mGuard Application Notes FL/TC MGUARD

Application Note

AH EN MGUARD APPNOTES

PHOENIX CONTACT
INSPIRING INNOVATIONS

Application Note

mGuard Application Notes – FL/TC MGUARD

AH EN MGUARD APPNOTES, Revision 04 2019-10-23

This application note is valid for mGuard security appliances of the series FL/TC MGUARD.

108391_en_04

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 Labeling of warning notes

This symbol together with the **NOTE** signal word alerts the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

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

1.2 Qualification of users

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

– Qualified electricians 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.

2 Update and flash FL/TC MGUARD devices

Document-ID: 108250_en_06
Document-Description: AH EN MGUARD UPDATE

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

Contents of this document

The following chapters describe:

1. which mGuard firmware versions can be updated to mGuard 8.8.0
2. the files you need to update your mGuard device
3. how a firmware update is carried out
4. how the flash procedure is carried out

2.1 Introduction

The firmware on mGuard devices can be updated in different ways:

1. Local Update
2. Online Update
3. Automatic Update
4. Flashing the firmware

In the case of a **firmware update**, the existing configuration of the mGuard device remains unchanged.

Flashing an mGuard device deletes the existing configuration, including all passwords, and resets the device to the default status.

Updating to **mGuard firmware version 8.8.0** is described in detail for all mGuard devices in Chapters 1.5 to 1.14. Table 2-1 briefly lists the required update files.

2.2 Update to mGuard firmware version 8.8.0

An update to **mGuard firmware version 8.8.0** is only possible from **mGuard firmware version 8.6.1** or later.

If you want to update from a **firmware version < 8.6.1**, you must perform the update in several steps by first updating to version 8.6.1 (see “Update to mGuard firmware version 8.6.1” on page 13). In the next step you can update this version to version 8.8.0.

An update to firmware version 8.8.0 is only possible if the function "**Encrypted State Synchronization**" (menu *Redundancy*) has been deactivated before.

The update to **mGuard firmware version 8.8.0** is described in detail in chapters 1.5 to 1.14, depending on the device (see „Contents of this document“). Table 2-1 briefly lists the required update files.

Table 2-1 Updating mGuard firmware version from **8.6.1** or later to **8.8.0**: Required files

Devices	Local Update	Flashing the firmware
FL MGUARD RS2000 FL MGUARD RS4000 (TX/TX) (incl. variants -B, -P, -M)	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
FL MGUARD RS2005 FL MGUARD RS4004 (TX respectively TX/DTX)	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
TC MGUARD RS2000 3G VPN TC MGUARD RS4000 3G VPN	**Download file**: *Update_8.8.0_TC3G_MPC.zip* **Update file**: *gemalto.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC3G_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* *pxs8_03001_0100617.usf.xz.p7s*
TC MGUARD RS2000 4G VPN TC MGUARD RS4000 4G VPN	**Download file**: *Update_8.8.0_TC4G_MPC.zip* **Update file**: *huaweigeneric.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC4G_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* *ME909u-521_UPDATE_12.636.12.01.00.BIN.xz.p7s*
TC MGUARD RS2000/4000 4G VZW VPN	**Download file**: *Update_8.8.0_TC4G_VZW_MPC.zip* **Update file**: *HL7518.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC4G_VZW_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* *RHL75xx.4.04.142600.201801231340.x7160_1_signed_dwl.dwl.xz.p7s*

Table 2-1 Updating mGuard firmware version from **8.6.1** or later to **8.8.0**: Required files

TC MGUARD RS2000/4000 4G ATT VPN	**Download file**: *Update_8.8.0_TC4G_ATT_MPC.zip* **Update file**: *HL7588.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC4G_ATT_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* *RHL75xx.A.2.15.151600.201809201422.x7160_3_signed_DWL.dwl.xz.p7s*
FL MGUARD PCI(E)4000	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
FL MGUARD SMART2	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
FL MGUARD CENTERPORT	**Download file**: *Update_8.8.0_x86.zip* **Update file**: *update-8.{6-7}-8.8.0.default.x86_64.tar.gz*	**Download file**: *FW_X86_8.8.0.zip* **Update (flash) file**: *firmware.img.x86_64.p7s* *install.x86_64.p7s*
FL MGUARD GT/GT	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_GTGT_8.8.0* **Update (flash) file**: *jffs2.img.mpc83xx.p7s* *install.mpc83xx.p7s*
FL MGUARD DELTA TX/TX	**Download file**: *Update_8.8.0_MPC.zip* **Update file**: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.8.0.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*

2.3 Update to mGuard firmware version 8.6.1

Possible from **mGuard firmware Version 7.6.0**.

The update to mGuard firmware version 8.6.1 is performed in the same way as described in chapters 1.5 to 1.14 (see „Contents of this document“). Table 2-2 briefly lists the required update files depending on the source firmware version.

Table 2-2 Updating mGuard firmware version **7.6.0 or later** to **8.6.1**: Required files

Devices	Lokales Update	Flashen der Firmware
FL MGUARD RS2000 FL MGUARD RS4000 (TX/TX) (incl. variants -B, -P, -M)	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz* *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
FL MGUARD RS2005 FL MGUARD RS4004 (TX respectively TX/DTX)	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
TC MGUARD RS2000 3G VPN TC MGUARD RS4000 3G VPN	**Download file**: *Update_8.6.1_TC3G_MPC.zip* **Update files**: *gemalto.update-8.{4-5}-8.6.1.default.mpc83xx.tar.gz* *gemalto.update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC3G_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* pxs8_03001_0100617.usf.xz.p7s
TC MGUARD RS2000 4G VPN TC MGUARD RS4000 4G VPN	**Download file**: *Update_8.6.1_TC4G_MPC.zip* **Update files**: *huaweigeneric.update-8.{4-5}-8.6.1.default.mpc83xx.tar.gz* *huaweigeneric.update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_TC4G_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s* ME909u-521_UPDATE_12.636.12.01.00.BIN.xz.p7s
FL MGUARD PCI(E)4000	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz* *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*
FL MGUARD SMART2	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz* *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*

Table 2-2 Updating mGuard firmware version **7.6.0 or later** to **8.6.1**: Required files

FL MGUARD CENTERPORT	**Download file**: *Update_8.6.1_x86.zip* **Update files**: *update-7.{6}-8.6.1.default.x86_64.tar.gz* *update-8.{0-5}-8.6.1.default.x86_64.tar.gz* *update-8.{6}-8.6.1.default.x86_64.tar.gz*	**Download file**: *FW_X86_8.6.1.zip* **Update (flash) file**: *firmware.img.x86_64.p7s* *install.x86_64.p7s*
FL MGUARD GT/GT	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz* *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_GTGT_8.6.1* **Update (flash) file**: *jffs2.img.mpc83xx.p7s* *install.mpc83xx.p7s*
FL MGUARD DELTA TX/TX	**Download file**: *Update_8.6.1_MPC.zip* **Update files**: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz* *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz* *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*	**Download file**: *FW_MPC_8.6.1.zip* **Update (flash) file**: *ubifs.img.mpc83xx* *install-ubi.mpc83xx.p7s*

2.4 General information about mGuard updates

2.4.1 PHOENIX CONTACT Web Shop

The available update files for each mGuard device are provided for download on the product page in the PHOENIX CONTACT Web Shop under: phoenixcontact.net/products.

