```
        "dst_netmas
        "dst_port":
        "id": 1,
        "protocol":
        "src_net":
        "src_netmas
        "verdict":
      }
    ]
  }
},
"hostname": "mGuard
"network": {
  "LAN0": {
    "address": "192
    "mask": 24,
    "router_mode":
  },
  "WAN": {
    "address": "10.
    "gateway": "10.
    "mask": 24,
    "router_mode":
  },
  "routes": [
```

FL MGUARD 1000
RESTful Configuration API
mGuardNT 1.8.x

User manual

UM EN MGUARD NT CONFIG API

User manual

FL MGUARD 1000 – RESTful Configuration API – mGuardNT 1.8.x

UM EN MGUARD NT CONFIG API, Revision 11 2024-05-16

This user manual is valid for:

Designation	Order No.
FL MGUARD 1102	1153079
FL MGUARD 1105	1153078

Firmware version mGuardNT 1.8.x

For further information see mGuardNT 1.8.x firmware – Release Notes.

108898_en_11

PHOENIX CONTACT GmbH & Co. KG • Flachsmarktstraße 8 • 32825 Blomberg • Germany
phoenixcontact.com

Table of contents

1 For your safety

Read this user manual carefully and keep it for future reference.

1.1 Identification of warning notes

❗ This symbol together with the **NOTE** signal word warns the reader of actions that might cause property damage or a malfunction.

ℹ Here you will find additional information or detailed sources of information.

1.2 About this user manual

The following elements are used in this user manual:

Bold	Designations of operating elements, variable names or other accentuations
Italic	– Product, module or component designations (e.g., *tftpd64.exe*, *Config API*) – Foreign designations or proper names – Other accentuations
–	Unnumbered list
1.	Numbered list
•	Operating instructions
⇒	Result of an operation

1.3 Qualification of users

The use of products described in this user manual is oriented exclusively to:

- Electrically skilled persons or persons instructed by them. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.
- Qualified application programmers and software engineers. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.

1.4 Intended use

- The devices of the FL MGUARD 1000 series are security routers for industrial use, with integrated stateful packet inspection firewall. They are suitable for distributed protection of production cells or individual machines against manipulation.
- The devices are intended for installation in a control cabinet.

1.5 Modifications to the product

Modifications to hardware and firmware of the device are not permitted.

- Incorrect operation or modifications to the device can endanger your safety or damage the device. Do not repair the device yourself. If the device is defective, please contact Phoenix Contact.

1.6 IT security

You have to protect components, networks, and systems against unauthorized access and ensure the integrity of data. As a part of this, you must take organizational and technical measures to protect network-capable devices, solutions, and PC-based software.

Phoenix Contact strongly recommends using an Information Security Management System (ISMS) to manage all of the infrastructure-based, organizational, and personnel measures that are needed to ensure compliance with information security directives.

Furthermore, Phoenix Contact recommends that at minimum the following measures are taken into consideration.

More detailed information on the measures described is available on the following websites (last accessed on 2024-04-15; partly only available in German):

- bsi.bund.de/it-sik.html
- ics-cert.us-cert.gov/content/recommended-practices

Use the latest firmware version

Phoenix Contact regularly provides firmware updates. Any firmware updates available can be found on the product page for the respective device.

- Ensure that the firmware on all devices used is always up to date.
- Observe the Change Notes for the respective firmware version.
- Pay attention to the security advisories published on Phoenix Contact‘s Product Security Incident Response Team (PSIRT) website regarding any published vulnerabilities.

Use up-to-date security software

- Install security software on all PCs to detect and eliminate security risks such as viruses, trojans, and other malware.
- Ensure that the security software is always up to date and uses the latest databases.
- Use whitelist tools for monitoring the device context.
- Use an Intrusion-Detection system for checking the communication within your system.

Take Defense-in-Depth strategies into consideration when planning systems

It is not sufficient to take measures that have only been considered in isolation when protecting your components, networks, and systems. Defense-in-Depth strategies encompass several coordinated measures that include operators, integrators, and manufacturers.

- Take Defense-in-Depth strategies into consideration when planning systems.

Perform regular threat analyses

- To determine whether the measures you have taken still provide adequate protection for your components, networks, and systems, threat analyses should be performed regularly.
- Perform a threat analysis on a regular basis.

Deactivate unneeded communication channels

- Deactivate unnecessary communication channels (e.g., SNMP, FTP, BootP, DCP, etc.) on the components that you are using.

Do not integrate components and systems into public networks

- Avoid integrating your components and systems into public networks.
- If you have to access your components and systems via a public network, use a VPN (Virtual Private Network).

Restrict access rights

- Restrict access rights for components, networks, and systems to those individuals for whom authorization is strictly necessary.
- Deactivate unused user accounts.

Secure access

- Change the default login information after initial startup.
- Use secure passwords reflecting the complexity and service life recommended in the latest guidelines.
- Change passwords in accordance with the rules applicable for their application.
- Use a password manager with randomly generated passwords.
- Wherever possible, use a central user administration system to simplify user management and login information management.

Use secure access paths for remote access

- Use secure access paths such as VPN (Virtual Private Network) or HTTPS for remote access.

Set up a firewall

- Set up a firewall to protect your networks and the components and systems integrated into them against external influences.
- Use a firewall to segment a network or to isolate a controller.

Activate security-relevant event logging

- Activate security-relevant event logging in accordance with the security directive and the legal requirements on data protection.

Secure access to SD cards

Devices with SD cards require protection against unauthorized physical access. An SD card can be read with a conventional SD card reader at any time. If you do not protect the SD card against unauthorized physical access (such as by using a secure control cabinet), sensitive data is accessible to all.

- Ensure that unauthorized persons do not have access to the SD card.
- When destroying the SD card, ensure that the data cannot be retrieved.

1.7 Latest security instructions for your product

Product Security Incident Response Team (PSIRT)

The Phoenix Contact PSIRT is the central team for Phoenix Contact as well as for its subsidiaries, authorized to respond to potential security vulnerabilities, incidents and other security issues related to Phoenix Contact products, solutions as well as services.

Phoenix Contact PSIRT manages the disclosure, investigation internal coordination and publishes security advisories for confirmed vulnerabilities where mitigations/fixes are available.

The PSIRT website (phoenixcontact.com/psirt) is updated regularly. In addition, Phoenix Contact recommends subscribing to the PSIRT newsletter.

Anyone can submit information on potential security vulnerabilities to the Phoenix Contact PSIRT by e-mail.

1.8 Support

ℹ For additional information on the device as well as release notes, user assistance and software updates, visit: phoenixcontact.net/products/<order number>.

In the event of problems with your device or with operating your device, please contact your supplier.

To get help quickly in the event of an error, make a snapshot of the device configuration immediately when a device error occurs, if possible (see Section 3.10, ““snapshot” end point”). You can then provide the snapshot to the support team.

2 Using the RESTful Configuration API

2.1 Introduction

The device can be configured via the web-based management, but also via the *RESTful Configuration API* (or *Config API* for short).

Only experienced users should use the *Config API*.

As a machine-to-machine interface, the *RESTful Configuration API* enables automated and dynamic control and configuration of the device.

The *Config API* is provided via a RESTful web server of the device.

The data is transmitted via the HTTP(S) protocol, which is also used to call up websites (see Figure 2-1).

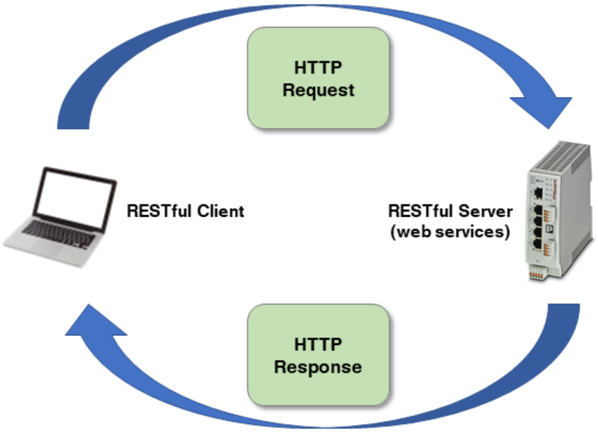


Figure 2-1 Data exchange between RESTful client and RESTful server (REST = *Representational State Transfer*)

Various RESTful clients can be used to access the RESTful server, e.g., to request the device configuration using a *GET request* or to change the device configuration using a *POST request*.

For example, appropriate management software, a command-line tool (e.g., *curl*), a graphical RESTful client for Windows (e.g., *Nightingale*) or a web browser extension (e.g., *YARC!* for *Google Chrome*) can be used as a RESTful client. However, they must be separately obtained and installed.

NOTE: Third-party software
Phoenix Contact does not undertake any guarantee or liability for the use of third-party products. Any reference to or descriptions of third-party software does not constitute a recommendation, rather serves as an example of a program that could be used.

The *RESTful Configuration API* and application examples for the available end points are described in the following sections:

- Section 2.2, “Structure of HTTP requests”
- Section 2.3, “Examples”
- Section 3, “Description of the end points”

2.2 Structure of HTTP requests

Certain elements are transferred to a RESTful server in an *HTTP request* (see also examples in Table 2-1).

URL escaping
To prevent the special characters in JSON strings being interpreted incorrectly by the server, the characters may have to be recoded (*URL escape*).

2.3 Examples

2.3.1 Login: Create CSRF token and session cookie

For secure device configuration and administration via the *Config API*, on login of the user first the device must generate a *CSRF token* and secure *session cookie* (= RESTful-Server) and transmit them to the RESTful client.

The **CSRF token** and **session cookie** must be entered again in later requests: Save the information in a suitable place.

Step 1

1) Request CSRF token (end point “csrf”)

A **login cookie** and a **CSRF token** to secure a session are created by the device (= RESTful server) and transferred to the RESTful client.

Proceed as follows:

- GET request on the “csrf” end point.

⇒ A *CSRF token* and a *login cookie* (e.g., *login_cookie*) are generated.

- Save or copy the *CSRF token* and the *login cookie* as well, if necessary.

Example:

```
curl -c login_cookie.txt -k -X GET https://192.168.1.1:443/api/v1/csrf
```

Respons e:

```
{"content":"ImIzYzk3N2UyYjFiYThlZmY5Yzc1M2FhZTQxYmE1MmYxZDQwZjQ3ZWYi_l2dMwHYPyJeVR1rFgli0Tww","
envelope":{"identifier":{"contentID":"d2c01a66","functionalID":"d2c01a66"},"version":1},"error":[],"schemes":[],"status":0}
```

CSRF token: The *CSRF token* is returned as “*content*” and is only valid in conjunction with a *session cookie* that must be created in the next step via the “*login*” end point. The current CSRF token must be entered in all following *POST requests* within the ongoing session.

Login cookie: The *login cookie* is saved on the configuration computer with option -c <*login_cookie_name*> when *curl* is used.
When graphical RESTful clients are used, the cookie is often automatically saved.

Step 2

2) Log in and start session (“login” end point)

As part of user log in, the following entries must be made:

- **Header:**
  - **Content type**: *application/json*
  - **X CSRF TOKEN**: the previously generated <*CSRF-Token*> or the variable
- **Login-Cookie (**may happen automatically): the previously generated <*login_cookie*>
- **Username/Password** (as content)

Proceed as follows:

- POST request on the “login” end point.

⇒ **Session cookie** (*session_cookie*) is generated.

- Save or copy the *session cookie*, if necessary.

⇒ The *session cookie* is necessary in order to implement following *GET* and *POST requests*.

Example:

```
curl -b login_cookie.txt -c session_cookie.txt -k -X POST https://192.168.1.1:443/api/v1/login -H "X-CSRF-Token:
ImIzYzk3N2UyYjFiYThlZmY5Yzc1M2FhZTQxYmE1MmYxZDQwZjQ3ZWYi_l2dMwHYPyJeVR1rFgIi0Tww" -H "Content-
Type: application/json" -d '{"content": {"username": "admin", "password": "private"}, "envelope": {"version": 1}}'
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
{"name":"login.login.c1a52347","url":"/v1/login/scheme/login.login.c1a52347"}],"status":0}
```

Session cookie: A session cookie secured via *login cookie* and *CSRF token* is generated and saved on the configuration computer (when *curl* is used) with the help of the *-c <session_cookie_name>* option.

GET and POST requests

In all subsequent GET and *POST requests* during a session, *curl* must be invoked with option *-b <session_cookie_name>* in order to use the saved *session cookie*.

Using different RESTful clients

curl

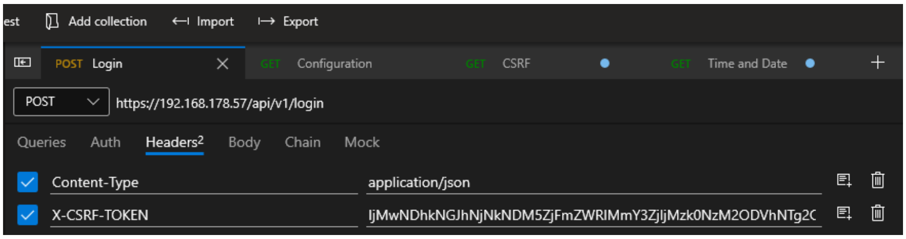
See Section 3.3, “End points “csrf” / “login” / “logout””.

YARC!

See Section 2.5, “Using the “YARC!” RESTful client (Chrome)”.

Nightingale

For **RESTful-Client “Nightingale”** (for Microsoft Windows), the *CSRF token* and *session cookie* are generated and used in a similar manner to *YARC!*.



POST https://192.168.178.57/api/v1/login

☑ Content-Type: application/json

☑ X-CSRF-TOKEN: IjMwNDhkNGJhNjNkNDM5ZjFmZWRlMmY3ZjljMzk0NzM2ODVhNTg2C

Figure 2-2 Example: RESTful client “*Nightingale*“

2.3.2 Change device configuration (POST request)

If you want to change a variable of the device configuration in the “*configuration*” end point using a *POST request*, you must also transfer the variables that do not need to be changed (i.e., all *keys* of the *frame*) to the RESTful server with the same *POST request*.

For example, if making a change to the *hostname*, the existing *firewall rules*, *network settings*, etc. must also be specified in the *POST request* (see under: “Example”).

Recommended procedure

- Execute a *GET request* on the “*v1/configuration*” end point to display the current device configuration.
- Edit the desired variables.
- Copy the configuration into a POST request.
- Send the modified configuration to the device as a *POST request*.

Please note:

Depending on the RESTful client used, you need to further adapt the *POST request* before sending it.

Some parts of the response to the *GET request* must not be sent in a *POST request*. For example, a *POST request* using the *curl* RESTful client ends with the following entry: "*envelope": {"version": 1}}*'

Observe the correct use of inverted commas at the start and end of the content block (content): ... -d '{”content": {"firewall" ... "*envelope": {"version": 1}}*'

Example

A change to the hostname of the device looks like this, for example.

(In this example, the *curl* RESTful client is used via the Linux command line.)

1. Request the current values of the “configuration” end point (**GET request**).

GET request:

curl -k **-b** *session_cookie* -X GET https://192.168.1.1:443/api/v1/configuration

Response: (For a structured view of the example with firmware 1.8.0), see Section 4.2.)

```
{"content": {"fileinfo": {"devtype": "0001010111020000", "firmware": "1.4.1"}, "firewall": {"forward": {"log_all_matches": "ON", "log_policy": "ON", "sanity_check": "ON", "stealth_allow_dhcp": "ON", "tables": [{"in_netzone": "NETZONE1", "out_netzone": "NETZONE2", "rules": []}, {"in_netzone": "NETZONE2", "out_netzone": "NETZONE1", "rules": [{"comment": "", "dst_network": "0.0.0.0/0", "dst_port": "ALL", "id": 0, "log": "OFF", "protocol": "ALL", "src_network": "0.0.0.0/0", "verdict": "ACCEPT"}]}], "testmode": "ON"}, "input": {"rules": [{"id": 0, "log": "OFF", "service": "HTTPS", "source": "NETZONE2", "verdict": "ACCEPT"}, {"id": 1, "log": "OFF", "service": "HTTPS", "source": "NETZONE1", "verdict": "ACCEPT"}]}, "port_forward": {"rules": [{"comment": "", "dst_ip": "0.0.0.0", "dst_port": 443, "inc_port": 5000, "protocol": "TCP", "src_interface": "NETZONE1"}, {"comment": "", "dst_ip": "0.0.0.0", "dst_port": 102, "inc_port": 5001, "protocol": "UDP", "src_interface": "NETZONE1"}]}}, "logging": {"remote": {"address": "syslog.my-mguard.com", "port": 513, "protocol": "UDP", "status": "ON"}}, "network": {"mode": "ROUTER", "nat": {"1_1_nat": [{"comment": "", "id": 0, "real_network": "192.168.1.100", "virt_network": "10.1.0.101"}, {"comment": "", "id": 1, "real_network": "192.168.1.200", "virt_network": "10.1.0.102"}], "masquerading": [{"from_ip": "0.0.0.0/0", "id": 0, "outgoing_on_if": "NETZONE1"}]}, "netzone1": {"mode": "DHCP"}, "netzone2": {"address": "192.168.1.1", "netmask": 24}, "routing": {"routes": [{"comment": "Production3", "gateway": "192.168.1.10", "network": "192.168.10.0/24"}]}, "stealth": {"management_address": "192.168.1.1", "management_gateway": "192.168.1.254", "management_netmask": 24}}, "service": {"dhcp_server": {"dns": "192.168.1.1", "gateway": "192.168.1.1", "lease_time": "12h", "netmask": 24, "range_high": "192.168.1.254", "range_low": "192.168.1.2", "status": "ON", "wins_server": ""}, "dnscache": {"allowed_requests": ["NETZONE2", "NETZONE1"], "dns_servers": "USER_DEFINED", "log": "ON", "user_defined": [{"comment": "", "ip": "212.2.220.212"}]}, "ntp": {"allow_client_requests": ["NETZONE2"], "server": [{"address": "0.pool.ntp.org", "comment": "", "port": 123}, {"address": "1.pool.ntp.org", "comment": "", "port": 123}, {"address": "2.pool.ntp.org", "comment": "", "port": 123}], "status": "ON"}, "snmp": {"allow_requests_from": ["NETZONE2"], "ro_community_string": "public", "status_v2c": "ON", "status_v3": "ON", "user": {"username": "snmp-v3-user"}}, "web": {"session_timeout": 450}}, "system": {"hostname": "OldName", "store_config_on_sdcard": "OFF", "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."}, "zoneinfo": "UTC"}, "envelope": {"identifier": {"contentID": "8effd771", "functionalID": "dc5a1dcc"}, "version": 1}, "error": [], "schemes": [{"name": "common.4710ab60", "url": "/v1/configuration/scheme/common.4710ab60"}, {"name": "common.types.f0bf23da", "url": "/v1/configuration/scheme/common.types.f0bf23da"}, {"name": "configuration.fileinfo.b3afd1b0", "url": "/v1/configuration/scheme/configuration.fileinfo.b3afd1b0"}, {"name": "configuration.firewall.62d07c99", "url": "/v1/configuration/scheme/configuration.firewall.62d07c99"}, {"name": "configuration.hostname.27e2cb1c", "url": "/v1/configuration/scheme/configuration.hostname.27e2cb1c"}, {"name": "configuration.logging.fce1b9ba", "url": "/v1/configuration/scheme/configuration.logging.fce1b9ba"}, {"name": "configuration.network.0edde642", "url": "/v1/configuration/scheme/configuration.network.0edde642"}, {"name": "configuration.service.69f74574", "url": "/v1/configuration/scheme/configuration.service.69f74574"}, {"name": "configuration.system.9df06664", "url": "/v1/configuration/scheme/configuration.system.9df06664"}, {"name": "configuration.zoneinfo.e8437e00", "url": "/v1/configuration/scheme/configuration.zoneinfo.e8437e00"}], "status": 0}
```

2. Copy the modified response into a **POST request**.

URL escaping

– Observe the correct use of inverted commas at the start and end of the content block (*content*): ... -d '{"content": {"fileinfo" ... "envelope": {"version": 1}}'

– Check whether or not specific characters you have used must be re-coded using *URL escaping*.

POST request:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/configuration -d '{{"content": {"fileinfo": {"devtype": "0001010111020000", "firmware": "1.4.1"}, "firewall": {"forward": {"log_all_matches": "ON", "log_policy": "ON", "sanity_check": "ON", "stealth_allow_dhcp": "ON", "tables": [{"in_netzone": "NETZONE1", "out_netzone": "NETZONE2", "rules": []}, {"in_netzone": "NETZONE2", "out_netzone": "NETZONE1", "rules": [{"comment": "", "dst_network": "0.0.0.0/0", "dst_port": "ALL", "id": 0, "log": "OFF", "protocol": "ALL", "src_network": "0.0.0.0/0", "verdict": "ACCEPT"}]}], "testmode": "ON"}, "input": {"rules": [{"id": 0, "log": "OFF", "service": "HTTPS", "source": "NETZONE2", "verdict": "ACCEPT"}, {"id": 1, "log": "OFF", "service": "HTTPS", "source": "NETZONE1", "verdict": "ACCEPT"}]}, "port_forward": {"rules": [{"comment": "", "dst_ip": "0.0.0.0", "dst_port": 443, "inc_port": 5000, "protocol": "TCP", "src_interface": "NETZONE1"}, {"comment": "", "dst_ip": "0.0.0.0", "dst_port": 102, "inc_port": 5001, "protocol": "UDP", "src_interface": "NETZONE1"}]}}, "logging": {"remote": {"address": "syslog.my-mguard.com", "port": 513, "protocol": "UDP", "status": "ON"}}, "network": {"mode": "ROUTER", "nat": {"1_1_nat": [{"comment": "", "id": 0, "real_network": "192.168.1.100", "virt_network": "10.1.0.101"}, {"comment": "", "id": 1, "real_network": "192.168.1.200", "virt_network": "10.1.0.102"}], "masquerading": [{"from_ip": "0.0.0.0/0", "id": 0, "outgoing_on_if": "NETZONE1"}]}, "netzone1": {"mode": "DHCP"}, "netzone2": {"address": "192.168.1.1", "netmask": 24}, "routing": {"routes": [{"comment": "Production3", "gateway": "192.168.1.10", "network": "192.168.10.0/24"}]}, "stealth": {"management_address": "192.168.1.1", "management_gateway": "192.168.1.254", "management_netmask": 24}}, "service": {"dhcp_server": {"dns": "192.168.1.1", "gateway": "192.168.1.1", "lease_time": "12h", "netmask": 24, "range_high": "192.168.1.254", "range_low": "192.168.1.2", "status": "ON", "wins_server": ""}, "dnscache": {"allowed_requests": ["NETZONE2", "NETZONE1"], "dns_servers": "USER_DEFINED", "log": "ON", "user_defined": [{"comment": "", "ip": "212.2.220.212"}]}, "ntp": {"allow_client_requests": ["NETZONE2"], "server": [{"address": "0.pool.ntp.org", "comment": "", "port": 123}, {"address": "1.pool.ntp.org", "comment": "", "port": 123}, {"address": "2.pool.ntp.org", "comment": "", "port": 123}], "status": "ON"}, "snmp": {"allow_requests_from": ["NETZONE2"], "ro_community_string": "public", "status_v2c": "ON", "status_v3": "ON", "user": {"username": "snmp-v3-user"}}, "web": {"session_timeout": 450}}, "system": {{"hostname": "NewName", "store_config_on_sdcard": "OFF", "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."}, "zoneinfo": "UTC"}, "envelope":  {"version": 1}}'
```

Response: (For a structured view of the example (Request and Response) with firmware 1.8.0, see Section 4.3)

2.3.3 Update device firmware (POST request)

In this example, the *curl* RESTful client is used via the Linux command line.

You can update the device firmware with a locally saved update file using a *POST request*.

