1 Using the CGI Interface

Document-ID: 108416_en_01
Document-Description: AH EN MGUARD CGI INTERFACE

Make sure you always use the latest documentation.
It can be downloaded using the following link phoenixcontact.net/products.

Contents of this document

This document describes the usage of the CGI interfaces (additional HTTPS interfaces) of the mGuard device.

1.1 Introduction

The additional HTTPS interfaces are implemented as CGI (**C**ommon **G**ateway **I**nterface) scripts, providing the following features and functionality.

Some commands are executed synchronously: they indicate the success or failure of their operation with their return code. When a VPN connection is to be established, also the progress is displayed with every significant step.

nph-vpn.cgi / nph-diag.cgi

- Accessible from a conventional HTTPS client.
- Enable/disable a VPN connection.
- Retrieve the connection status of a VPN connection.
- Triggering a "download test" in order to check whether the mGuard is able to download a configuration file from a specified HTTPS server.
- Retrieve firmware version and hardware revision of the mGuard.
- Download a support snapshot.

nph-action.cgi / nph-status.cgi

The CGI interfaces *nph-action.cgi* and *nph-status.cgi* provide an extended range of features and functionality (see Section 1.6, “Interface nph.action.cgi” and Section 1.7, “Interface nph.status.cgi”).

1.2 Usage

The CGI scripts on the mGuard can be accessed via HTTPS through the same IP addresses and port on which the web interface is available. Only a different URL has to be used. Each access to a CGI script executes a single particular command. Each command responds with an UTF-8 text in the body of the HTTP reply, except for the command *snapshot*, which returns binary data. Some error conditions are signaled within the SSL respectively within the HTTP response. For example, an authorization failure is indicated by HTTP status code 401.

1.2.1 Available commands

nph-vpn.cgi / nph-diag.cgi

Table 1-1 Commands provided by the CGI scripts *nph-vpn.cgi* and *nph-diag.cgi*

CGI script	Command	Purpose
nph-vpn.cgi	*synup*	Activate a VPN connection (synchronous command)
	syndown	Deactivate a VPN connection (synchronous command)
	synstat	Determine the status of a VPN connection (synchronous command)
	sysinfo	Retrieve firmware version and hardware revision of the mGuard
	up	Enable a VPN connection (asynchronous command)
	down	Disable a VPN connection (asynchronous command)
	status	Determine the status of a VPN connection (asynchronous command)
	clear	Clears the instance of a VPN connection
nph-diag.cgi	*testpull*	Trigger a “download test” from an HTTPS server
	snapshot	Download a snapshot from the mGuard

nph-action.cgi / nph-status.cgi

For commands provided by the CGI scripts *nph-action.cgi* and *nph-status.cgi* see Section 1.6, “Interface nph.action.cgi” and Section 1.7, “Interface nph.status.cgi”.

1.2.2 Command syntax

Using the command line tool *wget* only functions in combination with mGuard firmware versions < 8.4.0. From mGuard firmware Version 8.4.0, the command line tool *curl* can be used (parameters and options differ!).

Example:

wget --no-check-certificate "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up"

curl --insecure "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up"

The option --no-check-certificate (*wget*) or --insecure (*curl*) ensures that the HTTPS certificate on the mGuard does not undergo any further checking.

The command line has the following syntax when using the utility *wget*:

*wget […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***'*
*wget […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***&name=***VPN_NAME***'*
*wget […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***&channel=***LNET_RNET***'*

The command line has the following syntax when using the utility *curl*:

*curl […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***'*
*curl […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***&name=***VPN_NAME***'*
*curl […] 'https://***MGUARD/CGI-SCRIPT***?cmd=***COMMAND***&channel=***LNET_RNET***'*

Table 1-2 Command syntax

wget […] or curl […]	Utility used to issue the HTTPS request and the required arguments. Please refer to the manual of the utility.
MGUARD	IP address and port number on which the mGuard listens for incoming HTTPS requests. The IP address may be preceded by username and password. **[<Username>:<Password>@]<IP Address>[:<Port>]** Example: admin:mGuard@192.168.1.254:443
CGI-SCRIPT	Name of the CGI script to be called, either *nph-vpn.cgi* or *nph-diag.cgi.*
COMMAND	Command to be executed, described in the following pages.
VPN_NAME	Name of the VPN connection to be enabled or disabled or which status is to be retrieved. Commands: *synup, syndown, synstat, up, down, status.*
LNET_RNET	Local and remote VPN network. Commands: status, clear.

Examples

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=synup&name=Service'
curl […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=synup&name=Service'

- Under Linux and other UNIX operating systems the string beginning with https:// starts and ends with single quote (‘). For other operating systems, like for example Windows, double quotes (“) may be used.
- Special characters, like a space, must be quoted according to the URL encoding rules if the VPN name contains such characters.
- If the URL includes the password as shown in the examples above, be aware that an intruder may read the password from the process list or the command line history. It could be advisable to use the user with the username *user*. This user has the rights to enable or disable a VPN connection or to retrieve its status by calling the CGI scripts described in this document, but this user has neither the rights to log onto the mGuard via HTTPS or SSH, nor to apply changes to the configuration.

