialup-interface>dialup-interface : interface name to
check
```

Results

Example

PPPUP

Description

Called when a PPP link is up.

Command

```
pppup <interface> inet <local-ip> <remote-ip> <authname> [dns1
ip] [dns2 ip]
<interface> : Interface name
<local-ip> : IP address of link's local endpoint
<remote-ip> : IP address of link's remote endpoint
<authname> : authentication name
<dns1 ip> : Domain name server primary IP address
<dns2 ip> : Domain name server secondary IP address
```

Results

Example

PPPUP2

Description

Called in background when a PPP link is up.

Command

```
pppup <interface> inet <local-ip> <remote-ip> <authname> [dns1
ip] [dns2 ip]
<interface> : Interface name
<local-ip> : IP address of link's local endpoint
<remote-ip> : IP address of link's remote endpoint
<authname> : authentication name
<dns1 ip> : Domain name server primary IP address
<dns2 ip> : Domain name server secondary IP address
```

Results

Example

PVMGENCONF

Description

Used by autoupdate in order to generate the configuration files for pvm from the downloaded files.

Command

```
pvmgenconf -d <autoupdate files dir> [-c <core dir>] [-s <sodb
dir>] [-b <banner dir>] [-v <vuln rules file>] [-V <vuln descs
file>] [-p <pof rules file>] [-l <us|fr>:<language file> [-l
...]]
-d <autoupd files dir> : Autoupdate download directory
-c <core dir> : Pvm main directory
-s <sodb dir> : Service OS Database directory
-b <banner dir> : Service Banner directory
-v <vuln rules file> : Vulnerability rules file
-V <vuln descs file> : Vulnerability description file
-p <pof rules file> : OS Signature file
-l <us|fr>:<language file> [-l ...] : language file
```

Results

generates pvm conf files for ASQ <= "ASQ_VERSION"

Example

REBOOT

Description

Reboot the IPS-Firewall. Warning !! No confirmation is requested. This action stops the HA monitoring.

Command

```
Reboot (no argument)
```

Results

Example

```
U2504C099999999999>reboot
Shutdown NOW!
shutdown: [pid 712]
*** FINAL System shutdown message from
admin@U2504C099999999999
***
System going down IMMEDIATELY
U2504C099999999999>System shutdown time has arrived
```

REMOTE_SHELL

Description

Shell for remote user. Redirect to nsrpc or csh given the CLIShell token in ConfigFiles/system [SSH]

Command

```
remote_shell
```

Results

Example

ROUTERCTL

Description

Client application used to control the routing management daemon (routerd)

Command

```
routerctl [-h] [-v] [-B] [-o] [-b] [-4] [-6] [-f] [--refresh
arg] [--router arg] [--gateway arg] [--state arg] [--dhcp
dhcp-mac-ifce-name] [--dialup dialup-mac-ifce-name] [--check-
config] [--dump-config] [--reload-config arg] [--host-status]
[--get-history] [--dump-state] [--health]

Command            Args                              Description
-h [ --help ]                                        Display a
help message
-v [ --verbose ]                                     Enable
verbose mode
-B [ --background]                                   Execute in
the background (nothing is printed)
-o [ --libxo ]     (text|html|xml|json)[,pretty]     Specify the
output format (text,pretty is used by default)
-b                                                   Boot mode
(won't call external scripts in case the firmware is booting
up)
-4 [ --ipv4 ]                                        Manage IPv4
routes or objects
-6 [ --ipv6 ]                                        Manage IPv6
routes or objects
-f                                                   Update the
routes in the kernel even if their state has not changed
-n                                                   Changes are
not applied and are printed instead
--refresh                                            Refresh
routes (all routes are refreshed, for both ip versions by
default)
--refresh-pbr                                        Refresh the
router's pbr usage.
--router           router-object                     Name of the
router object to update (which ip version to operate on must
be specified with option -4 or -6)
--gateway          gateway                           Gateway of an
host to request
--state            UP|DEGRADED|DOWN                  New state of
the specified gateway (case insensitive)
--dhcp             dhcp-mac-ifce-name                Name of the
DHCP interface to update, can only be used for DHCPv4
interfaces ( ex: eth0, IPv4 option -4 must be specified)
--dialup           dialup-mac-ifce-name              Name of the
dialup interface to update ( ex: ng0, which ip version to
operate on must be specified with option -4 or -6 )
```

```
--check-config                                          Check routing
rules validity in configuration (exclusive with other queries)
--dump-config                                           Dump current
routerd configuration (exclusive with other queries)
--reload-config    all|verbose|objects                  Reload
current routerd configuration (every kind of configuration is
refreshed by default, exclusive with other queries)
--host-status      host-name                            Show the
links(gateway/ha link) status of a host (HA peer/router
object) given in parameter
--get-history      host-name                            Retrieves the
measurements history of a host (HA peer/router object) given
in parameter. Can be paired with --gateway to request a
specific gateway of this host.
--dump-state                                            Dump the
state of the differentes routes
--health                                                Get the
current global status
```

Results

Example

```
Refresh IPv4 or IPv6 static and default routes: routerctl
[-v] [-b] [-4] [-6] [-f] --refresh
Update the state of a gateway of a given router: routerctl
[-v] [-b] [-4] [-6] [-f] --router <router-object> --gateway
<gateway-host> --state <UP|DEGRADED|DOWN>
Update the state of a generated object of type Firewall_
<dhcp-ifce>_router and all router objects using this object
as a gateway: routerctl [-v] [-b] [-6] [-f] --dhcp <dhcp-
mac-ifce-name> --state <UP|DEGRADED|DOWN>
Update the state of a generated object of type Firewall_
<dialup-ifce>_peer and all router objects using this object
as a gateway: routerctl [-v] [-b] [-6] [-f] --dialup
<dialup-mac-ifce-name> --state <UP|DEGRADED|DOWN>
```

ROUTERD

Description

Routing management daemon that manage static and default routes

Command