Depending on the installed firmware version, different files must be used for an update.

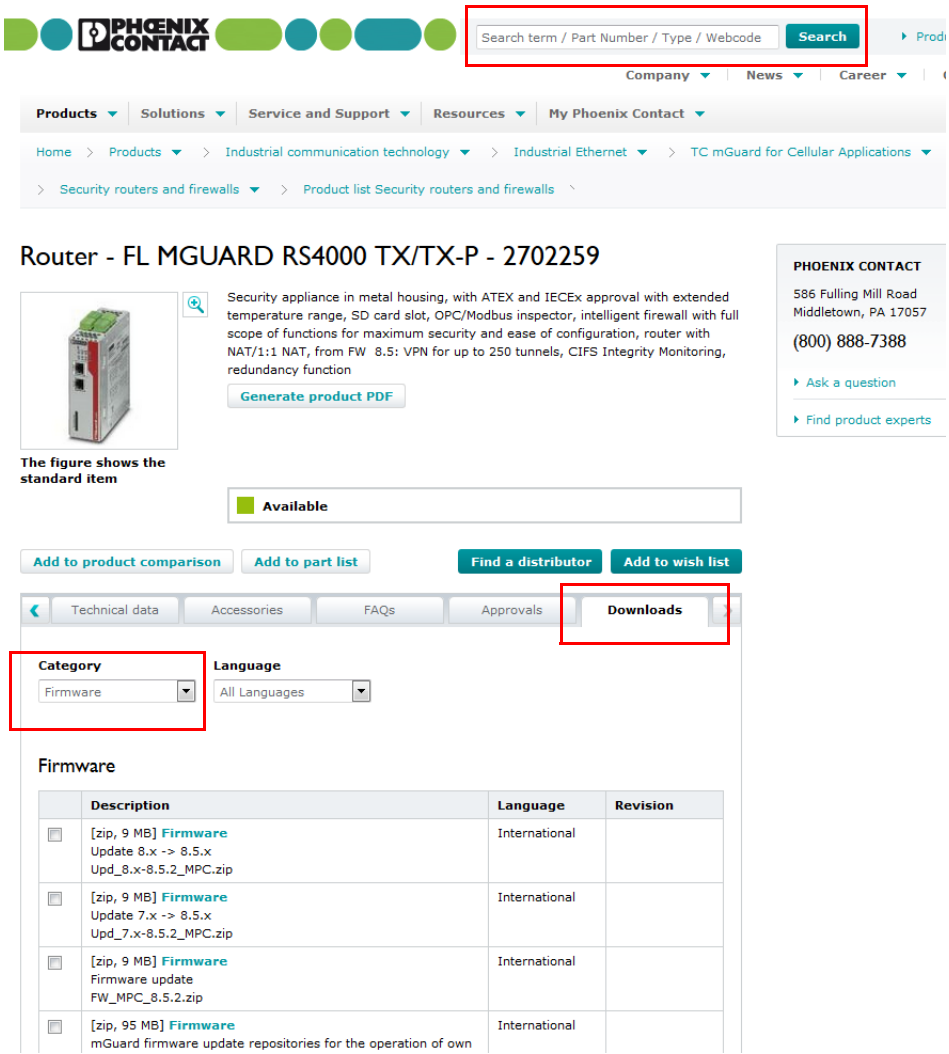


Figure 2-1 PHOENIX CONTACT Web Shop – Product page

2.4.2 Versioning: Major, Minor and Patch Releases

The following designations are used in the versioning of the mGuard firmware:

1. Major release (major version number)

   Major releases supplement the mGuard with new properties and contain mostly larger and more fundamental changes to the mGuard firmware . Their version number changes in the first digit position. Version **8**.6.1, for example, is a major release for Version **7**.6.8.

2. Minor release (minor version number)

   Minor releases supplement mGuard with new properties. Their version number changes in the second digit position. Version 8.**6.**0**,** for example, is a minor release for Version 8.**4.**2.

3. **Patch release** (troubleshooting)

   Patch releases resolve errors in previous versions and have a version number which only changes in the third digit position. Version 8.6.**1,** for example, is a patch release for Version 8.6.**0**.

2.4.3 Designation of the update files

The file that must be used to update your mGuard device depends on the firmware version installed on the device.

In the file name of the respective update file, it is indicated in **curly brackets** which firmware versions can be updated with this file.

Example "Local Update" RS4000

Using the update file "*update*-**8.{0-5}**-*8.6.1.default.mpc83xx.tar.gz",* firmware versions **8.0.0** to **8.5.3** can be updated to version 8.6.1.

In this case the download file is named "*Update_8.6.1_MPC.zip*".

Example "Online Update" RS4000

With the specification of the package set name "*update*-**7.{6}**-*8.6.1.default",* firmware versions **7.6.0** to **7.6.8** can be updated to version 8.6.1.