1.2.3 Access rights

Table 1-3 Access rights

Command	User				
	root	**admin**	**user**	**netadmin**	**audit**
up, down, synup, syndown	x	x	x	-	-
status, synstat, sysinfo	x	x	x	x	x
status & channel, clear (central VPN gateway)	x	x	-	-	-
testpull, snapshot	x	x	-	-	-

1.3 Preconditions and restrictions

When executing the CGI scrips *nph-vpn.cgi*, *nph-diag.cgi, nph-status.cgi* and *nph-action.cgi*, only the following characters may be used in user names, passwords, and other user-defined names (for example, the name of a VPN connection):

– Letters: A - Z, a - z
– Digits: 0 - 9
– Special characters: - . _ ~

If other special characters, such as "space" or the "question mark", are used, they must be encoded accordingly (URL encoding).

Using the command line tool *wget* only functions in combination with mGuard firmware versions < 8.4.0. From mGuard firmware Version 8.4.0, the command line tool *curl* can be used (parameters and options differ!).

Example:

wget --no-check-certificate "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up"

curl --insecure "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up"

The option --no-check-certificate (*wget*) or --insecure (*curl*) ensures that the HTTPS certificate on the mGuard does not undergo any further checking.

1.3.1 Preconditions

The commands *synup, syndown, up* and *down* can only be used to trigger a VPN connection if it is configured as follows:

1. The VPN connection is disabled (menu **IPsec VPN >> Connections**).
2. At least one VPN tunnel of the VPN connection is enabled (menu **IPsec VPN >> Connections**, tab *General,* section *Transport and Tunnel Settings*).
3. Connection startup must be set to *Initiate* or *Initiate on traffic* (menu **IPsec VPN >> Connections**, tab *General*, section *Options*).

1.3.2 Restrictions

– Commands which are executed via the CGI interface may conflict with other activities of the mGuard and with other commands executed through different interfaces.
– A VPN connection should be triggered either by CMD contact or by the CGI interface. A combination of both is not supported.
– The commands *synup, syndown, up* and *down* are not supported for VPN connections which wait (*Connection startup* = *Wait*) for incoming VPN connections.
– The CGI interface should not be used during a firmware update or a restart of the mGuard.

1.4 Interface nph-vpn.cgi

1.4.1 cmd=(up|down), name=<VPN name>

These commands enable or disable the specified VPN connection. The name of the VPN connection must be specified with the parameter *name*.

The return value does not provide any information about the status of the VPN connection due to the asynchronous execution of these commands. Thus these commands should be followed by an execution of the command status to determine the status of the VPN connection.

Examples:

*wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?***cmd=up***&name=Service'*
*wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?***cmd=down***&name=Service'*

These commands return one of the following values in the HTTP reply:

Return value	Meaning
unknown	A VPN connection with the specified VPN name does not exist.
void	The VPN connection is inactive either due to an error or because it was not enabled using the CGI interface.
ready	The VPN connection is ready to establish tunnels or allow incoming queries regarding tunnel establishment.
active	At least one VPN tunnel of the VPN connection is established for the connection.

1.4.2 cmd=status, [name=(<VPN name>|*)]

This command retrieves, depending on the parameter *name*, the status either

1. of a specified VPN connection (name=[VPN name]), or
2. of all configured VPN connections (name=*), or
3. of all enabled or via *synup* activated VPN connections (parameter name not specified), providing also additional information.

In case of (1) and (2) the command returns one of the following values:

Return value	Meaning
unknown	A VPN connection with the specified VPN name does not exist.
void	The VPN connection is inactive either due to an error or because it was not enabled using the CGI interface.
ready	The VPN connection is ready to establish tunnels or allow incoming queries regarding tunnel establishment.
active	At least one VPN tunnel of the VPN connection is established for the connection.

1.4.2.1 cmd=status, name=<VPN name>

This command retrieves the status of the specified VPN connection.

Example:

*wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?***cmd=status***&name=Service1'*

Return value
active

1.4.2.2 cmd=status, name=*

This command retrieves the status of all configured VPN connections.

Example:

*wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?***cmd=status***&name=*'*

Return value
Service 1: active
Service 2: void

1.4.2.3 cmd=status (without parameter *name*)

This command retrieves the status of all enabled VPN connections, providing also additional information.

Example:

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=status'

(Parameter *name* not specified)

Return value	
fullname	Service1
name	MAI0003584192_1 instance
leftnet	192.168.1.0/24
leftgw	10.1.0.48
leftnatport	
leftid	O=Innominate, OU=Support, CN=mGuard 3
leftproto	
leftport	
rightnet	192.168.2.0/24
rightgw	77.245.33.67
rightnatport	
rightid	O=Innominate, OU=Support, CN=Central Gateway
rightproto	
rightport	

Return value	
isakmp	6
isakmp-txt	STATE_MAIN_I4 (ISAKMP SA established)
isakmp-ltime	157s
isakmp-algo	3DES_CBC_192-MD5-MODP1536
ipsec	7
ipsec-txt	STATE_QUICK_I2 (sent QI2, IPsec SA established)
ipsec-ltime	25526s
ipsec-algo	3DES_0-HMAC_MD5

The status of the VPN connection *Service2* is not returned in this example because this connection is not enabled.