```
curl -v -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type:multipart/form-data" -X POST -F
update_info='{"content": {}, "envelope": {"version": 1}}' -F update_file=@/home/update/mGuard-image-1.8.0.mguard3.update.signed -k https://192.168.1.1:443/api/v1/update
```

Comment

- The *update_info* parameter does not contain any data about the JSON frame and is left empty.
- The *update_file* parameter contains the path to the update file.

2.3.4 Insert table rows in the JSON file

Individual table rows (e.g., firewall rules) are separated by commas.

Example (extract)

*{"***firewall***": {"forward": {"rules": [*

- *{*"dst_network": " 0.0.0.0/0*", "dst_port": "* ALL*", "id": 0, "protocol":* "ALL*",* "src_network": " *192.168.1.0/24, "verdict": "ACCEPT"},*
  {"dst_network": " 0.0.0.0/0*", "dst_port": "* ALL*", "id": 1, "protocol":* "ALL*",* "src_network": " *192.168.2.0/24, "verdict": "DROP"}*

2.4 Using RESTful client “curl“ (Linux)

NOTE: Third-party software
Phoenix Contact does not undertake any guarantee or liability for the use of third-party products. Any reference to or descriptions of third-party software does not constitute a recommendation, rather serves as an example of a program that could be used.

Table 2-1 Elements that can be used for a request to a RESTful server (e.g., client = *curl*)

Element	Options	Description
RESTful client *(curl)*	*-k* *--insecure*	Ensures that the HTTPS security certificate of the device is not checked.
	-H *--header*	Inserts an additional header in the HTTP request of *curl*. For a *POST request* to the RESTful server of the device, the header “*Content-Type: application/json*” must be specified to change the configuration and “*Content-Type: multipart/form-data*” must be specified for the upload of an update file.
	-X <cmd> *--request <cmd>*	Specifies a user-defined request method.
	-c <file name> *--cookie-jar*	Saves the session cookie transferred from the RESTful server on the configuration computer.
	-b <file name> *--cookie*	Uses the saved session cookie in the cookie header of a *GET* or *POST request* to the RESTful server.
	-d *--data*	Sends the specified data in a *POST request* to the RESTful server in the form in which a web browser would send a completed HTML form (see also option: *-F / --form*).
	-O *--remote-name*	The output is written to a local file and saved in the current working directory. The file name is extracted from the specified URL. If the file name is to be defined by the server, i.e., the device, you must also use option *-J / --remote-header-name*. Option *-o / --output <file name>* must not be used in this case.
	-J *--remote-header-name*	Option -J can only be used together with option -O. Option *-O / --remote-name* is instructed to use the file name specified by the server instead of extracting a file name from the URL. Option *-o / --output <file name>* must not be used in this case.
	-o <file> *--output <file>*	Writes the output to file *<file>* and not after *stdout*.
	-v *--verbose*	Results in *curl* returning additional information regarding an active *request* (e.g., warnings or information on the data sent).
	-F *--form*	Results in *curl* being able to send data with the help of “*Content-Type:multipart/form-data*” using a *POST request* (see also option: -d / --data).

Table 2-1 Elements that can be used for a request to a RESTful server (e.g., client = *curl*)

Element	Options	Description
Content type	*application/json*	To change files in JSON format using a *POST request*, "*Content-Type:application/json*" must be specified in the header of the request to the RESTful server.
	multipart/form-data	To initiate file upload using a *POST request*, "*Content-Type:multiform/form-data*" must be specified in the header of the request to the RESTful server. The *form-data key* "*update_info*" in the request contains an empty JSON frame; the *form-data key "update_file"* contains the actual update file.
HTTP request (method)	*GET*	The RESTful server is instructed to transfer the data (objects) clearly specified in the *HTTP request* to the RESTful client. Example: the device configuration is read.
	POST	Contents (objects) are transmitted from the RESTful client to the RESTful server in a data block. Example: the device configuration is created from scratch or changed.
URL	*https://username:password@<IP address>:<Port>/api/v1/<endpoint>* The address can be reached via the *Config API* of the device. Variables are configured in the *endpoints*. A username and password are used for authentication.	
Endpoint	Part of the URL for invoking the RESTful web service. The device variables are configured in the *keys* of a frame of the available endpoints (see Section 3).	
Argument	*content*	Contains the frame data (the structure is defined in *schemes*).
	envelope	Contains general information regarding the frame.
	version	Version of the *Config API* (also part of the endpoint, e.g., v1/configuration).
	identifier	Contains two hash values, which can be used to detect changes in the configuration.
	contentID	Describes a hash value about a generally formatted (and organized) input file in order to point out any changes in the monolithic configuration.
	functionalID	Describes a hash value about the effectively configured functionality of the device in order to point out any changes in the functionality (user permissions).
	error	Contains an error description (see "Error messages (RESTful server)".
	schemes	Contains the reference to the *schemes* for the current endpoint.
	status	Contains the status of the current *request* (based on the error index). In the event of an error, indicates the minimum error ID from the error list. If successful, *0* is displayed.

2.5 Using the “YARC!” RESTful client (Chrome)

The *YARC!* browser extension for *Google Chrome* can be used to perform simple *GET* and *POST requests* in the web browser (last tested: October 2021).

NOTE: Third-party software
Phoenix Contact does not undertake any guarantee or liability for the use of third-party products. Any reference to or descriptions of third-party software does not constitute a recommendation, rather serves as an example of a program that could be used.

2.5.1 Starting a secure session and user login

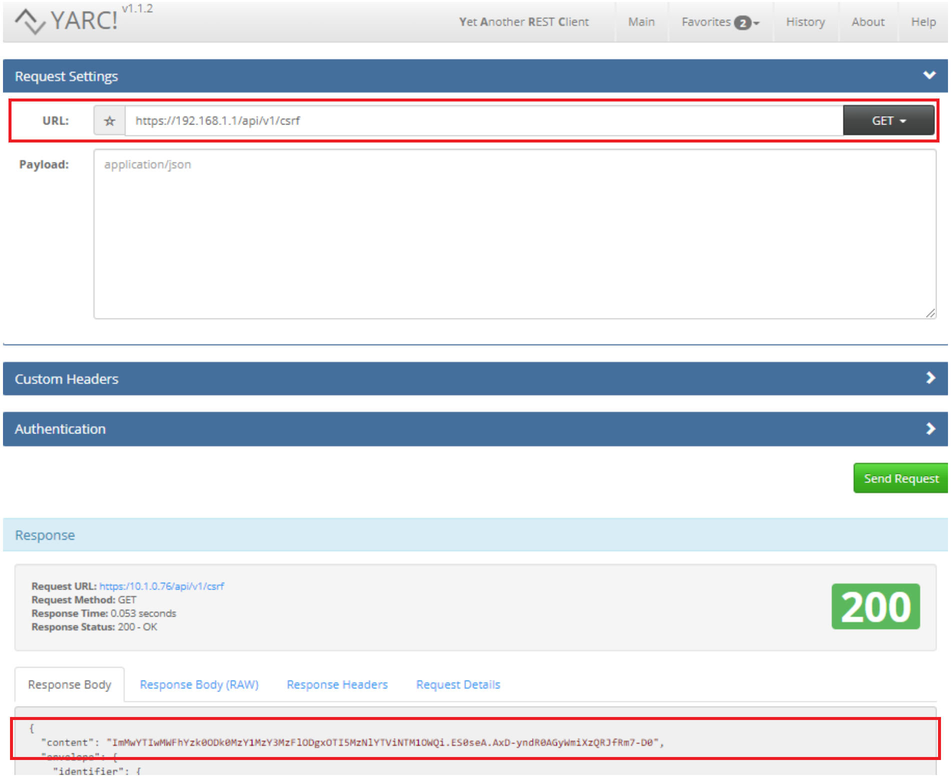
Before you can retrieve or change the configuration using a *GET* or *POST request*, you must log in the *admin* user during a secure session. The web browser automatically saves the session cookie used for this.

Proceed as follows:

1. **Create CSRF token**

- **Request**: GET
- **URL**: *https://192.168.1.1/api/v1/csrf*
- The token is specified in the response as *content*:

  Response*: "content":* "ImI1ZTY5NzhjNjhlOWY2ZDk4N2JjMDVkYmRkNTQ4Njgw-ZGViZDYwODgi.ESghMQ.vUlJDwWK20p8OJYbs5GVhzcvwM8"



URL: https://192.168.1.1/api/v1/csrf GET

Payload: application/json

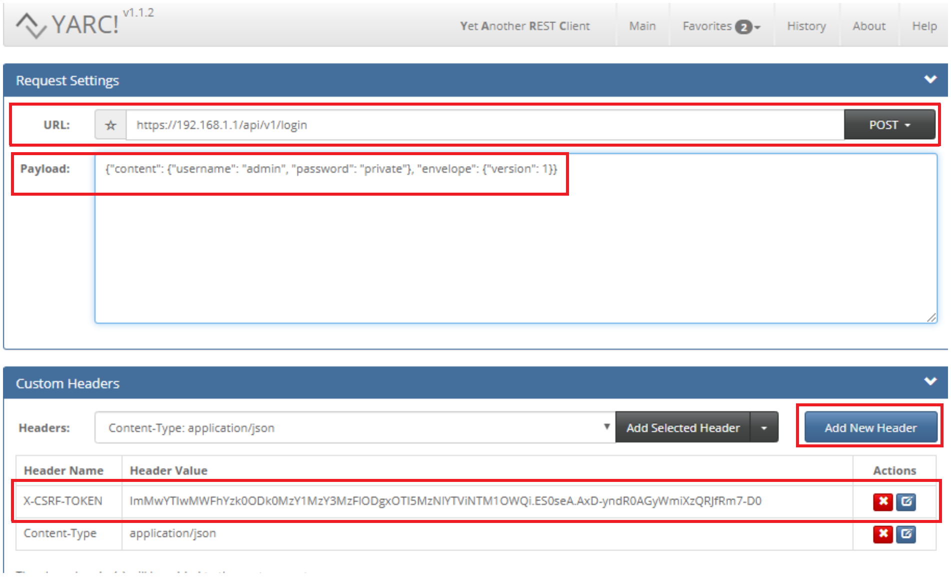
Request URL: https:/10.1.0.76/api/v1/csrf
Request Method: GET
Response Time: 0.053 seconds
Response Status: 200 - OK

```
{
  "content": "ImMwYTIwMWFhYzk0ODk0MzY1MzY3MzFlODgxOTI5MzNlYTViNTM1OWQi.ES0seA.AxD-yndR0AGyWmiXzQRJfRm7-D0",
  "envelope": {
    "identifier": {
```

Figure 2-3 *GET request* to *csrf* endpoint: create CSRF token

2. Login

- **Request**: POST
- **URL**: *https://192.168.1.1/api/v1/login*
- **Payload**: {"content": {"username": "admin", "password": "private"}, "envelope": {"version": 1}}
- **Custom Headers**:
  - Content-Type: application/json
  - X-CSRF-TOKEN: ImMwYTIwMWFhYzk0ODk0MzY22MZY3MzFlOSgxO-TISMzNIYTViNTM1OWQi.ES0seA.AxD-yndRoAGyWmiXzQRJfRm7-D0



URL: https://192.168.1.1/api/v1/login

Payload: {"content": {"username": "admin", "password": "private"}, "envelope": {"version": 1}}

Headers: Content-Type: application/json

Header Name	Header Value	Actions
X-CSRF-TOKEN	ImMwYTIwMWFhYzk0ODk0MzY1MzY3MzFlODgxOTI5MzNlYTViNTM1OWQi.ES0seA.AxD-yndR0AGyWmiXzQRJfRm7-D0	✖ ☑
Content-Type	application/json	✖ ☑

Figure 2-4 *POST request* to *login* endpoint: user login (with CSRF token)

⇒ A *session cookie* has been created following a successful POST request to the *login* endpoint.

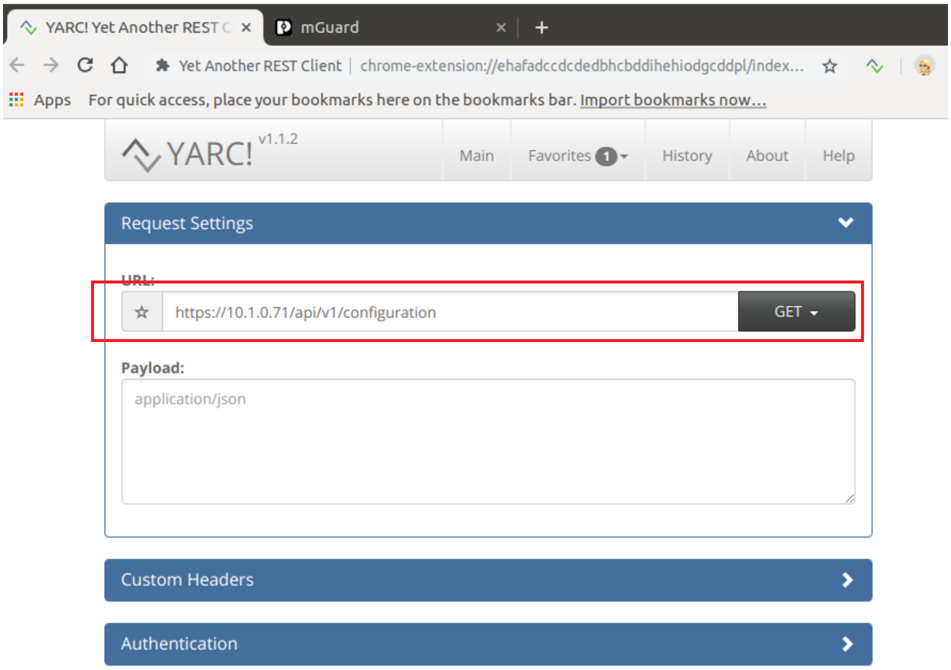
⇒ You can now send *GET* and *POST requests* during a secure session.

3. Send GET and POST requests (see Section 2.5.2)

- **Request**: GET
- **URL**: *https://192.168.1.1/api/v1/configuration*
- **Payload**: empty (GET) or content (POST)
- **Custom Headers** (*POST requests* only):
  - Content-Type: application/json
  - X-CSRF-TOKEN: ImMwYTIwMWFhYzk0ODk0MzY22MZY3MzFlOSgxO-TISMzNIYTViNTM1OWQi.ES0seA.AxD-yndRoAGyWmiXzQRJfRm7-D0

2.5.2 Example: Change configuration using POST request

- Log in by creating a *CSRF token* and a *session cookie*.
- Perform a *GET request* to the *configuration* endpoint (see Figure 2-5).

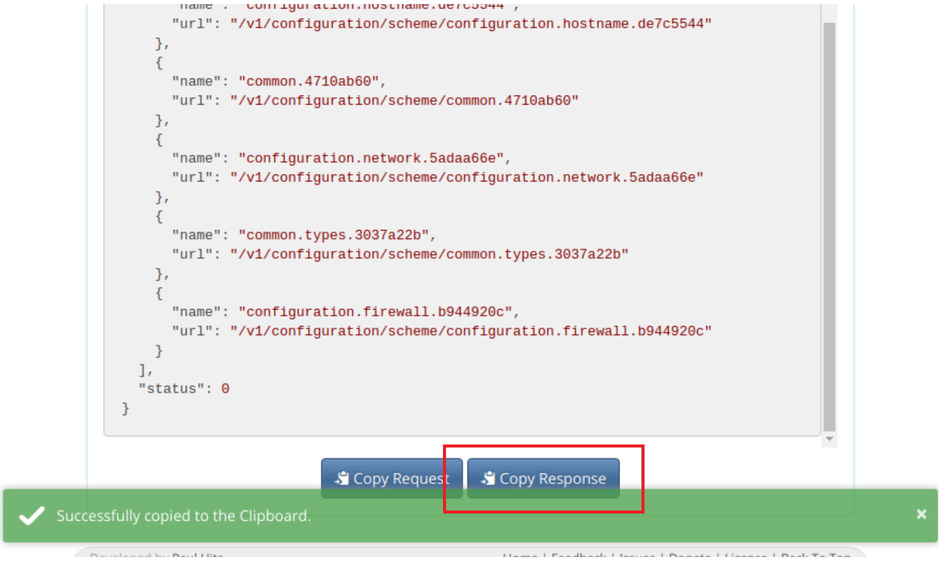

YARC! Yet Another REST C
mGuard
Yet Another REST Client
chrome-extension://ehafadccdcdedbhcbddihehiodgcddpl/index...
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
YARC! v1.1.2
Main
Favorites 1
History
About
Help
Request Settings
Custom Headers
Authentication

URL:
https://10.1.0.71/api/v1/configuration
GET

Payload:
application/json

Figure 2-5 *GET request* to the *configuration* endpoint

⇒ The response to the *GET request* is displayed (see Figure 2-6).



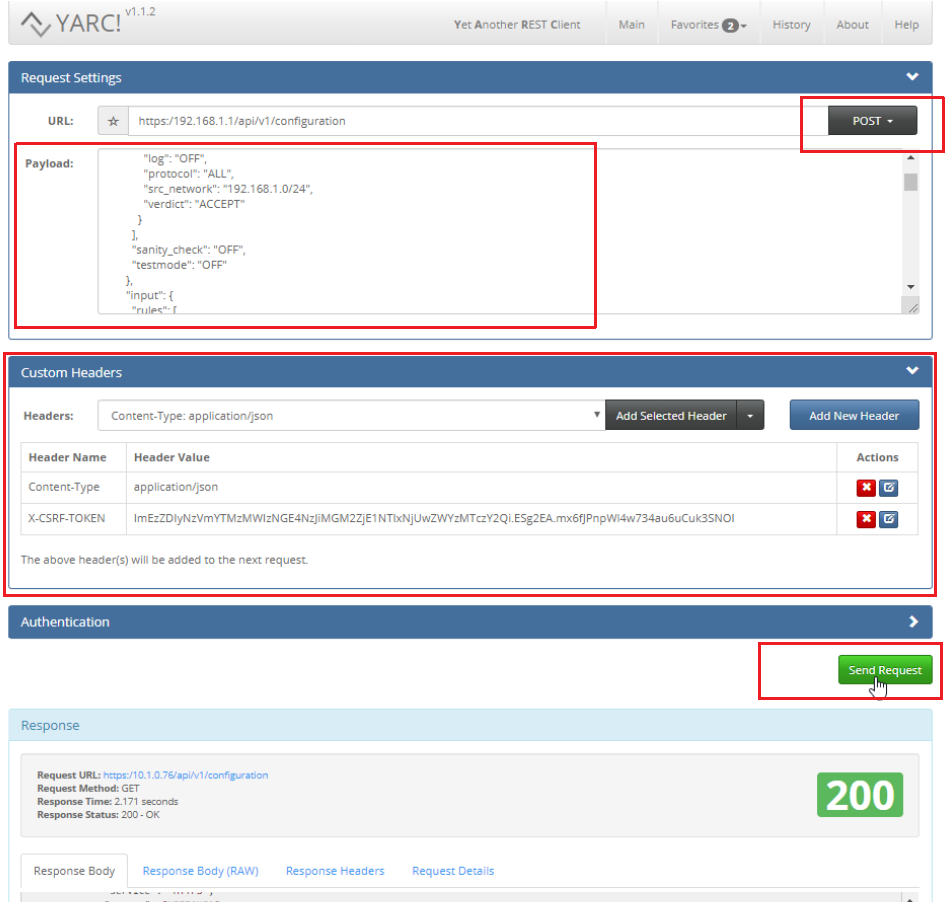
```
        "name": "configuration.hostname.de7c5544",
        "url": "/v1/configuration/scheme/configuration.hostname.de7c5544"
      },
      {
        "name": "common.4710ab60",
        "url": "/v1/configuration/scheme/common.4710ab60"
      },
      {
        "name": "configuration.network.5adaa66e",
        "url": "/v1/configuration/scheme/configuration.network.5adaa66e"
      },
      {
        "name": "common.types.3037a22b",
        "url": "/v1/configuration/scheme/common.types.3037a22b"
      },
      {
        "name": "configuration.firewall.b944920c",
        "url": "/v1/configuration/scheme/configuration.firewall.b944920c"
      }
    ],
    "status": 0
  }
```

Successfully copied to the Clipboard.

Figure 2-6 The response to the *GET request* is displayed and can be copied

- Copy the response to the *Payload* area (see Figure 2-6 and 2-7).

- In the response, change the variable values that you wish to reconfigure (see Figure 2-7).
- In the response, delete all keys that are not allowed in a POST request. These are, for example, the keys *error*, *shemes* and *status*. The content of the response must be terminated with the key *envelope* as follows:
  "envelope": {"version": 1}}
- Select *POST* from the drop-down menu (*Request Settings*) to send a *POST request*.
- Click on **Send Request**.



URL: https:/192.168.1.1/api/v1/configuration

Payload:

```
            "log": "OFF",
            "protocol": "ALL",
            "src_network": "192.168.1.0/24",
            "verdict": "ACCEPT"
          }
        ],
        "sanity_check": "OFF",
        "testmode": "OFF"
      },
      "input": {
        "rules": [
```

Headers: Content-Type: application/json

Header Name	Header Value	Actions
Content-Type	application/json	✖ ☑
X-CSRF-TOKEN	ImEzZDlyNzVmYTMzMWIzNGE4NzJiMGM2ZjE1NTIxNjUwZWYzMTczY2Qi.ESg2EA.mx6fJPnpWl4w734au6uCuk3SNOI	✖ ☑

The above header(s) will be added to the next request.

Request URL: https:/10.1.0.76/api/v1/configuration
Request Method: GET
Response Time: 2.171 seconds
Response Status: 200 - OK

Figure 2-7 Sending the adapted configuration as a *POST request* (*Payload* window)

⇒ The entire configuration, including the modified variable values, is sent to the RESTful server and the device configuration is changed accordingly.

- Send another *GET request* to check that the desired changes have been applied.

2.6 Common errors (troubleshooting)

1. **CSRF token**

   Before user login (via the "*login*" end point,) a *CSRF token* must be generated via the "*csrf*" end point. The *CSRF token* must be specified during log in and again for every *POST request* (see Section 3.3).

2. **Login cookie**

   When a *CSRF token* is generated (see above), a *login cookie* is generated that must be used to generate the session cookie when users log in (see below) (see Section 3.3).

3. **Session cookie**

   Before user login via the "*login*" end point, a *session cookie* must be generated. The previously generated *CSRF token* must also be entered in this step (see above).

   The *session cookie* must be specified after successful log in (= session start) and again for every GET and *POST request* (see Section 3.3).

4. **Quotation marks**

   In the event of an incorrect entry, check that quotation marks have been used correctly. Please note that some variable values must be specified without quotation marks (e.g., netmasks).

5. **Brackets**

   In the event of an incorrect entry, check that all opening and closing brackets are correct.

6. **URL escaping**

– Check the correct use of inverted commas at the start and end of the content block (*content*): ... *-d '{"content": {"firewall" ... "envelope": {"version": 1}}'*

– Check whether or not specific characters you have used must be re-coded using *URL escaping*.

7. **Illegal keys in POST requests**

   POST requests with non-allowed entries (*keys*) are rejected.

   For example, the identifier, *error*, *shemes*, and *status* keys returned by a GET request to the *"configuration"* endpoint must not be used in a POST request to the *"configuration"* endpoint.

   The contents of a POST request must be terminated with the key *envelope* as follows: *"envelope": {"version": 1}}' or "envelope": {"version": 1}}*

   Translated with www.DeepL.com/Translator (free version)

2.7 Error messages (RESTful server)

Tabelle 2-2 RESTful Configuration API – Error messages (RESTful server)

ID (status)	Error message
0	OK – No error: Request successful
1	Request error
2	Interface not found
3	Server Error
4	Necessary key is missing from the request
5	The Firewall Assistant is running. Only GET- and HEAD-requests are allowed
6	No valid user session
7	Too many sessions
8	CSRF Token invalid or missing
9	Unauthorized
10001	IO Error
10002	Unknown Schema
10003	Validation Error
10004	Callback Error
10005	Apply Error
10006	System Error
10007	IP change is in progress, new IP will be:
20001	No Data entry found
20002	Wrong or missing envelope version
20003	No envelope
20010	Unexpected data entry found
20011	Duplicate JSON keys found
30001	Validation Error
30002	Schema Error
30003	Error on applying the configuration
30004	Gateway with address and netmask do not match
30005	The networks of the net zones 1 and 2 are not allowed to overlap.
40001	Something went wrong in the updater script
40002	Content-Type needs to be multipart/form-data
40003	File is too small
40004	File could not be saved
40006	Updater script can't be reached
50001	Validation Error
50002	Schema Error
50003	Error on applying the passwords

Tabelle 2-2 RESTful Configuration API – Error messages (RESTful server)

ID (status)	Error message
50004	Error on updating eds node
50005	Error: only on device managed users are allowed to change their password
60001	Validation Error
60002	Schema Error
60003	Error on applying the datetime
60004	Error on syncing datetime to RTC
70001	Unknown module requested
80001	Snapshot Error
90001	Software License Error
100001	Can't start ping
100002	Invalid arguments
110001	Can't start tcpdump
110002	Can't stop tcpdump
110003	Can't delete tcpdump
110004	No data available
110005	Tcpdump is already running
110006	Invalid arguments passed
120001	Validation Error
120002	Login failed
120003	Unknown username or password
120004	„“
130001	Logout failed
140001	Internal Error
150001	Use Notification Error
160001	Validation Error
160002	Schema Error
160003	Error on applying the user changes
160004	Error on updating eds node
170001	Cannot start the Firewall Assistant
170002	Cannot stop the Firewall Assistant
180001	Validation Error
180002	Schema Error
180003	Error reading log information
190001	Error while generating certificate
190002	Error getting certificate
190003	Error reloading/restarting logger
200001	Error while storing configurations
210001	Can't start unblockUser action

Tabelle 2-2 RESTful Configuration API – Error messages (RESTful server)

ID (status)	Error message
210002	Invalid arguments
210003	User is manual blocked by admin. Automatically blocking state can not resolved. Please unblock user by change 'block_user' in users config
210004	User not found in Database
220001	Error while migrating the configuration
220002	Validation error after migration
220003	The configuration has no valid firmware version for migration
230001	Error while rebooting via configapi

3 Description of the end points

The individual firmware variables (*keys*) are configured in the end points of the mGuard RESTful Configuration API.

End points represent different areas of the firmware, for example where a firmware update can be started or the firewall configuration can be changed.

This section describes the RESTful variables and the corresponding menu items in the web-based management (WBM) (see Section 3.1).

3.1 Available end points

Table 3-1 Available end points of the RESTful server (mGuardNT 1.8.x)

End point	Method	What is displayed/configured	Description
v1/csrf	**GET**	A *login cookie* and a *CSRF token* to secure a session are created by the RESTful server and transferred to the RESTful client.	Section 3.3
v1/login	**POST**	A user is logged in with their access data (*username* and *password*). The session is started and a *session cookie* is created.	
v1/logout	**POST**	The logged in user is logged out. All information regarding the current session (*session data*) is deleted together with the *session cookie*.	
v1/configuration	**GET**	The configuration in the *Network*, *Firewall*, and *System* areas is displayed or changed.	Section 3.4
	POST		
v1/configuration/default	**GET**	The configuration of the default device setting in the *Network*, *Firewall*, and *System* areas is displayed or restored.	Section 3.5
	POST		
v1/users	**GET**	The properties of the existing users are displayed.	Section 3.6
	POST	Users are added, edited, or deleted.	
v1/password	**POST**	The registered user’s current password is changed.	Section 3.7
v1/update	**POST**	Uploading a firmware update file and subsequent execution of the firmware update is initiated.	Section 3.8
v1/datetime	**GET**	The current date and time of the device is displayed or changed.	Section 3.9
	POST		
v1/snapshot	**GET**	A snapshot of the current device configuration is created and downloaded.	Section 3.10
v1/logging	**GET**	All log entries on the device are retrieved and displayed.	Section 3.11
	POST	Only log entries of events relating to the firewall are retrieved and displayed.	
v1/status	**GET**	The status information regarding certain device functions is retrieved (e. g. current firmware version).	Section 3.12

Table 3-1 [...]Available end points of the RESTful server (mGuardNT 1.8.x)

End point	Method	What is displayed/configured	Description
v1/actions/fwassist/start	**POST**	Acquisition of connection data using the *Firewall Assistant* starts.	Section 3.13
v1/actions/fwassist/stop	**POST**	Acquisition of connection data using the *Firewall Assistant* stops. Acquired connections are automatically converted into firewall rules.	
v1/actions/ping	**POST**	An ICMP request is sent to the connected network clients.	Section 3.14
v1/actions/tcpdump/start	**POST**	The contents of the network packets are analyzed (*tcpdump*). The analysis can be restricted by specifying filter options.	Section 3.15
v1/actions/tcpdump/stop	**POST**	The analysis of network packets is stopped. The result of the analysis is automatically saved to a file (**.pcap*) and downloaded.	
v1/actions/pki/renew/logging	**GET** **POST**	The client certificate that is used for authenticating the device in a remote syslog server is created and/or downloaded.	Section 3.16
v1/actions/storeconfig/sdcard	**POST**	The configuration currently saved on the device is written to the SD card inserted.	Section 3.17
v1/actions/reboot	**POST**	The device reboots.	Section 3.18
v1/actions/unblockuser	**POST**	An automatically blocked user is unblocked.	Section 3.19
v1/actions/migration	**POST**	A configuration that was created with an older firmware version is migrated to a configuration that corresponds to the current firmware version.	Section 3.20
v1/usenotification	**GET**	System use notification is displayed.	Section 3.21
v1/softwarelicense	**GET**	The *Software License Terms* (SLT) for the product are created and downloaded.	Section 3.22
v1/licenses	**GET**	The third-party software components (modules) used on the device are displayed.	Section 3.23
v1/licenses/module/<module name>	**GET**	The license information for the third-party software components (modules) used on the device is displayed.	Section 3.24

3.2 Nomenclature

Table 3-2 Nomenclature used for the description of the end points

Format	Description
<ip>	IPv4 address (in quotation marks) *Example: "192.168.1.102"*
<nw_cidr>	IPv4 network in CIDR format (in quotation marks) *Example: "192.168.1.0/24"* **Note:** When specifying an IP address <ip>, the netmask **/32** may not be used. An IP address must be specified without netmask (see above).
<nm_num>	Subnet mask in numeric format *Example: 24*
<num>	Numeric value *Example: 443*
<string>	Alphanumeric value (in quotation marks) *Example: "mGuard-076"* The permitted special characters depend on the relevant configured variable.
<YYYY-MM-DD_hh:mm:ss>	Date and time (in quotation marks) – YYYY = Year I MM = Month I DD = Day – hh = Hour I mm = Minute I ss = Second *Example: "2018-06-24_18:05:09"*
<time_dhm>	Time specification (in quotation marks) Alphanumeric value not equal to zero that can be used to specify the time in days, hours, **or** minutes. d =Day, h = Hour, m = Minute *Example: "12h"*
<time_minute>	Time specification in minutes Numeric value not equal to zero that can be used to specify the time in minutes. *Example: 60*
<timezone>	The time zone is specified in accordance with the harmonized international time zones (see appendix: Section 5.1).
<start:end>	Some values can be specified as ranges. A range is entered by entering the start and end of the range separated by a colon (Start:End). *Example: "110:220"*

3.3 End points “csrf” / “login” / “logout”

For secure device/firmware configuration and administration, the device (RESTful server) must first generate a secure *session cookie* and transmit this to the RESTful client (e.g., web browser).

To prevent CSRF (*Cross-Site Request Forgery*) attacks, each *session* is additionally secured using a *CSRF token*.

Procedure

1. Request *CSRF token*.
2. User log in and start secure session with *CSRF token* (*session cookie*).
3. Execute GET and *POST requests* during the current session.

Request CSRF token

A *GET request* to the “*csrf*” end point creates a *CSRF token* and a *login cookie* is created with option -c (see Section 3.3.1).

The *CSRF token* must then be specified at user log in (“*login*” end point) and in all subsequent *POST requests* during the session.

User log in and create session cookie

A *POST request* to the “*login*” end point, which contains the *login cookie*, the *CSRF token*, and the username (*admin*) and user password (e.g., *private*), generates a *session cookie* and starts the session (see Section 3.3.2).

The *session cookie* must next be specified for all *POST* and *GET requests* during the session in order to ensure their integrity.

The *session cookie* is deleted when a *POST request* is sent to the “*logout*” end point. This also ends the session.

Execute GET and POST requests during a session (see Section 3.4)

GET request: Only the *session cookie* must be specified.

POST request: Both the *session cookie* and the *CSRF token* must be specified.

3.3.1 “csrf” end point

A *login cookie* and a *CSRF token* to secure a session are created by the RESTful server and transferred to the RESTful client.

The *login cookie* is saved on the configuration computer with option *-c <login_cookie>*.

The *CSRF token* is returned as “*content*” and is only valid in conjunction with the *session cookie* that is created in the next step via the “*login*” end point.

Example

```
curl -k -c login_cookie -X GET https://192.168.1.1:443/api/v1/csrf
```

Response:

```
{"content":"ImIzYzk3N2UyYjFiYThlZmY5Yzc1M2FhZTQxYmE1MmYxZDQwZjQ3ZWYi.ESVZMA.wKC_l2dMwHYPyJeV
R1rFgIi0Tww","envelope":{"identifier":{"contentID":"d2c01a66","functionalID":"d2c01a66"},"version":1},"error":[],"scheme
s":[],"status":0}
```

3.3.2 “login” end point

Users can be logged in with their access data (username and password) via this end point.

A secure *session cookie* is generated by the *login cookie* and *CSRF token* and is saved on the configuration computer with option *-c <session_cookie_name>*.

In all subsequent GET and *POST requests* during a session, *curl* must be invoked with option *-b <session_cookie_name>* in order to use the saved *session cookie*.

Example

```
curl -k -X POST https://192.168.1.1:443/api/v1/login -b login_cookie -c session_cookie -H "X-CSRF-Token:
ImIzYzk3N2UyYjFiYThlZmY5Yzc1M2FhZTQxYmE1MmYxZDQwZjQ3ZWYi.ESVZMA.wKC_l2dMwHYPyJeVR1rFgIi0T
ww" -H "Content-Type: application/json" -d '{"content": {"username": "admin", "password": "private"}, "envelope":
{"version": 1}}'
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
{"name":"login.login.c1a52347","url":"/v1/login/scheme/login.login.c1a52347"}],"status":0}
```

3.3.3 “configuration” end point (GET request)

Example (GET “configuration”)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/configuration
```

Response:

See Section 4.2, “GET Request (Endpoint: "configuration")”

3.3.4 “logout” end point

A logged in user can be logged out via this end point.