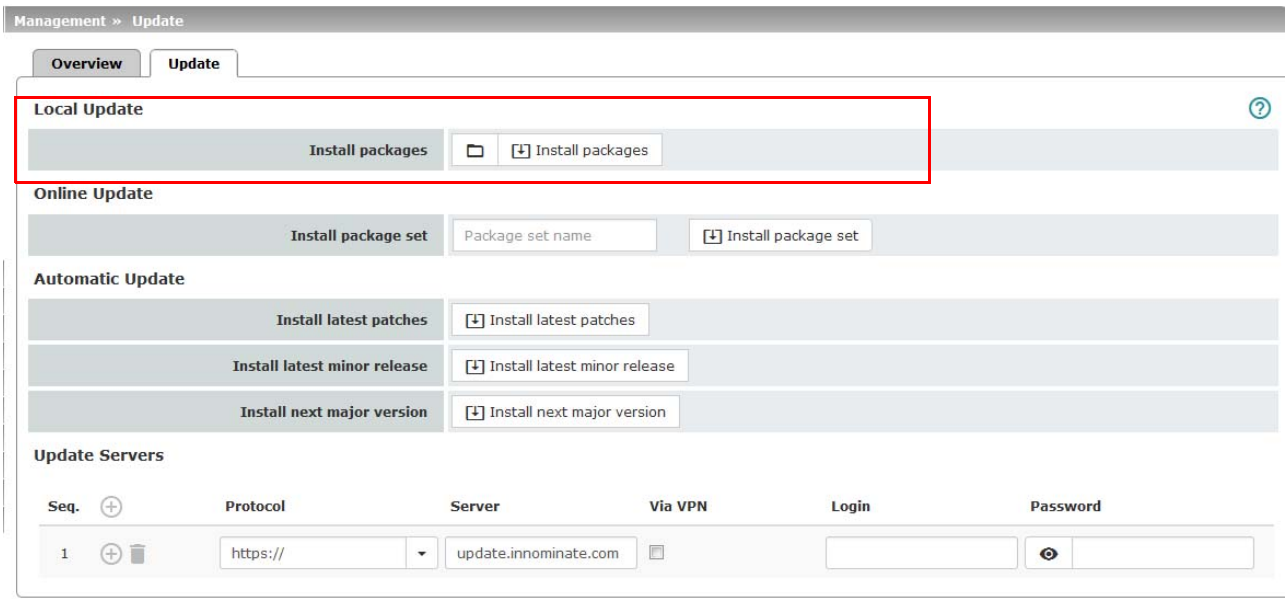
2.4.4 Description of the update procedure

NOTE: Do not interrupt the power supply of the mGuard device during the update process! Otherwise, the device could be damaged.

You can find more information on installation, operation and updates for mGuard devices in the firmware reference manual "UM EN MGUARD" and in the mGuard device manual "UM EN MGUARD DEVICES" available in the PHOENIX CONTACT Web Shop under phoenixcontact.net/products or help.mguard.com.

2.4.4.1 Local Update

The update file (*tar.gz* format) is loaded from the locally connected configuration computer onto the mGuard device and installed via the mGuard web interface (**Management >> Update >> Update**).



The firmware versions which can be updated with the update file are indicated in the file names of the update file in curly brackets.

Example (FL MGUARD RS4000):

Major release update: 7.6.8 to 8.6.1:

– Download file: *Update_8.6.1_MPC*
– Update file: *update-**7.{6}**-8.6.1.default.mpc83xx.tar.gz*

Minor release update: 8.**4**.2 to 8.6.1:

– Download file: *Update_8.6.1_MPC*
– Update file: *update-**8.{0-5}**-8.6.1.default.mpc83xx.tar.gz*

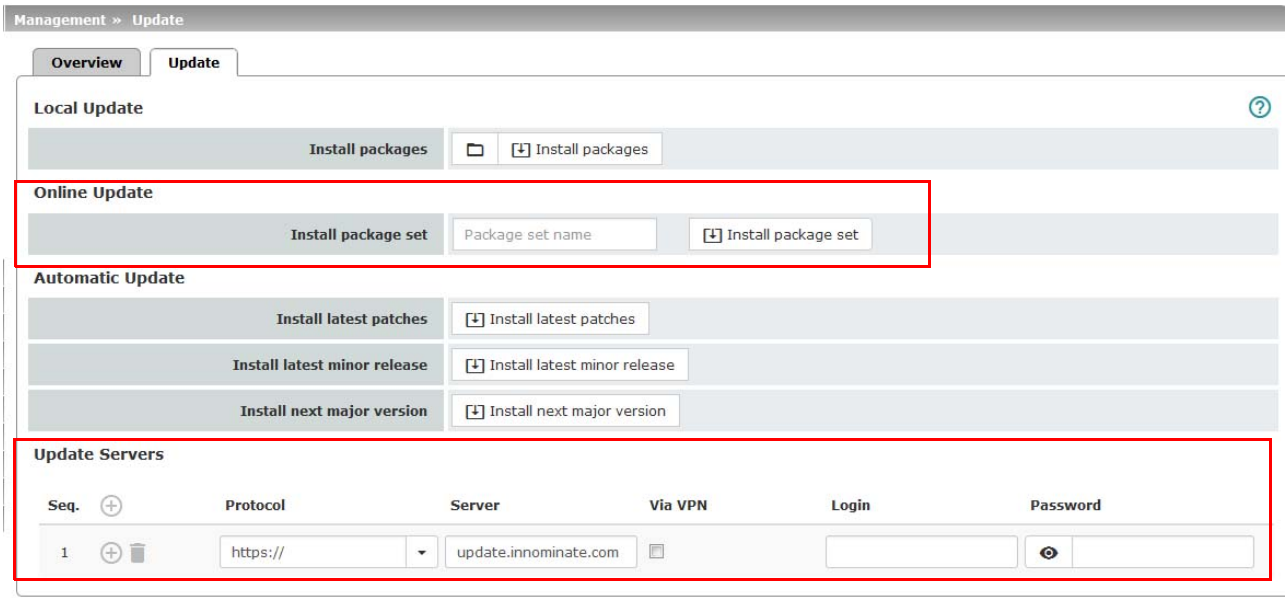
Patch release update: 8.6.**0** to 8.6.1:

– Download file: *Update_8.6.1_MPC*
– Update file: *update-**8.{6}**-8.6.1.default.mpc83xx.tar.gz*

2.4.4.2 Online Update

The update file is loaded from a configurable update server and installed.

The update is initialized through the request of a **package set** on the mGuard web interface (**Management >> Update >> Update**).



The firmware versions which can be updated by means of the selection of the package set name are indicated in the package set names in curly brackets.