```
routerd [-h] [-d] [-D] [-n] [-s]

Command             Args Description
-h [ --help ]            Display a help message
```

```
-D [ --daemonize ]      Will daemonize
-d [ --debug ]          Debug mode (start routerd with verbose
if not running, otherwise activate the verbose of the current
routerd process)
```

Results

Example

SECADM

Description

Used to configure Hardened BSD rules

Command

```
secadm <command> [[modifiers] args]
Command     Args                    Description
show        [-f json|ucl|xml]     show loaded ruleset
list        [-f json|ucl|xml]     alias for "show" command
load        <file>                    load ruleset
validate    <file>                    validate ruleset
version                            show version number
flush                              flush ruleset
add         pax <path> <options>   add PaX rule
del         <id>                      del rule
enable      <id>                      enable rule
disable     <id>                      disable rule
set         <options>              set various secadm options
get         <options>              get various secadm options
```

Results

Example

SENDALARM

Description

Used to send alarms from shell scripts

Command

```
sendalarm -i <id> [-b] [-m message] [-u login] [-s src_addr]
[-d -dst_addr]
-i <id>  id of the alarm message.
-b            store an early alarm that will be logged at next
start of asqd. -u, -s and -d cannot be used with this option.
-m <message>  alarm message related to the issue.
-u <login>  user login.
-s <addr>  source address.
-d <addr>  destination address.
```

Results

Example

SENDFILE

Description

Used to send file from shell scripts

Command

```
sendfile -s <server> -p <port> -f <path> -t <protocol> -m
(basic|digest|post) -d <directory> -n <name> [-c
<controlname>] [-b] [-u <username>] [-a <password>] [-x
<ca:cert>] [-r <ca:cert>] [-v]
-s server : object http server
-f path : filepath on server
-t protocol : http | https
-m mode : basic | digest | post
-d directory : file directory
-n name : filename
-c controlname : http control name
-u username : username for http authentication
-a password : password for http authentication
-x ca:cert : client certificate (default : fw certificate)
-r ca:cert : reference server certificate
-b : bypass proxy server settings
-v : verbose
```

Results

Example

SERVERD

Description

Configuration of the daemon. Configuration is set by the user with commands lines.

Command

```
usage: serverd [<-b | -B> ipaddr] [-p port] [-r user][-d]
-b ipaddr Bind to the specified ipaddr (ipv4).
-B ipdaddr Bind to the specified ipaddr (ipv6).
-p port Attach to the specified port.
-r user Run as the specified user.
-d debug Set or launch serverd in verbose mode.
```

Results

Example

SERVICE_CLIENT

Description

Test binary that use the internal messaging to communicate. It will create a client, send and receive messages from a specific service.

Command

```
service_client
-h [ --help ]: display this message
-v [ --verbose ]: Enable verbosity
-t [ --service ]: service_name Set the service name
-m [ --message ]: arg Set the message
-s [ --startup ]: arg Set the delay in seconds at startup
before the first message (default: 1 second)
-i [ --interval ]: arg Set the interval in seconds between
successive sends (default: 1 second)
-c [ --count ]: arg Set the number of times to send the
message before exiting (default: do not stop sending)
```

Results

Responses received from the service.

Example

```
$> service_client --message test_request --service test_
service --count 3
Received response: <test_response>Received response: <test_
response>Received response: <test_response>
```

SERVICE_SERVER

Description

Test binary that use the internal messaging to communicate. It will create a server, recevie and send messages to a specific service.

Command

```
service_server
-h [ --help ] Display this message
-v [ --verbose ] service_name Enable verbosity
-s [ --service ] service_name Set the service name
-m [ --message ] arg Set the message
```

Results

Requests received from the service.

Example

```
$> service_server --service test_service -m test_response
Got request: "test_request"
Got request: "test_request"
Got request: "test_request"
...
```

SETBOOT

Description

Used to select the boot partition for the next reboot. During the boot, if you select manually the partition on which you want to boot, it has the same effect that this command.

Command

```
setboot <Main|Backup>Main: set main partition for next reboot
Backup: set Backup partition for next reboot.
```

Results

Example

SETCONF

Description

Write a section value to a configuration file. This command is generally called from scripts.

Command

```
setconf <file> <section> [<token>] <value> [<comment>]
       Adds <token>=<value> to <section> in configuration file
<file>  If <token> is not set, the section is appended with
<value> <comment> is only available if <token> is set.
setconf
-n, --no-protect <file> <section> <value>       Sets <section> to
<value> in configuration file <file> without protecting with
\"\"
setconf
-d, --delete <file> <section> [<token> [<value>]]
       Removes section <section> from configuration file <file>        If
<token> is set, removes only the token from <section>   If
<value> is set, check token value before removing
```

Results

Example

```
U2504C099999999999>setconf /usr/Firewall/ConfigFiles/network
Ethernet1 Address 10.x.x.x
U2504C099999999999>
```

SETKEY

Description

PFKEYv2 userland tool used to manage kernel information related to IPSec.

Command

```
setkey [-v] file ...
setkey [-nv] -c
```

```
setkey [-nv] -f filename
setkey [-Palpv] -D
setkey [-Pv] -F
setkey [-H] -x
setkey [-V] [-h]
```

Results

Example

SETPERMISSIONS

Description

Used to check and repair permissions on specific files

Command

```
setpermissions [-h] [-r] [-t] [-p] [-o] [-g]
-h [ --help ]: Display this message
-r [ --repair ]: Repair permissions errors if it is possible
-t [ --ignore-type ]: Ignore the type of the file
-p [ --ignore-permissions ]: Ignore the permissions of the
file
-o [ --ignore-owner ]: Ignore the owner of the file
-g [ --ignore-group ]: Ignore the group of the file
-v [ --verbose ]: Enable verbose
```

Results

Example

SETSAREPLAYCOUNTER

Description

Userland troubleshooting tool used to change an IPSec SA replay counter.

Command