All information regarding the session (*session data*) is deleted together with the *session cookie*. For a new login, a new *CSRF token* and new *session cookie* must be generated.

Example

```
curl -k -X POST https://192.168.1.1:443/api/v1/logout -b session_cookie -H "X-CSRF-Token:
ImIzYzk3N2UyYjFiYThlZmY5Yzc1M2FhZTQxYmE1MmYxZDQwZjQ3ZWYi.ESVZMA.wKC_l2dMwHYPyJeVR1rFgIi0T
ww" -H "Content-Type: application/json" -d '{"content": {}, "envelope": {"version": 1}}'
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
],"status":0}
```

3.4 “configuration” end point

Via this end point, the configuration of the end point elements can be

1. displayed (*GET request*) or
2. changed (*POST request*).

ℹ Locally saved passwords are not transmitted with a GET request.

ℹ The following applies for POST requests:

1. The configuration may not have been created with a minor version that is higher than the one that is already installed on the device.
2. If a configuration created with an older firmware version is restored on the device the variable values which were not yet present in the older firmware version are kept.

The following elements are part of the end point

- Firewall (continuous data traffic) (Section 3.4.1)
- Input firewall (device access) (Section 3.4.2)
- Port forwarding (Section 3.4.3)
- Remote logging (Section 3.4.4)
- Network mode (Section 3.4.5)
- Network configuration (Section 3.4.6)
- NAT masquerading (Section 3.4.7)
- 1:1 NAT (Section 3.4.8)
- Default gateway (Section 3.4.9)
- Additional static routes (Section 3.4.10)
- Network services:
  - DHCP server (Section 3.4.11)
  - DNS server/DNS cache (Section 3.4.12)
  - NTP server/NTP client (Section 3.4.13)
  - SNMP server (Section 3.4.14)
  - Web (session timeout) (Section 3.4.15)
- System (Section 3.4.16)
  - Hostname of the device
  - Automatically save configuration
  - System use notification
- Time zone (Section 3.4.17)

Example: Display configuration (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/configuration
```

Response:

⇒ (Result/response: see Section 4.2)

Example (1.4.1): Change configuration (POST)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/configuration -d '{{"content": {"fileinfo": {"devtype": "0001010111020000", "firmware": "1.4.1"}, "firewall": {"forward": {"log_all_matches": "ON", "log_policy": "ON", "sanity_check": "ON", "stealth_allow_dhcp": "ON", "tables": [{"in_netzone": "NETZONE1", "out_netzone": "NETZONE2", "rules": []}, {"in_netzone": "NETZONE2", "out_netzone": "NETZONE1", "rules": [{"comment": "", "dst_network": "0.0.0.0/0", "dst_port": "ALL", "id": 0, "log": "OFF", "protocol": "ALL", "src_network": "0.0.0.0/0", "verdict": "ACCEPT"}]}], "testmode": "ON"}, "input": {"rules": [{"id": 0, "log": "OFF", "service": "HTTPS", "source": "NETZONE2", "verdict": "ACCEPT"}, {"id": 1, "log": "OFF", "service": "HTTPS", "source": "NETZONE1", "verdict": "ACCEPT"}]}, "port_forward": {"rules": [{"comment": "", "dst_ip": "0.0.0.0", "dst_port": 443, "inc_port": 5000, "protocol": "TCP", "src_interface": "NETZONE1"}, {"comment": "", "dst_ip": "0.0.0.0", "dst_port": 102, "inc_port": 5001, "protocol": "UDP", "src_interface": "NETZONE1"}]}}, "hostname": "NewName", "logging": {"remote": {"address": "syslog.my-mguard.com", "port": 513, "protocol": "UDP", "status": "ON"}}, "network": {"mode": "ROUTER", "nat": {"1_1_nat": [{"comment": "", "id": 0, "real_network": "192.168.1.100", "virt_network": "10.1.0.101"}, {"comment": "", "id": 1, "real_network": "192.168.1.200", "virt_network": "10.1.0.102"}], "masquerading": [{"from_ip": "0.0.0.0/0", "id": 0, "outgoing_on_if": "NETZONE1"}]}, "netzone1": {"mode": "DHCP"}, "netzone2": {"address": "192.168.1.1", "netmask": 24}, "routing": {"routes": [{"comment": "Production3", "gateway": "192.168.1.10", "network": "192.168.10.0/24"}]}, "stealth": {"management_address": "192.168.1.1", "management_gateway": "192.168.1.254", "management_netmask": 24}}, "service": {"dhcp_server": {"dns": "192.168.1.1", "gateway": "192.168.1.1", "lease_time": "12h", "netmask": 24, "range_high": "192.168.1.254", "range_low": "192.168.1.2", "status": "ON", "wins_server": ""}, "dnscache": {"allowed_requests": ["NETZONE2", "NETZONE1"], "dns_servers": "USER_DEFINED", "log": "ON", "user_defined": [{"comment": "", "ip": "212.2.220.212"}]}, "ntp": {"allow_client_requests": ["NETZONE2"], "server": [{"address": "0.pool.ntp.org", "comment": "", "port": 123}, {"address": "1.pool.ntp.org", "comment": "", "port": 123}, {"address": "2.pool.ntp.org", "comment": "", "port": 123}], "status": "ON"}, "snmp": {"allow_requests_from": ["NETZONE2"], "ro_community_string": "public", "status_v2c": "ON", "status_v3": "ON", "user": {"username": "snmp-v3-user"}}, "web": {"session_timeout": 450}}, "system": {"store_config_on_sdcard": "OFF", "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."}, "zoneinfo": "UTC"}, "envelope":  {"version": 1}}'
```

Response: (For a structured view with firmware 1.8.0, see Section 4.3)

3.4.1 Firewall (for continuous data traffic)

Setting options

1. “Logging”
2. “Consistency check”
3. “Forwarding DHCP packets”
4. “Connection tracking helper (FTP)”
5. “Firewall tables”
6. “Firewall rules”
7. “Firewall test mode”

Example

"**firewall**": {"**forward**": {"**log_all_matches**": "ON", "**log_policy**": "ON", "**sanity_check**": "ON", "**stealth_allow_dhcp**": "ON", "**ftp_allow_field**": "ON", "**tables**": [{"in_netzone": "NETZONE2", "out_netzone": "NETZONE1", "**rules**": [{"dst_network": "0.0.0.0/0", "dst_port": "ALL", "id": 0, "protocol": "ALL", "src_network": "192.168.1.0/24, "verdict": "ACCEPT", "log": "OFF“, “comment“: ““}, {"dst_network": "192.168.1.55", "dst_port": 443, "id": 1, "protocol": "TCP", "src_network": "0.0.0.0", "src_netmask": 0, "verdict": "ACCEPT", "log": "OFF“, “comment“: “This rule belongs to the machine B“}]}, {"in_netzone": "NETZONE1", "out_netzone": "NETZONE2", "**rules**": [] }], "**testmode**": "ON"}}

Logging

Table 3-3 End point **configuration**, key(s): **firewall >> forward**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward)	log_all_matches	"ON" "OFF"	**Log all configured rules** When this function is activated, a corresponding log entry is created for each data connection to which any firewall rule applies. This also applies to rules where logging is deactivated using the “*“Log”*” function. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-forward-* *Example: "OFF"*
	log_policy	"ON" "OFF"	**Log unknown connection attempts** When this function is activated, a corresponding log entry is created for each data connection to which no configured firewall rules apply. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-forward-policy-* *Example: "OFF"*

Consistency check

Table 3-4 End point **configuration**, key(s): **firewall >> forward >> (sanity_check)**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward)	sanity_check	"ON" "OFF"	**TCP/UDP/ICMP consistency check** The consistency check increases the protection of connected network clients against *Denial of Service* (DoS) attacks. When this function is activated, data packets that are routed through the device and forwarded to connected network clients are checked for malicious elements: **ICMP packets** Only known ICMP code is used. **UDP packets** Destination port in the UDP packet is not equal to zero. **TCP packets** Source and destination port in the TCP packet are not equal to zero. **IPv4 packets** Protocol is not set to zero. Data packets that do not meet the specified requirements are dropped by the firewall. *Example:* "*ON*"

Forwarding DHCP packets

Table 3-5 End point **configuration**, key(s): **firewall >> forward >> (stealth_allow_dhcp)**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward)	stealth_allow_dhcp	"ON" "OFF"	**Allow forwarding of DHCP packets** In stealth mode, the following applies: When the function is activated, clients in net zone 2 can obtain their IP configuration **automatically and independently of the settings in the firewall tables** from a DHCP server in net zone 1. Firewall rules configured in the firewall table that would block this DHCP data traffic are not considered. It is not necessary to manually configure firewall rules to allow DHCP data traffic. *Example:* "*ON*"

Connection tracking helper (FTP)

Table 3-6 End point **configuration**, key(s): **firewall >> forward >> (ftp_allow_field)**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward)	ftp_allow_field	"ON" "OFF"	**Connection tracking helper (FTP)** Activating this function helps to enable desired data connections via the FTP protocol that are blocked by the firewall. If a connection is established via the FTP protocol, data can be transferred in two ways: 1. With "active FTP", the called FTP server establishes an additional counter-connection to the caller (FTP client) in order to transfer the data via this connection. 2. With "passive FTP", the caller (FTP client) establishes an additional connection to the server in order to transfer the data. To ensure that the additional connection is not blocked by the firewall, the connection tracking helper for FTP must be activated in both cases. The activated function is also applied to data packets that are forwarded using port forwarding. ! **NOTE: No connection in stealth mode with "active FTP".** For connections in stealth mode with "active FTP", no connection is established even if the connection tracking helper is activated. In this case, either use "passive FTP" or create an additional firewall rule that allows a data connection from the server to the client according to your requirements (e.g. Allow: *Net zone 1* → *Net zone 2*, Protocol: *TCP*, From IP: *192.168.1.100*, To IP: *192.168.1.200*). *Example:* "*ON*"

Firewall tables

Table 3-7 End point **configuration**, key(s): **firewall >> forward >> tables**

Key(s)	**Variable (key)**	**Value (format)**	**Designation (WBM)/description**
firewall (forward, tables)	in_netzone	"NETZONE1" "NETZONE2"	**Net zone X → Net zone Y** The firewall rules are configured in two different tables depending on the direction of the initial data traffic: – *Net zone 1 → Net zone 2* *(in_netzone → out_netzone)* – *Net zone 2 → Net zone 1* (*in_netzone → out_netzone*) The rules in a firewall table are only applied to the data traffic that is *routed* through the device in the specified direction from one net zone to the other. The data traffic direction that the following configured rules are to be applied to is defined for each table via the key **firewall >> forward >> tables**. **Note**: 1. Both tables must be configured (see “Example”). 2. The values for the variables *in_netzone* and *out_netzone* must be different within one table. *Example:* "*NETZONE1*"
	out_netzone	"NETZONE1" "NETZONE2"	

Firewall rules

Table 3-8 End point **configuration**, key(s): **firewall >> forward >> tables >> rules**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward, tables, rules)	dst_network	<nw_cidr> <ip>	**To IP/network** Destination (network or IP address) to which the data packets have to be sent so that the rule applies here. If "0" is specified as the subnet mask, the rule applies to all sources (all IP addresses and networks) here. **Note:** When specifying an IP address <ip>, the netmask **/32** may not be used. An IP address must be specified without netmask. *Example: "10.1.0.0/24"* *Example: "10.1.0.50"*
	dst_port	<num> "ALL" <start:end>	**To port** Destination port or port range where the data packets have to be sent so that the rule applies here. "ALL" = all ports *Example: 443* *Example (port range): 110:120*
	id	<num>	**ID** ID number of the rule The ID determines the order in which the rules are queried, starting with the lowest ID. *Example: 33*
	log	"ON" "OFF"	**Log** When this function is activated, a corresponding log entry is created for each data connection this rule applies to. For rules in which the function is deactivated, a log entry is not created unless the *"Log all configured rules"* function is activated. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-forward-* *Example: "OFF"*
	protocol	"TCP" "UDP" "ICMP" "GRE" "ALL"	**Protocol** Network protocol that must be used to transmit the data packets so that the rule applies here. "ALL" = all protocols *Example: "TCP"*

Table 3-8 End point **configuration**, key(s): **firewall >> forward >> tables >> rules**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
	src_network	<nw_cidr> <ip>	**From IP/network** Source (network or IP address) from which the data packets have to be sent so that the rule applies here. **Note:** When specifying an IP address <ip>, the netmask **/32** may not be used. An IP address must be specified without netmask. If "0" is specified as the subnet mask, the rule applies to all sources (all IP addresses and networks) here. *Example: "192.168.1.0/24"* *Example: "10.168.1.50"*
	verdict	"ACCEPT" "DROP" "REJECT"	**Action** The action that will be performed if all parameters configured in the access rule apply to a packet. **Accept:** The data packets may pass through. **Reject:** The data packets are rejected. The sender is informed. **Drop:** The data packets are dropped. The sender is not informed. **Note (Stealth mode):** In *Stealth mode*, selection of the *Reject* action leads to the same behavior as that of the action *Drop.* Because the device does not have its own IP address in *Stealth mode*, data packets are dropped in both cases and the sender is not informed. In these cases, the log entries will be listed as the action “*Drop*” and not “*Reject*”. *Example: "ACCEPT"*
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

Firewall test mode

Table 3-9 End point **configuration**, key(s): **firewall >> forward >> testmode**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (forward)	testmode	"ON" "OFF"	**Firewall test mode** Data traffic unintentionally rejected by the firewall can be easily identified and permitted through the automated creation of corresponding firewall rules. **NOTE: The firewall is deactivated.** In *Firewall test mode*, data packets that are not acquired by any of the already configured firewall rules will not be discarded, as is normally the case, but instead will be forwarded. **Prerequisite** For the *firewall test mode* to be able to generate entries, the existing firewall table must not contain an overriding rule that rejects all data traffic. **Method of operation** When this function is activated, the data traffic *routed* through the device is analyzed by the firewall. If an already configured firewall rule applies to a data packet, the rule is applied to the data packet **as normal** (*Accept, Reject*, or *Drop*). If none of the configured rules apply to a data packet, the packet is **not discarded, as is usually the case**, but forwarded. At the same time, the user is informed via an event: 1. The “PF2” LED on the device lights up red. 2. The “O1” switching output on the “XG2” COMBICON connector of the device switches to *high level.* (If a signal light is connected, it would light up in this case““status” end point”.) 3. An entry is generated in the ““status” end point” that can be analyzed by the user. If the data traffic that has triggered a *test mode alarm* is to be allowed in the future, users can automatically create an appropriate firewall rule via **web-based management** from the corresponding entry in the *test mode alarms* table (WBM). (See the “UM DE MGUARD NT” user manual, available at phoenixcontact.net/product/1153079)

Table 3-9 End point **configuration**, key(s): **firewall >> forward >> testmode**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
			Creating firewall rules from test mode alarms In **web-based management** entries in the *Test-mode alarms* table can be selected and automatically added as new firewall rules at the end of the existing firewall tables. The newly added rules would then allow the relevant data traffic in the future (*Action* = *Accept*).
			Deactivating Firewall test mode If the *firewall test mode* is deactivated, all corresponding entries in the ““status” end point”or in the *test mode alarms* table will be deleted and signaling via the “PF2” LED and the “O1” switching output will stop.

3.4.2 Input firewall (device access)

Setting options

1. “Logging”
2. “Input firewall rules”

Example

"**firewall**": {"**input**": {"log_all_matches": "ON", "log_policy": "ON", "**rules**": [{"id": 0, "service": "HTTPS", "source": "NETZONE2", "verdict": "ACCEPT"}, {"id": 1, "service": "HTTPS", "source": "NETZONE1", "verdict": "ACCEPT", "log": "ON“}]}}

Logging

Table 3-10 End point **configuration**, key(s): **firewall** >> **input** >> **(log_all_matches / log_policy)**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (input)	log_all_matches	"ON" "OFF"	**Log all configured rules** When this function is activated, a log entry is created for each data connection to which any input firewall rule applies. This also applies to rules where logging is deactivated using the “*“Log”*” function. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-input-* *Example: "OFF"*
	log_policy	"ON" "OFF"	**Log unknown connection attempts** When this function is activated, a corresponding log entry is created for each data connection to which no configured input firewall rules apply. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-input-policy-* *Example: "OFF"*

Input firewall rules

Table 3-11 End point **configuration**, key(s): **firewall >> input >> rules**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (input, rules)	id	<num>	**ID** ID number of the rule The ID determines the order in which the rules are queried, starting with the lowest ID. *Example: 33*
	log	"ON" "OFF"	**Log** When this function is activated, a corresponding log entry is created for each data connection this rule applies to. For rules in which the function is deactivated, a log entry is not created unless the "*"Log all configured rules"*" function is activated. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). Log prefix: *fw-input-* *Example: "OFF"*
	service	"HTTPS"	**Service** The network service running on the device, for which an access rule should be created. The web server of the device (web-based management and *Config API*) can be accessed via HTTPS. *Example: HTTPS*
	source	"NETZONE1" "NETZONE2"	**HTTPS access from net zone 1/2** Access to the device web server (HTTPS) is permitted from the specified net zone (TCP port 443). *Example: "NETZONE2"*
	verdict	"ACCEPT"	**Action** The action that will be performed if all parameters configured in the access rule apply to a packet. *Example: "ACCEPT"*

3.4.3 Port forwarding

Port forwarding rules are applied before firewall rules

The rules for port forwarding are applied and executed before the configured firewall rules for continuous/routed data traffic are applied (see Section 3.4.1).

This means that a firewall rule that blocks all incoming data traffic is not applied if a port forwarding rule applies.

Example

"**firewall**": "**port_forward**": {"**rules**": [{"dst_ip": "192.168.1.200", "dst_port": 5000, "inc_port": 115, "protocol": "ALL", "src_interface": "NETZONE1", “comment“: “This rule refers to production B“}]}

Table 3-12 End point **configuration**, key(s): **firewall** >> **port_forward** >> **rules**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
firewall (port_forward, rules)	inc_port	<num>	**Incoming port** Device network port to which the data packets must be sent so that the rule is applied. Data packets sent to this port are usually forwarded to the defined destination IP address (*dst_ip*) and the defined destination port (*dst_port*): – The destination IP address in the header of the data packet is translated into the destination IP address defined in the rule (*dst_ip*). – The destination port in the header of the data packet is translated into the destination port defined in the rule (*dst_port*). **Note:** The ports available are 1–65535, except the following ports, because they are used by device services: DNS (53), HTTPS (443), NTP (123), SNMP (161), DHCP (67, 68) *Example: 115*
	protocol	"TCP" "UDP"	**Protocol** Network protocol that must be used to transmit the data packets so that the rule is applied. *Example: "TCP"*
	src_interface	“NETZONE1“ “NETZONE2“	**Off** Net zone from which the data packets must be sent to the device so that the rule is applied. *Example: "NETZONE1"*

Table 3-12 End point **configuration**, key(s): **firewall >> port_forward >> rules**

Key(s)	**Variable (key)**	**Value (format)**	**Designation (WBM)/description**
	dst_ip	<ip>	**To IP** IP address of the destination client to which the incoming data packets are forwarded when the rule is applied. The original destination address in the header of the data packet is translated into this IP address. *Example: "192.168.1.200"*
	dst_port	<num>	**To port** Network port to which the incoming data packets are forwarded if the rule is applied. The original destination port in the data packet header (see *“inc_port”*) is translated into this port. *Example: 5000*
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

3.4.4 Remote logging

Example

"**logging**": {"remote": {"address": "192.168.1.254", "port": 514, "protocol": "TLS", "ca": "-----BEGIN CERTIFICATE-----\nMIID4jdQibqcmC/Q9xueMwDQYJKoZIhvcNAQEL\nBQAwb-DELMAkG [...] g92ibqcaZmC/Q9Oys=\n-----END CERTIFICATE-----", "status": "OFF"}}

Table 3-13 End point **configuration**, key(s): **logging**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
logging (remote)	address	<ip> <string>	**IP/hostname (log server)** IP address or hostname of the remote server (*syslog* server) to which the log entries are to be sent. *Example: "192.168.1.254"*
	port	<num>	**Port (log server)** Network port via which the remote server accepts data packets (standard port: *514/UDP*). *Example: 514*
	status	"ON" "OFF"	**Remote logging** When this function is activated, all device log entries will be transmitted to a remote server using the *syslog protocol* (see RFC 5424) (see Above). You can choose whether the information is transmitted using the unencrypted UDP protocol or encrypted using the TCP protocol. *Example: "OFF"*
	protocol	"UDP" "TLS"	**Transmission protocol** Network protocol that is used to establish a connection to the remote server (*syslog* server). **Note:** For reasons of security, an encrypted TLS connection should always be used between the device (mGuard) and the *syslog* server. **UDP** The data are transmitted unencrypted using the UDP protocol. Mutual authentication between the device and the remote server does not take place. **TLS over TCP** The data are transmitted encrypted via a TCP connection. Mutual authentication between the device and the remote server takes place via X.509 certificates. The necessary client certificate can be displayed via the following end point or a new certificate can be created: ““actions/pki/renew/logging” end point”

Table 3-13 End point **configuration**, key(s): **logging**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
			Prerequisite: Conditions needed to ensure the integrity and the authenticity of the encrypted TCP connection: 1. A server certificate (CA certificate) for the remote server must be installed on the device (see Below) 2. A client certificate must be generated on the device, downloaded, and installed on the remote server (see Section 3.16) *Example: "TLS"*
	ca	<string>	**Upload server CA certificate to the device** The CA certificate with which the device authenticates the remote server (*syslog* server) is uploaded to the device. The CA certificate is provided by the remote server operator and must be uploaded to the device (X.509 certificate with *public* key). An encrypted TCP connection to the remote server can only be established successfully if it in turn has a certificate issued by the CA certificate (with the *secret* key) or a valid certificate chain with the CA certificate as the highest instance. **Format:** The maximum file size allowed is 1 MB. *Example:* *"-----BEGIN CERTIFICATE-----\nMIID4jCCAsqgAwIBAgIUfFtWt2Ytv88GdQibqcmC/Q9xueMwDQYJKoZIhvcNA* [...] *EmQxzWgTz8ljR4VgmTXFOC2yqXOys=\n-----END CERTIFICATE-----"*

3.4.5 Network (mode)

Example **"network":** {"mode": "STEALTH"}

Table 3-14 End point **configuration**, key(s): **network >> mode >> stealth**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network	mode	ROUTER STEALTH	**Mode** The device can be operated in two network modes. **ROUTER** If the device is in Router mode, it acts as a gateway between different subnets. The data traffic is *routed* between the two network interfaces (net zones) of the device. Clients in the subnet of one net zone can communicate and exchange data with clients in the subnet of the other net zone. The security and firewall functions of the device are applied to incoming and *routed* data traffic. **STEALTH** Stealth mode is used to protect one or more local clients in an existing subnet (e.g., machine controls in a production network) against unwanted network access without having to change their IP settings. To do this, the device is added between the clients and the surrounding subnet via its two network interfaces (net zones) so that all the data traffic to and from the clients is routed through the device. The network configuration of the connected clients does not have to be changed. The server services DHCP, NTP, and DNS server are deactivated on the device. The security and firewall functions of the device are applied to incoming and routed data traffic (e.g., DHCP *requests*). The device is configured via the *stealth management IP address*, which can be accessed via the WBM and the *Config API* of the device.

Mode: STEALTH

Example **"network": {"mode":** "STEALTH", "stealth": {"management_address": "192.168.1.1", "management_netmask": 24, "management_gateway": "192.168.1.254"}}

Table 3-15 End point **configuration**, key(s): **network >> mode >> (STEALTH) >> stealth**

Key(s)	**Variable (key)**	**Value (format)**	**Designation (WBM)/description**
network (mode: STEALTH) (stealth)	management_ address	<ip>	**Management IP address** IP address via which the device is reachable in Stealth mode and can be managed. The management IP address is available on all network interfaces (net zones). The device is configured via the WBM or the *Config API*. **Note:** Changing the IP address that you are currently using to access the device will cause the device to no longer be available at this address after the configuration is saved. Log back in via the changed IP address. *Example: "192.168.1.1"*
	management_ netmask	<nm_num>	**Netmask** Subnet mask that defines the subnet where the device can be reached in Stealth mode via the management IP address. *Example: 16*
	management_ gateway	<ip>	**Default gateway** IP address of the default gateway to which the device sends connection requests to reach unknown subnets or the Internet. In Stealth mode, the device can use it to send requests as a client, for example, to an NTP or DNS server. When a management IP address is assigned, the default gateway of the network in which the device is located must be specified. The default gateway can be reached via net zone 1 (XF1) and net zone 2 (XF2–XF5). *Example: "192.168.1.254"*

Mode: ROUTER The individual functions in Router mode are described in separate sections.