1.4.3 cmd=(synup|synstat|syndown), name=<VPN name>

These commands enable, disable, or retrieve the status of the specified VPN connection. In contrast to the commands *up, down,* and *status*, these commands Mare executed synchronously which means that the operation returns once a certain status has been reached.

The first character of the response indicates whether the operation could be executed successfully. Further information is provided within the rest of the response line. The reply text consists of one line only, except for the command *synup*, which establishes a VPN connection. For this command the returned text contains progress messages about the establishment of the VPN connection and a final message with the overall result.

1.4.3.1 Response message format

Each message has the format: <TYPE> <CODE> <MESSAGE BODY>

TYPE	Message type, one character: P, R or F: **P** – progress message (command *synup* only) **R** – final message, operation terminated successfully **F** – final message, operation terminated with a failure
CODE	Max. 12 characters, an abbreviation about what was done in this step (for progress messages) respectively what the final result was (for final messages). Please refer to the next chapter.
MESSAGE BODY	A sequence of text fields delimited by blanks. Each field consists of an identifier and a value, separated by an equal sign. At the beginning of a MESSAGE BODY there is often the field “uptime=…“ or “tstamp=…”. “uptime=” indicates the operation time of the mGuard in seconds, with fractional digits since its last start up. “tstamp=” indicates the date and time when the message was generated.

1.4.3.2 Response code

The response may contain one of the following codes:

Response code	Description
EAMBIGUOUS	The specified name of the VPN connection was ambiguous because there are several VPN connections having the same name.
EBUSY	The called CGI script is currently busy with another task or it is blocked due to a running firmware update.
ECONFPULL	The test download of a configuration profile from the HTTPS server failed.
EINVAL	The CGI command or the parameters contain syntactical errors.
EVLOOKUPGW	The host name of the remote VPN gateway could not be resolved into an IP address.
EVLOOKUPROUT	No route known to the IP address of the remote VPN gateway.
ENOENT	The specified object does not exist (e.g. a VPN connection with the specified name does not exist).
ESYNVPN001	The VPN connection was established successfully but then it was interrupted (e.g. due to a network outage). The connection should be deactivated and established again. Use the command *synstat* to determine the status of the VPN connection.
EVDIFFALG1	During the handshaking at the beginning of establishing the VPN connection (negotiation of the ISAKMP SA) the devices did not agree on the strength of the keys or the cryptographic algorithms to be used in the first phase.
EVDIFFALG2	During the handshaking at the beginning of the establishment of the VPN connection (negotiation of the IPsec SA) the devices did not agree on the strength of the keys or the cryptographic algorithms to be used in the second phase.
EVIFDOWN	The network interface, through which the VPN connections should be established, does not have an uplink.
EVPEERNOENT1	The remote VPN peer does not know a VPN connection matching the criteria for the first IKE phase (negotiation of the ISAKMP SA). Probably the mGuard's or the peer's configuration is not correct.
EVPEERNOENT2	The VPN peer does not know a VPN connection which matches the criteria for the second IKE phase (negotiation of the IPsec SA). Probably the mGuard's or the peer's configuration is not correct.
EVTOUT1RESP	The mGuard did not receive a response from the remote VPN peer to his first message for establishing the VPN connection.
EVTOUTWRESP	The mGuard did not receive a response from the remote VPN peer after it has responded at least to one message.
OKCONFPULL	The test download of a configuration profile from the HTTPS server succeeded.
OKVACT	The VPN connection was already established when the *synup* command was called.
OKVDOWN	The VPN connection was disabled successfully.
OKVNOTACT	The VPN connection, which should be disabled by the *syndown* command, was already disabled.
OKVST1	The status of the specified VPN connection could be retrieved successfully.
OKVUP	The VPN connection could be established successfully.

1.4.3.3 cmd=synup

This command enables a VPN connection. The name of the VPN connection must be specified with the parameter name. This command is executed synchronously and returns once a certain status has been reached. The returned text contains progress messages about the establishment of the VPN connection and a final message with the overall result.

Example: Activate the VPN connection with the name *Service*

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=synup&name=Service'

Response:

```
P synup name=Service1
P deviceinfo uptime=9508.73 tstamp= 20120907095258a serial=2004010272 hostname=mguard
P vpnconn uptime=9508.79 id=MAI0003584192 gw=77.245.33.67
P dnslookup uptime=9508.83 ip=77.245.33.67
P routeinfo uptime=9508.87 via=ext1(10.1.0.48) ifstate=up
...
P IKEv1 uptime=9509.33 newstate=main-i2
...
P IKEv1 uptime=9509.88 newstate=main-i4
P IKEv1 uptime=9509.93 isakmp-sa=established id=#13
...
P IKEv1 uptime=9510.31 newstate=quick-i2 dpd=on
P IKEv1 uptime=9510.34 ipsec-sa=established id=#14 msg=IPsec SA 1 out of 1 is established  on this side.
R OKVUP uptime=9510.36 msg=The connection is established on this side.
```

When the mGuard executes the command synup, it performs the following steps:

1. Resolve the name of the remote VPN gateway into an IP address (if required).
2. Determine the network interface through which the VPN connection should be established and its connectivity.