```
setsareplaycounter <ip src> <ip dst> <spi> <replay counter>
```

Results

Example

SETURL

Description

Set the field "URLFiltering" in the file /usr/Firewall/ConfigFiles/proxy for CLOUDURL case : Cloudurl State is set to 1 and URLFiltering State is set to 0 for STORMSHIELD NETWORK case : Cloudurl State 0 URLFiltering State is set to 1 for NONE case : both Cloudurl and URLFiltering State are set to 0

Command

```
seturl [SN|CLOUDURL|NONE]
SN: set value "SN"
CLOUDURL: set value "CLOUDURL"
NONE: set value "SN"
```

Results

Example

SFCTL

Description

Get or set ASQ module parameters. Waring This command uses some advanced functions of the firewall. Its usage must be done very carefully and with some very good knowledges. Some commands can cut current network connexions.

Command

```
sfctl
Opt  Arg            Description
-e                  set module state
                    1 = enable
                    0 = disable
-T                  top alike mode
-f                  force operation
-v                  verbose mode
-n                  disable the reverse object lookup
-O   level          optimize ruleset at level
                    0 = none
                    1 = skip rules
-F   modifier       flush one of the following
                    addrlist = flush address list
```

```
                        assoc = flush SCTP assoc information
                        filter = flush filter rules
                        state = flush state information
                        etherstate = flush all ether state
information
                        count = flush count rule
                        stat = flush statistics
                        fpstat = flush fastpath statistics
                        pof = flush os signature  list (pof)
                        qosq = flush qos queues
                        host = flush host (see -H hstate=...)
                        sipr = flush the sip requests
                        sip = flush the sip register table
                        ipstate = flush flows managed by ipstate
                        fpstate = flush fastpath state
                        hproperties = flush hostproperties
                        assoc = flush SCTP assoc informations
                        all = all the above
-b    t,o,a[,to]        manage blacklist entry
                        t = BlackList|WhiteList...
                        o = add or delete
                        a = string identifier or '*'
                        to = timeout
-C    configdir         load and activate a ASQ configuration
-R    rulefile          load a filter rule file and activate it
-c                      commit filter rules even if equal to old
ones
-P    rulefile          load finger printing rule file and
activate it
-Q                      load QoS queues config and activate it
-q                      set QoS state
                        1 = enable
                        0 = disable
-s    modifier          dump one of the following
                        addrlist = show address list
                        assoc = show SCTP association table
content
                        conn = show connection table content
                        connstat = show TCP conn stats per state
                        count = show count rule
                        etherstate = show Ethernet connection
table content
                        filter = show current filter rules
                        fpstat = show fastpath statistics
                        fpstate = show fastpath state table
                        global = show if statistics
                        ha = show ha cluster info
                        host = show host table content
                        if = show interface information
                        ioctl = show ioctl statistics
                        ipstate = show flows managed by ipstate
                        limit = show ASQ limits
                        log = show last log message
```

```
                        mem = show memory stats
                        nat = show current nat rules
                        natpool = show reserved nat ports
                        pof = show os signature list (pof)
                        protaddr = show protected address list
                        qos = show QoS rule
                        revrt = show reverse router table
                        route = show route information
                        rulestat = show rulesmatch
                        sip = show sip register table (nat)
                        sipr = show sip request table
                        stat = show statistics
                        state = show state table content
                        table = show filter tables content
                        user = show user table content
                        all = all the above
-l    modifier          write a log entry
                        count = log count rule
                        stat = log statistics
                        all = all the above
-H    type=modifier     modify output. type can be
                        host = display information for host
                        shost = display information for client
                        dhost = display information for server
                        port = display information for port
                        sport = display information for source
                        dport = display information for
                        plugin = display information associated
                        iface = display information associated
                        siface = display information associated
                        diface = display information associated
                        proto = display information associated
                        section = filter information for show
                        state = display information according
                        hstate = display information for host
                        htype = display information for host
                        sigid = display information for host
                        ctype = display connections of a given
                        qid = display connections of a given
                        rtname = display connections of a given
                        auth = display users authenticated
                        name = display user table for a given
                        conn = all to flush all connections
                        rule = filter the connections by the
                        natrule = filter the connections by the
                        macaddr = display information for mac
                        iptype = display information by IP type
                        cpu = display information by CPU
                        bytes = display connections with total
                        lastuse = display connections used within
                        bandwidth = display host with a total
                        hostrep = display host with reputation
                        maxcount = limit number of elements
```

```
returned by -s
                    geo = geo location filter
                    iprep = iprep filter
-A   <key>[=<val>]
     [,<key>[=<val>]
        [, ...]];[...] manually add/update authenticated user(s)
                    address = user address
                    name = user name
                    domain = user domain
                    group = group membership ("g_a,g_b")
                    timeout = timeout
                    multiuser = adress is multi-user (no
value)
                    authmethod = authentication method
                    admin = user is an admin (no value)
                    sslvpn = user have access to sslvpn (no
value)
                    sslrdr = user have access to sslrdr (no
value)
                    openvpn = user have access to openvpn (no
value)
                    ipsec = user have access to ipsec (no
value)
                    sponsoring = user has the rights to
sponsor (no value)
                    ports_add = add ports to this user (1:4-
8:12), for terminal server users only (authmethod 17)
                    ports_del = delete ports from this user
(1:4-8:12), for terminal server users only (authmethod 17)
                    hostchecking = user host checking state
                    sslvpnclientversion = user SN SSL VPN
client version
                    ostype = user operating system


-a   <key>[=<val>]
     [,<key>[=<val>]
     [, ...]];[...] manually remove authenticated user(s)
                    name = user name
                    domain = user domain
                    address = user address
                    all = all authenticated user (no value)
-r old,new          rename a user domain
-t op,val           manually add/remove objects from filter
tables (experimental)
                    name = name of the table
                    op = add or del
                    val = addresses separated by comma
-B   op,host,conn,assoc backup operation
                    op = backup or restore
                    host = host filename
                    conn = conn filename
                    assoc = assoc filename
-h   modifier       HA ethernet mode
```