Example (FL MGUARD RS4000):

Major release update: 7.6.8 to 8.6.1

– Package set name: *update-***7.{6}***-8.6.1.default*

Minor release update: 8.**4**.2 to 8.6.1

– Package set name: *update-***8.{0-5}***-8.6.1.default*

Patch release update: 8.6.**0** to 8.6.1

– Package set name: *update-***8.{6}***-8.6.1.default*

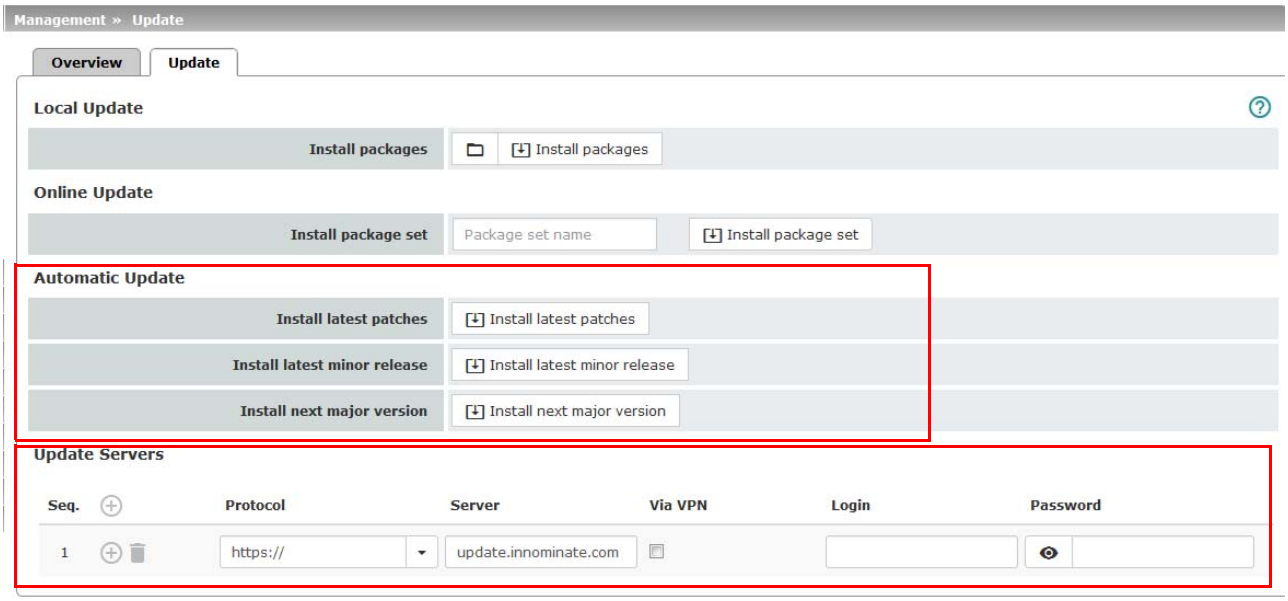
NOTE: Online or Automatic Updates from the installed source firmware version **7.6.8** can lead to an error (see note in Section “Setting up mGuard firmware update repositories” on page 78).

The login information (login + password) does not have to be specified if the update server which has been preset ex-works (https://update.innominate.com) is used.

2.4.4.3 Automatic Update

The update file is automatically determined from the selected update option and loaded and installed by a configurable update server.

The update is initialized via the mGuard web interface (**Management >> Update >> Update**) or the mGuard command line "*mg update*" .



Update options:

a) Install latest patches
b) Install latest minor release
c) Install next major release

NOTE: Online or Automatic Updates from the installed source firmware version **7.6.8** can lead to an error (see note in Section “Setting up mGuard firmware update repositories” on page 78).

It may occur that a **direct Automatic Update** to the current minor or the next major release is not possible from an installed firmware version. In this case, first perform one or more updates on authorized minor or patch releases. Afterwards, you can update to the current minor or the next major release in the last step.

The login information (login + password) does not have to be specified if the update server which has been preset ex-works (https://update.innominate.com) is used.

2.4.4.4 Flashing the firmware

The mGuard firmware is loaded from an SD card, USB flash memory (both with vfat file system) or from a TFTP update server, and installed onto the mGuard device.

Installed licenses remain on the device after flashing (in the case of devices with installed firmware version 5.0.0 or higher).

Configuration profiles and licenses can be installed and activated during the flash process (see “mGuard Flash Guide” on page 68).

NOTE: Flashing the firmware deletes all data, passwords and configurations on the mGuard device. The device is reset to its default setting. Save any existing configuration as a configuration profile at a safe location before flashing.

NOTE: Downgrading the pre-installed default firmware version is not supported.

For mGuard devices produced starting in January 2018, a *downgrade* of the pre-installed default firmware version to an earlier firmware version may fail. If this is the case, flash the device again with the firmware version that was originally installed or a higher version.

2.5 FL MGUARD RS2000/4000 TX/TX (incl. -B, -P, -M)

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.5.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.5.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD RS 4000).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.5.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.5.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.5.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.5.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.5.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.5.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.5.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.5.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_8.8.0.zip*

Update files (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

2.5.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD RS 4000).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the */Firmware* directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.5.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Sig* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod,* and *Sig* flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.6 FL MGUARD RS2005/4004 TX bzw. TX/DTX

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.6.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.6.1.1 Download update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD RS 4004).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.6.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.6.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.6.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.6.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.6.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.6.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.6.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.6.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_8.8.0.zip*

Update files (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

2.6.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD RS 4004).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the */Firmware* directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.6.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Info2* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod,* and *Info2* flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.7 TC MGUARD RS2000/4000 3G VPN

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

A **Local Update** to mGuard firmware Version **8.6.1** is possible from Version 8.4.0.

Online Update and **Automatic Update** are possible from version 8.0.0.

2.7.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_TC3G_MPC.zip*

Update files (= unpacked Zip file):

– *gemalto.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *gemalto.update-8.{4-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *gemalto.update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.7.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 3G).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_TC3G_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *gemalto.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.7.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *gemalto.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*.
5. Click the **Install packages** button to start the update.

2.7.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.7.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.7.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.7.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.7.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.7.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.7.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_TC3G_8.8.0.zip*

Update files, including modem firmware (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

– *pxs8_03001_0100617.usf.xz.p7s*

2.7.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 3G).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s* and *pxs8_03001_0100617.usf.xz.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the /Firmware directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.7.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from an TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Info2* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod* and *Info2* flash green simultaneously, the flash process has been concluded successfully (differs when uploading a configuration profile).
3. Restart the device.

2.8 TC MGUARD RS2000/4000 4G VPN

Order number: 2903588 (RS2000) / 2903586 (RS4000)

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.8.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_TC4G_MPC.zip*

Update files (= unpacked Zip file):

– *huaweigeneric.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

– (To 8.6.1: *huaweigeneric.update-8.{4-5}-8.6.1.default.mpc83xx.tar.gz*)

– (To 8.6.1: *huaweigeneric.update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.8.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_TC4G_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *huaweigeneric.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.8.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *huaweigeneric.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.8.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-8.{4-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.8.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.8.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.8.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.8.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.8.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.8.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_TC4G_8.8.0.zip*

Update files, including modem firmware (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

– *ME909u-521_UPDATE_12.636.12.01.00.BIN.xz.p7s*

2.8.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_TC4G_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s* and *ME909u-521_UPDATE_12.636.12.01.00.BIN.xz.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the /Firmware directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.8.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Info2* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod* and *Info2* flash green simultaneously, the flash process has been concluded successfully (differs when uploading a configuration profile).
3. Restart the device.