Table 3-16 End point **configuration**, key(s): **network >> mode >> (ROUTER)**

Key(s)	Variable (key)	Designation (WBM)/description
network (mode: ROUTER)	netzone1 netzone2	See: – Section 3.4.6, “Network (net zone 1/2)”
	nat	See: – Section 3.4.7, “Network (NAT, IP masquerading)” – Section 3.4.8, “Network (NAT, 1:1 NAT)”
	routing	See: – Section 3.4.9, “Network (routing, gateway)” – Section 3.4.10, “Network (routing, additional routes)”

3.4.6 Network (net zone 1/2)

Example

"network": "netzone1": {"mode": "DHCP"}, "netzone2": {"address": "192.168.1.1", "netmask": 24}

ℹ The DHCP- or static-configured networks of the two net zones must not overlap.

Net zone 1

Table 3-17 End point **configuration**, key(s): **network >> netzone1**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (netzone1)	mode	"DHCP" "STATIC"	**Router mode** Mode that is used to determine how a network configuration is assigned to the net zone. **DHCP** The net zone is automatically assigned a network configuration (IP address, subnet mask, and, as an option, a default gateway and DNS server) by a DHCP server if a DHCP server is available in the network. **Static** Users have to manually assign a static network configuration to the net zone (IP address, subnet mask, and, as an option, a default gateway). *Example:* "*STATIC*"
	address	<ip>	**IP address** IP address of network interface XF1 (net zone 1). **Note:** Changing the IP address that you are currently using to access the device will cause the device to no longer be available at this address after the configuration is saved. Log back in via the changed IP address. *Example:* "*10.1.0.100*"
	netmask	<nm_num>	**Netmask** Subnet mask that defines the subnet where the device is located. *Example: 16*

Net zone 2

Table 3-18 End point **configuration**, key(s): **network >> netzone2**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (netzone2)	address	<ip>	**IP address** IP address of network interface XF2–XF5 (net zone 2). **Note:** Changing the IP address that you are currently using to access the device will cause the device to no longer be available at this address after the configuration is saved. Log back in via the changed IP address. *Example:* "*192.168.1.1*"
	netmask	<nm_num>	**Netmask** Subnet mask that defines the subnet where the device is located. *Example: 24*

3.4.7 Network (NAT, IP masquerading)

Deviating settings in Config API and WBM are possible

IP masquerading can be activated or deactivated for each net zone via web-based management.

In the *Config API* it is also possible to specify that only the data traffic from defined networks is masqueraded.

The device uses this type of configuration, but it is not displayed in the web-based management.

Example

"**network**": {"**nat**": {"**masquerading**": [{"from_ip": "0.0.0.0/0", "id": 0, "outgoing_on_if": "NETZONE1"}, {"from_ip": "0.0.0.0/0", "id": 1, "outgoing_on_if": "NETZONE2"}, {"from_ip": "10.1.1.0/24", "id": 2, "outgoing_on_if": "NETZONE2"}]}}

Table 3-19 End point **configuration**, key(s): **network >> nat >> masquerading**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (nat, masquerading)	id	<num>	ID number of the rule The ID determines the order in which the rules are applied, starting with the lowest ID. *Example: 0*
	from_ip	<nw_cidr>	The NAT masquerading rule is applied to data packets that are sent from the specified network and *routed* through the device. If "0" (e.g., 0.0.0.0/0) is specified as the subnet mask, the NAT rule applies to all IP addresses and networks. **Note:** If the function is activated in **web-based management**, the variables are assigned the value 0.0.0.0/0. *Example:* "*10.1.1.0/24*"
	outgoing_on_if	"NETZONE1" "NETZONE2"	**Masquerade to net zone 1/2** The NAT masquerading rule is applied to data packets (requests) that leave the device via the selected network interface (net zone). In the data packet, the sender’s IP address is translated into the IP address of the selected network interface (net zone). *Example: "NETZONE1"*

3.4.8 Network (NAT, 1:1 NAT)

Example

"**network**": {"**nat**": {"**1_1_nat**": [{"id": 0, "real_network": "192.168.1.0/24", "virt_network": "10.1.0.0/24", “comment“: “This rule refers to to production B“}]}}

Table 3-20 End point **configuration**, key(s): **network >> nat >> 1_1_nat**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (nat, 1_1_nat)	id	<num>	**ID** ID number of the rule The ID determines the order in which the rules are applied, starting with the lowest ID.
	real_network	<nw_cidr> <ip>	**Real IP/network** Data traffic sent from or to network clients of the real network are subject to the 1:1 NAT rule. **1:1 NAT** With 1:1 NAT, the network part (red) of the IP addresses of clients in the real network are translated to the network part of another (translated) network (see example). The host part (green) of the IP addresses assigned to the clients remain unchanged. **Example** **1:1 NAT rule:** 192.168.1.0/**24** <-> 10.1.0.0/**24** ⇒ **Translation:** 192.168.1.100 <-> 10.1.0.100 ⇒ **Translation:** 192.168.1.200 <-> 10.1.0.200 The network part and host part of an IP address are defined by the subnet mask (e.g., 192.168.70.80/**16** or 10.1.1.30/**24**). **Real IP** If the netmask is 32, individual IP addresses and not networks are translated by the 1:1 NAT rule: **Note:** When specifying an IP address <ip>, the netmask **/32** may not be used. An IP address must be specified without netmask. **1:1 NAT rule:** 192.168.1.40 <-> 10.1.5.40 ⇒ **Translation:** 192.168.1.40 <-> 10.1.5.40 **In practice** Clients in both networks can communicate with each other in both directions. At the same time, the real (mostly private) network is not visible in the other (mostly public) network:

Table 3-20 End point **configuration**, key(s): **network >> nat >> 1_1_nat**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
			– The respective translated client IP addresses in the real network appear as the sender address to the network participants in the other network. – To reach clients in the real network from the other network, their translated IP addresses must be used. – ARP requests to the translated client addresses in the real network are automatically responded to by the device as the representative. **Prerequisite** – Both the real and the translated networks must use the same subnet mask. – The translated IP client addresses in the real network must not yet be assigned in the other (translated) network. – Firewall rules are generally also applied to translated IP addresses. *Example:* "*192.168.1.0/24*" *Example:* "*192.168.1.50*"
	virt_network	<nw_cidr> <ip>	**Translated IP/network** The network to which the real IP addresses of the clients in the real network are to be translated (see “real_network”). **Prerequisite** – Both the real and the translated networks must use the same subnet mask. – The translated IP client addresses in the real network must not yet be assigned in the other (translated) network. **Translated IP** If the netmask is 32, individual IP addresses and not networks are translated by the 1:1 NAT rule. **Note:** When making configuration changes via the Config API, the netmask **/32** may not be used. An IP address must be specified without netmask instead. **Input format:** IPv4 address, IPv4 network (CIDR notation) *Example:* "*192.168.2.0/24*" *Example:* "*10.1.0.50*"

Table 3-20 End point **configuration**, key(s): **network >> nat >> 1_1_nat**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

3.4.9 Network (routing, gateway)

Example

"**network**": "**routing**": {"**gateway**": "192.168.1.144", "routes": [{"network": "10.2.2.0/24", "gateway": "192.168.1.200", “comment“: “This route leads to cell B“}]}

Table 3-21 End point **configuration**, key(s): **network >> routing >> gateway**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (routing, gateway)	gateway	<ip>	**Default gateway** IP address of the default gateway to which the device sends connection requests to reach unknown subnets or the Internet. A device in the subnet of net zone 1 (XF1) or in the subnet of net zone 2 (XF2–XF5) can be specified as the default gateway. **Note:** This only has to be specified for the “Static” router mode (see Section 3.4.6). *Example:* "*10.1.0.254*"

3.4.10 Network (routing, additional routes)

Example

"**network**": "**routing**": {"gateway": "192.168.1.144", "**routes**": [{"network": "192.168.3.0/24", "gateway": "192.168.1.200", “comment“: “This route leads to cell B“}]}

Table 3-22 End point **configuration**, key(s): **network >> routing >> routes**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
network (mode: ROUTER) (routing, routes)	network	<nw_cidr> <ip>	**IP/network** Destination (network or IP address) that should be reached via an additional route. **Note:** When making configuration changes via the Config API, the netmask **/32** may not be used. An IP address must be specified without netmask instead. *Example:* "*192.168.3.0/24*" *Example:* "*192.168.4.100*"
	gateway	<ip>	**Gateway** IP address of the gateway via which the destination can be reached using the additional route. *Example: "192.168.1.200"*
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

3.4.11 Service (DHCP server)

Example

"**service**": {"**dhcp_server**": {"dns": "192.168.1.1", "gateway": "192.168.1.1", "lease_time": "12h", "range_high": "192.168.1.254", "range_low": "192.168.1.2", "status": "ON", "wins_server": "192.168.1.252"}}

Table 3-23 End point **configuration**, key(s): **service >> dhcp_server**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (dhcp_server)	dns	<ip>	**DNS server** IP address of a DNS server that the DHCP server assigns to requesting clients. A DNS (*Domain Name System*) server allows clients to resolve hostnames into IP addresses. If the DNS server of the device is to be used, the IP address of the net zone on which this service is active must be specified (default setting: net zone 2 = 192.168.1.1). *Example:* "*192.168.1.1*"
	gateway	<ip>	**Default gateway** IP address of the default gateway the DHCP server assigns to requesting clients. Usually this is the internal IP address of the device. *Beispiel:* "*192.168.1.1*"
	lease_time	<time_dhm>	Period of time during which the network configuration assigned to a client is valid for the client. Even if the client temporarily does not have a network connection to the DHCP server, the same network configuration will always be assigned to the client when another request is made within this time period. The client should renew its assigned configuration shortly before this period expires. Otherwise, the configuration may be assigned to another client. The period of time can be specified in days (d), hours (h), **or** minutes (m). *Example:* "*12h*"
	netmask	<nm_num>	**Local netmask** Subnet mask the DHCP server assigns to requesting clients. The range from which network clients are assigned IP addresses should be chosen so that the IP addresses can be reached in the assigned subnet (see keys: *range_low, range_high*). *Example: 24*

Table 3-23 End point **configuration**, key(s): **service >> dhcp_server**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
	range_low	<ip>	**IP range start** Start of the IP address range from which the DHCP server assigns IP addresses to requesting clients. The range should be chosen so that the IP addresses it contains can be reached in the assigned subnet (see key: *netmask*). *Example:* "*192.168.1.2*"
	range_high	<ip>	**IP range end** End of the IP address range from which the DHCP server assigns IP addresses to requesting clients. The range should be chosen so that the IP addresses it contains can be reached in the assigned subnet (see key: *netmask*). *Example:* "*192.168.1.249*"
	status	"ON" "OFF"	**DHCP server for net zone 2** When this function is activated, requesting clients that are connected to the device via net zone 2 are assigned a network configuration. **Note:** The requests to UDP port 67 are always accepted regardless of the firewall table settings of the device if the DHCP server is activated. The server then assigns IP addresses to the clients from the configured IP address range. *Example:* "*ON*"
	wins_server	<ip>	**WINS server** IP address of a WINS server that the DHCP server assigns to requesting clients. A WINS (*Windows Internet Naming Service*) server allows clients to resolve hostnames (*NetBIOS* names) into IP addresses. *Example:* "*192.168.1.252*"

3.4.12 Service (DNS cache/DNS server)

Example

"**service**": "dnscache": {"allowed_requests": ["NETZONE1", "NETZONE2"], "dns_servers": "USER_DEFINED", "log": "ON", "user_defined": [{"ip": "192.168.1.150", “comment“: “Company DNS server A“}, {"ip": "192.168.1.160", “comment“: “DNS server fallback“}]}

Table 3-24 End point **configuration**, key(s): **service >> dnscache**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (dnscache)	allowed_requests	"NETZONE1" "NETZONE2"	**DNS server reachable from net zone 1/2** Access to the DNS server of the device is permitted from the specified net zone (UDP/TCP port 53). *Example: "NETZONE2"*
	dns_servers	"USER_DEFINED" "ROOT_DNS_SERVER"	Users can select whether the preset “root DNS servers” or “user-defined DNS servers” are used in the device for the resolution of hostnames. **Note:** This choice is only available if the device **does not receive its network configuration from a DHCP server** (see Section 3.4.6). **Root DNS server** **Only** the default root DNS servers in the device are used for the resolution of hostnames. The first available root DNS server will be used. **User-defined** **Only** the user-defined DNS servers are used for the resolution of hostnames. Several DNS servers can be specified. If a DNS server is not specified, hostnames are not resolved. *Example: "ROOT_DNS_SERVER"*
	log	"ON" "OFF"	**Log DNS requests** When this function is activated, a log entry is created for all requests (UDP/TCP) to the DNS server of the device. Log entries can be analyzed via the *logging* end point (see Section 3.11) or in the *journal* file, which can be created and downloaded via a snapshot (see Section 3.10). For allowed requests (““*allowed_requests*”” variable): – Log prefix: *fw-input-dnscache-* For all other requests: – Log prefix: *fw-input-policy-* *Example: "OFF"*

Table 3-24 End point **configuration**, key(s): **service >> dnscache**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (dnscache, user_defined)	ip	<ip>	**User-defined DNS server** IP address of one or more DNS servers that are queried by the device for resolving hostnames. *Example:* "*46.182.19.48*"
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

3.4.13 Service (NTP server/NTP client)

Example

"**service**": "ntp": {"allow_client_requests": ["NETZONE1", "NETZONE2"], "server": [{"address": "0.pool.ntp.org", "port": 123, “comment“: “Company NTP 1“}, {"address": "1.pool.ntp.org", "port": 123, “comment“: “Company NTP fallback“}], "status": "ON"}

Table 3-25 End point **configuration**, key(s): **service >> ntp**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (ntp)	status	"ON" "OFF"	**NTP** This function can be used to activate the NTP client and the NTP server of the device. The NTP server of the device is only activated if access to the NTP server is permitted from at least one net zone (see *"allow_client_requests"*). **NTP client** When this function is activated, the device obtains its system time (time and date) from one or more NTP servers and continuously synchronizes itself with them. The NTP server transmits the *Universal Time Coordinated* (UTC). The time on the device (system time) will be displayed in accordance with the configured time zone and used (e.g., in log entries). The *real-time clock (RTC)* of the device is automatically synchronized with the time data obtained from the NTP servers. Initial time synchronization can take up to 15 minutes or more. During this time, the device continuously compares the time data of the external NTP servers to its own system time so that they can be adjusted as accurately as possible. **NTP server** When this function is activated, connected network clients can synchronize their system time via the NTP server of the device. The NTP server transmits the *Universal Time Coordinated* (UTC). Access to the NTP server can be limited to selected sources (net zones, IP addresses or networks) (see "*"allow_client_requests"*"). *Example:* "ON"
	allow_client_requests	"NETZONE1" "NETZONE2"	**NTP server can be reached from net zone 1/2** Access to the NTP server of the device is permitted from the specified net zone (UDP port 123). The NTP server of the device is only activated if access from at least one net zone is permitted. *Example: "NETZONE1"*

Table 3-25 End point **configuration**, key(s): **service >> ntp**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (ntp, server)	address	<ip> <string>	**IP/Hostname** IP address or hostname of the external NTP server (time server) to which the device is to send NTP requests to obtain the current time (time and date). If several NTP servers are specified, the device automatically connects to all of them to determine the current time from all values received. **Input format**: IPv4 address or hostname *Example: "0.pool.ntp.org"*
	port	<num> (or empty)	**Port** Port on which the external NTP server accepts NTP requests. Specifying a port is optional *Example: 123*
	comment	<string>	**Comment** Freely selectable comment. Permitted characters: max. 128

3.4.14 Service (SNMP server)

Example

"**service**": "snmp": {"allow_requests_from": ["NETZONE1", "NETZONE2"], ro_community_string": "public", "status_v2c": "ON", "status_v3": "ON", "user": {"new_password": "My-Password_123", "repeat_password": "My-Password_123", "username": "SNMP-mGuard_01"}}

Table 3-26 End point **configuration**, key(s): **service >> snmp**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (snmp)	allow_request_from	"NETZONE1" "NETZONE2“	**SNMP server can be reached from net zone 1/2** Access to the SNMP server of the device is permitted from the specified net zone (UDP port 161). The SNMP server is not activated until access from at least one net zone is permitted. *Example: "NETZONE1"*
	ro_community_string	<string>	**Read-only community** With the SNMPv1/SNMPv2c version, SNMP encodes the access data as part of what is referred to as a *community*. Here, the *read-only community* string is used as a password or access key. Authentication via the *read-only community* string allows limited SNMP write access. **Input format:** The string must begin with a letter. Permitted characters (min. 6, max. 255): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789_- *Example: "public"*
	status_v2c	“ON“ “OFF“	**SNMPv2c** When this function is activated, the device can be monitored via the SNMPv2c protocol (read access). **NOTE: Non-secure protocol** The unencrypted SNMPv1/2 protocol should only be used in a secure network environment that is entirely under the control of the operator. When SNMPv2c is activated, the SNMPv1 protocol is also supported. The SNMP server is only activated if access from at least one net zone is permitted (see Above). *Example: "OFF"*

Table 3-26 End point **configuration**, key(s): **service >> snmp**

<table>
<tr><th>Key(s)</th><th>Variable (key)</th><th>Value (format)</th><th>Designation (WBM)/description</th></tr>
<tr><td></td><td>status_v3</td><td>“ON“
“OFF“</td><td><b>SNMPv3</b>
When this function is activated, the device can be monitored via the SNMPv3 protocol (read access).
ℹ Unlike the SNMPv1/v2c protocols, the SNMPv3 protocol is considered secure because it provides the option for user authentication and for encryption. Encryption and hash algorithms used:
– AES-128
– SHA-2 (SHA-256) with SNMPv3 USM
The SNMP server is not activated until access from at least one net zone is permitted (see above).
<i>Example: "OFF"</i></td></tr>
<tr><td><b>service</b>
(snmp, user)</td><td>new_password</td><td><string></td><td><b>Password</b>
The new password of the corresponding SNMP user.
<b>Note:</b> Once the configuration is saved, the configured password will no longer be shown.
<b>Input format:</b> To increase security, the password should contain uppercase and lowercase characters, numbers, and special characters.
Permitted characters (min. 8, max. 200):
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
<i>Example: "My-Password_123"</i></td></tr>
<tr><td></td><td>repeat_password</td><td><string></td><td><b>Confirm password</b>
Enter the password again.</td></tr>
<tr><td></td><td>username</td><td></td><td><b>Username</b>
Username of the SNMPv3 user who would like to access the device SNMP server via the SNMPv3 protocol.
The addition of further SNMPv3 users is not supported.
<b>Input format:</b> Permitted characters (min. 1, max. 200):
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789_.-
<i>Example: "SNMP-mGuard_01"</i></td></tr>
</table>

3.4.15 Service (session timeout)

Example "**service**": "web": {"session_timeout“: 60, "user_blocking_time": 10, "user_max_failed_logins": 5}

Table 3-27 End point **configuration**, key(s): **service >> web**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
service (web)	session_timeout	<time_minute>	**Session timeout (hh:mm)** Length of the *session timeout* (time period). A user session is limited in time by a *session timeout*. The configurable time period of the *session timeout* is between 5 minutes and 8 hours. After the session times out, the user is logged out automatically. The *session timeout* period begins when the user logs in (default setting: 30 minutes). If the user executes an action during a session, the *session timeout* period is reset to the configured start value. **Input format:** Minutes (min. 5, max. 480) *Example: 60*
	user_blocking_time	<time_minute>	**Period for which a user will be blocked (hh:mm)** Period for which a user will be blocked after unsuccessful login attempts. Users are automatically blocked after a configurable number of unsuccessful login attempts (incorrect password entry) for up the configured period (see below). **Note:** The block can be prematurely removed by an administrator with the “*Super Admin*” role (see Section 3.19). **Note:** An automatic user block is also removed by rebooting the device. **Input format:** Minutes (min. 1, max. 480) *Example: 10*

Table 3-27 End point **configuration**, key(s): **service >> web**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
	user_max_failed_logins	<string>	**Number of unsuccessful login attempts until a user is blocked** Number of unsuccessful login attempts until a user is blocked. Users are automatically blocked after the configured number of unsuccessful login attempts (incorrect password entry) for up to 8 hours (see above). **Note:** The block can be prematurely removed by an administrator with the "*Super Admin*" role (see Section 3.19). **Note:** An automatic user block is also removed by rebooting the device. **Input format:** Number (min. 5, max. 200) *Example: 3*

3.4.16 System

Via the "*System*" end point, you can

1. Change the hostname of the device
2. Save the current configuration to an SD card (e.g. for a device replacement)
3. Configure the system use notification

Example

"**system**": {"hostname": "mGuard-production-01", "store_config_on_sdcard": "ON", "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."}

Table 3-28 End point **configuration**, key(s): **system**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
system	hostname	<string>	**Hostname** Name under which the device is always visible and reachable in the network. If the hostname is resolved using the *Domain Name System* (DNS), network devices can address the device directly via its hostname. **Input format:** The name must begin and end with a letter or a number. Permitted characters (min. 1, max. 63): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789- *Example: "mGuard-production-01"*
	usenotification	<string>	**System use notification** Freely selectable text for a system use notification that is displayed before logging onto the device (maximum 512 characters). Is displayed for: – Logging on via web-based management (WBM) *Example: “The usage of this mGuard security appliance is reserved to authorized staff only.“*

Table 3-28 End point **configuration**, key(s): **system**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
	store_config_on_sdcard	"ON" "OFF"	**Automatically save configuration** If this function is activated, every configuration change that is saved in the WBM or via the *Config API* will be saved to the inserted SD card automatically. Three files will be saved: – *users_pass.json* – *snmp-pass.conf* – *configuration.json* ℹ **Re-importing the saved configuration into the device via SD card:** The following applies to all **new devices** or devices that are reset to the factory settings via smart mode: A configuration/user management saved on the inserted SD card is automatically imported into the device and used there when the device is started or commissioned. **Prerequisite:** – The firmware version of "SD card" in the minor version is lower than/equal to the firmware version of "device". – The SD card contains the three files (individually or bundled as *mGuard.tar.gz:* Use the individual files as first priority!). If an error occurs during the import, the device will boot with default values. The FAIL and PF1 LEDs will also light up.

3.4.17 Time zone

Example "**zoneinfo**": "Europe/Berlin"

Table 3-29 End point **configuration**, key(s): **zoneinfo**

Key(s)	Variable (key)	Value (format)	Designation (WBM)/description
system	zoneinfo	<timezone>	**Time zone** The manually set or NTP-obtained system time will be displayed in accordance with the configured time zone and used (e.g., in log entries). See Section 5.1 for available time zones *Example: "Europe/Berlin"*

3.5 “configuration/default” end point

Via this end point, the default setting of the end point elements can be
1. displayed (*GET request*) or
2. restored (*POST request*).

The current administrator password and certificates do not change.

Example: Display configuration (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/configuration/default
```

Response:

⇒ (Result/response: see Section 4.1)

Example: Apply factory setting (POST)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -X POST https://192.168.1.1:443/api/v1/configuration/default
```

3.6 “users” end point

Only visible for users with the *Super Admin* user role.

Locally saved passwords are not transmitted with a GET request.

The following settings can be made via this end point:

- An external LDAP server can be configured (**key: ldap**, see Section 3.6.1, “Users >> LDAP”)
- The properties of the existing local users can be displayed and local users can be added, edited, and deleted (**key: user_mgmt**, see Section 3.6.2, “Users >> User management”)

User roles and permissions

Table 3-30 User roles and permissions

Permission/role	Super Admin	Admin	Audit
Manage users	x		
Configure LDAP	x		
Change configuration	x	x	
Execute actions	x	x	
Install firmware updates	x	x	
Check configuration	x	x	x
Change own password	x	x	x
Request device status	x	x	x
Read log entries	x	x	x

Example: Configuration – LDAP server/user management – Display (GET) configuration

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/users
```

Response:

```
{"content":

{"ldap": {"ldap_server": {"base_dn": "DC=mguard,DC=management", "ca": "-----BEGIN CERTIFICATE-----\nMII [...]
nF\nBW5/87JeonwLYiT0JjajXDGLAf0t4O\n-----END CERTIFICATE-----\n", "hostname": "192.168.2.100", "port": 389,
"tls": "ON", "username": "admin_ldap" }, "status": "ON", "user_role_mapping": {"admin": "Role_2", "audit": "Role_3",
"ldap_attribute": "Role", "super_admin": "Role_1" } },

"user_mgmt": {"current_user": "admin",

"users": [{"block_user": "OFF", "name": "", "old_username": "admin", "role": "SUPERADMIN", "username": "admin" },
{"block_user": "OFF", "name": "", "old_username": "admin_production", "role": "ADMIN", "username": "admin_production"
}] } },

"envelope": {"identifier": {"contentID": "4b7a11b1", "functionalID": "4b7a11b1" }, "version": 1 }, "error": [], "schemes":
[{"name": "users.manageusers.e52f65cd", "url": "/v1/users/scheme/users.manageusers.e52f65cd" }], "status": 0 }
```

Response: (For a structured view of another example, see Section 4.5)

Example: Change (POST) user properties and passwords

The standard user “*admin*” (role: *Super Admin*) is logged in and, using a POST request, wants to:

1. Change their user name to "*superadmin*"
2. Changing the "*admin_production*" user’s current password
3. Adding the "*audit_production*" user with the respective roles
4. (The settings for the LDAP server (key: "*ldap*") will not be changed.) It is only necessary to enter the LDAP password if a change is made!)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type:application/json" -X POST
https://192.168.1.1:443/api/v1/users -d

'{"content": {"ldap": {"ldap_server": {"base_dn": "DC=mguard,DC=management", "ca": "-----BEGIN CERTIFICATE-----
\nMII [...] nF\nBW5/87JeonwLYiT0JjajXDGLAf0t4O\n-----END CERTIFICATE-----\n", "hostname": "192.168.2.100",
"password": "ldap_server_password", "port": 389, "tls": "ON", "username": "server-admin"}, "status": "ON", "user_role_-
mapping": {"ldap_attribute":"Role", "admin": "Role_2", "audit": "Role_3", "super_admin": "Role_1"}},

"user_mgmt": {"old_password": "private", "current_user": "admin",

"users": [{"block_user": "OFF", "name": "", "old_username": "admin", "role": "SUPERADMIN", "username": "superadmin"},
{"block_user": "OFF", "name": "", "old_username": "admin_production", "role": "ADMIN", "username": "admin_production",
"new_password": "secret_production_password","repeat_password": "secret_production_password" },{ "block_user":
"OFF", "name": "", "old_username": "", "role": "AUDIT", "username": "secret_audit_production","new_password":
"secret_audit_password","repeat_password": "secret_audit_password"}]}}, "envelope": { "version": 1 }}'
```

3.6.1 Users >> LDAP

Table 3-31 End point: **users,** key(s): **ldap**

End point	Variable (key)	Value (format)	Designation (WBM)/description
users, ldap (ldap_server)	base_dn	<string>	**Base DN** Base address in the directory on the LDAP server. The search for the desired objects (e.g., user data) is restricted to a smaller area in the LDAP server directory tree. This takes place exclusively below the specified base address (node). **Input format:** directory path (*DC=x,DC=y,DC=z*) Permitted characters (min. 1, max. 1024): The entry must begin with one of the following characters: ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789._ These characters can each be connected by one of the following four characters: -_=, **Example:** DC=mguard,DC=management,DC=user
	hostname	<ip> <string>	**IP/Hostname** IP address or hostname of the external LDAP server to which the device is supposed to send requests for user authentication. **Input format**: IPv4 address or hostname *Example: "my-ldap-server.com"*
	password	<string>	**Password** Password with which the device logs into and authenticates the LDAP server. **Note:** Once the configuration is saved, the configured password will no longer be shown. **Input format:** To increase security, the password should contain uppercase and lowercase characters, numbers, and special characters. Permitted characters (min. 6, max. 200): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789!#$%&()*+,-./:;<=>?[]^_`{|}~@ *Example: "ldap_password_183"*
	port	<num>	**Port** Port on which the external LDAP server accepts requests. *Example: 389*

Table 3-31 End point: **users,** key(s): **ldap**

End point	Variable (key)	Value (format)	Designation (WBM)/description
	username	<string>	**Username** Username with which the device logs into and authenticates the LDAP server. Permitted characters (min. 1, max. 200): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789_.- *Example: "mGuard_183"*
	tls	"ON" "OFF"	**LDAP over TLS** When this function is activated, the data is transmitted with encryption using a TCP connection. **Note:** For reasons of security, an encrypted TLS connection should always be used between the device (mGuard) and the LDAP server. **Prerequisite:** To ensure the integrity and authenticity of the encrypted TCP connection, the server certificate (CA certificate) of the remote server must be installed on the device (see below).
	ca	<string>	**Upload server CA certificate to the device** CA certificate with which the device authenticates the remote server (LDAP server). The CA certificate is provided by the remote server operator and must be uploaded to the device (X.509 certificate with public key). An encrypted TCP connection to the remote server can only be established successfully if it in turn has a certificate issued by the CA certificate (with the secret key) or a valid certificate chain with the CA certificate as the highest instance. **Format:** The maximum file size allowed is 1 MB.