The results of both steps are reported in the lines *dnslookup* and *routeinfo*. Only if those steps were executed successfully, the mGuard continues establishing the VPN connection. If the mGuard did not receive any response from the remote VPN peer, it sends an *IKE ping* to check its availability and reports the result.

Response pattern

A response of the *synup* command consists of several progress messages and a final message with the overall result. The following structure reflects the case of a successful established VPN connection.

Response consisting of progress messages (*P*) and one final message (*R*).

P synup name=**vpn_name**
P deviceinfo uptime=… tstamp=… serial=**XXXX** hostname=**string**
P vpnconn uptime=… id=**vNNN** gw=**hostname/IP**
P dnslookup uptime=… ip=**IP**
P routeinfo uptime=… via=**IF(IP)** ifstate=**up/down/error**
P IKEv1 uptime=… newstate=**status** [**key=value**…] send=…
P IKEv1 uptime=… state=**status** [**key=value** …] rcvd=…
P IKEv1 uptime=… newstate=**status**
P IKEv1 uptime=… newstate=**status** [**key=value** …] send=…
P IKEv1 uptime=… state=status [**key=value** …] rcvd=…
P IKEv1 uptime=… newstate=**status**
P IKEv1 uptime=… isakmp-sa=**status** [**key=value** …] info=…
P IKEv1 uptime=… newstate=**status** [**key=value**…] send=…
P IKEv1 uptime=… state=**status** [key=value …] rcvd=…
P IKEv1 uptime=… newstate=**status**
P IKEv1 uptime=… newstate=**status** [**key=value** …] send=…
P IKEv1 uptime=… ipsec-sa=**status** [**key=value** …] info=…
R OKVUP tstamp=… msg=VPN connection is established.

Progress messages

The response always starts with the five progress messages *synup, deviceinfo, vpnconn, dnslookup* and *routeinfo:*

synup	Displays the given *synup* command with its parameter *name*

deviceinfo	This message displays information about the mGuard. The format of this message is: **P deviceinfo uptime=… tstamp=… serial=XXXX hostname=string** The meaning of the fields are:			
	uptime=	Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. Example: uptime=75178.32		
	tstamp=	Date and time when the message was generated. Format: YYYYMMDDhhmmssx The date is followed by the time (UTC), and a lowercase letter. The meaning of the letters is as follows:		
		YYYY	4 digits indicating the year	
		MM	2 digits indicating the month	
		DD	2 digits indicating the day in the month	
		hh	2 digits indicating the hour of the day	
		mm	2 digits indicating the minute of the hour	
		ss	2 digits indicating the second of the minute	
		x	Lowercase letter indicating the state of system time and date of the mGuard.	
			a	System time and date are not yet synchronized.
			b	System time was set manually or synchronized by means of an imprecise timestamp recorded every 2 hours in the mGuard's file system.
			c	System time is synchronized by the battery buffered real time clock which had been synchronized manually or via NTP once.
			d	System time synchronized with an NTP server once.
			e	System time synchronized frequently with an NTP server.
			If more than one case applies, the last one of the alphabetical order is displayed.	
	serial= Serial number of the device. Spaces are substituted by underscores.			
	hostname= Hostname of the mGuard.			

vpnconn	Particular configuration properties of the VPN connection. The format of this message is as follows: **P vpnconn uptime=… id=vNNN gw=hostname/IP** The meaning of the fields are:	
	uptime=	Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. Example: uptime=75178.32
	id=	mGuard's internal name of the VPN connection under which the connection is maintained.
	gw=	Remote VPN gateway of the VPN connection.

dnslookup	Result of resolving the host name of the remote VPN peer into an IP address. The format of this message is as follows: **P dnslookup uptime=… ip=IP** The meaning of the fields are:	
	uptime=	Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. Example: uptime=75178.32
	ip=	IP address of the remote VPN peer.

routeinfo	Network interface, through which the mGuard will try to establish the VPN connection and interface status. The format of this message is as follows: **P routeinfo uptime=… via=IF(IP) ifstate=up/down/error** The meaning of the fields are:		
	uptime=	Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. Example: uptime=75178.32	
	via=	Network interface, through which the mGuard will try to establish the VPN connection. Possible values are "ext1", "ext2", "int" "dmz0" and "dial-in".	
	ifstate=	Status of the network interface. Possible values are:	
		up	Network interface is ready for operation.
		down	Network interface will become ready when traffic arrives that needs to be forwarded through it.
		error	Network interface is not ready to operate. In this case the *synup* command will return EVIFDOWN in the final message.

If the mGuard does not succeed to connect to the remote VPN peer although the previous steps were executed successfully, the mGuard checks with an IKE-ping, whether the remote site answers to IKE messages. The check will be skipped, if IKE messages had already been exchanged with the peer during the connection establishment.