```
                        active = set as active mode
                        passive = set as passive mode
                        show = display current mode
                        swap = do a swap
                        bulk = send a bulk update to peer
                        <local IP>,<peer IP>,mtu = configure HA
sync in IPS
-o    filename          write output data to filename (work only
with -s)
-i    source            data source (work only with -s)
                        asq = use ASQ data (default)
-p    <key>[=<val>]
      [,<key>[=<val>]
      [, ...]];[...] manually add or tweak a host
                        addr = mandatory address of the host
                        if = interface name
                        state = desired state
                        mac = MAC address
                        geo = geo IP ("eu:fr")
                        iprep = IP reputation ("botnet,spam")
                        hostrep = host reputation
                        dns = DNS cache
                        nogeo = remove geo IP from host (no value)
                        noiprep = remove IP reputation from host
(no value)
                        nohostrep = remove reputation from host
(no value)
                        nodns = remove DNS cache from host (no
value)
--libxo params        Pass params to libxo, see libxo possible
parameters http://juniper.github.io/libxo/libxo-
manual.html#option-keywords.
                        color = Enable colors/effects for display
styles (TEXT, HTML)
                        colors=xxxx = Adjust color output values
                        dtrt = Enable "Do The Right Thing" mode
                        flush = Flush after every libxo function
call
                        flush-line = Flush after every line (line-
buffered)
                        html = Emit HTML output
                        indent=xx = Set the indentation level
                        info = Add info attributes (HTML)
                        json = Emit JSON output
                        keys = Emit the key attribute for keys
(XML)
                        log-gettext = Log (via stderr) each
gettext(3) string lookup
                        log-syslog = Log (via stderr) each syslog
message (via xo_syslog)
                        no-humanize = Ignore the {h:} modifier
(TEXT, HTML)
                        no-locale = Do not initialize the locale
```

```
setting
                    no-retain = Prevent retaining formatting
information
                    no-top Do = not emit a top set of braces
(JSON)
                    not-first = Pretend the 1st output item
was not 1st (JSON)
                    pretty = Emit pretty-printed output
                    retain = Force retaining formatting
information
                    text = Emit TEXT output
                    underscores = Replace XML-friendly "-"s
with JSON friendly "_"s
                    units = Add the 'units' (XML) or 'data-
units (HTML) attribute
                    warn = Emit warnings when libxo detects
bad calls
                    warn-xml = Emit warnings in XML
                    xml = Emit XML output
                    xpath = Add XPath expressions (HTML)
```

Results

Example

```
S
U2504C099999999999>sfctl -s host
Host (ASQ):
host if state packet bytes throughput
10.1.20.249 in active 0.00 p 0.00 B 1.26MB 0.00 b/s 0.00 b/s
10.1.20.10 in active 0.00 p 0.00 B 490KB 0.00 b/s 12.2Kb/s
10.1.20.103 in active 0.00 p 0.00 B 2.13KB 0.00 b/s 984 b/s
10.1.20.254 in active 5.00 p 320 B 400 B 0.00 b/s 0.00 b/s
10.1.20.251 in active 0.00 p 0.00 B 8.75KB 0.00 b/s 0.00 b/s
204.13.248.112 learning learning / / /
10.1.4.50 in active 0.00 p 0.00 B 80.4KB 0.00 b/s 0.00 b/s
10.1.204.11 in active 0.00 p 0.00 B 189KB 0.00 b/s 2.69Kb/s
10.1.20.101 in active 0.00 p 0.00 B 2.13KB 0.00 b/s 16.0 b/s
10.1.6.1 in active 51.0 p 15.7KB 6.86KB 3.38Kb/s 4.11Kb/s
10.1.20.102 in active 0.00 p 0.00 B 2.13KB 0.00 b/s 16.0 b/s
10.1.5.1 in active 0.00 p 0.00 B 328KB 0.00 b/s 7.25Kb/s
U2504C099999999999>
```

SLAPD

Description

LDAP daemon

Command

```
slapd <options>
```

Options

-4 IPv4 only -6 IPv6 only -T {acl|add|auth|cat|dn|index|passwd|test} : Run in Tool mode -c cookie : Sync cookie of consumer -d level : Debug level -f filename : Configuration file -F dir : Configuration directory -g group : Group (id or name) to run as -h URLs : List of URLs to serve -l facility : Syslog facility (default: LOCAL4) -n serverName : Service name -o <opt>[=val] : Generic means to specify options supported options: slp[={on|off|(attrs)}] enable/disable SLP using (attrs) -r directory : Sandbox directory to chroot to -s level : Syslog level -u user : User (id or name) to run as -V : Print version info (-VV exit afterwards, -VVV print info about static overlays and backends)

Results

Example

SLD

Description

Daemon sld.

Command

```
sld [-d] [-i] [-s] [-v]
-d : Toogle verbose
-i : Show information
-s : Show config
-h : Help
-v : Version
```

Results

Example

SLOTINFO

Description

Manage the different slots of configuration of the firewall (filtering, translation, VPN, ...)

Command

```
Slotinfo [-A index [-v]] [-g index] [-f] [-a] [-n] [-S] [-s
state] <slotname>-h : This help message
-A : Set Active SlotNumber / -v verify
-f : Get Current Slot Filename
-a : Get Current SlotNumber
-g : Get Slot Filename from index
-i : Get Slot index from Filename
-n : Get Current SlotName
-S : Get Sync
-s : Set Sync
The list of <slotname> =
       globalfilter
       globalvpn
       filter
       vpn
```

Results

Example