2.9 TC MGUARD RS2000/4000 4G VZW VPN

Order number: 1010462 (RS2000) / 1010461 (RS4000)

2.9.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_TC4G_VZW_MPC.zip*

Update files (= unpacked Zip file):

– *HL7418.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.9.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G *VZW VPN*).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_TC4G_VZW_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *HL7518.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.9.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *HL7518.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.9.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.9.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.9.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.9.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.9.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.9.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.9.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_TC4G_VZW_8.8.0.zip*

Update files, including modem firmware (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

– *RHL75xx.4.04.142600.201801231340.x7160_1_signed_dwl.dwl.xz.p7s*

2.9.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G VZW VPN).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_TC4G_VZW_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s* and RHL75xx.4.04.142600.201801231340.x7160_1_signed_dwl.dwl.xz.p7s) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the /Firmware directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.9.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Info2* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod* and *Info2* flash green simultaneously, the flash process has been concluded successfully (differs when uploading a configuration profile).
3. Restart the device.

2.10 TC MGUARD RS2000/4000 4G ATT VPN

Order number: 1010464 (RS2000) / 1010463 (RS4000)

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.10.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_TC4G_ATT_MPC.zip*

Update files (= unpacked Zip file):

– *HL7588.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.10.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G ATT VPN).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_TC4G_ATT_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *HL7588.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.10.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *HL7588.update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.10.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.10.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.10.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.10.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.10.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.10.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.10.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_TC4G_ATT_8.8.0.zip*

Update files, including modem firmware (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

– *RHL75xx.A.2.15.151600.201809201422.x7160_3_signed_DWL.dwl.xz.p7s*

2.10.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. TC MGUARD RS 4000 4G ATT VPN).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_TC4G_ATT_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s* and RHL75xx.A.2.15.151600.201809201422.x7160_3_signed_DWL.dwl.xz.p7s) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the /Firmware directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.10.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the *Stat*, *Mod*, and *Info2* LEDs light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the LEDs *Stat, Mod* and *Info2* flash green simultaneously, the flash process has been concluded successfully (differs when uploading a configuration profile).
3. Restart the device.

2.11 FL MGUARD PCI(E)4000

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.11.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.11.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD PCI4000).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.11.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.11.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.11.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.11.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.11.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.11.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.11.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.11.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_8.8.0.zip*

Update files (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

2.11.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD PCI4000).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the */Firmware* directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.11.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device: The two WAN LEDs and the upper LAN LED light up green simultaneously. Release the Reset button during this green light phase.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the two WAN LEDs and the upper LAN LED flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.12 FL MGUARD SMART2

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.12.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.12.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD SMART2).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.12.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.12.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.12.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.12.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.12.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.12.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.12.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.12.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_8.8.0.zip*

Update files (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

2.12.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD SMART2).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx, install-ubi.mpc83xx.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.12.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

To flash the firmware from a TFTP server, a TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until all three LEDs light up green.
   – The device starts the flash process: The device searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the TFTP server.
2. If all three LEDs flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.13 FL MGUARD CENTERPORT

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.13.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_X86.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.x68_64.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.x68_64.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.x68_64.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.x68_64.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.13.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD CENTERPORT).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_X86.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.x68_64.tar.gz* .

2.13.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.x68_64.tar.gz*
5. Click the **Install packages** button to start the update.

2.13.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.13.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.13.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.13.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.13.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.13.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.13.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_X86_8.8.0.zip*

Update files (= unpacked Zip file):

– *firmware.img.x86_64.p7s*

– *install.x86_64_p7s*

2.13.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD CENTERPORT).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_X86_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*firmware.img.x86_64.p7s*, *install.x86_64_p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the *Firmware* directory on the SD card or the USB flash drive.

2.13.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card / USB flash drive first. The firmware is only loaded from a TFTP server if no SD card / USB flash drive is found.

The TFTP server must be installed on the locally connected computer.

1. Connect a USB keyboard and a monitor to the device.
2. Restart the device.
3. As soon as the device boots, press one of the arrow keys on the USB keyboard several times until the boot process is interrupted: ↑, ↓, ← or →.
4. The boot menu is displayed.

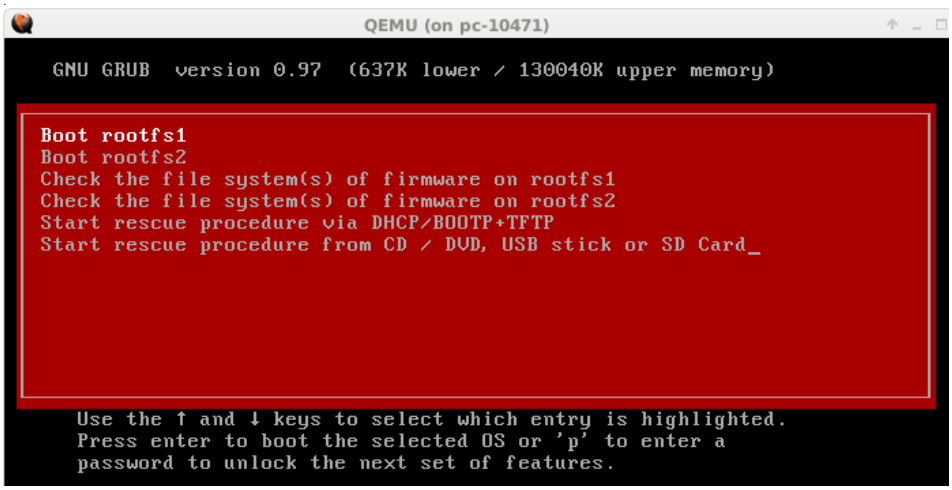


Figure 2-2 Boot menu

5. Select one of the two options to perform the flash procedure (rescue procedure) using the arrow keys ↓ or ↑:

– **Start rescue procedure via DHCP / BOOTP+TFTP**
– **Start rescue procedure from CD / DVD, USB stick or SD card**

To apply the selection, press the **Enter** key.

Start rescue procedure via DHCP / BootP+TFTP

Effect: The device downloads the necessary files from the TFTP server:

– *install.x86_64_p7s*
– *firmware.img.x86_64.p7s*

After the flash process concludes, the device is in the delivery state (default setting).