Table 3-31 End point: **users,** key(s): **ldap**

End point	Variable (key)	Value (format)	Designation (WBM)/description
users, ldap (status)	status	"ON" "OFF"	**LDAP authentication** When this function is activated, the device can access a configured LDAP server via the LDAP protocol. Users managed on the LDAP server can be authenticated when logging into the device via the LDAP protocol and entering their LDAP access data. ℹ When a user logs in (log in), the device first checks whether the user has been configured as a **local user** on the device. If this is the case, the local user can only be logged in with the **locally configured user password**. In this case, the LDAP server is not queried. ℹ A user logged in via LDAP is automatically logged out when the function is deactivated during the ongoing session. *Example: "ON"*
users, ldap (user_role_mapping)	ldap_attribute	<string>	**LDAP attribute** Name of the attribute in which the role/user class is specified for each LDAP user. To be able to assign the roles, they must be assigned the same LDAP attribute on both the LDAP server and on the device. **Example configuration:** Configuration on the LDAP server**:** – **Role:** *Role_1* – **Role:** *Role_2* – **Role:** *Role_3* LDAP attribute to be specified on the mGuard device**:** – *Role* Permitted characters (min. 1, max. 200): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789_.- *Example: "Role"*

Table 3-31 End point: **users,** key(s): **ldap**

End point	Variable (key)	Value (format)	Designation (WBM)/description
	admin	<string>	When logging in via LDAP, the user role (or user roles) assigned to the LDAP user on the LDAP server must be assigned to at least one of the three available user roles on the device (see also“User roles and permissions” on page 76). If the user role of the LDAP user cannot be assigned, it is not possible for this user to log in. **Example:** **Device <-> LDAP server** Super Admin <-> Role_1 Admin <-> Role_2 Audit <-> Role_3 If several user roles are assigned to one LDAP user, the user is logged in with the role with the highest possible authorization level when logging in. Permitted characters (min. 1, max. 200): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789_.- *Example: "Role_1"*
	audit	<string>	
	super_admin	<string>	

3.6.2 Users >> User management

Table 3-32 End point **users,** keys: **user_mgmt**

End point	Variable (key)	Value (format)	Designation (WBM)/description
user_mgmt (current_users)	old_password	<string>	The password of the logged-in user must be specified if changes are made in the “users” end point and these are to be sent to the device via a POST request. **Note:** Once the configuration is saved, the configured password will no longer be shown. *Example: "current_password"*
	current_user	<string>	The user name of the logged-in user.
user_mgm (users)	username	<string>	**Username** Unique username that the user uses to log into the device. **Input format:** The name must begin with a letter or a number. It must not end with a dot. Permitted characters (min. 2, max. 200): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789_.- *Example: "admin_01_dep-1.15"*
	role	SUPERADMIN ADMIN AUDIT	**Role** The selection of a user role assigns certain permissions to the user. The standard user in the default “admin” setting has the “*Super Admin*” role. However, users with the “*Super Admin*” role cannot delete themselves. *Example: "SUPERADMIN"*
	name	<string> (or empty)	**Real name** Freely assignable name for simplification of management. *Example: "Administrator 01"*

Table 3-32 End point **users,** keys: **user_mgmt**

End point	Variable (key)	Value (format)	Designation (WBM)/description
	new_password	<string>	**New password** The new password for the corresponding user. **Note:** Once the configuration is saved, the configured password will no longer be shown. **Input format:** To increase security, the password should contain uppercase and lowercase characters, numbers, and special characters. Permitted characters (min. 6, max. 64): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ *Example: "My-Password_123"*
	repeat_password	<string>	**Confirm new password** Enter the new password again.
	block_user	"ON" "OFF"	**Block user** When this function is activated, the associated user is blocked and cannot log back into the device. Users cannot block themselves. **Note:** Logged in users remain logged in during their ongoing session even if they are blocked by another instance. **Note:** Users authenticated by an LDAP server can only be blocked via the LDAP server user management function. *Example: "ON"*

3.7 “password” end point

The password of the registered user can be changed via this end point.

ℹ Locally saved passwords are not transmitted with a GET request.

Example

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type:application/json" -X POST
https://192.168.1.1:443/api/v1/password -d '{"content": {"old_password": "private", "new_password": "My-Pass-
word_123", "repeat_password": "My-Password_123"}, "envelope": {"version": 1}}'
```

Table 3-33 “**password**” end point

End point	Method	Variable (key)	Value (format)	Designation (WBM)/description
password	**POST**	old_password	<string> (password in plain format)	**Current password** The logged in user’s current password that is to be changed.
		new_password	<string> (password in plain format)	**New password** The new password for the logged in user. **Note:** Once the configuration is saved, the configured password will no longer be shown. **Input format:** To increase security, the password should contain uppercase and lowercase characters, numbers, and special characters. Permitted characters (min. 6, max. 64): ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ *Example:* "*My-Password_123*"
		repeat_password	<string> (password in plain format)	**Confirm new password** Enter the new password again.

3.8 “update” end point

The upload of a signed update file provided by Phoenix Contact can be initiated via this end point and the firmware update can be started.

All settings, passwords, and certificates are retained on the device.

Downgrading from a higher to a lower firmware version is not possible.

Example

```
curl -v -b session_cookie -H "X-CSRF-Token: <TOKEN> "-H "Content-Type:multipart/form-data" -X POST -F
update_info='{"content": {}, "envelope": {"version": 1}}' -F update_file=@/home/update/mGuard-image-1.8.0.mguard3.update.signed -k https://192.168.1.1:443/api/v1/update
```

Following successful installation of the update, the device automatically reboots after a few seconds. **Wait until the device has completely booted.**

Comment

- The *update_info* parameter does not contain any data about the JSON frame and is left empty.
- The *update_file* parameter contains the path to the update file.

3.8.1 Difference between update types

Table 3-34 Difference between update types (Example)

Update type	Property	Effect on the existing configuration
Patch release **Patch update**	Fixes errors from previous versions. The version number changes in the third digit position: – Version 1.7.**2**, for example, is a patch release for Version 1.7.**1** or 1.7.**0**.	The existing configuration remains unchanged.
Minor release **Minor update**	Extends the device with additional new properties and functions. The version number changes in the second digit position: Version 1.**8**.0, for example, is a minor release for Version 1.**7**.2 or 1.**6**.2.	1. If the device is in factory settings, then: – After the update, the device will be configured with the **new** firmware version’s settings. – It is possible that standard values of the existing firmware version could change or that properties and variables could be added or removed. 2. If changes have already been made to the existing device configuration, then: – The existing configuration will be applied unchanged. – New properties and variables from the **new** firmware version will be added to the existing configuration (in the factory setting). **Note:** The update can only be executed if any necessary adjustments are made to the existing configuration before the update. **Note:** If the update fails due to an incompatible configuration, an error message and/or log entry will inform the user of the reason for the error.
Major release **Major update**	Extends the device with completely new properties and functions. The version number changes in the first digit position: Version **2**.0.0, for example, is a major release for Version **1**.5.0 or **1**.4.2.	

3.9 “datetime” end point

Using this end point, the current time (UTC) of the device can be

1. displayed (*GET request*) or
2. set (*POST request*).

The manually set or NTP-obtained time (UTC) of the configured time zone is displayed via *GET request*.
The **time zone** can be changed in the “*configuration*” end point (see Section 3.4.17).

To set the time via *POST request*, the NTP client must first be deactivated (see Section 3.4.13).

Example: Display time (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/datetime
```

Response:

```
{"content": {"datetime": "2018-03-28_14:04:59"}, "envelope": {"identifier": {"contentID": "00bfc976", "functionalID": "00bfc976"}, "version": 1}, "error": [], "schemes": [{"name": "datetime.datetime.0020c25e", "url": "/v1/datetime/scheme/datetime.datetime.0020c25e"}], "status": 0}
```

Example: Set time (POST)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type:application/json" -X POST
https://192.168.1.1:443/api/v1/datetime -d '{"content": {"datetime": "2018-03-28_14:04:59"}, "envelope": {"version": 1}}'
```

Table 3-35 **datetime** end point

End point	Method	Key	Value (format)	Designation (WBM)/description
datetime	**POST**	datetime	<YYYY-MM-DD_hh:mm:ss>	**Set time and date** The device system time is configured and saved to the *real-time clock* (RTC). Permitted range: >= 2018-01-01_00:00:00 <= 2069-01-01_00:00:00 The system time will be displayed in accordance with the configured time zone and used (e.g., in log entries). *Example: "2018-03-28_14:04:59"*

3.10 “snapshot” end point

A snapshot can be created and downloaded via this end point.

The snapshot can be used for error diagnostics and communication with the support team. It contains the current configuration and other system information of the device (see Table 3-36):

Table 3-36 Content of a snapshot

File name	Content/description
File format: *json*	
config.json	Shows the current device configuration.
serdata.json	Shows the serialization data that was linked to the device during creation.
ldap.json	Shows the current configuration for LDAP authentication via LDAP server.
users.json	Shows current informations about the local users on the device.
File format: *txt*	
bootloader_version	Shows the version of the currently installed bootloader.
conntrack	Shows the current content of the status table (*connection tracking table*).
df	Shows the current amount of disk space available on the file system
eds	Shows the current dynamic status information of certain device functions.
ethtool_eth0	Shows information about the Ethernet port *eth0* (XF1 / net zone 1).
ethtool_eth1	Shows information about the Ethernet port *eth0* (XF2–5 / net zone 2).
ipset_list	Shows information about the currently used IP set.
ip_neight	Shows the current connection information for connected (*neighbored*) devices.
ip_route	Shows the current routing table.
ip_link	Shows the current connection status of the network interfaces.
ip_addr	Shows the current network configuration.
issue	Information on the firmware image.
journal	Shows the current log file of the system.
ls_mnt_hfs	Shows the files and directories currently in the device file system (/mnt/hfs).
mount	Shows the mounted file systems
nft_ruleset	Shows the firewall rules currently configured.
nft_tables	Shows the firewall tables currently configured.
proc_net_dev	Shows current information about the network traffic of all network interfaces (file */proc/net/dev)*.
proc_net_snmp	Shows information about the network traffic via the SNMP protocol (file */proc/net/snmp*).
pstree	Shows information about currently running processes.
services	Shows the services currently started on the system (*systemd*).
tpm2_fixed	Shows fiexed information about the TPM chip that cannot be changed.
tpm2_variable	Shows variable information of the TPM chip that can be changed.

Table 3-36 [...]Content of a snapshot

File name	Content/description
uptime	Shows the current operating time and the load average of the system.
userid	Shows the user ID and the group membership.
version	Shows the firmware version currently installed.

Sensitive data and security-relevant information (e.g., passwords or secret cryptographic/hashed keys) are not included in the snapshot.

Example: Create and download snapshot

```
curl -k -O -J -b session_cookie -X GET https://192.168.1.1:443/api/v1/snapshot
```

Response:

```
% Total  % Received  % Xferd  Average   Speed   Time     Time     Time     Current
                              Dload     Upload  Total    Spent    Left     Speed
100 31225  100 31225  0  0    4158          0   0:00:07  0:00:07  --:--:--  7256
curl: Saved to filename 'snapshot_2019-12-24_22_00_00.tar.gz'
```

The time the snapshot was created is specified in the file name as follows: <YYYY-MM-DD_hh:mm:ss> (also see Section 3.2)

3.11 “logging” end point

All or selected log entries on the device can be retrieved and displayed via this end point.

Firewall logging
Log entries are only created for packets with *Ether type IPv4*. Packets with other *Ether types* (e.g., *ARP, IPv6*) are not recorded in the log files. (Exception: Entries that affect the rate limit – *fw-input-rate-limit*)

With data connections (e.g., UDP, TCP, or ICMP), only the first packet of the connection will be logged (if logging is activated), because the connection is subject to *connection tracking*.

Remote logging (log server)
A remote server (*syslog* server) can be configured in the “*configuration*” end point (see Section 3.4.4).

In rare cases, generating a large number of log entries may result in a log entry not being transmitted. To be able to check this, each log entry, as described in the syslog protocol, is assigned a consecutive sequence ID (e.g., meta sequenceId="728").

Sensitive data and security-relevant information (e.g., passwords or secret cryptographic/hashed keys) are not included in the log files.

Unlike in the WBM, the time at which the log entry was created is always displayed in UTC, regardless of the time zone set.

Example: Retrieve all log entries (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/logging
```

Response:

```
{"content":{"logging":{"logs" :"Jan 27 08:09:44 kernel:

[…]

Jan 27 08:13:32 configapi[1963]:127.0.0.1 - - [27/Jan/2020 08:13:32] \"GET /v1/logging HTTP/1.1\" 200 -\n"}}, "error":[], "envelope":{ "version":1, "identifier":{ "contentID":"66db9094", "functionalID":"66db9094" }}, "status":0, "schemes":[ { "url":"/v1/logging/scheme/logging.logging.17ef3f7f", "name":"logging.logging.17ef3f7f" } ]}
```

Example: Retrieve only firewall log entries (POST)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST https://192.168.1.1:443/api/v1/logging -d '{"content": {"logging": {"features": ["firewall"]}}, "envelope": {"version":1}}'
```

Response:

```
{"content": {"logging": {"logs":"Mar 28 14:12:00 systemd[1]: Started Firewall Logger.\nMar 28 14:14:32 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\nMar 28 14:14:32 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\nMar 28 14:14:34 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\nMar 28 14:14:34 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\nMar 28 14:14:36 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\nMar 28 14:14:36 firewall-log[1618]: fw-forward-policy: IPv4 PROTO=ICMP SRC=10.1.0.68 DST=192.168.1.2 SPT=0 DPT=0\n"}}, envelope": {"identifier": {"contentID":"993a659f","functionalID":"993a659f"},"version": 1},"error":[],"schemes": [{"name": "logging.logging.17ef3f7f","url":"/v1/logging/scheme/logging.logging.17ef3f7f"}],"status": 0}
```

Table 3-37 **logging** end point

End point	Method	Key	Value (format)	Designation (WBM)/description
logging	**POST**	"features"	"firewall"	**Only firewall** Only log entries of events relating to the firewall are retrieved and displayed. *Example: "firewall"*

3.12 “status” end point

Dynamic status information regarding certain device functions can be retrieved and displayed in JSON format via this end point.

For example:

- Current firmware version
- Test mode alarms
- Status of the *Firewall Assistant*
- DHCP client (network configuration data received from the DHCP server)

Example: Retrieve dynamic status information (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/status
{ "content": {
  "firewall": {
   "forward": {
    "testmode": {
     "hit": "" } } },
  "fwassist": {
   "status": "OFF"
  },
  "network": {
   "dhcp": "",
   "noroute": "0",
   "ntp_state": "NOT_SYNCED"
  },
  "system": {
   "admin_password_is_default": "FALSE",
   "firmwareversion": "1.8.0"
  },
  "tcpdump": {
   "status": "OFF"
  },
  "users": {
   "admin": {
    "block_start_time": "",
    "block_status": "UNBLOCKED"
   },
   "admin_extern": {
    "block_start_time": "",
    "block_status": "BLOCKED_BY_ADMIN"
   },
   "audit_production": {
    "block_start_time": "",
    "block_status": "BLOCKED_BY_AUTO"
   } } }, "envelope": { "identifier": { "contentID": "facbc43c", "functionalID": "facbc43c" }, "version": 1 }, "error": [], "schemes": [], "status": 0}
```

3.13 “actions/fwassist” end point

The *Firewall Assistant* can be started and stopped via the “*/v1/actions/fwassist/start*” and “*/v1/actions/fwassist/stop*” end points respectively.

Description

If activated, the *Firewall Assistant* analyzes and acquires the data traffic *routed* through the device (**Net zone 1 ←→ Net zone 2**).

In the process, the firewall is open in both directions.

The acquired packet data is used to derive firewall rules that are automatically entered into the corresponding firewall table of the device when the *Firewall Assistant* is exited.

The data traffic defined in these firewall rules will be allowed in the future (**Action = Accept**). All other connections will be dropped.

The firewall tables created using the *Firewall Assistant* can be adapted and extended as required.

Table 3-38 Firewall Assistant: Conversion of packet data into firewall rules

Header entry	Entry in firewall rule	Example
Source IP address	**src_network**	*10.1.1.55*
Destination IP address	**dst_network**	*192.168.1.100*
The respective netmask of the source and destination network is not acquired. Only the individual IP addresses are acquired and applied in the firewall rule.		
Destination port	**dst_port**	*443*
If no destination port is transmitted (e.g., as for the *ICMP* protocol), no value is entered in the firewall rule.		
Protocol	**protocol**	*ALL*
The following protocols can be applied as values in the firewall rule: – *TCP, UDP, ICMP, GRE, ESP* For all other protocols, the value “*ALL*” is entered in the firewall rule.		
—	**verdict**	*ACCEPT*
In all firewall rules created via the *Firewall Assistant* or *Firewall test mode*, “*Accept*” is always entered as the action value.		

3.13.1 Start Firewall Assistant ("actions/fwassist/start")

NOTE: The firewall is deactivated.
If the *Firewall Assistant* is activated, connected network clients are no longer protected by the firewall.

The *Firewall Assistant* can only be started if **all firewall rules** in all firewall tables were previously deleted (see Section 3.4.1).

The *Firewall Assistant* can be activated via this end point:

⇒ Data traffic is analyzed and acquired.

⇒ The firewall is open in both directions.

Example:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/fwassist/start
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
],"status":0}
```

3.13.2 Stop Firewall Assistant ("actions/fwassist/stop")

NOTE: The automatically created firewall rules are active without prior checking.
Immediately check the newly created firewall rules and adapt them based on your security requirements.

The activated *Firewall Assistant* can be stopped via this end point:

⇒ The acquired packet data is used to automatically create firewall rules, which are entered in the corresponding firewall tables (WBM menu: **Network security >> Firewall >> Rules**, see Table 3-8).

⇒ The entered rules immediately and permanently allow the corresponding data traffic (**Action = Accept**) (see Table 3-38).

Example:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/fwassist/stop
```

Response (extract):

```
{"content": "{\"fileinfo\": {\"devtype\": \"0001010111020000\", \"firmware\": \"1.8.0\"}, \"firewall\": {\"forward\": {\"sanity_check\": \"ON\", \"stealth_allow_dhcp\": \"ON\", \"tables\": [{\"in_netzone\": \"NETZONE1\", \"out_netzone\": \"NETZONE2\", \"rules\": [{\"dst_network\": \"192.168.1.1\", \"dst_port\": \"ALL\", \"id\": 1, \"protocol\": \"ALL\", \"src_network\":
\"10.1.0.68\", \"verdict\": \"ACCEPT\"}]}, {\"in_netzone\": \"NETZONE2\", \"out_netzone\": \"NETZONE1\", \"rules\": []}],
\"testmode\": \"OFF\"},
...
```

3.14 “actions/ping” end point

This end point can be used to check whether a network client is connected to a device interface via its IP address and can be reached via the ICMP protocol.

Example

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/ping -d '{"content": {"dst_ip": "192.168.1.3"}, "envelope": {"version": 1}}'
```

Response:

```
{"content": {
"result": "PING 192.168.1.3 (192.168.1.3): 56 data bytes\n64 bytes from 192.168.1.3: seq=0 ttl=128 time=1.801 ms\n64
bytes from 192.168.1.3: seq=1 ttl=128 time=1.670 ms\n64 bytes from 192.168.1.3: seq=2 ttl=128 time=1.521 ms\n64
bytes from 192.168.1.3: seq=3 ttl=128 time=1.515 ms\n64 bytes from 192.168.1.3: seq=4 ttl=128 time=1.486 ms\n\n---
192.168.1.3 ping statistics ---\n5 packets transmitted, 5 packets received, 0% packet loss\nround-trip min/avg/max =
1.486/1.598/1.801 ms\n"
},
  "envelope": {
    "identifier": {
      "contentID": "28e2909c",
      "functionalID": "28e2909c"
    },
    "version": 1
  },
  "error": [],
  "schemes": [],
  "status": 0
}
```

Table 3-39 **actions/ping** end point

End point	Method	Key	Value (format)	Designation (WBM)/description
actions/ping	**POST**	dst_ip	<ip>	**Ping** A ping request (*ICMP request*) is sent to the specified IP address of a network client. If the client can be reached via the ICMP protocol and any net zone of the device, it sends a response back to the device. *Example: "192.168.1.254"*

3.15 “actions/tcpdump” end point

The content of network packets that are sent or received via a specified network interface can be analyzed via this end point (*tcpdump*).

Filter options are used to define which network packets are to be analyzed.

The result of the analysis is saved to a file (**.pcap*), downloaded, and deleted from the device.

If the device is restarted while an analysis is running, the data acquired until then is deleted.

If the file (**.pcap*) exceeds a size of 50 MB, the analysis is aborted with an error. The data acquired until then is deleted.

3.15.1 Start network analysis ("actions/tcpdump/start")

Packet analysis (*tcpdump*) can be activated via this end point.

Example: Acquisition data

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/tcpdump/start -d '{"content": {"interface": "eth0", "options": "tcp and net
192.168.1.0/24 and not port 443"}, "envelope": {"version": 1}}'
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
],"status":0}
```

Table 3-40 **actions/tcpdump/start** end point

End point	Method	Key	Value (format)	Designation (WBM)/description
actions/tcpdump/start	**POST**	interface	*“lan(n)“* *“eth(n)“*	Only data packets that are sent or received via the selected network interface are analyzed. *Example: "eth0"*
		Options can be used to limit the packet analysis to a selection of the elements listed below. Options can be linked via the logical operators “*and*, *or*, *not*”. *Example: "tcp and net 192.168.1.0/24 and not port 443"*		
		options	tcp	TCP protocol
			udp	UDP protocol
			arp	ARP protocol
			icmp	ICMP protocol
			esp	ESP protocol
			host <ip>	IPv4 address
			port <1-65535>	Network port (single port number)
			net <nw_cidr>	Network (in CIDR format, e.g., 192.168.1.0/24)
			and, or, not	Logical operators

3.15.2 Stop network analysis ("actions/tcpdump/stop")

A running analysis (*tcpdump*) can be stopped via this end point. The acquired packet contents are compiled in a file (**.pcap*) and downloaded automatically from the device. Afterwards, the file is deleted from the device.

Example: Stop data acquisition and download data

```
curl -k -J -O -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/tcpdump/stop
```

Response:

```
% Total  % Received  % Xferd  Average  Speed   Time    Time    Time      Current
                              Dload    Upload  Total   Spent   Left      Speed
100 361   100 361     0  0    4158          0  0:00:07 0:00:07 --:--:--  7256
curl: Saved to filename 'tcpdump_2019-12-24_18_20_53.pcap'
```

The time of the file download is indicated in the file name as follows: <YYYY-MM-DD_hh:mm:ss> (see also Section 3.2)

3.16 “actions/pki/renew/logging” end point

The client certificate that is used for authenticating the device in a remote syslog server can be created or downloaded via this end point.

The self-signed client certificate with which the device authenticates itself to the remote server (*Syslog*-Server) is created on the device and saved there.

The operator must download and upload it to the remote server (X.509 certificate with *public* key).

GET request: The existing certificate will be downloaded

POST request: The certificate will be newly created and downloaded. The existing certificate will be dropped.

NOTE: The current certificate is deleted
When you create a new client certificate, the certificate currently saved on the device will be deleted permanently. The newly created certificate must be uploaded to the remote server again.

3.16.1 Download/displayclient certificate

Example: Downloading/Displaying a client certificate (GET)

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/actions/pki/renew/logging
```

Response:

{ "content": {
   "result": "-----BEGIN CERTIFICATE-----
\nMIIC+jCCAoCgAwIBAgIUYAZPkn8RAgJAmkIOnLDm+onXOScwCgYIKoZIzj0EAwIw\ngY4xIzAhBgNVBAMMGjB4ZD
c1ZmFiMDhfOUIraTVVNnJfc3lzbG9nMScwJQYDVQQK\nDB5QSE9FTklYIENPTlRBQ1QgQ3liZXIgU2VjdXJpdHkxDzA
NBgNVBAsMBm1HdWFy\nZDEPMA0GA1UEBwwGQmVybGluMQ8wDQYDVQQIDAZCZXJsaW4xCzAJBgNVBAYTAk
RF\nMB4XDTIwMDkyMzA5MDQwMloXDTIxMDkyMzA5MDQwMlowgY4xIzAhBgNVBAMMGjB4\nZDc1ZmFiMDhfOUIr
aTVVNnJfc3lzbG9nMScwJQYDVQQKDB5QSE9FTklYIENPTlRB\nQ1QgQ3liZXIgU2VjdXJpdHkxDzANBgNVBAsMBm
1HdWFyZDEPMA0GA1UEBwwGQmVy\nbGluMQ8wDQYDVQQIDAZCZXJsaW4xCzAJBgNVBAYTAkRFMHYwEAYH
KoZIzj0CAQYF\nK4EEACIDYgAEZ6tFsUk5fQFCz/9BiCUWnpugLfMukOFqvA7LxTfgCrm/m205vFjB\n8XioQ/6K7l/u46Q
xFkvFRVFCReSp42igsQPIB9UovrTS5QHFl1co8bZ0olHEYret\nc9mPYokYCRYIo4GcMIGZMB0GA1UdDgQWBBQsG
CVeZr4OqdwrUFNg+YeFB7mYPTAf\nBgNVHSMEGDAWgBQsGCVeZr4OqdwrUFNg+YeFB7mYPTAPBgNVHRMBAf
8EBTADAQH/\nMEYGA1UdEQQ/MD2CBm1HdWFyZIIJbG9jYWxob3N0hxD+gAAAAAAAAAAAAAAAAAAB\nhwR/AA
ABhxAAAAAAAAAAAAAAAAAAAAABMAoGCCqGSM49BAMCA2gAMGUCMQDsbX2a\nyUmuqqjOQD+5AzMNiAFl5h
aDmHkl0pEmvcLY9f8nNHQ8Me58PuZyw4VgKowCMDL3\nBsYB4Kc3fIirQUy7hn0RjV2OH1OQjGNS2cHopSQXC9In-
eNrTjuWfVe9Hr2RzKA==\n-----END CERTIFICATE-----\n"
  },
  "envelope": {"identifier": {"contentID": "d90879c6", "functionalID": "d90879c6"}, "version": 1}, "error": [], "schemes": [],
"status": 0}

3.16.2 Generating and downloading/displaying a new clientcertificate

Example: Generating and downloading/displaying a new client certificate (POST)

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/pki/renew/logging
```

Response:

```
{"content":{"result":"-----BEGIN CERTIFICATE-----
\nMIIC/jCCAoOgAwIBAgIUOqSd61vLE05FFS3HTUj7R95SLQowCgYIKoZIzj0EAwIw\ngY4xIzAhBgNVBAMMGjB4ZDd
hN2M1MDhfU3BNMVpVLWdfc3lzbG9nMScwJQYDVQQK\nDB5QSE9FTklYIENPTlRBQ1QgQ3liZXIgU2VjdXJpdHkxD
zANBgNVBAsMBm1HdWFy\nZDEPMA0GA1UEBwwGQmVybGluMQ8wDQYDVQQIDAZCZXJsaW4xCzAJBgNVBAYT
AkRF\nMB4XDTIwMDkyNDA4MzUyMloXDTIxMDkyNDA4MzUyMlowgY4xIzAhBgNVBAMMGjB4\nZDdhN2M1MDhfU3
BNMVpVLWdfc3lzbG9nMScwJQYDVQQKDB5QSE9FTklYIENPTlRB\nQ1QgQ3liZXIgU2VjdXJpdHkxDzANBgNVBAsM
Bm1HdWFyZDEPMA0GA1UEBwwGQmVy\nbGluMQ8wDQYDVQQIDAZCZXJsaW4xCzAJBgNVBAYTAkRFMHYwEA
YHKoZIzj0CAQYF\nK4EEACIDYgAEbfRgAuCxBh6iOq38GZVKbdlxC7OLSOZRB+C8RYDUu+f/+bzon/bP\nMZhjmuWh
whVFmFm0CBjRa4DuEN1LhJwKwmpTmKX4W2x8UCdLhEaAtbSXrFelCYo8\ngZSVs/Em8415o4GfMIGcMB0GA1Ud
DgQWBBT/HhB3tU2HOs0Xm3wg/PRGaAhvcjAf\nBgNVHSMEGDAWgBT/HhB3tU2HOs0Xm3wg/PRGaAhvcjAPBgNV
HRMBAf8EBTADAQH/\nMEkGA1UdEQRCMECCCXRyZXRzdHN0c4IJbG9jYWxob3N0hxD+gAAAAAAAAAAAAAAA\n
AAABhwR/AAABhxAAAAAAAAAAAAAAAAAAAAABMAoGCCqGSM49BAMCA2kAMGYCMQD4\ncX0FsoGCrfCD5kkx
r9TDZzul9bCu9O6mw6/BahwNmtVbKOflNrPXcYNIrNhgc7sC\nMQCsAvMxmMzkxVQoTLL2CDTCfQmyccZufdF5lSjH-
q6Hd+VVUls+2xUxs1R63D9sb\nsqE=\n-----END CERTIFICATE-----
\n"},"envelope":{"identifier":{"contentID":"e039f5fa","functionalID":"e039f5fa"},"version":1},"error":[],"schemes":[],"status":
0}
```

3.17 “actions/storeconfig/sdcard” end point

The configuration currently saved on the device can be written to the inserted SD card via this end point.