```
U2504C099999999999>slotinfo -a filter
10
U2504C099999999999>slotinfo -n filter
pass all
U2504C099999999999>slotinfo -f filter
/usr/Firewall/ConfigFiles/Filter/10
U2504C099999999999>
```

SMARTCK

Description

Check Utility for SMART Disks

Command

```
smartck -h | -H [device(s)] | -A [device(s)]
-h: print this help and exit
-H: check disk health
-A: dump information about disk state
If device is not defined, all disks are checked.
```

Results

Example

SMARTCTL

Description

Control and Monitor Utility for SMART Disks.

Command

```
Usage: smartctl [options] device
-h --help : Display this help and exit
-V --version : Print license, copyright, and version
information and exit
-i --info : Show identity information for device
--identify : Show words and bits from IDENTIFY DEVICE data
           (ATA)
-g --get : NAME Get device setting: all, aam, apm, lookahead,
security, wcache, rcache, wcreorder
-a --all : Show all SMART information for device
-x --xall : Show all information for device
--scan : Scan for devices
--scan-open : Scan for devices and try to open each device
-q --quietmode <TYPE> : Set smartctl quiet mode to one of:
errorsonly, silent, noserial
-d --device <TYPE> : Specify device type to one of: ata, scsi,
sat[,auto][,N][+TYPE], usbcypress[,X], usbjmicron[,p][,x][,N],
usbsunplus, 3ware,N, hpt,L/M/N, cciss,N, areca,N/E, atacam,
auto, test
-T --tolerance <TYPE> : Tolerance: normal, conservative,
permissive, verypermissive
-b --badsum <TYPE> : Set action on bad checksum to one of:
warn, exit, ignore
-r --report <TYPE> : Report transactions (see man page)
-n --nocheck <MODE> : No check if: never, sleep, standby, idle
(see man page)
-s --smart <VALUE> : Enable/disable SMART on device (on/off)
-o --offlineauto <VALUE> : Enable/disable automatic offline
testing on device (on/off)
-S --saveauto <VALUE> : Enable/disable Attribute autosave on
device (on/off)
-s --set <NAME[,VALUE]> : Enable/disable/change device
setting: aam,[N|off], apm,[N|off], lookahead,[on|off],
security-freeze, standby,[N|off|now], wcache,[on|off], rcache,
[on|off], wcreorder,[on|off]
-H --health : Show device SMART health status
-c --capabilities : Show device SMART capabilities
-A --attributes : Show device SMART vendor-specific Attributes
and values
-f --format <FORMAT> : Set output format for attributes: old,
```

```
brief, hex[,id|val]
-l --log <TYPE> : Show device log. TYPE: error, selftest,
selective, directory[,g|s], xerror[,N][,error], xselftest[,N]
[,selftest], background, sasphy[,reset], sataphy[,reset],
scttemp[sts,hist], scttempint,N[,p], scterc[,N,M], devstat
[,N], ssd, gplog,N[,RANGE], smartlog,N[,RANGE]
-v --vendorattribute <N,OPTION> : Set display OPTION for
vendor Attribute N (see man page)
-F --firmwarebug <TYPE> : Use firmware bug workaround: none,
nologdir, samsung, samsung2, samsung3, xerrorlba, swapid
-P --presets <TYPE> : Drive-specific presets: use, ignore,
show, showall
-B --drivedb <[+]FILE> : Read and replace [add] drive database
from FILE and then
/usr/local/share/smartmontools/drivedb.h]
-t --test <TEST> : Run test. TEST: offline, short, long,
conveyance, force, vendor,N, select,M-N, pending,N,
afterselect,[on|off]
-C --captive : Do test in captive mode (along with -t)
-X --abort : Abort any non-captive test on device
```

Results

Example

```
smartctl -a /dev/ad0
(Prints all SMART information)
smartctl --smart=on --offlineauto=on --saveauto=on
/dev/ad0
Enables SMART on first disk)
smartctl -t long /dev/ad0
(Executes extended disk self-test)
smartctl --attributes --log=selftest --quietmode=errorsonly
/dev/ad0
(Prints Self-Test & Attribute errors)
smartctl -a --device=3ware,2 /dev/twa0
smartctl -a --device=3ware,2 /dev/twe0
(Prints all SMART information for ATA disk on third port of
first 3ware RAID controller)
smartctl -a --device=cciss,0 /dev/ciss0
(Prints all SMART information for first disk on Common
Interface for SCSI-3 Support driver)
```

SMCROUTERD

Description

Daemon smcrouterd.

Command

```
smcrouterd [-v] [-i] [-f <file>]
-i: get info on the configuration and exit
-h: show this help
-f: force config file
-v: activate verbose mode
```

Results

Example

SNMPD

Description

Daemon snmp.

Command

```
snmpd [
```

Options

] [LISTENING ADDRESSES] -a : log addresses -A : append to the logfile rather than truncating it -c FILE[,...] : read FILE(s) as configuration file(s) -C : do not read the default configuration files (config search path: /usr/local/etc/snmp:/usr/local/share/snmp:/usr/local/lib/snmp:/usr/Firewall/.snmp) -d : dump sent and received SNMP packets -D[TOKEN[,...]] : turn on debugging output for the given TOKEN (s) (try ALL for extremely verbose output). Don't put space(s) between -D and TOKEN(s). -f : do not fork from the shell -g GID : change to this numeric gid after opening transport endpoints -h, --help : display this usage message -H : display configuration file directives understood -I [-]INITLIST : list of mib modules to initialize (or not) (run snmpd with -Dmib_init for a list) -L <LOGOPTS> : toggle options controlling where to log to e: log to standard error o: log to standard output n: don't log at all f file: log to the specified file s facility: log to syslog (via the specified facility) (variants) [EON] pri: log to standard error, output or /dev/null for level 'pri' and above [EON] p1-p2: log to standard error, output or /dev/null for levels 'p1' to 'p2' [FS] pri token: log to file/syslog for level 'pri' and above [FS] p1-p2 token: log to file/syslog for levels 'p1' to 'p2' -m MIBLIST : use MIBLIST instead of the default MIB list -M DIRLIST : use DIRLIST as the list of locations to look for MIBs (default no) -p FILE : store process id in FILE -q : print information in a more parsable format -r : do not exit if files only accessible to root cannot be opened -u UID : change to this uid (numeric or textual) after opening transport endpoints -v, --version : display version information -V : verbose display -x ADDRESS : use ADDRESS as AgentX address -X : run as an AgentX subagent rather than as an SNMP master agent Deprecated options: -l FILE : use -Lf <FILE> instead -P : use -p instead -s : use -Lsd instead -S d|i|0-7 : use -Ls <facility> instead

Results

Example

SSLINIT

Description

Initialize some SSL/SSH secure keys.

Command