<table>
<tr><td rowspan="6">ikeping</td><td colspan="3">Result of the IKE ping. The format of this message is as follows:
P ikeping uptime=… to=IP:PORT via=IF response=yes|no|error
The meaning of the fields are:</td></tr>
<tr><td>uptime=</td><td colspan="2">Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. Example: uptime=75178.32</td></tr>
<tr><td>to=</td><td colspan="2">IP address and port number of the IKE ping target.</td></tr>
<tr><td>via=</td><td colspan="2">Network interface through which the IKE ping was sent. Possible values are: “ext1”, “ext2”, “int”, "dmz0" and “dial-in”.</td></tr>
<tr><td rowspan="4">response=</td><td colspan="2">Tells whether the mGuard has received a reply to the IKE ping in time. Possible values are:</td></tr>
<tr><td>yes</td><td>The mGuard has received a reply from the remote VPN peer.</td></tr>
<tr><td>no</td><td>The mGuard did not receive any reply from the remote VPN peer within a certain period of time.</td></tr>
<tr><td>error</td><td>The mGuard failed to send an IKE ping.</td></tr>
</table>

Further progress messages are displayed during the establishment of the VPN connection. A final message will be displayed immediately upon failure.

<table>
<tr><td>IKEv1</td><td colspan="3">This message is displayed if:
– The mGuard has received or sent an IKEv1 packet.
– A phase of the connection establishment has been completed.

The message may contain several text fields with values. Some of them may indicate the crypto algorithms that are offered or selected.

The format of this message is as follows:

P IKEv1 uptime=… newstate=state [key=value …] send=…

P IKEv1 uptime=… state=state [key=value …] rcvd=…

P IKEv1 uptime=… newstate=state

P IKEv1 uptime=… isakmp-sa=status id=NN info=… or

P IKEv1 uptime=… ipsec-sa=established id=NN info=…

The meaning of the fields that may occur is as follows:</td></tr>
<tr><td></td><td>uptime=</td><td colspan="2">Operation time of the mGuard since its last start up. The value is displayed in seconds with fractional digits. For example: uptime=75178.32</td></tr>
<tr><td></td><td>newstate=</td><td colspan="2">Status change during the establishment of the VPN connection. The value is the name of the new status.</td></tr>
<tr><td></td><td>state=</td><td colspan="2">Current status of the VPN connection.</td></tr>
<tr><td></td><td>send=</td><td colspan="2">Details about a sent packet.</td></tr>
<tr><td></td><td>rcvd=</td><td colspan="2">Details about a received packet.</td></tr>
<tr><td></td><td rowspan="3">isakmp-sa=</td><td colspan="2">Completion status of the first phase. Possible values are:</td></tr>
<tr><td></td><td>established</td><td>A new ISAKMP Security Association (ISAKMP SA) has been established.</td></tr>
<tr><td></td><td>reused</td><td>A suitable ISAKMP SA had already been established for another VPN connection. It was reused for this one.</td></tr>
<tr><td></td><td>ipsec-sa=</td><td colspan="2">Completion status of the second phase. The value is always “established”.</td></tr>
<tr><td></td><td>id=</td><td colspan="2">Identifier of the first or the second phase. These identifiers are used by the mGuard internally during runtime. If an ISAKMP SA was reused, this identifier may be used to find the synup command, which established it.</td></tr>
</table>

Final message

If the VPN connection was established successfully, the command returns either **OKVUP** or **OKVACT**.

Otherwise one of the following values is returned: **EINVAL, EAMBIGUOUS, ENOENT, ESYNVPN001, EBUSY, EVLOOKUPGW, EVLOOKUPROUT, EVIFDOWN, EVTOUT1RESP, EVTOUTWRESP, EVDIFFALG1, EVDIFFALG2, EVPEERNOENT1, EVPEERNOENT2.**

Please refer to “Response code” on page 9 for an explanation about those codes.

1.4.3.4 cmd=synstat

This command retrieves the status of a VPN connection. The name of the VPN connection must be specified with the parameter *name*.

Example: Retrieve the status of the VPN connection with the name *Service*

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=synstat&name=Service'

Response:

```
R OKVST1 id=MAI0003584192 enabled=no activated=yes ike=OK ipsec=OK
```

If the status of the VPN connection could be retrieved successfully, **OKVST1** is returned with the following additional information:

OKVST1	The mGuard succeeded to determine the status of the VPN connection. The format of the message is as follows: **R OKVST1 id=id enabled=yesno1 activated=yesno2 ike=stat1 ipsec=stat2** The meaning of the fields are:		
	id=	Internal identifier of the VPN connection, which is used by the mGuard at runtime. It is not the configured name of the VPN connection.	
	enabled=	Indicates whether the VPN connection is configured on the mGuard as “enabled” or not. Possible values are:	
		yes	VPN connection is enabled.
		no	VPN connection is disabled.
	activated=	Indicates whether the VPN connection is “temporarily active”, which is the case if the VPN connection was established with the commands **synup** or **up** through the *CGI-script nph-vpn.cgi* or if it was established with the CMD contact. Possible values are:	
		yes	Temporarily active
		no	Not temporarily active
	ike=	Status of the ISAKMP Security Association (ISAKMP SA) which belongs to this VPN connection. The field is only present if the VPN connection is "temporarily active". Possible values are:	
		NAME	The ISAKMP SA is currently being established. The ISAKMP SA is in the state called **NAME**. The value of **NAME** differs from the other values “OK”, “EXP” or “DEAD”.
		OK	ISAKMP SA is established and can be used.
		EXP	ISAKMP SA expired. It has not yet been renewed.
		DEAD	ISAKMP SA does not exist for this VPN connection.
	ipsec=	Status of the IPsec Security Association (IPsec SA) which belongs to this VPN connection. Displayed only if the VPN connection is "temporarily active". Possible values and their meaning are:	
		NAME	The IPsec SA is currently being established. The IPsec SA is in the state called **NAME**. The value of **NAME** differs from the other values “OK”, “EXP” or “DEAD”.
		OK	IPsec SA is established and can be used.
		EXP	IPsec SA is expired. It is not yet renewed.
		DEAD	IPsec SA does not exist for this VPN connection.