Start rescue procedure from CD/DVD, USB stick or SD card

General requirements:

1. A CD/DVD drive connected to the USB port or
2. A USB stick (USB Flash drive) connected to the USB port or
3. An SD memory card inserted into the SD card drive.
4. The necessary update files were copied onto the installation medium in the following directories:
   – */Firmware/install.x86_64_p7s*
   – */Firmware/firmware.img.x86_64.p7s*

Effect: After the flash process has been started by pressing the Enter key, the required data is downloaded from the selected medium. After the flash process concludes, the device is in the delivery state (default setting).

2.14 FL MGUARD GT/GT

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.14.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.14.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD GT/GT).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.14.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.14.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.14.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.14.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.14.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.14.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.14.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.14.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_GTGT_8.8.0.zip*

Update files (= unpacked Zip file):

– *jffs2.img.mpc83xx.p7s*

– *install.mpc83xx.p7s*

2.14.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD GT/GT).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_GTGT_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*jffs2.img.mpc83xx.p7s, install.mpc83xx.p7s*) from the directory *GTGT* into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server.

2.14.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

To flash the firmware from a TFTP server, a TFTP server must be installed on the locally connected computer.

1. Start the flash process by pressing the mode button (see ""Function selection by means of mode button (Smart mode)" on page 62").
   – The device searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the TFTP server.
2. If **05** is shown in the display, and the LEDs flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.14.4.3 Function selection by means of mode button (Smart mode)

Activate Smart mode

The Mode button is used to call/exit Smart mode and to select the desired function. The three mode LEDs indicate the mode that is currently set and the mode which will apply when exiting Smart mode.

Call up Smart mode

- Disconnect the device from the power supply.
- As soon as the supply voltage is switched on, hold down the Mode button for **more than ten seconds**. The three mode LEDs flash briefly three times and indicate that Smart mode is active.
- When Smart mode is started, the device is initially in the “Exit without changes” state (“51” in the display).

Select the desired setting

- To select the different settings, press the Mode button briefly and select the desired operating mode using a binary light pattern of the mode LEDs and a code on the 7-segment display.

Exit Smart mode and activating the selection

- To exit, press and hold down the Mode button for at least five seconds. The previously selected function is executed.

Possible functions in Smart mode

The device supports the selection of the following functions in Smart mode (see also example below):

Table 2-3 Functions in Smart mode

Function	7-segment display	ACT LED 1	SPD LED 2	FD LED 3
Exit Smart mode without changes	51	Off	Off	**On**
Activate the recovery procedure	55	**On**	Off	**On**
Activate the flash procedure	56	**On**	**On**	Off
Apply customized default profile	57	**On**	**On**	**On**

2.15 FL MGUARD DELTA TX/TX

An update to mGuard firmware Version 8.8.0 is possible from Version 8.6.1 or later.

If necessary, perform the update in two steps, by first updating Version < 8.6.1 to Version 8.6.1. In the next step, you can update this version to Version 8.8.0.

2.15.1 Local Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Required files (*depending on installed firmware version!***):**

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *Update_8.8.0_MPC.zip*

Update files (= unpacked Zip file):

– *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
– (To 8.6.1: *update-7.{6}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default.mpc83xx.tar.gz*)
– (To 8.6.1: *update-8.{6}-8.6.1.default.mpc83xx.tar.gz*)

The curly bracket indicates which installed source firmware versions can be updated with the update file (see Section 2.4.3).

2.15.1.1 Download the update file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD DELTA).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the **download file** *Update_8.8.0_MPC.zip.*
6. Unpack the Zip file.
7. Use the **update file** provided for the firmware version installed on your device (see Section 2.4.3):
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*

2.15.1.2 Install Local Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. In the **Local Update** section, click the **No file selected** symbol under **Install packages**.
4. Select the downloaded **update file**:
   – e. g. Minor update: *update-8.{6-7}-8.8.0.default.mpc83xx.tar.gz*
5. Click the **Install packages** button to start the update.

2.15.2 Online Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

Package set name to be used (*depending on installed firmware version!***):**

A package set name describes from which firmware versions updates can be made to the current firmware version.

– *update-8.{6-7}-8.8.0.default*
– (To 8.6.1: *update-7.{6}-8.6.1.default*)
– (To 8.6.1: *update-8.{0-5}-8.6.1.default*)
– (To 8.6.1: *update-8.{6}-8.6.1.default*)

The curly bracket indicates which installed source firmware versions can be updated by specifying the package set name (see Section 2.4.3).

2.15.2.1 Prepare online updates

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.15.2.2 Perform online update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Enter the name of the desired package set in the **Online Update** section under **Install package set**:
   – e.g., Minor update: *update-8.{6-7}-8.8.0.default*
4. Click the **Install package set** button to start the update.

2.15.3 Automatic Update to 8.8.0

Possible from installed firmware Version **8.6.1** or later.

2.15.3.1 Prepare Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Make sure that at least one valid update server is entered in the **Update Servers** section.

2.15.3.2 Start Automatic Update

1. Log on as *admin* user on the web interface of the mGuard device.
2. Select **Management** >> **Update** >> **Update**.
3. Click the button of the desired update process in the **Automatic Update** section to start the update:
   a) Install latest patches
   b) Install latest minor release
   c) Install next major release

2.15.4 Flash firmware Version 8.8.0

Required files:

Download file on the device-specific product page in the Phoenix Contact Web Shop:

– *FW_MPC_8.8.0.zip*

Update files (= unpacked Zip file):

– *ubifs.img.mpc83xx*

– *install-ubi.mpc83xx.p7s*

2.15.4.1 Download flash file

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD DELTA).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the following **download file**: *FW_MPC_8.8.0.zip*
6. Unpack the Zip file.
7. Copy all unpacked files (*ubifs.img.mpc83xx*, *install-ubi.mpc83xx.p7s*) from the *mpc* directory into a freely selected directory (e.g. *mGuard-Firmware*) on your TFTP server or in the */Firmware* directory on the SD card.

The *ubifs.img.mpc83xx* and *install-ubi.mpc83xx.p7s* files can be used to flash all of the devices described in this document, with the exception of FL MGUARD CENTERPORT and FL MGUARD GT/GT.

2.15.4.2 Flash mGuard device

NOTE: Flashing the firmware deletes all passwords and configurations on the mGuard device. The device is reset to its default setting.

During flashing, the firmware is always loaded from an SD card first. The firmware is only loaded from a TFTP server if no SD card is found.

The TFTP server must be installed on the locally connected computer.