```
sslinit [-p | -s] [-f] [-v]
No arg : configure all required  keys and Certification
Autorities
-p : only configure proxy Certification Authorities
-s : only regenerate ssh host key
-v : activate verbose mode
-f : Do not perform any check on CA generation conditions and
force ssh host key regeneration
```

Results

Example

SSOD

Description

SSO agent daemon

Command

```
ssod [-Ddh]
-D, --daemonize: run in background

-d, --dump      : dump conf

-h, --help      : show this help
```

Results

Example

STATECTL

Description

Command line utility to set state daemon parameters when firewall is in HA mode.

Command

```
statectl
All usage:
-v : verbose mode
-t <0-9999> : timeout
-s <infos> dump information
<infos> :
cluster = show HA cluster node info
sync = show HA node sync status
interfaces = show interfaces HA status
all = all the above
(default target host: all)
-c <command> send a command to the cluster.
       <command>:
       halt                                stop firewall
       reboot                              reboot firewall
       force_active                        force firewall to become
the active one
       force_passive                       force firewall to become
the passive one
       unforce                             cancel previous forcing
       relink                              reactivate faulty links
       sync[,<type>[,<source>[,nowait]]] synchronize files
              Synchronizations options (-c sync[,<type>[,<source>]]):
              type : Type of synchronization
                     everything (default)
                     config
                     ldap
                     ssh
                     cert
                     ha
                     au_Clamav
                     au_AdvancedAV
                     au_Antispam
                     au_RootCertificates
                     au_Patterns
                     au_URLFiltering
                     au_Vaderetro
                     au_Pvm
                     pvmdb
                     utm_secrets
```

```
                source : specify from which node the files must be
downloaded
                        <serial> = specific host
                        local = from local firewall
                        active = from an active firewall (default)
       dumproot                              run dumproot
       enha                                  run enha
       ennetwork                             run ennetwork
       pause_balancing[<,reason>[<,duration>]] will freeze HA
balancing
                <reason> : [enha|enfilter|ennetwork|enswitch|forced]
                <duration> : max time during which the HA will be frozen
                (target host: all)
       resume_balancing                      resume HA balancing if
frozen
       has_logdisk                           indicates if the firewall
has a log disk
-w <channel> watch HA message between cluster <channel>:
'SYNC-<serial>' or 'command', or 'all' (default target host:
all)
-S <serial> specify a target cluster member
<serial>:
specific host
local = local host
all = all cluster members
-a (re)generate Corosync authentification key file
-d display Corosync statistics and diagnostics info
-W <nb fw> wait for the HA cluster to be operationnal <nb fw>
number of firewalls to wait for
```

Results

Example

STATED

Description

State daemon. Monitors various firewall states like connected host, connections in progress, connected users, HA, network interfaces, etc... Allows HA configuration synchronization.

Command

```
stated [-d] [-t <option1>(,<option2>(,...))] [-k]
-d Activate debugging
-t <option1>(,<option2>(,...)) Testing options:
       'generate_events' : generate random events/connections
       'no_passive_eth' : never switch ethernet interfaces to
passive mode
```

```
        'no_asq_events' : do no get connections lists from the ASQ
        'no_asq_restoration' : do not restore peer connections into
the ASQ when becoming active
-k : Kill all SSH redirections
```

Results

Example

STRONGSWAN_AUTH

Description

Control user access.

Command

```
strongswan_auth [-v] <user_id>-v : verbose mode
user_id : id of the user to be checked
```

Results

Example

STRONGSWAN_SSO

Description

Insert/remove user from IPS.

Command

```
strongswan_sso --add|--delete --name <name> --address <IP> [--
domain <domain>] [--timeout <seconds>] [--group <groups>]
--add : insert user
--delete : delete user
--name <name> : name of the user
--address <IP> : IP address of the user
--domaine <domain> : domain of the user
--timeout <seconds> : timeout in seconds of the user
--group <groups> : comma separated list of groups of the user
```

Results

Example

SWANINFO

Description

Display current configuration and connection status in strongSwan

Command

```
swaninfo <element> [--noresolve] [--verbose]
<element> is one of the following:
       conn: Display configured connections
       conn-status: Display connection status
       ike-sa [--state=<value>]: Display IKE SAs and associated
CHILD SAs
       get-counters [--name=<value>]: Display counters for all of 1
(named) connection(s)
       stats: Display statistics based on IKE status and all
connections counters
       logstat: Log IKE_SA / CHILD_SA rekeyings counters and reset
them plus ESTABLISHED/CONNECTING IKE_SA
```

Results

Example

SWITCHCTL

Description

Manages switch (Only models with switch).

Command

```
switchctl [-e "cmd"] [-s] [-r]
-e "cmd" : send cmd command to switch and display result
-r : reboot the switch
-s : spy on communications with the switch. Commands can be
input from stdin (leave with ^C)
-b : prevent network traffic from going through the switch
```

Results

Example

SWITCHD

Description

Switch daemon. It is not possible to run two instances of switchd without argument. (Only models with switch)

Command