If the status of the VPN connection could not be retrieved successfully, one of the following values is returned: **EINVAL**, **EAMBIGUOUS**, **ENOENT**.

Please refer to “Response code” on page 9 for an explanation about those codes.

1.4.3.5 cmd=syndown

This command disables a VPN connection. The name of the VPN connection must be specified with the parameter *name*.

Example: Disable the VPN connection with the name *Service*

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=syndown&name=Service'

Response:

```
R OKVDOWN
```

If the VPN connection was disabled successfully, the command returns either **OKVDOWN** or **OKVNOTACT**.

Otherwise one of the following values is returned: **EINVAL, EAMBIGUOUS, ENOENT, EBUSY**.

Please refer to “Response code” on page 9 for an explanation about those codes.

1.4.4 Central VPN gateway commands

The commands explained in the previous chapters are used on remote mGuards which initiate VPN connections to a central VPN gateway. Two more commands are available especially for using them on a central VPN gateway which uses the *VPN Tunnel Group* feature. The *VPN Tunnel Group* feature allows lots of remote mGuards to establish the VPN connection to one single configured VPN connection on the central VPN gateway.

A *VPN Tunnel Group* connection has *%any* as peer address and the specified remote VPN network is a large network (e.g. 192.168.0.0/16), including all networks of the remote mGuards (e.g. 192.168.**1**.0/24, 192.168.**2**.0/24, 192.168.**3**.0/24, etc.).

The VPN connection accepts ISAKMP SAs from many different remote mGuards at the same time. Each remote mGuard is expected to establish one or more IPsec SAs in tunnel mode where the remote mGuard requests a unique subnet of the configured remote network for each of its tunnel ends.

If the central VPN gateway has only one single *VPN Tunnel Group* configured, where all remote mGuards connect to, there is no way to determine whether there exists an active connection to an individual remote mGuard. Of course, *cmd=status* can be used without a specified VPN connection name (refer to Section 1.4.2.3) but this command would determine the status of all tunnels which is rather inefficient for querying the state of one single tunnel.

Sometimes it is also desired that the administrator of the central VPN gateway can clear the VPN connection of a specific remote VPN peer. This is in particular helpful if the remote VPN peer cannot establish a new tunnel for whatever interoperability reason. IPsec is a standard but sometimes other vendors are not fully compliant to it. Without an option to clear one specific VPN connection, it is only possible to restart the complete *VPN Tunnel Group* configuration. This would mean that all VPN tunnels are dropped and need to be reestablished.

1.4.4.1 cmd=status, channel=<LNet:RNet>

This command retrieves the status of the specified VPN tunnel. LNet stands for the local VPN network, *RNet* for the VPN network of the remote peer.

Return value	Meaning
unknown	This return value could have two reasons: – A matching tunnel currently does not exist. There is neither a configured and active tunnel which has the specified networks nor a matching established tunnel of a *VPN tunnel group*. – A matching channel is inactive due to an error (e.g. the external network is down or the hostname of the remote peer could not be resolved to an IP address (DNS)).
ready	A connection allows incoming queries regarding the tunnel establishment.
active	The tunnel is established.

Example: *wget […] 'https://admin:mGuard@77.245.33.67/nphvpn.cgi?cmd=status& channel=10.1.0.0/16:192.168.23.0/24'*

Response:

active

1.4.4.2 cmd=clear, channel=<LNet:RNet>

This command clears the specified VPN tunnel. *LNet* stands for the local VPN network, *RNet* for the VPN network of the remote peer.

Return value	Meaning
unknown	A matching tunnel currently does not exist.
Deleting connection …	The tunnel is being deleted.

Example:

wget […] 'https://admin:mGuard@77.245.33.67/nph-vpn.cgi?cmd=clear&channel=10.1.0.0/16:192.168.23.0/24'

Response:

```
002 "MAI1693250436_1"[2] 77.245.32.76: deleting connection "MAI1693250436_1"[2] instance    with peer 77.245.32.76 {isakmp=#0/ipsec=#0} cleared
```

1.4.5 cmd=sysinfo

This command retrieves the mGuard's software version, hardware name and hardware revision.