Three files will be saved:

– *users_pass.json, snmp-pass.conf, configuration.json*

Ensure that only authorized persons are able to access the SD card.

Do not remove the SD card until the write process has been completed.

Re-importing the saved configuration into the device via SD card

The following applies to all **new devices** or devices that are reset to the factory settings via smart mode:

A configuration/user management saved on the inserted SD card is automatically imported into the device and used there when the device is started or commissioned.

Prerequisite:

– The firmware version of "SD card" in the minor version is lower than/equal to the firmware version of "device".

– The SD card contains the three files (individually or bundled as *mGuard.tar.gz:* Use the individual files as first priority!).

If an error occurs during the import, the device will boot with default values. The FAIL and PF1 LEDs will also light up.

The saved configuration contains security-relevant information, such as local users, authorizations, passwords (hashed), and certificates (public keys). The password for the LDAP server is included in plain text.
Exception: Private keys are not included in the configuration.

Example: Writing the currently saved configuration to the SD card

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/storeconfig/sdcard
```

Response:

```
{ "content": "",
  "envelope": {
    "identifier": {
      "contentID": "330b153b",
      "functionalID": "330b153b"
    }, "version": 1}, "error": [], "schemes": [], "status": 0}
```

3.18 “actions/reboot” end point

This end point can be used reboot the device.

All changes that have not been saved will be lost.

Example: Rebooting the device

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/reboot
```

Response:

```
{
  "content": "",
  "envelope": {
    "identifier": {
      "contentID": "330b153b",
      "functionalID": "330b153b"
    },
    "version": 1
  },
  "error": [],
  "schemes": [],
  "status": 0
}
```

3.19 “actions/unblockuser” end point

This end point can be used by a user with the "*Super Admin*" role to unblock an automatically blocked user (see Section 3.4.15) before the blocking time has elapsed.

Example: Unblock user admin2

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/unblockuser -d '{"content": {"username": "admin2"}, "envelope": {"version": 1}}'
```

Response:

```
{"content":{},"envelope":{"identifier":{"contentID":"a3a6bf43","functionalID":"a3a6bf43"},"version":1},"error":[],"schemes":[
   {
"name": "unblockuser.unblockuser.228955d0",
"url": "/v1/actions/unblockuser/scheme/unblockuser.unblockuser.228955d0"
   }
],"status":0}
```

Table 3-41 **actions/tcpdump/start** end point

End point	Method	Key	Value (format)	Designation (WBM)/description
actions/unblockuser	**POST**	username	<string>	An automatically blocked user is unblocked before the blocking time has elapsed. The status of the user in the "Status" end point changes from "BLOCKED_BY_AUTO" to "UNBLOCKED" (see Section 3.12). **Note:** An automatic user block is also removed by rebooting the device.

3.20 “actions/migration” end point

Using this end point, a configuration that was created with an older firmware version is migrated to a configuration that corresponds to the firmware version currently installed on the device.

The migrated configuration is only displayed as a response to the POST request. A configuration can be uploaded and activated via the "*configuration*" end point (see Section 3.4).

The migrated configuration can be adapted with a text editor and then uploaded to the device via the "*configuration*" end point and activated there.

Example: Migrating a configuration (1.5.1) to a device with installed firmware version 1.8.0

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X POST
https://192.168.1.1:443/api/v1/actions/migration -d '
{{"content": {
  "fileinfo": {
   "devtype": "0001010111020000",
   "firmware": "1.5.1"
  },
  "firewall": {
   "forward": {
     "log_all_matches": "OFF",
     "log_policy": "OFF",
[...]
```

Response: (The response shows the migrated configuration: For a structured view, see Section 4.4)

```
"content": {
  "fileinfo": {
   "devtype": "0001010111020000",
   "firmware": "1.8.0"
  },
  "firewall": {
   "forward": {
     "ftp_allow_field": "OFF",
     "log_all_matches": "OFF",
     "log_policy": "OFF",
[...]
```

3.21 “usenotification” end point

The system use notification can be displayed via this end point.

Session cookies and *session tokens* are not required on this end point for a *GET request*.

The **system use notification is configured** via the "*configuration/system*" end point (see Section 3.4.16).

Example: Displaying the system use notification (GET)

```
curl -k -X GET https://192.168.1.1:443/api/v1/usenotification
```

Response:

```
{"content": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited.", "envelope": {"identifier": {"contentID": "00bfc976", "functionalID": "00bfc976"}, "version": 1}, "error": [ ], "schemes": [ ], "status": 0}
```

3.22 “softwarelicense” end point

The *Software License Terms* (SLT) currently valid for the product can be created and downloaded via this end point. SLTs are provided as a PDF file.

Example:

```
curl -k -O -J -b session_cookie -X GET https://192.168.1.1:443/api/v1/softwarelicense
```

Response:

```
% Total   % Received  % Xferd  Average  Speed   Time     Time     Time    Current
                               Dload    Upload  Total    Spent    Left    Speed
100 2186k 100 2186k   0     0  12.4M         0  0:00:07  0:00:07  --:--:--  12.3M
curl: Saved to filename 'Phoenix_Contact_Software_License_Terms_date_of_May_2018.pdf'
```

3.23 “licenses” end point

The third-party software components (modules) used on the device can be displayed via this end point.

Example:

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/licenses
```

Response:

```
{"content": {"modules": ["acl", "attr", "base-files", "base-passwd", "busybox", "bzip2", "ca-certificates", "conntrack-tools", "cracklib", "curl", "dbus", "dbus-glib", "dtc", "e2fsprogs", "ethtool", "expat", "gcc-runtime", "gdbm", "glib-2.0", "glibc", "gmp", "gnutls", "gptfdisk", "jq", "kmod", "libcap", "libffi", "libgcc", "libgcrypt", "libgpg-error", "libidn2", "libmnl", "libnetfilter-conntrack", "libnetfilter-cthelper", "libnetfilter-cttimeout", "libnfnetlink", "libnftnl", "libpam", "libpcre", "libseccomp", "libunistring", "libxcrypt", "libxml2", "linux-yocto", "mdio-tool", "ncurses", "netbase", "nettle", "nftables", "nginx", "openssh", "openssl", "opkg-utils", "os-release", "packagegroup-core-boot", "packagegroup-tpm2", "parted", "perl", "popt", "python3", "python3-click", "python3-flask", "python3-itsdangerous", "python3-jinja2", "python3-jsonmerge", "python3-jsonpointer", "python3-jsonschema", "python3-markupsafe", "python3-rfc3987", "python3-setuptools", "python3-simplejson", "python3-strict-rfc3339", "python3-werkzeug", "readline", "rng-tools", "run-postinsts", "shadow", "shadow-securetty", "shared-mime-info", "sqlite3", "systemd", "systemd-compat-units", "systemd-conf", "systemd-serialgetty", "tpm2-abrmd", "tpm2-tools", "tpm2-tss", "tpm2-tss-engine", "u-boot-tools", "update-rc.d", "util-linux", "volatile-binds", "xz", "zlib"]}, "envelope": {"identifier": {"contentID": "39632d7a", "functionalID": "39632d7a"}, "version": 1}, "error": [], "schemes": [{"name": "licenses.licenses.1362f8b6", "url": "/v1/licenses/scheme/licenses.licenses.1362f8b6"}], "status": 0}
```

3.24 “licenses/module/<module name>” end point

The license information for the third-party software components (modules) used on the device can be displayed via this end point.

Example: Displaying license information for the “curl” component

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/licenses/module/curl | python -m json.tool
```

Response:

```
{
  "content": {
    "license": [
      "COPYRIGHT AND PERMISSION NOTICE\n\nCopyright (c) 1996 - 2018, Daniel Stenberg, <daniel@haxx.se>, and many\ncontributors, see the THANKS file.\n\nAll rights reserved.\n\nPermission to use, copy, modify, and distribute this software for any purpose\nwith or without fee is hereby granted, provided that the above copyright\nnotice and this permission notice appear in all copies.\n\nTHE SOFTWARE IS PROVIDED \"AS IS\", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR\nIMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,\nFITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN\nNO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,\nDAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR\nOTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE\nOR OTHER DEALINGS IN THE SOFTWARE.\n\nExcept as contained in this notice, the name of a copyright holder shall not\nbe used in advertising or otherwise to promote the sale, use or other dealings\nin this Software without prior written authorization of the copyright holder.",
      "MIT License\n\nCopyright (c) <year> <copyright holders>\n\nPermission is hereby granted, free of charge, to any person obtaining a copy\nof this software and associated documentation files (the \"Software\"), to deal\nin the Software without restriction, including without limitation the rights\nto use, copy, modify, merge, publish, distribute, sublicense, and/or sell\ncopies of the Software, and to permit persons to whom the Software is\nfurnished to do so, subject to the following conditions:\n\nThe above copyright notice and this permission notice shall be included in\nall copies or substantial portions of the Software.\n\nTHE SOFTWARE IS PROVIDED \"AS IS\", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR\nIMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,\nFITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE\nAUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER\nLIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,\nOUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN\nTHE SOFTWARE."
    ]
  },
  "envelope": {
    "identifier": {
      "contentID": "ed097814",
      "functionalID": "ed097814"
    },
    "version": 1
  },
  "error": [],
  "schemes": [
    {
      "name": "licenses.licenses.1362f8b6",
      "url": "/v1/licenses/scheme/licenses.licenses.1362f8b6"
    }
  ],
  "status": 0}
```

4 Examples

4.1 GET Request (Endpoint: "configuration/default")

Get Request:

curl -k -b *session_cookie* -X GET https://192.168.1.1:443/api/v1/configuration/default

Response:

```
{
  "content": {
    "fileinfo": {
      "devtype": "0001010111020000",
      "firmware": "1.8.0"
    },
    "firewall": {
      "forward": {
        "ftp_allow_field": "OFF",
        "log_all_matches": "OFF",
        "log_policy": "OFF",
        "sanity_check": "ON",
        "stealth_allow_dhcp": "ON",
        "tables": [
          {
            "in_netzone": "NETZONE2",
            "out_netzone": "NETZONE1",
            "rules": [
              {
                "comment": "",
                "dst_network": "0.0.0.0/0",
                "dst_port": "ALL",
                "id": 0,
                "log": "OFF",
                "protocol": "ALL",
                "src_network": "0.0.0.0/0",
                "verdict": "ACCEPT"
              }
            ]
          },
          {
            "in_netzone": "NETZONE1",
            "out_netzone": "NETZONE2",
            "rules": []
    }
   ],
   "testmode": "OFF"
  },
  "input": {
   "rules": [
    {
     "id": 0,
     "log": "OFF",
     "service": "HTTPS",
     "source": "NETZONE2",
     "verdict": "ACCEPT"
    }
   ]
  },
  "port_forward": {
   "rules": []
  }
 },
 "logging": {
  "remote": {
   "address": "192.168.1.254",
   "port": 514,
   "protocol": "UDP",
   "status": "OFF"
  }
 },
 "network": {
  "mode": "ROUTER",
  "nat": {
   "1_1_nat": [],
   "masquerading": [
    {
     "from_ip": "0.0.0.0/0",
     "id": 0,
     "outgoing_on_if": "NETZONE1"
    }
   ]
  },
  "netzone1": {
   "mode": "DHCP"
  },
  "netzone2": {
      "address": "192.168.1.1",
      "netmask": 24
    },
    "routing": {
      "routes": []
    },
    "stealth": {
      "management_address": "192.168.1.1",
      "management_gateway": "192.168.1.254",
      "management_netmask": 24
    }
  },
  "service": {
    "dhcp_server": {
      "dns": "192.168.1.1",
      "gateway": "192.168.1.1",
      "lease_time": "12h",
      "netmask": 24,
      "range_high": "192.168.1.254",
      "range_low": "192.168.1.2",
      "status": "ON",
      "wins_server": ""
    },
    "dnscache": {
      "allowed_requests": [
        "NETZONE2"
      ],
      "dns_servers": "ROOT_DNS_SERVER",
      "log": "OFF",
      "user_defined": []
    },
    "ntp": {
      "allow_client_requests": [
        "NETZONE2"
      ],
      "server": [
        {
          "address": "0.pool.ntp.org",
          "comment": "",
          "port": 123
        },
        {
          "address": "1.pool.ntp.org",
      "comment": "",
      "port": 123
     },
     {
      "address": "2.pool.ntp.org",
      "comment": "",
      "port": 123
     },
     {
      "address": "3.pool.ntp.org",
      "comment": "",
      "port": 123
     }
    ],
    "status": "ON"
   },
   "snmp": {
    "allow_requests_from": [
     "NETZONE2"
    ],
    "ro_community_string": "public",
    "status_v2c": "OFF",
    "status_v3": "OFF"
   },
   "web": {
    "session_timeout": 30,
    "user_blocking_time": 10,
    "user_max_failed_logins": 5
   }
  },
  "system": {
   "hostname": "mGuard",
   "store_config_on_sdcard": "OFF",
   "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only.
Any intrusion and its attempt without permission is illegal and strictly prohibited."
  },
  "zoneinfo": "UTC"
 },
 "envelope": {
  "identifier": {
   "contentID": "72f6b081",
   "functionalID": "be532724"
  },
      "version": 1
    },
    "error": [],
    "schemes": [
      {
        "name": "common.4710ab60",
        "url": "/v1/configuration/scheme/common.4710ab60"
      },
      {
        "name": "common.types.f0bf23da",
        "url": "/v1/configuration/scheme/common.types.f0bf23da"
      },
      {
        "name": "configuration.fileinfo.b3afd1b0",
        "url": "/v1/configuration/scheme/configuration.fileinfo.b3afd1b0"
      },
      {
        "name": "configuration.firewall.62d07c99",
        "url": "/v1/configuration/scheme/configuration.firewall.62d07c99"
      },
      {
        "name": "configuration.logging.fce1b9ba",
        "url": "/v1/configuration/scheme/configuration.logging.fce1b9ba"
      },
      {
        "name": "configuration.network.0edde642",
        "url": "/v1/configuration/scheme/configuration.network.0edde642"
      },
      {
        "name": "configuration.service.1f00d993",
        "url": "/v1/configuration/scheme/configuration.service.1f00d993"
      },
      {
        "name": "configuration.system.ef2e081a",
        "url": "/v1/configuration/scheme/configuration.system.ef2e081a"
      },
      {
        "name": "configuration.zoneinfo.e8437e00",
        "url": "/v1/configuration/scheme/configuration.zoneinfo.e8437e00"
      }
    ],
    "status": 0
  }
```

4.2 GET Request (Endpoint: "configuration")

GET Request:

```
curl -k -b session_cookie -X GET https://192.168.1.1:443/api/v1/configuration
```

Response:

```
{
  "content": {
    "fileinfo": {
      "devtype": "0001010111020000",
      "firmware": "1.8.0"
    },
    "firewall": {
      "forward": {
        "ftp_allow_field": "ON",
        "log_all_matches": "ON",
        "log_policy": "ON",
        "sanity_check": "ON",
        "stealth_allow_dhcp": "ON",
        "tables": [
          {
            "in_netzone": "NETZONE1",
            "out_netzone": "NETZONE2",
            "rules": [
              {
                "comment": "",
                "dst_network": "192.168.1.20",
                "dst_port": "ALL",
                "id": 0,
                "log": "OFF",
                "protocol": "ALL",
                "src_network": "0.0.0.0/0",
                "verdict": "REJECT"
              },
              {
                "comment": "Office",
                "dst_network": "0.0.0.0/0",
                "dst_port": "ALL",
                "id": 1,
                "log": "ON",
                "protocol": "ALL",
                "src_network": "192.168.2.0/24",
                "verdict": "ACCEPT"
              },
              {
```

```
            "comment": "Production",
            "dst_network": "192.168.1.0/24",
            "dst_port": "ALL",
            "id": 2,
            "log": "ON",
            "protocol": "ALL",
            "src_network": "10.10.0.0/24",
            "verdict": "ACCEPT"
          }
        ]
      },
      {
        "in_netzone": "NETZONE2",
        "out_netzone": "NETZONE1",
        "rules": [
          {
            "comment": "",
            "dst_network": "0.0.0.0/0",
            "dst_port": "ALL",
            "id": 0,
            "log": "OFF",
            "protocol": "ALL",
            "src_network": "0.0.0.0/0",
            "verdict": "ACCEPT"
          }
        ]
      }
    ],
    "testmode": "ON"
  },
  "input": {
    "log_all_matches": "OFF",
    "log_policy": "OFF",
    "rules": [
      {
        "id": 1,
        "log": "OFF",
        "service": "HTTPS",
        "source": "NETZONE1",
        "verdict": "ACCEPT"
      }
    ]
  },
  "port_forward": {
    "rules": [
      {
```

```
      "comment": "",
      "dst_ip": "192.168.5.99",
      "dst_port": 162,
      "inc_port": 1515,
      "protocol": "TCP",
      "src_interface": "NETZONE1"
     }
    ]
   }
  },
  "logging": {
   "remote": {
    "address": "192.168.1.254",
    "ca": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAlc5enlYQSFe-
KohrV0cjaOOmnC1NgnSCENMp0Yt16iKtUuYSI\nLI3xxrBmmeaYcRvWpuy3WDUYrHPMgIyWdmpF
XhxxK2oO3g1eqsNKnvYQAXUQeIdS\nbbMZejfwsgrsFo0gK3dU9AXZe20FCGdfnmzhfrmVNlIZAMJ
ZhwSz2RvbsQss2gPF\nHddJC6nHzsmrEnoEQN+Z0173N9OhUQKG5WSZOPsOKDfIHLBvFxsmm
6oisTZM4g+w\n9eXG4EhWVfkxJmFUMWXa+nFm37Px1eTDFEW5hpJC1/SPUPEO51/nhrAtxre/FR
Rg\nAuh26x/D8+t/cAxtLpB2eht21Cfo/I9F7kwdywIDAQABAoIBAGeSsgpo2IMbu2bO\nhOyxIGde7D
ZBZDfepmlVXFiKZlCdnEtTnU6oqPPFPHrFWrpBFAx+91hOBYwd19M8\n4D5oxSMHKRtqDXNq7Pv
FYA89ct1/EW8zqELeJAxDJvAB6y7ATfCfZaX9cVsLigJA\nbnS7NMCIEOjopA7JUFqwXqQJxb/GOm
DrENr9eTC4fp8elCgDF+gyBY6bcc19L4ab\nFDF5fcAb+mRFG4GNE1NIToBi5R5bZchjwt1wp174HA
4XyY4cBS+9COMON74MFM6t\nWvnxACzh0UCB7PvrONgSj0yZ9UfZ3OvVYXURxdxtKN8G3LYYR
PSEFumjFY/sTWFA\nW7+xHwkCgYEAxkFBu+RozZ1m39EMNjYSrs2YIRJLDKwxv3Fr0erFw5fM78
SZ9wAo\nig5EzqprD+qAitnee4rAvDZajdeYnH/gREco6ca6bCx37ksMewCDtMo6SlWSRieY\nuEHKz
gGa8/DWp04FXgpYwhkRwye32cJucDUtPBxlLQi66nSYBJ9RAW8CgYEAwXCP\n/bLG6wleRaI+f61
T54lhu8qr1R2vvcWnCrH85EyB64Q8YBJDBKSzmSSuKF5U9swN\nXsqLhQHx7KPkouwoZcQFcidL
ur+Bww/kXldujAzTX7OsEegsSEcQXafyVrxI4Ela\nhCV7YOtTiilF6iM3/cWigmuIFp+8fdGlm/cxw2UCg
YBcaEBOZslexXYU7qiFgDC3\nH4dAKvmmP4C0nhZGcuqZH2FbhMTK91zt9Han6ZEbiw89KQ3lga
gSUjdlE8/DamtL\nB+wPAx0TnKqN/JclofjBxNzklvwmDQDHKYtw+BiUiXZT5y7jRWlXIz3LO/Ea4+B8\
nFp0t/oI+Omp9K7lLtkKYqwKBgBjronFBpeTDuTRqSJS0RLnwdfnWe1qiT3C4VPPI\nyFa1ElvB5nFO
CPpBKa4SDqm+tV1yHkrW9zB0drFQz/S5Td8GaNky/MubPmFd28LX\nqrM6N8T9ha5s5b+OACrAp
zTLXuweJx4dIg7zYjjLZmlqjh0QaAY7SjX38DWZW6eD\nKhMNAoGBAKD7QKKk7UoVTbocOlTdxak
5DUTmO5NPbnoHo3aj5rq57v1STutHNi2w\nZiYHgDGvlflyHzwU2MEIXV0S3ZEVI76kaffZn7Nmyhc
6ByibbbqRmyDqzTqnwzSR\ntLUEox056XsJRfKrBNhj0e9utJ1wLrPmAF7EqqCT1+2IXmGbTFU3\n--
---END RSA PRIVATE KEY-----\n-----BEGIN CERTIFICATE-----\nMIIDhTCCAm2gAwIBAgIBFzANBg-
kqhkiG9w0BAQsFADB9MQswCQYDVQQGEwJERTEN\nMAsGA1UECBMEdGVzdDENMAsGA1U
EBxMEdGVzdDENMAsGA1UEChMEdGVzdDENMAsG\nA1UECxMEdGVzdDEVMBMGA1UEAxM
Mc2VydmVyLmlwLmRlMRswGQYJKoZIhvcNAQkB\nFgx0ZXN0QHRlc3QuZGUwHhcNMjAxMTAyM
TAyNDAwWhcNMjExMTAyMTAyNDAwWjB9\nMQswCQYDVQQGEwJERTENMAsGA1UECBMEd
GVzdDENMAsGA1UEBxMEdGVzdDENMAsG\nA1UEChMEdGVzdDENMAsGA1UECxMEdGVzdD
EVMBMGA1UEAxMMc2VydmVyLmlwLmRl\nMRswGQYJKoZIhvcNAQkBFgx0ZXN0QHRlc3QuZG
UwggEiMA0GCSqGSIb3DQEBAQUA\nA4IBDwAwggEKAoIBAQCVzl6eVhBIV4qiGtXRyNo46acLU2
CdIIQ0ynRi3XqIq1S5\nhIgsjfHGsGaZ5phxG9am7LdYNRisc8yAjJZ2akVeHHErag7eDV6qw0qe9hA
BdRB4\nh1Jtsxl6N/CyCuwWjSArd1T0Bdl7bQUIZ1+ebOF+uZU2UhkAwlmHBLPZG9uxCyza\nA8Ud
10kLqcfOyasSegRA35nTXvc306FRAoblZJk4+w4oN8gcsG8XGyabqiKxNkzi\nD7D15cbgSFZV+TE
mYVQxZdr6cWbfs/HV5MMURbmGkkLX9I9Q8Q7nX+eGsC3Gt78V\nFGAC6HbrH8Pz639wDG0ukH
Z6G3bUJ+j8j0XuTB3LAgMBAAGjEDAOMAwGA1UdEwQF\nMAMBAf8wDQYJKoZIhvcNAQELBQA
DggEBACYKsvmIu0Yqb+YBrXGbpCm36S0dfgms\n74KbIqYTKRrx2aMQc7HAhyJgCbnZPrZ/reDHb
sMjAvhMc+uXmuDbsamlvP90G80E\nj/2eCKafcPbvnqI1mU4eV7VcjDlkqN2x3NTAUcRHTWssFoIG
g5DYW0vN1KjKjjly\nHEaFW71o6iQwxWWrC5gJKP+t6HZ8sfJKvGT2jHIOuLwqI3WUsas5DTh5pyu
bGxQS\nb6ngF3YV/t/PuC43i3UkYcGtczrVLrA3WJB1Eyncu6kMQKJp87+bCUlY2ajn1twc\nkx1HCr9
vXeTBolubJgsPfeDEYEihsBHbrlhRRcNBO4EZfY4LMebN820=\n-----END CERTIFICATE-----\n",
    "port": 514,
    "protocol": "TLS",
   "status": "ON"
  }
 },
 "network": {
  "mode": "ROUTER",
  "nat": {
   "1_1_nat": [
    {
     "comment": "",
     "id": 0,
     "real_network": "192.168.180.0/24",
     "virt_network": "192.168.5.0/24"
    }
   ],
   "masquerading": [
    {
     "from_ip": "0.0.0.0/0",
     "id": 0,
     "outgoing_on_if": "NETZONE1"
    }
   ]
  },
  "netzone1": {
   "address": "192.168.178.57",
   "mode": "DHCP",
   "netmask": 24
  },
  "netzone2": {
   "address": "192.168.1.1",
   "netmask": 24
  },
  "routing": {
   "gateway": "192.168.178.1",
   "routes": [
    {
     "comment": "Route to Machine Net 2",
     "gateway": "192.168.1.1",
     "network": "192.168.5.0/24"
    }
   ]
  },
  "stealth": {
   "management_address": "192.168.178.57",
   "management_gateway": "192.168.178.1",
   "management_netmask": 24
  }
  },
  "service": {
    "dhcp_server": {
      "dns": "192.168.1.1",
      "gateway": "192.168.1.1",
      "lease_time": "12h",
      "netmask": 24,
      "range_high": "192.168.1.254",
      "range_low": "192.168.1.2",
      "status": "ON",
      "wins_server": ""
    },
    "dnscache": {
      "allowed_requests": [
        "NETZONE2"
      ],
      "dns_servers": "USER_DEFINED",
      "log": "OFF",
      "user_defined": []
    },
    "ntp": {
      "allow_client_requests": [
        "NETZONE2"
      ],
      "server": [
        {
          "address": "0.pool.ntp.org",
          "comment": "",
          "port": 123
        },
        {
          "address": "1.pool.ntp.org",
          "comment": "",
          "port": 123
        },
        {
          "address": "2.pool.ntp.org",
          "comment": "",
          "port": 123
        },
        {
          "address": "3.pool.ntp.org",
          "comment": "",
          "port": 123
        }
      ],
      "status": "ON"
    },
    "snmp": {
      "allow_requests_from": [
        "NETZONE2"
      ],
      "ro_community_string": "public",
      "status_v2c": "ON",
      "status_v3": "ON",
      "user": {
        "username": "snmp-user-mGuardNT"
      }
    },
    "web": {
      "session_timeout": 90,
      "user_blocking_time": 30,
      "user_max_failed_logins": 4
    }
  },
  "system": {
    "hostname": "OldName",
    "store_config_on_sdcard": "OFF",
    "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."
  },
  "zoneinfo": "Europe/Berlin"
},
"envelope": {
  "identifier": {
    "contentID": "66e8539e",
    "functionalID": "59d3ff2b"
  },
  "version": 1
},
"error": [],
"schemes": [
  {
    "name": "common.4710ab60",
    "url": "/v1/configuration/scheme/common.4710ab60"
  },
  {
    "name": "common.types.f0bf23da",
    "url": "/v1/configuration/scheme/common.types.f0bf23da"
  },
  {
    "name": "configuration.fileinfo.b3afd1b0",
    "url": "/v1/configuration/scheme/configuration.fileinfo.b3afd1b0"
  },
  {
    "name": "configuration.firewall.62d07c99",
    "url": "/v1/configuration/scheme/configuration.firewall.62d07c99"
  },
  {
    "name": "configuration.logging.fce1b9ba",
    "url": "/v1/configuration/scheme/configuration.logging.fce1b9ba"
  },
  {
    "name": "configuration.network.0edde642",
    "url": "/v1/configuration/scheme/configuration.network.0edde642"
  },
  {
    "name": "configuration.service.1f00d993",
    "url": "/v1/configuration/scheme/configuration.service.1f00d993"
  },
  {
    "name": "configuration.system.ef2e081a",
    "url": "/v1/configuration/scheme/configuration.system.ef2e081a"
  },
  {
    "name": "configuration.zoneinfo.e8437e00",
    "url": "/v1/configuration/scheme/configuration.zoneinfo.e8437e00"
  }
],
"status": 0
}
```

4.3 POST Request (Endpoint "configuration")

(For the answer to the POST Request; see below „“Response:”“.)