```
switchd [-i] [-c] [-f/-F file] [-d]
-i : create ethX interfaces (no daemon)
-c : write /var/switch (no daemon)
-f/-F <firmware> : reset switch and flash it (DANGEROUS)
-d : run in verbose mode (no daemon)
```

Results

Example

SYSDBG

Description

Active the debugging. Launch each line from command_list file and log it in /dbg/..

Command

```
/usr/Firewall/sbin/sysdbg [-q] [-c <commands>] [-S <hastate>]
/usr/Firewall/sbin/sysdbg -h
When run without arguments, simply create the /dbg directory
and if it already exists, compress its content.
-c <commands> : execute the commands listed in <commands>-h :
display help and exit
-q : quiet, no output
-S <hastate> : expected licence HA state.
```

Results

Example

SYSINFO

Description

Displays a detailed list of the configuration and activity of the firewall.

Command

```
sysinfo [-arp] [-ndp] [-host] [-conn] [-safety] [-proxy] [-
global] [-ipmi] [-time] [-fastpath] [-ipstate] [-sysctl] [-
vmstat] [-socket] [-wifi] | [-a]
-arp: add ARP table
-ndp: add NDP table
-host: add ASQ host table
-conn: add ASQ Connection table
-safety: add Safety mode information
-proxy: add PROXY informations
-global: add GLOBAL informations
-ipmi: add IPMI informations
-time: display time objects informations
-fastpath: add FASTPATH information
-ipstate: add IPSTATE information
-sysctl: display sysctl informations
-vmstat: display vmstat informations
-socket: add SOCKET INET informations
-wifi: display WIFI informations
-a: add all optional informations
WARNING: Dumping all informations can overload the appliance!
```

Results

There is a great amount of information returned by this command, it is then advised to output the results in a file: sysinfo > /tmp/sysinfo for example.

Example

```
U2504C099999999999>sysinfo
##############################
# Software information #
##############################
current date : "2011-04-06 18:35:44" zone=CEST tz=+0200
ntp=Off
Serial : U250XA0A0803770
Model : U250-A
Software : Stormshield Network Security Firewall software
version
trunk.dev-2011-03-29-10:56-NO_OPTIM
ASQ : Firewall ASQ version 5.0.0
Branch/Build : INTERNE / M
Partitions : Active=Main BackupVersion="8.1.2.beta-8-NO_OPTIM"
BackupBranch="INTERNE" Boot=Main
```

...

SYSUTIL

Description

Provide general information about the system.

Command

```
sysutil [ -h ] [ -p ] [ -d ] [-k]
-h --help
-p --labeltopartition
-d --labeltodisk
-k --keyconvert
```

Results

Example

```
U2504C099999999999>sysutil -p ufs/main
ad0s1a
```

TCPICK

Description

tcpick is a textmode sniffer libpcap-based that can track, reassemble and reorder tcp streams.

Command

```
tcpick [ -a ] [ -n ] [ -C ] [ -i interface ] [ -yH ] [ -yP ] [
-yR ] [ -yU ] [ -yx ] [ -yX ] [ -bH ] [ -bP ] [ -bR ] [ -bU ]
[ -bx ] [ -bX ] [ -wH ] [ -wP ] [ -wR ] [ -wU ] [ -v [
verbosity ]] [ -S ] [ -h ] [ --separator ] [ "filter" ] [ -r
file ] [ --help ] [ --version ]
```

Results

Example

```
U2504C099999999999>tcpick -i eth1 -yP -C -h "port 22"
Starting tcpick 0.2.1 at 2011-04-11 16:54 CEST
Timeout for connections is 600
```

```
tcpick: listening on eth1
ERROR: eth1: no IPv4 address assigned
setting filter: "port 22"
172.17.6.1:62278 AP > 172.17.6.254:ssh (48)
|....(..'06.c..............-..`$\.{z...-.k.x(.G.
172.17.6.254:ssh AP > 172.17.6.1:62278 (48)
.......E...ku.w.......4.....t.u.....#yj..).....
^C
2 packets captured
0 tcp sessions detected
U2504C099999999999>
```

TELEMETRYD

Description

Telemetry daemon.

Command

```
telemetryd [-D] [-d] [-h]
-D: will daemonize
-d: debug mode
-h: show help message
```

Results

Example

```
U2504C099999999999>telemetryd -d
telemetryd (pid 2444) is already running
Signal SIGINFO was sent to current process
Verbose status is modified
```

TESTLDAPBASE

Description

Check if openldap is up and accessible.

Command

```
testldapbase [-n number] [-t delay][-v]
-n: number of tests
-t: delay in milliseconds between tests
-v: verbose
```

Results

Example

```
U2504C099999999999>testldapbase
U2504C099999999999>
```

THIND

Description

Threat intelligence daemon.

Command

```
thind
```

Results

Example

TOPIC_MONITOR

Description

Binary that uses the internal messaging to communicate. It will create a subscriber and receive messages from a specific topic, and then dump them in a readable format.

Command

```
topic_monitor
-h [ --help ]: display this message
-v [ --verbose ]: enable verbosity
-t [ --topic ] topic_name: set the topic name
--dump arg: Specify the message dump format, arg may be
"asc|hex|all" (default is "asc")
--width arg: Specify the message dump width, arg is an integer
(default is 16)
```

Results

Messages from the topic.

Example

TOPIC_READER

Description

Test binary that use the internal messaging to communicate. It will create a subscriber and receive message from a specific topic.

Command

```
topic_reader
-h [ --help ]: Display this message
-v [ --verbose ]: Enable verbosity
-t [ --topic ] topic_name: Set the topic name
```

Results

Messages from the topic.

Example

```
$> topic_reader --topic test_topic
test
test
test
...
```

TOPIC_SENDER

Description

Test binary that use the internal messaging to communicate. It will create a publisher and send messages to a specific topic.

Command

```
topic_sender
-h [ --help ]: Display this message
-v [ --verbose ]: Enable verbosity
-t [ --topic ] topic_name: Set the topic name
-m [ --message ] arg: Set the message
-s [ --startup ] arg: Set the delay in seconds at startup
before the first message (default: 1 second)
-i [ --interval ] arg: Set the interval in seconds between
successive sends (default: 1 second)
```