Example:

wget […] 'https://admin:mGuard@192.168.1.1/nph-vpn.cgi?cmd=sysinfo'

Response:

```
mGuardProductName=mGuard smart2
mGuardHardware=MGUARD2
mGuardHardwareVersion=00003000
mGuardVersion=8.6.1.default
```

1.5 Interface nph-diag.cgi

1.5.1 cmd=snapshot

The body of the HTTP response produced by the command snapshot is binary content. It should be saved to a file, preferable as snapshot.tar.gz. When using *wget*, use the option *output-document* to do so (*wget … --output-document=snapshot.tar.gz …*).

The snapshot contains the current configuration of the mGuard, the runtime parameters, and all log entries. The file also contains the VPN diagnostic messages described in this document of the last 100 VPN connection establishments at most, if the VPN connection is triggered by CMD contact or by the script nph-vpn.cgi and if the option **Archive diagnostic messages for VPN connections** (menu **IPsec VPN >> Global**, tab *Options*) is enabled. The file does not contain private information such as private keys or passwords.

Example: *wget […] 'https://admin:mGuard@192.168.1.1/nph-diag.cgi?cmd=snapshot'*

1.5.2 cmd=testpull

The mGuard can retrieve new configuration profiles from a HTTPS server in configurable time intervals, provided that the server makes them available as configuration profile for the mGuard (*.atv). When a new mGuard configuration differs from the current configuration, it will be downloaded and activated automatically. This option is configured through the web interface in the menu **Management >> Central Management**.

With this command it can be tested whether a configuration file can be downloaded from the configuration server according to the current settings of the mGuard. The mGuard does not apply the profile if execution of this command succeeded.

This command returns one of the following values in the HTTP reply:

<table>
<tr><td rowspan="3">OKCONFPULL</td><td colspan="2">The mGuard succeeded in downloading the configuration. The format of the message is:
R OKCONFPULL d=digest
The meaning of the fields are:</td></tr>
<tr><td>digest</td><td>Alphanumerical string the mGuard sends to the IDM (MGUARD DM, MGUARD Device Manager) with the HTTP request in order to indicate which version of the configuration file has been downloaded.</td></tr>
<tr><td></td></tr>
<tr><td rowspan="4">ECONFPULL</td><td colspan="2">Downloading the configuration file failed. The format of the message is as follows:
F ECONFPULL http-code=code msg=message
The meaning of the fields are:</td></tr>
<tr><td>code</td><td>HTTP status code returned by the HTTPS server. Empty, if the HTTP status code could not be transferred due to an error on another layer, e.g. on the Secure Socket Layer (SSL).</td></tr>
<tr><td>message</td><td>This message indicates the cause of the error and may also contain further information. It contains also the error message of the HTTPS server if the HTTP status code is known.</td></tr>
<tr><td></td></tr>
</table>

Example: *wget […] 'https://admin:mGuard@192.168.1.1/nph-diag.cgi?cmd=testpull'*

Response:

```
R OKCONFPULL tstamp=20120515094007e d=d12851f0b9801e0df45c5794c7f392c5
```

1.6 Interface nph.action.cgi

User “root“ and “admin“

The following commands are executable by the users **root** and **admin**.

Row actions

https://admin:mGuard@192.168.1.1/nph-action.cgi?action=<ACTION>&name=<NAME>

https://admin:mGuard@192.168.1.1/nph-action.cgi?action=<ACTION>&rowid=<ROWID>

Table 1-4 Row actions – Parameters

Parameter	Description
name	Name of the connection, rule record, integrity check
rowid	Unique ID from the configuration. (gaiconfig --goto VPN_CONNECTION:0 --get-rowid)

Table 1-5 Row actions – Actions

Action	Description
fwrules/inactive	Deactivates a firewall rule record
fwrules/active	Activates a firewall rule record
vpn/stop	Also stops an IPsec connection like "nph-vpn.cgi" but with less complexity
vpn/start	Also starts an IPsec connection like "nph-vpn.cgi" but with less complexity
openvpn/stop	Stops an OpenVPN connection
openvpn/start	Starts an OpenVPN connection
cifsim/validaterep	Validates the report of a CIFS/IM scan
cifsim/check-start	Starts a CIFS/IM check
cifsim/init-start	Intializes a new CIFS/IM integrity-database
cifsim/cancel	Cancels a running CIFS/IM job
cifsim/erase-db	Deletes the CIFS/IM database
cifsim/access-scan	Starts a quick file permission check of a share

User firewall logout

https://admin:mGuard@192.168.1.1/nph-action.cgi?action=userfw/logout&name=<NAME> &ip=<IP>

Table 1-6 User firewall logout – Parameters

Parameter	Description
name	Username of the logged in user of the user firewall
ip	The actual IP-Address of the logged in user of the user firewall

Table 1-7 User firewall logout – Actions

Action	Description
userfw/logout	Logs out the logged in firewall user

Simple commands

(Parameters *name* or *ID* not required)

https://admin:mGuard@192.168.1.1/nph-action.cgi?action=<ACTION>

Table 1-8 Simple commands – Actions

Action	Description
switch/purge-arlt	Resets the Address Resolution Table in the internal switch
switch/reset-phy-counters	Resets the PHY counters inside the switch

User "mobile", "root" and "admin"

The following commands are executable by the users **mobile**, **root** and **admin**. The user **mobile** is available since firmware version 8.3.0.