1. Hold down the reset button of the device until the three lower LEDs on the left (ERR, FAULT, INFO) light up green.
   – The device starts the flash process: It first searches for an inserted SD card and for the corresponding update file in the */Firmware* directory. If the device does not find an SD card, it searches for a DHCP server via the LAN interface in order to obtain an IP address. The required files are loaded and installed from the SD card or the TFTP server.
2. If the three lower LEDs on the right (ERR, FAULT, INFO) flash green simultaneously, the flash process has been concluded successfully. (The flashing behavior is different in the case of simultaneous uploading of a configuration profile).
3. Restart the device.

2.16 FL MGUARD CORE TX

Please contact Support at your local PHOENIX CONTACT subsidiary.

2.17 mGuard Flash Guide

2.17.1 Flashing mGuard devices

The mGuard firmware is loaded and installed onto the mGuard device from an SD card, USB flash memory (both with vfat file system) or from a TFTP update server. All data, passwords, and configurations on the device are deleted. The device is reset to its default setting.

Carrying out the flash process is described individually for every mGuard device in this document (see the device-specific Section "*Flashing firmware Version 8.8.0*").

NOTE: Downgrading the pre-installed default firmware version is not supported.

For mGuard devices produced starting in January 2018, a *downgrade* of the pre-installed default firmware version to an earlier firmware version may fail. If this is the case, flash the device again with the firmware version that was originally installed or a higher version.

2.17.2 Problems with incompatible SD cards

When you flash the mGuard device with an SD card from a manufacturer other than PHOENIX CONTACT, the flashing procedure described in this document may fail.

To avoid problems flashing with SD cards of other manufacturers, proceed as follows during the described flashing procedure:

1. Push the card lightly into the device without engaging it.
2. Start the flash procedure as described for your device.
3. Hold down the reset button of the device until the corresponding LEDs light up.
4. Release the reset button.
5. Immediately push the card firmly into the slot until it engages.
6. Wait until the flashing procedure is over, then restart the device.

2.17.3 Uploading configuration profile during the flash process

You can automatically upload and activate a created configuration profile (ATV profile) onto the mGuard device during the flash process.

The flashing behavior of the LEDs after the flash process deviates in this case from the standard flashing behavior.

2.17.3.1 Preparation

Create the file *preconfig.sh* with the following contents:

For unencrypted ATV profiles

```
#!/bin/sh
exec gaiconfig --silent --set-all < /bootstrap/preconfig.atv
```

For encrypted ATV profiles

```
#!/bin/sh
/Packages/mguard-tpm_0/mbin/tpm_pkcs7 < /bootstrap/preconfig.atv.p7e | gaiconfig \ --factory-default --set-all
```

If you wish to upload a configuration profile encrypted with the device certificate, you should change the file's name from **.atv* to **.atv.p7e.* Encrypted and unencrypted configuration profiles can be kept apart easier in this way.

The mGuard device treats the ATV profile equally, independent of the file ending.

During the flash process, the device searches for the following files and uploads them:

- */Rescue Config/<Seriennummer>.atv*
- */Rescue Config/<Seriennummer>.atv.p7e*
- */Rescue Config/preconfig.atv*
- */Rescue Config/preconfig.atv.p7e*
- */Rescue Config/preconfig.sh*

2.17.3.2 Loading configuration profile from SD card

In order to upload and activate a configuration profile during the flash process, proceed as follows:

1. Besides the *Firmware* directory, also create the *Rescue Config.* directory.
2. Rename the saved configuration profile as *preconfig.atv or <Seriennummer>.atv*.
3. Copy the configuration profile to the *Rescue Config*. directory.
4. Copy the *preconfig.sh* file (UNIX-Format) to the *Rescue Config*. directory.
5. Carry out the flash process as described for your device.

2.17.3.3 Loading configuration profile from the TFTP server

In order to load and activate a configuration profile during the flash process, see the description in Section “Setting up DHCP and TFTP servers” on page 71.

2.17.4 Uploading licence file during the flash process

A licence file can be uploaded onto the mGuard device and activated during the flash process as follows (e. g. a licence for more VPN connections *FL MGUARD LIC VPN-10* or for a lifetime software update *FL MGUARD LIC LIFETIME FW*).

2.17.4.1 From SD card

In order to upload and activate a licence file during the flash process, proceed as follows:

1. Create the *Rescue Config.* directory on the installation medium.
2. Copy the licence file in the *Rescue Config.* directory.
3. Rename the licence file as *license.lic or <Seriennummer>.lic*.
4. Carry out the flash process as described for your device.

2.17.4.2 From the TFTP server

In order to load and activate a licence file during the flash process, see “Setting up DHCP and TFTP servers” on page 71.

2.17.5 Setting up DHCP and TFTP servers

Network problems

If you install a second DHCP server in a network, this could affect the configuration of the entire network.

Third-party software

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

2.17.5.1 Under Windows

If you wish to use the third-party program "*TFTPD32.exe*", obtain the program from a trustworthy source, and proceed as follows:

1. If the Windows PC is connected to a network, disconnect it from the network.
2. On the Windows PC, create a directory that you wish to use for the flash process of mGuard devices. This directory is later selected as root directory of the TFTP server. All the files required are loaded from this directory during the flash process.
3. Copy the desired firmware image file(s) into the created directory.
4. **(Uploading licence file)** If a **licence file** is to be uploaded and installed onto the mGuard device during the flash process, copy the file into the directory that has been created. Name the file as follows:
   - *license.lic* or
   - *<Serial number>.lic.*
5. **(Uploading configuration profile)** If a configuration profile is to be uploaded and activated on the mGuard device during the flash process, copy the corresponding **rollout script** (*rollout.sh*, see “Sample script: rollout.sh” on page 75) and the **configuration profile** in the directory that has been created. Name the configuration profile as follows:
   - *preconfig.atv* (if all mGuard devices should receive the same configuration) or
   - *<Seriennummer>.atv* (if each mGuard device should receive an individual configuration).
6. Start the *TFTPD32.exe* program.

   The host IP to be specified is: **192.168.10.1**. It must also be used as the address for the network card.
7. Click the **Browse** button to switch to the directory where the mGuard image files are saved: (e. g. *install-ubi.mpx83xx.p7s, ubifs.img.mpc.p7s*).

8. Make sure that this really is the correct licence file for the device (under “Management >> Update” on the web interface).