```
-c [ --count ] arg: Set the number of times to send the
message before exiting (default: do not stop sending)
```

Results

Nothing without verbose.

Example

```
$> topic_sender --topic test_topic --message test --count 3
$>
```

TPMCTL

Description

Control TPM (initialization, configuration,reset).

Command

```
tpmctl [-v] [-i [-n]|-r|-a|-f|-o|-t|-s [-g]|-C|-c
<newtpmpassword> [-n]] [-p <tpmpassword>] [-w]
       -v: verbose mode
       -i: initialize TPM (tpm password is mandatory)
       -n: do not derive TPM key from password
       -r: reset TPM (tpm password is mandatory)
       -C: change TPM password (prompt for newpassword, tpmpassword
is mandatory)
       -c: change TPM password (tpmpassword is mandatory)
       -n: no decrypt (leave pkeys as is (troubleshooting only))
       -a: run TPM diagnostic
       -f: flush TPM session authentication handles
       -p: password associated with TPM (will prompt if not provided
and tpmpassword is needed)
       -s: rehash symmetric key PCRs (tpmpassword is mandatory)
       -g: force rehash even if secureboot is disabled
       -w: enable TPM TSS verbose
       -o: check if the TPM exists
       -t: return an error code representing TPM status
```

Results

Example

TPROXYD

Description

Display information about each proxy used on the Firewall (HTTP, SMTP, POP3, FTP, SSL).

Command

```
tproxyd [-d] [ -L | -s <opt> | -v | -h ]
-d: debug mode

-h, -?: help
-L: shows ICAP proxy licences
-s <http|smtp|pop3|ftp|ssl|av|antispam|rules|all>: shows
config
-v: version
```

Results

Example

```
U2504C099999999999>tproxyd -L
[2011-04-07 10:49:29] Icap url (reqmod) licence ok
[2011-04-07 10:49:29] Icap virus (respmod) licence ok
U2504C099999999999>U2504C099999999999>tproxyd -s http
OEM groups loaded
URL groups loaded
CN groups loaded
-- Http proxy : enabled
. BindAddr=0.0.0.0
. FullTransparent=1
. Postprocessing :
- policy: pass on failed
- datasize limit of 100000 Ko
. Antivirus:
- using default antiviral solution
- policy: block on failed
- policy: block on infected
. BindAddr=0.0.0.0
----- URL Filtering part -----
(Default action = Block) :
/usr/Firewall/ConfigFiles/URLFiltering/02
1: bypass_proxy ==> Pass
5: anonymizers ==> Blockpage
6: anorexia_and_bulimia ==> Blockpage
7: antivirus_bypass ==> Blockpage
8: art ==> Pass
...
...
...
```

```
U2504C099999999999>
```

TSD

Description

tsd daemon.

Command

```
tsd [-h] [-D]
-h [ --help ]: Display this message.
-D [ --daemonize ]: Daemonize, run in background.
```

Results

Example

```
VMSNSX08K0013A9>tsd -D
```

UDPSYNC

Description

Factory tool.

Command

```
udpsync [-s] [-p <port>] [-i <phase>] [-t <timeout>] [-v]
[<host>]
-s: Server
-p <port>: host port (default: 1991)
-i <phase>: ???
-t <timeout>: time before timeout in seconds (default: 60s)
-v: verbose mode enabled
```

Results

Example

URLCTL

Description

Manages the URL classification daemon

Command

```
urlctl [-v] [-o] [-q] [-b] [-B] [-r <reload arg>] [-R
<reason>] [-s <slotid>] [-u <URL>] [-U <arg>] [-c <CN>] [-C
<arg>] [-d] [-g <arg>]
-v Enable verbosity
-o Specify the output format, arg may be "text|html|xml|json
[,pretty]" (default is "text,pretty")
-q Do not print the results to standard output
-b Bypass CloudURL categorization (oem category will be
'Private IP Adresses')
-B Execute in background (will not print the results)
-r Make urld reload partially or totally its configuration.
arg may be "all", "engine", "filter", "verbose"
-R Text to explain why the reload was requested
-s Classify only for the categories defined in the provided
slot
-u Classify the given url
-U Classify all the URLs found in the file name given in arg
-c Classify the given CN
-C Classify all the CNs found in the file name given in arg
-d Display the current loaded configuration of urld
-g Display classification groups. arg can be "all", "url",
"cn"
```

Results

A command is sent to urld. Execution will wait until a response is received from urld unless background exection is requested

Example

URLD

Description

Url and CN classification daemon.

Command

```
urld [-d] [-D]
-d If an other process is already running, send it a signal to
switch its verbose mode, otherwise start with verbose mode
```

```
enabled.
-D Daemonize, run in background.
```

Results

Example

USERREQD

Description

User Requests daemon.

Command

```
userreqd [-d] [-D] [-h]
-D: will daemonize
-d: debug mode
-h: show help message
```

Results

Example

```
U2504C099999999999>userreqd -d
userreqd (pid 2517) is already running
Signal SIGINFO was sent to current process
Verbose status is modified
```

VMREPORT

Description

PAYG virtual machine reporting utility.

Command

```
vmreport -S
vmreport -U
vmreport -E
-S, --start: report Start event
-U, --up: report UP event
-E, --stop: report Stop event

-v, --verbose: verbose in console
```

```
-q, --quiet: quiet mode

-h, --help: display help
Whithout parameters, sync the events if needed.
```

WIZARDINIT

Description

First install wizard on VMs.

Command

```
wizardinit
```

Results

Example

documentation@stormshield.eu

All images in this document are for representational purposes only, actual products may differ.

Copyright © Stormshield 2025. All rights reserved. All other company and product names contained in this document are trademarks or registered trademarks of their respective companies.