POST Request:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X
POST https://192.168.1.1:443/api/v1/configuration -d '
{
  "content": {
    "fileinfo": {
      "devtype": "0001010111020000",
      "firmware": "1.8.0"
    },
    "firewall": {
      "forward": {
        "ftp_allow_field": "ON",
        "log_all_matches": "ON",
        "log_policy": "ON",
        "sanity_check": "ON",
        "stealth_allow_dhcp": "ON",
        "tables": [
          {
            "in_netzone": "NETZONE1",
            "out_netzone": "NETZONE2",
            "rules": [
              {
                "comment": "",
                "dst_network": "192.168.1.20",
                "dst_port": "ALL",
                "id": 0,
                "log": "OFF",
                "protocol": "ALL",
                "src_network": "0.0.0.0/0",
                "verdict": "REJECT"
              },
              {
                "comment": "Office",
                "dst_network": "0.0.0.0/0",
                "dst_port": "ALL",
                "id": 1,
                "log": "ON",
                "protocol": "ALL",
                "src_network": "192.168.2.0/24",
                "verdict": "ACCEPT"
              },
              {
                "comment": "Production",
                "dst_network": "192.168.1.0/24",
                "dst_port": "ALL",
                "id": 2,
                "log": "ON",
                "protocol": "ALL",
                "src_network": "10.10.0.0/24",
                "verdict": "ACCEPT"
              }
            ]
          },
          {
            "in_netzone": "NETZONE2",
            "out_netzone": "NETZONE1",
            "rules": [
              {
                "comment": "",
                "dst_network": "0.0.0.0/0",
                "dst_port": "ALL",
                "id": 0,
                "log": "OFF",
                "protocol": "ALL",
                "src_network": "0.0.0.0/0",
                "verdict": "ACCEPT"
              }
            ]
          }
        ],
        "testmode": "ON"
      },
      "input": {
        "log_all_matches": "OFF",
        "log_policy": "OFF",
        "rules": [
          {
            "id": 1,
            "log": "OFF",
            "service": "HTTPS",
            "source": "NETZONE1",
            "verdict": "ACCEPT"
```

```
        }
      ]
    },
    "port_forward": {
      "rules": [
        {
          "comment": "",
          "dst_ip": "192.168.5.99",
          "dst_port": 162,
          "inc_port": 1515,
          "protocol": "TCP",
          "src_interface": "NETZONE1"
        }
      ]
    }
  },
  "logging": {
    "remote": {
      "address": "192.168.1.254",
      "ca": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAlc5enlYQSFeKohrV0cjaOOmnC1NgnSCENMp0Yt16iKtUuYSI\nLI3xxrBmmeaYcRvWpuy3WDUYrHPMgIyWdmpF
XhxxK2oO3g1eqsNKnvYQAXUQeIdS\nbbMZejfwsgrsFo0gK3dU9AXZe20FCGdfnmzhfrmVNlIZAMJ
ZhwSz2RvbsQss2gPF\nHddJC6nHzsmrEnoEQN+Z0173N9OhUQKG5WSZOPsOKDflHLBvFxsmm
6oisTZM4g+w\n9eXG4EhWVfkxJmFUMWXa+nFm37Px1eTDFEW5hpJC1/SPUPEO51/nhrAtxre/FR
Rg\nAuh26x/D8+t/cAxtLpB2eht21Cfo/I9F7kwdywIDAQABAoIBAGeSsgpo2IMbu2bO\nhOyxIGde7D
ZBZDfepmlVXFiKZlCdnEtTnU6oqPPFPHrFWrpBFAx+91hOBYwd19M8\n4D5oxSMHKRtqDXNq7Pv
FYA89ct1/EW8zqELeJAxDJvAB6y7ATfCfZaX9cVsLigJA\nbnS7NMCIEOjopA7JUFqwXqQJxb/GOm
DrENr9eTC4fp8elCgDF+gyBY6bcc19L4ab\nFDF5fcAb+mRFG4GNE1NIToBi5R5bZchjwt1wp174HA
4XyY4cBS+9COMON74MFM6t\nWvnxACzh0UCB7PvrONgSj0yZ9UfZ3OvVYXURxdxtKN8G3LYYR
PSEFumjFY/sTWFA\nW7+xHwkCgYEAxkFBu+RozZ1m39EMNjYSrs2YIRJLDKwxv3Fr0erFw5fM78
SZ9wAo\nig5EzqprD+qAitnee4rAvDZajdeYnH/gREco6ca6bCx37ksMewCDtMo6SlWSRieY\nuEHKz
gGa8/DWp04FXgpYwhkRwye32cJucDUtPBxlLQi66nSYBJ9RAW8CgYEAwXCP\n/bLG6wleRal+f61
T54lhu8qr1R2vvcWnCrH85EyB64Q8YBJDBKSzmSSuKF5U9swN\nXsqLhQHx7KPkouwoZcQFcidL
ur+Bww/kXldujAzTX7OsEegsSEcQXafyVrxI4Ela\nhCV7YOtTiilF6iM3/cWigmuIFp+8fdGlm/cxw2UCg
YBcaEBOZslexXYU7qiFgDC3\nH4dAKvmmP4C0nhZGcuqZH2FbhMTK91zt9Han6ZEbiw89KQ3lga
gSUjdlE8/DamtL\nB+wPAx0TnKqN/JclofjBxNzklvwmDQDHKYtw+BiUiXZT5y7jRWlXIz3LO/Ea4+B8\
nFp0t/oI+Omp9K7lLtkKYqwKBgBjronFBpeTDuTRqSJS0RLnwdfnWe1qiT3C4VPPI\nyFa1ElvB5nFO
CPpBKa4SDqm+tV1yHkrW9zB0drFQz/S5Td8GaNky/MubPmFd28LX\nqrM6N8T9ha5s5b+OACrAp
zTLXuweJx4dIg7zYjjLZmlqjh0QaAY7SjX38DWZW6eD\nKhMNAoGBAKD7QKKk7UoVTbocOlTdxak
5DUTmO5NPbnoHo3aj5rq57v1STutHNi2w\nZiYHgDGvIfIyHzwU2MEIXV0S3ZEVI76kaffZn7Nmyhc
6ByibbbqRmyDqzTqnwzSR\ntLUEox056XsJRfKrBNhj0e9utJ1wLrPmAF7EqqCT1+2IXmGbTFU3\n--
---END RSA PRIVATE KEY-----\n-----BEGIN CERTIFICATE-----\nMIIDhTCCAm2gAwIBAgIBFzANBgkqhkiG9w0BAQsFADB9MQswCQYDVQQGEwJERTEN\nMAsGA1UECBMEdGVzdDENMAsGA1U
EBxMEdGVzdDENMAsGA1UEChMEdGVzdDENMAsG\nA1UECxMEdGVzdDEVMBMGA1UEAxM
Mc2VydmVyLmlwLmRlMRswGQYJKoZIhvcNAQkB\nFgx0ZXN0QHRlc3QuZGUwHhcNMjAxMTAyM
TAyNDAwWhcNMjExMTAyMTAyNDAwWjB9\nMQswCQYDVQQGEwJERTENMAsGA1UECBMEd
GVzdDENMAsGA1UEBxMEdGVzdDENMAsG\nA1UEChMEdGVzdDENMAsGA1UECxMEdGVzdD
EVMBMGA1UEAxMMc2VydmVyLmlwLmRl\nMRswGQYJKoZIhvcNAQkBFgx0ZXN0QHRlc3QuZG
UwggEiMA0GCSqGSIb3DQEBAQUA\nA4IBDwAwggEKAoIBAQCVzl6eVhBIV4qiGtXRyNo46acLU2
CdIIQ0ynRi3XqIq1S5\nhIgsjfHGsGaZ5phxG9am7LdYNRisc8yAjJZ2akVeHHErag7eDV6qw0qe9hA
BdRB4\nh1Jtsxl6N/CyCuwWjSArd1T0Bdl7bQUIZ1+ebOF+uZU2UhkAwlmHBLPZG9uxCyza\nA8Ud
10kLqcfOyasSegRA35nTXvc306FRAoblZJk4+w4oN8gcsG8XGyabqiKxNkzi\nD7D15cbgSFZV+TE
```

```
mYVQxZdr6cWbfs/HV5MMURbmGkkLX9I9Q8Q7nX+eGsC3Gt78V\nFGAC6HbrH8Pz639wDG0ukH
Z6G3bUJ+j8j0XuTB3LAgMBAAGjEDAOMAwGA1UdEwQF\nMAMBAf8wDQYJKoZIhvcNAQELBQA
DggEBACYKsvmIu0Yqb+YBrXGbpCm36S0dfgms\n74KbIqYTKRrx2aMQc7HAhyJgCbnZPrZ/reDHb
sMjAvhMc+uXmuDbsamlvP90G80E\nj/2eCKafcPbvnqI1mU4eV7VcjDlkqN2x3NTAUcRHTWssFolG
g5DYW0vN1KjKjjly\nHEaFW71o6iQwxWWrC5gJKP+t6HZ8sfJKvGT2jHIOuLwqI3WUsas5DTh5pyu
bGxQS\nb6ngF3YV/t/PuC43i3UkYcGtczrVLrA3WJB1Eyncu6kMQKJp87+bCUlY2ajn1twc\nkx1HCr9
vXeTBolubJgsPfeDEYEihsBHbrlhRRcNBO4EZfY4LMebN820=\n-----END CERTIFICATE-----\n",
    "port": 514,
    "protocol": "TLS",
    "status": "ON"
   }
  },
  "network": {
   "mode": "ROUTER",
   "nat": {
    "1_1_nat": [
     {
      "comment": "",
      "id": 0,
      "real_network": "192.168.180.0/24",
      "virt_network": "192.168.5.0/24"
     }
    ],
    "masquerading": [
     {
      "from_ip": "0.0.0.0/0",
      "id": 0,
      "outgoing_on_if": "NETZONE1"
     }
    ]
   },
   "netzone1": {
    "address": "192.168.178.57",
    "mode": "DHCP",
    "netmask": 24
   },
   "netzone2": {
    "address": "192.168.1.1",
    "netmask": 24
   },
   "routing": {
    "gateway": "192.168.178.1",
    "routes": [
     {
      "comment": "Route to Machine Net 2",
      "gateway": "192.168.1.1",
      "network": "192.168.5.0/24"
     }
    ]
   },
   "stealth": {
    "management_address": "192.168.178.57",
    "management_gateway": "192.168.178.1",
    "management_netmask": 24
   }
  },
  "service": {
   "dhcp_server": {
    "dns": "192.168.1.1",
    "gateway": "192.168.1.1",
    "lease_time": "12h",
    "netmask": 24,
    "range_high": "192.168.1.254",
    "range_low": "192.168.1.2",
    "status": "ON",
    "wins_server": ""
   },
   "dnscache": {
    "allowed_requests": [
     "NETZONE2"
    ],
    "dns_servers": "USER_DEFINED",
    "log": "OFF",
    "user_defined": []
   },
   "ntp": {
    "allow_client_requests": [
     "NETZONE2"
    ],
    "server": [
     {
      "address": "0.pool.ntp.org",
      "comment": "",
      "port": 123
     },
     {
      "address": "1.pool.ntp.org",
      "comment": "",
            "port": 123
          },
          {
            "address": "2.pool.ntp.org",
            "comment": "",
            "port": 123
          },
          {
            "address": "3.pool.ntp.org",
            "comment": "",
            "port": 123
          }
        ],
        "status": "ON"
      },
      "snmp": {
        "allow_requests_from": [
          "NETZONE2"
        ],
        "ro_community_string": "public",
        "status_v2c": "ON",
        "status_v3": "ON",
        "user": {
          "username": "snmp-user-mGuardNT"
        }
      },
      "web": {
        "session_timeout": 90,
        "user_blocking_time": 30,
        "user_max_failed_logins": 4
      }
    },
    "system": {
      "hostname": "NewName",
      "store_config_on_sdcard": "OFF",
      "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only.
Any intrusion and its attempt without permission is illegal and strictly prohibited."
    },
    "zoneinfo": "Europe/Berlin"
  },
"envelope":  {"version": 1}}'
```

Response:

```
{
 "content": {
  "fileinfo": {
   "devtype": "0001010111020000",
   "firmware": "1.8.0"
  },
  "firewall": {
   "forward": {
    "ftp_allow_field": "ON",
    "log_all_matches": "ON",
    "log_policy": "ON",
    "sanity_check": "ON",
    "stealth_allow_dhcp": "ON",
    "tables": [
     {
      "in_netzone": "NETZONE1",
      "out_netzone": "NETZONE2",
      "rules": [
       {
        "comment": "",
        "dst_network": "192.168.1.20",
        "dst_port": "ALL",
        "id": 0,
        "log": "OFF",
        "protocol": "ALL",
        "src_network": "0.0.0.0/0",
        "verdict": "REJECT"
       },
       {
        "comment": "Office",
        "dst_network": "0.0.0.0/0",
        "dst_port": "ALL",
        "id": 1,
        "log": "ON",
        "protocol": "ALL",
        "src_network": "192.168.2.0/24",
        "verdict": "ACCEPT"
       },
       {
        "comment": "Production",
        "dst_network": "192.168.1.0/24",
        "dst_port": "ALL",
            "id": 2,
            "log": "ON",
            "protocol": "ALL",
            "src_network": "10.10.0.0/24",
            "verdict": "ACCEPT"
          }
        ]
      },
      {
        "in_netzone": "NETZONE2",
        "out_netzone": "NETZONE1",
        "rules": [
          {
            "comment": "",
            "dst_network": "0.0.0.0/0",
            "dst_port": "ALL",
            "id": 0,
            "log": "OFF",
            "protocol": "ALL",
            "src_network": "0.0.0.0/0",
            "verdict": "ACCEPT"
          }
        ]
      }
    ],
    "testmode": "ON"
  },
  "input": {
    "log_all_matches": "OFF",
    "log_policy": "OFF",
    "rules": [
      {
        "id": 1,
        "log": "OFF",
        "service": "HTTPS",
        "source": "NETZONE1",
        "verdict": "ACCEPT"
      }
    ]
  },
  "port_forward": {
    "rules": [
      {
            "comment": "",
            "dst_ip": "192.168.5.99",
            "dst_port": 162,
            "inc_port": 1515,
            "protocol": "TCP",
            "src_interface": "NETZONE1"
          }
        ]
      }
    },
    "logging": {
      "remote": {
        "address": "192.168.1.254",
        "ca": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAlc5enlYQSFe-
KohrV0cjaOOmnC1NgnSCENMp0Yt16iKtUuYSI\nLI3xxrBmmeaYcRvWpuy3WDUYrHPMgIyWdmpF
XhxxK2oO3g1eqsNKnvYQAXUQeIdS\nbbMZejfwsgrsFo0gK3dU9AXZe20FCGdfnmzhfrmVNlIZAMJ
ZhwSz2RvbsQss2gPF\nHddJC6nHzsmrEnoEQN+Z0173N9OhUQKG5WSZOPsOKDflHLBvFxsmm
6oisTZM4g+w\n9eXG4EhWVfkxJmFUMWXa+nFm37Px1eTDFEW5hpJC1/SPUPEO51/nhrAtxre/FR
Rg\nAuh26x/D8+t/cAxtLpB2eht21Cfo/I9F7kwdywIDAQABAoIBAGeSsgpo2IMbu2bO\nhOyxIGde7D
ZBZDfepmlVXFiKZlCdnEtTnU6oqPPFPHrFWrpBFAx+91hOBYwd19M8\n4D5oxSMHKRtqDXNq7Pv
FYA89ct1/EW8zqELeJAxDJvAB6y7ATfCfZaX9cVsLigJA\nbnS7NMCIEOjopA7JUFqwXqQJxb/GOm
DrENr9eTC4fp8elCgDF+gyBY6bcc19L4ab\nFDF5fcAb+mRFG4GNE1NIToBi5R5bZchjwt1wp174HA
4XyY4cBS+9COMON74MFM6t\nWvnxACzh0UCB7PvrONgSj0yZ9UfZ3OvVYXURxdxtKN8G3LYYR
PSEFumjFY/sTWFA\nW7+xHwkCgYEAxkFBu+RozZ1m39EMNjYSrs2YIRJLDKwxv3Fr0erFw5fM78
SZ9wAo\nig5EzqprD+qAitnee4rAvDZajdeYnH/gREco6ca6bCx37ksMewCDtMo6SlWSRieY\nuEHKz
gGa8/DWp04FXgpYwhkRwye32cJucDUtPBxlLQi66nSYBJ9RAW8CgYEAwXCP\n/bLG6wleRal+f61
T54lhu8qr1R2vvcWnCrH85EyB64Q8YBJDBKSzmSSuKF5U9swN\nXsqLhQHx7KPkouwoZcQFcidL
ur+Bww/kXldujAzTX7OsEegsSEcQXafyVrxI4Ela\nhCV7YOtTiilF6iM3/cWigmuIFp+8fdGlm/cxw2UCg
YBcaEBOZslexXYU7qiFgDC3\nH4dAKvmmP4C0nhZGcuqZH2FbhMTK91zt9Han6ZEbiw89KQ3lga
gSUjdlE8/DamtL\nB+wPAx0TnKqN/JclofjBxNzklvwmDQDHKYtw+BiUiXZT5y7jRWlXlz3LO/Ea4+B8\
nFp0t/oI+Omp9K7lLtkKYqwKBgBjronFBpeTDuTRqSJS0RLnwdfnWe1qiT3C4VPPI\nyFa1ElvB5nFO
CPpBKa4SDqm+tV1yHkrW9zB0drFQz/S5Td8GaNky/MubPmFd28LX\nqrM6N8T9ha5s5b+OACrAp
zTLXuweJx4dIg7zYjjLZmlqjh0QaAY7SjX38DWZW6eD\nKhMNAoGBAKD7QKKk7UoVTbocOlTdxak
5DUTmO5NPbnoHo3aj5rq57v1STutHNi2w\nZiYHgDGvlfIyHzwU2MEIXV0S3ZEVI76kaffZn7Nmyhc
6ByibbbqRmyDqzTqnwzSR\ntLUEox056XsJRfKrBNhj0e9utJ1wLrPmAF7EqqCT1+2IXmGbTFU3\n--
---END RSA PRIVATE KEY-----\n-----BEGIN CERTIFICATE-----\nMIIDhTCCAm2gAwIBAgIBFzANBg-
kqhkiG9w0BAQsFADB9MQswCQYDVQQGEwJERTEN\nMAsGA1UECBMEdGVzdDENMAsGA1U
EBxMEdGVzdDENMAsGA1UEChMEdGVzdDENMAsG\nA1UECxMEdGVzdDEVMBMGA1UEAxM
Mc2VydmVyLmlwLmRlMRswGQYJKoZIhvcNAQkB\nFgx0ZXN0QHRlc3QuZGUwHhcNMjAxMTAyM
TAyNDAwWhcNMjExMTAyMTAyNDAwWjB9\nMQswCQYDVQQGEwJERTENMAsGA1UECBMEd
GVzdDENMAsGA1UEBxMEdGVzdDENMAsG\nA1UEChMEdGVzdDENMAsGA1UECxMEdGVzdD
EVMBMGA1UEAxMMc2VydmVyLmlwLmRl\nMRswGQYJKoZIhvcNAQkBFgx0ZXN0QHRlc3QuZG
UwggEiMA0GCSqGSIb3DQEBAQUA\nA4IBDwAwggEKAoIBAQCVzl6eVhBIV4qiGtXRyNo46acLU2
CdIIQ0ynRi3XqIq1S5\nhIgsjfHGsGaZ5phxG9am7LdYNRisc8yAjJZ2akVeHHErag7eDV6qw0qe9hA
BdRB4\nh1JtsxI6N/CyCuwWjSArd1T0Bdl7bQUIZ1+ebOF+uZU2UhkAwlmHBLPZG9uxCyza\nA8Ud
10kLqcfOyasSegRA35nTXvc306FRAoblZJk4+w4oN8gcsG8XGyabqiKxNkzi\nD7D15cbgSFZV+TE
mYVQxZdr6cWbfs/HV5MMURbmGkkLX9I9Q8Q7nX+eGsC3Gt78V\nFGAC6HbrH8Pz639wDG0ukH
Z6G3bUJ+j8j0XuTB3LAgMBAAGjEDAOMAwGA1UdEwQF\nMAMBAf8wDQYJKoZIhvcNAQELBQA
DggEBACYKsvmIu0Yqb+YBrXGbpCm36S0dfgms\n74KbIqYTKRrx2aMQc7HAhyJgCbnZPrZ/reDHb
sMjAvhMc+uXmuDbsamlvP90G80E\nj/2eCKafcPbvnqI1mU4eV7VcjDlkqN2x3NTAUcRHTWssFoIG
g5DYW0vN1KjKjjly\nHEaFW71o6iQwxWWrC5gJKP+t6HZ8sfJKvGT2jHIOuLwqI3WUsas5DTh5pyu
bGxQS\nb6ngF3YV/t/PuC43i3UkYcGtczrVLrA3WJB1Eyncu6kMQKJp87+bCUlY2ajn1twc\nkx1HCr9
vXeTBolubJgsPfeDEYEihsBHbrlhRRcNBO4EZfY4LMebN820=\n-----END CERTIFICATE-----\n",
        "port": 514,
        "protocol": "TLS",
        "status": "ON"
      }
    },
    "network": {
      "mode": "ROUTER",
      "nat": {
        "1_1_nat": [
          {
            "comment": "",
            "id": 0,
            "real_network": "192.168.180.0/24",
            "virt_network": "192.168.5.0/24"
          }
        ],
        "masquerading": [
          {
            "from_ip": "0.0.0.0/0",
            "id": 0,
            "outgoing_on_if": "NETZONE1"
          }
        ]
      },
      "netzone1": {
        "address": "192.168.178.57",
        "mode": "DHCP",
        "netmask": 24
      },
      "netzone2": {
        "address": "192.168.1.1",
        "netmask": 24
      },
      "routing": {
        "gateway": "192.168.178.1",
        "routes": [
          {
            "comment": "Route to Machine Net 2",
            "gateway": "192.168.1.1",
            "network": "192.168.5.0/24"
          }
        ]
      },
      "stealth": {
        "management_address": "192.168.178.57",
        "management_gateway": "192.168.178.1",
        "management_netmask": 24
      }
    },
    "service": {
      "dhcp_server": {
        "dns": "192.168.1.1",
        "gateway": "192.168.1.1",
        "lease_time": "12h",
        "netmask": 24,
        "range_high": "192.168.1.254",
        "range_low": "192.168.1.2",
        "status": "ON",
        "wins_server": ""
      },
      "dnscache": {
        "allowed_requests": [
          "NETZONE2"
        ],
        "dns_servers": "USER_DEFINED",
        "log": "OFF",
        "user_defined": []
      },
      "ntp": {
        "allow_client_requests": [
          "NETZONE2"
        ],
        "server": [
          {
            "address": "0.pool.ntp.org",
            "comment": "",
            "port": 123
          },
          {
            "address": "1.pool.ntp.org",
            "comment": "",
            "port": 123
          },
          {
            "address": "2.pool.ntp.org",
            "comment": "",
            "port": 123
```

```
          },
          {
            "address": "3.pool.ntp.org",
            "comment": "",
            "port": 123
          }
        ],
        "status": "ON"
      },
      "snmp": {
        "allow_requests_from": [
          "NETZONE2"
        ],
        "ro_community_string": "public",
        "status_v2c": "ON",
        "status_v3": "ON",
        "user": {
          "username": "snmp-user-mGuardNT"
        }
      },
      "web": {
        "session_timeout": 90,
        "user_blocking_time": 30,
        "user_max_failed_logins": 4
      }
    },
    "system": {
      "hostname": "NewName",
      "store_config_on_sdcard": "OFF",
      "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."
    },
    "zoneinfo": "Europe/Berlin"
  },
  "envelope": {
    "identifier": {
      "contentID": "d311f50a",
      "functionalID": "ec2a59bf"
    },
    "version": 1
  },
  "error": [],
  "schemes": [
    {
      "name": "common.4710ab60",
      "url": "/v1/configuration/scheme/common.4710ab60"
    },
    {
      "name": "common.types.f0bf23da",
      "url": "/v1/configuration/scheme/common.types.f0bf23da"
    },
    {
      "name": "configuration.fileinfo.b3afd1b0",
      "url": "/v1/configuration/scheme/configuration.fileinfo.b3afd1b0"
    },
    {
      "name": "configuration.firewall.62d07c99",
      "url": "/v1/configuration/scheme/configuration.firewall.62d07c99"
    },
    {
      "name": "configuration.logging.fce1b9ba",
      "url": "/v1/configuration/scheme/configuration.logging.fce1b9ba"
    },
    {
      "name": "configuration.network.0edde642",
      "url": "/v1/configuration/scheme/configuration.network.0edde642"
    },
    {
      "name": "configuration.service.1f00d993",
      "url": "/v1/configuration/scheme/configuration.service.1f00d993"
    },
    {
      "name": "configuration.system.ef2e081a",
      "url": "/v1/configuration/scheme/configuration.system.ef2e081a"
    },
    {
      "name": "configuration.zoneinfo.e8437e00",
      "url": "/v1/configuration/scheme/configuration.zoneinfo.e8437e00"
    }
  ],
  "status": 0
}
```

4.4 POST Request (Endpoint "actions/migration")

POST Request:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type: application/json" -X
POST https://192.168.1.1:443/api/v1/actions/migration -d '
{"content": {
  "fileinfo": {
    "devtype": "0001010111020000",
    "firmware": "1.5.1"
  },
  "firewall": {
    "forward": {
      "log_all_matches": "OFF",
      "log_policy": "OFF",

[...]

},"envelope":  {"version": 1}}'
```