Mobile actions (User: mobile / root / admin)

– **Only mGuard firmware version 8.3:**
https://admin:mGuard@192.168.1.1/nph-action.cgi?action=gsm/call&dial=<NUMBER>&timeout=<TIMEOUT>

– **mGuard firmware version 8.3 and 8.4:**
https://admin:mGuard@192.168.1.1/nph-action.cgi?action=gsm/sms&dial=<NUMBER> &msg=<MESSAGE>

Table 1-9 Mobile actions – Parameters

Parameter	Description
dial	Telephone number of the destination
timeout	Time in seconds until the call is finished
msg	Content of the short message (should be cleaned of special characters like umlauts)

Table 1-10 Mobile actions – Actions

Action	Description
gsm/call	Starts a phone call
gsm/sms	Sends a text message (SMS)

1.7 Interface nph.status.cgi

The following commands are executable by the users **root** and **admin**.

Table 1-11 CGI status

Parameter	Description
/network/modem/state	**Modem state**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/network/modem/state	
Answer: *online* | *offline*	
/network/ntp_state	**NTP time synchronization state**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/network/ntp_state	
Answer: *disabled* | *not_synced* | *synchronized*	
/system/time_sync	**State of the system time synchronization**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/system/time_sync	
Answer: *not_synced* | *manually* | *stamp* | *rtc* | *ntp* | *gps* | *gpslost*	
/ecs/status	**State of the ECS**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/ecs/status	
Answer: "1" for not present, "2" for removed, "3" for present an in synchronization, "4" for not in synchronization and "8" for generic error	
/vpn/con	**State of a VPN connection**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/vpn/con&name=<Verbindungsname>	
Answer: – */vpn/con/<rowid>/armed=[yes|no]* Shows whether the connection is started or not – */vpn/con/<rowid>/ipsec=[down|some|up]* Shows the IPsec state. – */vpn/con/<rowid>/isakmp=[up|down]* Shows the ISAKMP state. – */vpn/con/<rowid>/sa_count=<number>* Number of configured tunnel – */vpn/con/<rowid>/sa_count_conf=<number>* Number of configured enabled tunnel	
/fwrules	**State of a firewall rule record**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/fwrules&name=<rule record >	
Answer: – */fwrules/<rowid>/expires=<seconds since 1.1.1970>* Expiration date – 0 for no expiration – */fwrules/<rowid>/state=[inactive|active]* Activation state of the firewall rule record	
/cifs/im	**State of a share in the context of CIFS**
https://admin:mGuard@192.168.1.1/nph-status.cgi?path=/cifs/im&name=<WS_SHARE>	

Table 1-11 CGI status

Parameter	Description
Answer: **Actual check** – */cifs/im/<rowid>/curr/all=<number>* Number of files – */cifs/im/<rowid>/curr/end=<seconds>* End time of the current check in seconds since 1.1.1970 – */cifs/im/<rowid>/curr/numdiffs=<number>* Currently found number of diffs. – */cifs/im/<rowid>/curr/operation=[none|suspend|check|idb_build]* Current operation – */cifs/im/<rowid>/curr/scanned=<number>* Number of currently checked files – */cifs/im/<rowid>/curr/start=<seconds>* Start time in seconds since 1.1.1970 **Last check** – */cifs/im/<rowid>/last/duration=<number>* Number of seconds of the last duration – */cifs/im/<rowid>/last/numdiffs=<number>* Number of differences found during the last check – */cifs/im/<rowid>/last/start=<seconds> start time in seconds since 1.1.1970* Start time in seconds since 1.1.1970 – */cifs/im/<rowid>/last/result*=<see "Last Results“ below"> **Log results** – */cifs/im/<rowid>/log/fname=<filename of the log file>* – */cifs/im/<rowid>/log/hash=<sha1 hash>* – */cifs/im/<rowid>/log/result*=<siehe "Log result" below>	

Table 1-11 CGI status

Parameter	Description
Last results	
– *-1:* The share has not yet been checked. Probably no integrity database exists. – *0:* Last check finished successfully. – *1:* The process failed due to an unforeseen condition, please consult the logs. – *2:* Last check was aborted due to timeout. – *3:* The integrity database is missing or incomplete. – *4:* The signature of the integrity database is invalid. – *5:* The integrity database was created with a different hash algorithm. – *6:* The integrity database is the wrong version. – *7:* The share which is to be checked is not available. – *8:* The share which is to be used as checksum memory is not available. – *11:* A file could not be read due to an I/O failure. Please consult the report. – *12:* The directory tree could not be traversed due to an I/O failure. Please consult the report.	
Log result	
– *unchecked* – The signature has not been verified, yet. – *valid* – The signature is valid. – *Emissing – ERROR: The report is missing.* – *Euuid_mismatch – ERROR:* The report does not belong to this device or is not up to date. – *Ealgo_mismatch – ERROR:* The report was created with a different hash algorithm. – *Etampered – ERROR:* The report was tampered with. – *Eunavail – ERROR:* The report is not available. For example the share might not be mounted. – *Eno_idb* – No report exists, because of a missing integrity database.	