Response:

```
{
 "content": {
  "fileinfo": {
    "devtype": "0001010111020000",
    "firmware": "1.8.0"
  },
  "firewall": {
    "forward": {
      "ftp_allow_field": "OFF",
      "log_all_matches": "OFF",
      "log_policy": "OFF",
      "sanity_check": "ON",
      "stealth_allow_dhcp": "ON",
      "tables": [
       {
         "in_netzone": "NETZONE2",
         "out_netzone": "NETZONE1",
         "rules": []
       },
       {
         "in_netzone": "NETZONE1",
         "out_netzone": "NETZONE2",
         "rules": []
       }
    ],
    "testmode": "ON"
   },
   "input": {
    "rules": [
     {
      "id": 0,
      "log": "OFF",
      "service": "HTTPS",
      "source": "NETZONE2",
      "verdict": "ACCEPT"
     },
     {
      "id": 1,
      "log": "OFF",
      "service": "HTTPS",
      "source": "NETZONE1",
      "verdict": "ACCEPT"
     }
    ]
   },
   "port_forward": {
    "rules": []
   }
  },
  "logging": {
   "remote": {
    "address":
"a.123456ddddd7890dddddddddddddddddddddddd12345678901234567890.AAA1234567890123
45678901234567890123456789012345678901234567890.AAA1234567890123456789012345678
901234567890123456789012345678901234567890.AAA1234567890123456789012345678901234567890123
45678901234567890",
    "port": 513,
    "protocol": "UDP",
    "status": "ON"
   }
  },
  "network": {
   "mode": "ROUTER",
   "nat": {
    "1_1_nat": [],
    "masquerading": [
     {
      "from_ip": "0.0.0.0/0",
        "id": 0,
        "outgoing_on_if": "NETZONE1"
      }
    ]
  },
  "netzone1": {
    "mode": "DHCP"
  },
  "netzone2": {
    "address": "10.1.1.1",
    "netmask": 24
  },
  "routing": {
    "routes": []
  },
  "stealth": {
    "management_address": "192.168.1.1",
    "management_gateway": "192.168.1.254",
    "management_netmask": 24
  }
},
"service": {
  "dhcp_server": {
    "dns": "192.168.1.1",
    "gateway": "192.168.1.1",
    "lease_time": "12h",
    "netmask": 24,
    "range_high": "192.168.1.254",
    "range_low": "192.168.1.2",
    "status": "OFF",
    "wins_server": ""
  },
  "dnscache": {
    "allowed_requests": [],
    "dns_servers": "USER_DEFINED",
    "log": "OFF",
    "user_defined": [
      {
        "comment": "",
        "ip": "212.2.220.212"
      }
    ]
  },
  "ntp": {
    "allow_client_requests": [
      "NETZONE2",
      "NETZONE1"
    ],
    "server": [
      {
        "address": "0.pool.ntp.org",
        "comment": "",
        "port": 123
      },
      {
        "address": "1.pool.ntp.org",
        "comment": "",
        "port": 123
      },
      {
        "address": "2.pool.ntp.org",
        "comment": "",
        "port": 123
      },
      {
        "address": "1.2.3.4",
        "comment": "",
        "port": 123
      }
    ],
    "status": "OFF"
  },
  "snmp": {
    "allow_requests_from": [
      "NETZONE2"
    ],
    "ro_community_string": "public",
    "status_v2c": "OFF",
    "status_v3": "OFF"
  },
  "web": {
    "session_timeout": 90,
    "user_blocking_time": 30,
    "user_max_failed_logins": 4
  }
},
    "system": {
      "hostname": "mGuard",
      "store_config_on_sdcard": "OFF",
      "usenotification": "The usage of this mGuard security appliance is reserved to authorized staff only. Any intrusion and its attempt without permission is illegal and strictly prohibited."
    },
    "zoneinfo": "Europe/Berlin"
  },
  "envelope": {
    "identifier": {
      "contentID": "5cc1731e",
      "functionalID": "d78399a9"
    },
    "version": 1
  },
  "error": [],
  "schemes": [],
  "status": 0
}
```

4.5 GET Request (Endpoint: "users")

GET-Request:

curl -k -b *session_cookie* - XGET https://192.168.1.1:443/api/v1/users

Response:

```
{
 "content": {
  "ldap": {
   "ldap_server": {
    "base_dn": "DC=mguard,DC=management",
    "ca": "-----BEGIN CERTIFICATE-----\nMIIDmzCCAoOgAwIBAgIU-
WYcWnmC15gUbcfq6Zx7c9MgYviEwDQYJKoZIhvcNAQEL\nBQAwXTELMAkGA1UEBhMCQlkxCz
AJBgNVBAgMAkJZMQ4wDAYDVQQHDAVNaW5zazEM\nMAoGA1UECgwDU0FNMQswCQYDVQ
QLDAJRQTEWMBQGA1UEAwwNMTkyLjE2OC4xLjEx\nNTAeFw0yMDEwMjkxMTE0NDBaFw0zM
DEwMjcxMTE0NDBaMF0xCzAJBgNVBAYTAkJZ\nMQswCQYDVQQIDAJCWTEOMAwGA1UEBww
FTWluc2sxDDAKBgNVBAoMA1NBTTELMAkG\nA1UECwwCUUExFjAUBgNVBAMMDTE5Mi4xNjg
uMS4xMTUwggEiMA0GCSqGSIb3DQEB\nAQUAA4IBDwAwggEKAoIBAQCyY2f6XAZoRkv2wIRv8
LQfXs+rkhxLQsy62oQcmMPt\nwVkg3NAgC69t3ESk91zFUZvhE7Of2NJbFQmtfJlUZlJWhYNg4gVR
28X/VrsKgkps\npzqemiKmj4aWWvk9+8IjPpvdng9TP5F4zTDF3W3Xy3v3THr3YixY80LqMHbPNFp
O\n7GnGe7YQMrWt3rZFkSEG3k3q4nTS8znPUS78qE96GAgspxLllcsdVKe6/9K8yYSb\nv5l0L6r8c
Cj+zel3EV9UxatyC1hGbZjcO+QfwNhz/nJb+5HOF6Kpxexl6rsIe/28\njE9LadvXAl+DDiX2gcStGj0Lw
9h7Uuu3hDkQVezyLKzrAgMBAAGjUzBRMB0GA1Ud\nDgQWBBSqPqzTnykG0FHJdijV7WeJLC5B
GzAfBgNVHSMEGDAWgBSqPqzTnykG0FHJ\ndijV7WeJLC5BGzAPBgNVHRMBAf8EBTADAQH/M
A0GCSqGSIb3DQEBCwUAA4IBAQBv\n4vnhipL0JOOoLwNsp6vW9Gzx9nVlkdSmlD3e6zqg5m2HlI
NbCvIf1fxMtKq5m+cR\n1tnb3fNUjp+Au30B/iPQD9LFaX0458XinOxYpyQcKRWDrXLgnMfSixUv96G
NQzoZ\ndjLI3O8IDFU0GsitQNAfepyH94+GDSsP2oKdAPTIUO5jgPKM5deSqeh0qCND8rhW\nYN6
viunYRKz/9y9pDDM6iLkBwZpjAzjj1e17tB06QPkrfwOn5ofYY0vcqRK6LsnF\nBW5/87JeogTAN2iLD
gVIIVuSe9+Q/Wm+okFObiIbECoh2L6zqojLwpp8GEqv3NhD\nwLYiT0JjajXDGLAf0t4O\n-----END
CERTIFICATE-----\n",
    "hostname": "192.168.2.100",
    "port": 389,
    "tls": "ON",
    "username": "admin_ldap"
   },
   "status": "ON",
   "user_role_mapping": {
    "admin": "Role_2",
    "audit": "Role_3",
    "ldap_attribute": "Role",
    "super_admin": "Role_1"
   }
  },
  "user_mgmt": {
   "current_user": "admin",
   "users": [
    {
     "block_user": "OFF",
     "name": "",
     "old_username": "admin",
     "role": "SUPERADMIN",
     "username": "admin"
    },
    {
     "block_user": "OFF",
     "name": "",
     "old_username": "admin_production",
     "role": "ADMIN",
     "username": "admin_production"
    }
   ]
  }
 },
 "envelope": {
  "identifier": {
   "contentID": "4b7a11b1",
   "functionalID": "4b7a11b1"
  },
  "version": 1
 },
 "error": [],
 "schemes": [
  {
   "name": "users.manageusers.e52f65cd",
   "url": "/v1/users/scheme/users.manageusers.e52f65cd"
  }
 ],
 "status": 0
}
```

4.6 POST Request (Endpoint "users")

POST Request:

```
curl -k -b session_cookie -H "X-CSRF-Token: <TOKEN>" -H "Content-Type:application/json" -X POST
https://192.168.1.1:443/api/v1/users -d '
{
 "content": {
  "ldap": {
   "ldap_server": {
    "base_dn": "DC=mguard,DC=management",
    "ca": "-----BEGIN CERTIFICATE-----\nMIIDmzCCAoOgAwIBAgIU-
WYcWnmC15gUbcfq6Zx7c9MgYviEwDQYJKoZIhvcNAQEL\nBQAwXTELMAkGA1UEBhMCQlkxCz
AJBgNVBAgMAkJZMQ4wDAYDVQQHDAVNaW5zazEM\nMAoGA1UECgwDU0FNMQswCQYDVQ
QLDAJRQTEWMBQGA1UEAwwNMTkyLjE2OC4xLjEx\nNTAeFw0yMDEwMjkxMTE0NDBaFw0zM
DEwMjcxMTE0NDBaMF0xCzAJBgNVBAYTAkJZ\nMQswCQYDVQQIDAJCWTEOMAwGA1UEBww
FTWluc2sxDDAKBgNVBAoMA1NBTTELMAkG\nA1UECwwCUUExFjAUBgNVBAMMDTE5Mi4xNjg
uMS4xMTUwggEiMA0GCSqGSIb3DQEB\nAQUAA4IBDwAwggEKAoIBAQCyY2f6XAZoRkv2wIRv8
LQfXs+rkhxLQsy62oQcmMPt\nwVkg3NAgC69t3ESk91zFUZvhE7Of2NJbFQmtfJlUZlJWhYNg4gVR
28X/VrsKgkps\npzqemiKmj4aWWvk9+8IjPpvdng9TP5F4zTDF3W3Xy3v3THr3YixY80LqMHbPNFp
O\n7GnGe7YQMrWt3rZFkSEG3k3q4nTS8znPUS78qE96GAgspxLllcsdVKe6/9K8yYSb\nv5l0L6r8c
Cj+zel3EV9UxatyC1hGbZjcO+QfwNhz/nJb+5HOF6Kpxexl6rsIe/28\njE9LadvXAI+DDiX2gcStGj0Lw
9h7Uuu3hDkQVezyLKzrAgMBAAGjUzBRMB0GA1Ud\nDgQWBBSqPqzTnykG0FHJdijV7WeJLC5B
GzAfBgNVHSMEGDAWgBSqPqzTnykG0FHJ\ndijV7WeJLC5BGzAPBgNVHRMBAf8EBTADAQH/M
A0GCSqGSIb3DQEBCwUAA4IBAQBv\n4vnhipL0JOOoLwNsp6vW9Gzx9nVlkdSmlD3e6zqg5m2HlI
NbCvIf1fxMtKq5m+cR\n1tnb3fNUjp+Au30B/iPQD9LFaX0458XinOxYpyQcKRWDrXLgnMfSixUv96G
NQzoZ\ndjLI3O8IDFU0GsitQNAfepyH94+GDSsP2oKdAPTIUO5jgPKM5deSqeh0qCND8rhW\nYN6
viunYRKz/9y9pDDM6iLkBwZpjAzjj1e17tB06QPkrfwOn5ofYY0vcqRK6LsnF\nBW5/87JeogTAN2iLD
gVIIVuSe9+Q/Wm+okFObiIbECoh2L6zqojLwpp8GEqv3NhD\nwLYiT0JjajXDGLAf0t4O\n-----END
CERTIFICATE-----\n",
    "hostname": "192.168.2.100",
    "password": "ldap_server_password",
    "port": 389,
    "tls": "ON",
    "username": "admin_ldap"
   },
   "status": "ON",
   "user_role_mapping": {
    "admin": "Role_2",
    "audit": "Role_3",
    "ldap_attribute": "Role",
    "super_admin": "Role_1"
   }
  },
  "user_mgmt": {
   "current_user": "admin", "old_password": "private",
   "users": [
    {
     "block_user": "OFF",
     "name": "",
     "old_username": "admin",
     "role": "SUPERADMIN",
     "username": "superadmin"
    },
    {
     "block_user": "OFF",
     "name": "",
     "old_username": "admin_production",
     "role": "ADMIN",
     "username": "admin_production",
     "new_password": "secret_production_password",
     "repeat_password": "secret_production_password"
    },
{
     "block_user": "OFF",
     "name": "",
     "old_username": "",
     "role": "AUDIT",
     "username": "secret_audit_production",
     "new_password": "secret_audit_password",
     "repeat_password": "secret_audit_password"

    }
   ]
  }
 },
 "envelope": {"version": 1}}'
```

5 Appendix

5.1 Available time zones

In alphabetic order:

Africa

Africa/Abidjan, Africa/Accra, Africa/Addis_Ababa, Africa/Algiers, Africa/Asmara, Africa/Asmera, Africa/Bamako, Africa/Bangui, Africa/Banjul, Africa/Bissau, Africa/Blantyre, Africa/Brazzaville, Africa/Bujumbura, Africa/Cairo, Africa/Casablanca, Africa/Ceuta, Africa/Conakry, Africa/Dakar, Africa/Dar_es_Salaam, Africa/Djibouti, Africa/Douala, Africa/El_Aaiun, Africa/Freetown, Africa/Gaborone, Africa/Harare, Africa/Johannesburg, Africa/Juba, Africa/Kampala, Africa/Khartoum, Africa/Kigali, Africa/Kinshasa, Africa/Lagos, Africa/Libreville, Africa/Lome, Africa/Luanda, Africa/Lubumbashi, Africa/Lusaka, Africa/Malabo, Africa/Maputo, Africa/Maseru, Africa/Mbabane, Africa/Mogadishu, Africa/Monrovia, Africa/Nairobi, Africa/Ndjamena, Africa/Niamey, Africa/Nouakchott, Africa/Ouagadougou, Africa/Porto-Novo, Africa/Sao_Tome, Africa/Timbuktu, Africa/Tripoli, Africa/Tunis, Africa/Windhoek

America

America/Adak, America/Anchorage, America/Anguilla, America/Antigua, America/Araguaina, America/Argentina/Buenos_Aires, America/Argentina/Catamarca, America/Argentina/ComodRivadavia, America/Argentina/Cordoba, America/Argentina/Jujuy, America/Argentina/La_Rioja, America/Argentina/Mendoza, America/Argentina/Rio_Gallegos, America/Argentina/Salta, America/Argentina/San_Juan, America/Argentina/San_Luis, America/Argentina/Tucuman, America/Argentina/Ushuaia, America/Aruba, America/Asuncion, America/Atikokan, America/Atka, America/Bahia, America/Bahia_Banderas, America/Barbados, America/Belem, America/Belize, America/Blanc-Sablon, America/Boa_Vista, America/Bogota, America/Boise, America/Buenos_Aires, America/Cambridge_Bay, America/Campo_Grande, America/Cancun, America/Caracas, America/Catamarca, America/Cayenne, America/Cayman, America/Chicago, America/Chihuahua, America/Coral_Harbour, America/Cordoba, America/Costa_Rica, America/Creston, America/Cuiaba, America/Curacao, America/Danmarkshavn, America/Dawson, America/Dawson_Creek, America/Denver, America/Detroit, America/Dominica, America/Edmonton, America/Eirunepe, America/El_Salvador, America/Ensenada, America/Fort_Nelson, America/Fort_Wayne, America/Fortaleza, America/Glace_Bay, America/Godthab, America/Goose_Bay, America/Grand_Turk, America/Grenada, America/Guadeloupe, America/Guatemala, America/Guayaquil, America/Guyana, America/Halifax, America/Havana, America/Hermosillo, America/Indiana/Indianapolis, America/Indiana/Knox, America/Indiana/Marengo, America/Indiana/Petersburg, America/Indiana/Tell_City, America/Indiana/Vevay, America/Indiana/Vincennes, America/Indiana/Winamac, America/Indianapolis, America/Inuvik, America/Iqaluit, America/Jamaica, America/Jujuy, America/Juneau, America/Kentucky/Louisville, America/Kentucky/Monticello, America/Knox_IN, America/Kralendijk, America/La_Paz, America/Lima, America/Los_Angeles, America/Louisville, America/Lower_Princes, America/Maceio, America/Managua, America/Manaus, America/Marigot, America/Martinique, America/Matamoros, America/Mazatlan, America/Mendoza, America/Menominee, America/Merida, America/Metlakatla, America/Mexico_City, America/Miquelon, America/Moncton, America/Monterrey, America/Montevideo, America/Montreal, America/Montserrat, America/Nassau, America/New_York, America/Nipigon, America/Nome, America/Noronha, America/North_Dakota/Beulah, America/North_Dakota/Center, America/North_Dakota/New_Salem, America/Ojinaga, America/Panama, America/Pangnirtung, America/Paramaribo, America/Phoenix, America/Port-au-Prince, America/Port_of_Spain, America/Porto_Acre, America/Porto_Velho, America/Puerto_Rico, America/Punta_Are-

	nas, America/Rainy_River, America/Rankin_Inlet, America/Recife, America/Regina, America/Resolute, America/Rio_Branco, America/Rosario, America/Santa_Isabel, America/Santarem, America/Santiago, America/Santo_Domingo, America/Sao_Paulo, America/Scoresbysund, America/Shiprock, America/Sitka, America/St_Barthelemy, America/St_Johns, America/St_Kitts, America/St_Lucia, America/St_Thomas, America/St_Vincent, America/Swift_Current, America/Tegucigalpa, America/Thule, America/Thunder_Bay, America/Tijuana, America/Toronto, America/Tortola, America/Vancouver, America/Virgin, America/Whitehorse, America/Winnipeg, America/Yakutat, America/Yellowknife
Antarctica	Antarctica/Casey, Antarctica/Davis, Antarctica/DumontDUrville, Antarctica/Macquarie, Antarctica/Mawson, Antarctica/McMurdo, Antarctica/Palmer, Antarctica/Rothera, Antarctica/South_Pole, Antarctica/Syowa, Antarctica/Troll, Antarctica/Vostok, Arctic/Longyearbyen
Asia	Asia/Aden, Asia/Almaty, Asia/Amman, Asia/Anadyr, Asia/Aqtau, Asia/Aqtobe, Asia/Ashgabat, Asia/Ashkhabad, Asia/Atyrau, Asia/Baghdad, Asia/Bahrain, Asia/Baku, Asia/Bangkok, Asia/Barnaul, Asia/Beirut, Asia/Bishkek, Asia/Brunei, Asia/Calcutta, Asia/Chita, Asia/Choibalsan, Asia/Chongqing, Asia/Chungking, Asia/Colombo, Asia/Dacca, Asia/Damascus, Asia/Dhaka, Asia/Dili, Asia/Dubai, Asia/Dushanbe, Asia/Famagusta, Asia/Gaza, Asia/Harbin, Asia/Hebron, Asia/Ho_Chi_Minh, Asia/Hong_Kong, Asia/Hovd, Asia/Irkutsk, Asia/Istanbul, Asia/Jakarta, Asia/Jayapura, Asia/Jerusalem, Asia/Kabul, Asia/Kamchatka, Asia/Karachi, Asia/Kashgar, Asia/Kathmandu, Asia/Katmandu, Asia/Khandyga, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Kuala_Lumpur, Asia/Kuching, Asia/Kuwait, Asia/Macao, Asia/Macau, Asia/Magadan, Asia/Makassar, Asia/Manila, Asia/Muscat, Asia/Nicosia, Asia/Novokuznetsk, Asia/Novosibirsk, Asia/Omsk, Asia/Oral, Asia/Phnom_Penh, Asia/Pontianak, Asia/Pyongyang, Asia/Qatar, Asia/Qostanay, Asia/Qyzylorda, Asia/Rangoon, Asia/Riyadh, Asia/Saigon, Asia/Sakhalin, Asia/Samarkand, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Srednekolymsk, Asia/Taipei, Asia/Tashkent, Asia/Tbilisi, Asia/Tehran, Asia/Tel_Aviv, Asia/Thimbu, Asia/Thimphu, Asia/Tokyo, Asia/Tomsk, Asia/Ujung_Pandang, Asia/Ulaanbaatar, Asia/Ulan_Bator, Asia/Urumqi, Asia/Ust-Nera, Asia/Vientiane, Asia/Vladivostok, Asia/Yakutsk, Asia/Yangon, Asia/Yekaterinburg, Asia/Yerevan
Atlantic	Atlantic/Azores, Atlantic/Bermuda, Atlantic/Canary, Atlantic/Cape_Verde, Atlantic/Faeroe, Atlantic/Faroe, Atlantic/Jan_Mayen, Atlantic/Madeira, Atlantic/Reykjavik, Atlantic/South_Georgia, Atlantic/St_Helena, Atlantic/Stanley
Australia	Australia/ACT, Australia/Adelaide, Australia/Brisbane, Australia/Broken_Hill, Australia/Canberra, Australia/Currie, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/LHI, Australia/Lindeman, Australia/Lord_Howe, Australia/Melbourne, Australia/NSW, Australia/North, Australia/Perth, Australia/Queensland, Australia/South, Australia/Sydney, Australia/Tasmania, Australia/Victoria, Australia/West, Australia/Yancowinna
Brazil	Brazil/Acre, Brazil/DeNoronha, Brazil/East, Brazil/West
CET/CST6CDT	CET, CST6CDT
Canada	Canada/Atlantic, Canada/Central, Canada/Eastern, Canada/Mountain, Canada/Newfoundland, Canada/Pacific, Canada/Saskatchewan, Canada/Yukon
Chile	Chile/Continental, Chile/EasterIsland
Cuba	Cuba

EET, EST, EST5EDT	EET, EST, EST5EDT
Egypt	Egypt, Eire
Etc	Etc/GMT, Etc/GMT+0, Etc/GMT+1, Etc/GMT+10, Etc/GMT+11, Etc/GMT+12, Etc/GMT+2, Etc/GMT+3, Etc/GMT+4, Etc/GMT+5, Etc/GMT+6, Etc/GMT+7, Etc/GMT+8, Etc/GMT+9, Etc/GMT-0, Etc/GMT-1, Etc/GMT-10, Etc/GMT-11, Etc/GMT-12, Etc/GMT-13, Etc/GMT-14, Etc/GMT-2, Etc/GMT-3, Etc/GMT-4, Etc/GMT-5, Etc/GMT-6, Etc/GMT-7, Etc/GMT-8, Etc/GMT-9, Etc/GMT0, Etc/Greenwich, Etc/UCT, Etc/UTC, Etc/Universal, Etc/Zulu
Europe	Europe/Amsterdam, Europe/Andorra, Europe/Astrakhan, Europe/Athens, Europe/Belfast, Europe/Belgrade, Europe/Berlin, Europe/Bratislava, Europe/Brussels, Europe/Bucharest, Europe/Budapest, Europe/Busingen, Europe/Chisinau, Europe/Copenhagen, Europe/Dublin, Europe/Gibraltar, Europe/Guernsey, Europe/Helsinki, Europe/Isle_of_Man, Europe/Istanbul, Europe/Jersey, Europe/Kaliningrad, Europe/Kiev, Europe/Kirov, Europe/Lisbon, Europe/Ljubljana, Europe/London, Europe/Luxembourg, Europe/Madrid, Europe/Malta, Europe/Mariehamn, Europe/Minsk, Europe/Monaco, Europe/Moscow, Europe/Nicosia, Europe/Oslo, Europe/Paris, Europe/Podgorica, Europe/Prague, Europe/Riga, Europe/Rome, Europe/Samara, Europe/San_Marino, Europe/Sarajevo, Europe/Saratov, Europe/Simferopol, Europe/Skopje, Europe/Sofia, Europe/Stockholm, Europe/Tallinn, Europe/Tirane, Europe/Tiraspol, Europe/Ulyanovsk, Europe/Uzhgorod, Europe/Vaduz, Europe/Vatican, Europe/Vienna, Europe/Vilnius, Europe/Volgograd, Europe/Warsaw, Europe/Zagreb, Europe/Zaporozhye, Europe/Zurich
Factory	Factory
GB	GB, GB-Eire
GMT	GMT, GMT+0, GMT-0, GMT0
Greenwich	Greenwich
HST	HST
Hongkong	Hongkong
Iceland	Iceland
Indian	Indian/Antananarivo, Indian/Chagos, Indian/Christmas, Indian/Cocos, Indian/Comoro, Indian/Kerguelen, Indian/Mahe, Indian/Maldives, Indian/Mauritius, Indian/Mayotte, Indian/Reunion
Iran	Iran, Israel, Jamaica, Japan, Kwajalein, Libya, MET, MST, MST7MDT, Mexico/BajaNorte, Mexico/BajaSur, Mexico/General, NZ, NZ-CHAT, Navajo, PRC, PST8PDT
Pacific	Pacific/Apia, Pacific/Auckland, Pacific/Bougainville, Pacific/Chatham, Pacific/Chuuk, Pacific/Easter, Pacific/Efate, Pacific/Enderbury, Pacific/Fakaofo, Pacific/Fiji, Pacific/Funafuti, Pacific/Galapagos, Pacific/Gambier, Pacific/Guadalcanal, Pacific/Guam, Pacific/Honolulu, Pacific/Johnston, Pacific/Kiritimati, Pacific/Kosrae, Pacific/Kwajalein, Pacific/Majuro, Pacific/Marquesas, Pacific/Midway, Pacific/Nauru, Pacific/Niue, Pacific/Norfolk, Pacific/Noumea, Pacific/Pago_Pago, Pacific/Palau, Pacific/Pitcairn, Pacific/Pohnpei, Pacific/Ponape, Pacific/Port_Moresby, Pacific/Rarotonga, Pacific/Saipan, Pacific/Samoa, Pacific/Tahiti, Pacific/Tarawa, Pacific/Tongatapu, Pacific/Truk, Pacific/Wake, Pacific/Wallis, Pacific/Yap

Poland	Poland
Portugal	Portugal
ROC	ROC
ROK	ROK
Singapore	Singapore
Turkey	Turkey
UCT	UCT
US	US/Alaska, US/Aleutian, US/Arizona, US/Central, US/East-Indiana, US/Eastern, US/Hawaii, US/Indiana-Starke, US/Michigan, US/Mountain, US/Pacific, US/Samoa
UTC	UTC
Universal	Universal
W-SU	W-SU
WET	WET
Zulu	Zulu

Please observe the following notes

General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

This manual, including all illustrations contained herein, is copyright protected. Any changes to the contents or the publication of extracts of this document is prohibited.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

How to contact us

Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:
phoenixcontact.com

Make sure you always use the latest documentation.
It can be downloaded at:
phoenixcontact.net/products

Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at phoenixcontact.com.

Published by

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
GERMANY

PHOENIX CONTACT Development and Manufacturing, Inc.
586 Fulling Mill Road
Middletown, PA 17057
USA

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:
tecdoc@phoenixcontact.com

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg, Germany
Phone: +49 5235 3-00
Fax: +49 5235 3-41200
E-mail: info@phoenixcontact.com
phoenixcontact.com

108898_en_11
Order No. —11