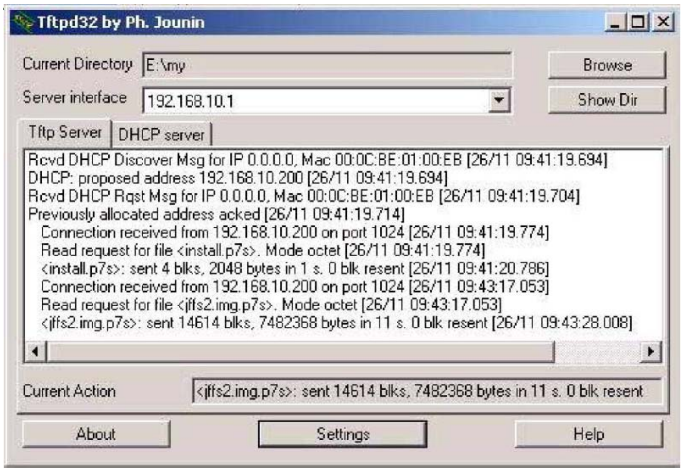
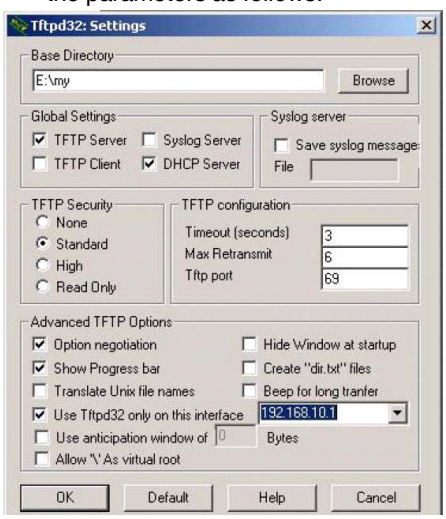


Figure 2-3 Entering the host IP

9. Switch to the “TFTP Server” or “DHCP Server” tab and click the “Settings” button to set the parameters as follows:



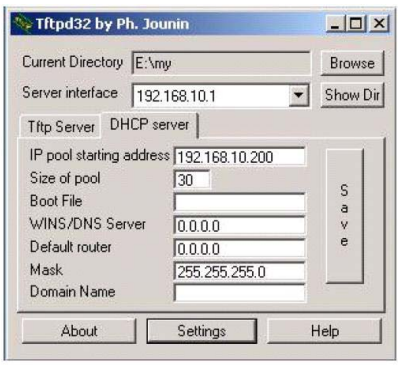


Figure 2-4 Settings

2.17.5.2 Under Linux

All current Linux distributions include DHCP and TFTP servers.

1. Install the corresponding packages according to the instructions provided for the respective distribution.
2. Configure the DHCP server by making the following settings in the */etc/dhcpd.conf* file:
   subnet 192.168.134.0 netmask 255.255.255.0 {
   range 192.168.134.100 192.168.134.119;
   option routers 192.168.134.1;
   option subnet-mask 255.255.255.0;
   option broadcast-address 192.168.134.255;}
   This example configuration provides 20 IP addresses (.100 to .119). It is assumed that the DHCP server has the address 192.168.134.1 (settings for ISC DHCP 2.0).

The required TFTP server is configured in the following file: */etc/inetd.conf*

3. In this file, insert the corresponding line or set the necessary parameters for the TFTP service. (Directory for data: */tftpboot*)
   tftp dgram udp wait root /usr/sbin/in.tftpd -s /tftpboot/
   The mGuard image files must be saved in the */tftpboot* directory: e. g. *install-ubi.mpx83xx.p7s, ubifs.img.mpc.p7s*.
4. **(Uploading licence file)** If a **licence file** is to be uploaded and installed onto the mGuard device during the flash process, copy the file into the */tftpboot* directory. Name the file as follows:
   – *license.lic* or
   – *<Serial number>.lic*.
5. **(Uploading configuration profile)** If a configuration profile is to be uploaded and activated on the mGuard device during the flash process, copy the corresponding **rollout script** (*rollout.sh*, see “Sample script: rollout.sh” on page 75) and the *configuration profile* in the */tftpboot* directory. Name the configuration profile as follows:
   – *preconfig.atv* (if all mGuard devices should receive the same configuration) or
   – *<Seriennummer>.atv* (if each mGuard device should receive an individual configuration).
6. Then restart the *inetd* process to apply the configuration changes.
7. If using a different mechanism, e.g., *xinetd*, please consult the corresponding documentation.

2.17.5.3 TFTP server: Error messages

During the flash process, the mGuard device searches by default for the files *rollout.sh, license.lic* and <*Seriennummer*>*.lic*. If these files are not available, a corresponding error message is displayed:

File rollout.sh: error 2 in system call CreateFile The system cannot find the file specified.

File <serial number>.lic : error 2 in system call CreateFile The system cannot find the file specified.

File licence.lic: error 2 in system call CreateFile The system cannot find the file specified.

The error message can be ignored if no licence file is uploaded, or the mGuard device should not be preconfigured via the *rollout.sh* script. The flash process is continued as planned in such cases.

2.17.6 Sample script: rollout.sh

Use of rollout scripts

The implementation and use of a rollout script is not a part of the mGuard product or mGuard firmware supported by PHOENIX CONTACT. Responsibility for the implementation and use of a rollout script lies solely with the customer and not PHOENIX CONTACT.

During the flash process, the mGuard device checks the presence of the *rollout.sh* file. This file must be located in the same directory as the firmware image file on the TFTP server. If the file exists, it is uploaded on the mGuard device and run there.

The *rollout.sh* file should be a UNIX shell script. The configuration data for the mGuard device can be requested from the TFTP server with the script, and the configuration program of the mGuard device (*gaiconfig*), started.

The rollout script documented here serves as a template and only can be used in a manner individually adapted by the customer. In principle, the rollout support can be implemented in two ways, so that

– "**all**" mGuard devices receive the same configuration (**static TFTP**), or
– "**every**" mGuard receives its own individual configuration depending on its serial number (**dynamic TFTP**).

2.17.6.1 Static TFTP (standard configuration for every mGuard device)

A sample *rollout.sh* script is documented below. This downloads a standard configuration file for installation on mGuard devices from the TFTP server via *tftp*. The name of the configuration file defined in the script is *preconfig.atv*.

```
#!/bin/sh -ex
# The IP address of the DHCP/TFTP server
# is supplied by install.p7s server=$1

# This is the filename of the user supplied static configuration file
# on the host in the TFTP-server directory cfg_name=preconfig.atv

export PATH=/bin:/bootstrap

# fetch the static configuration-file "preconfig.atv"
tftp -g -l - -r "$cfg_name" "${server}" | dd bs=1M of=/bootstrap/preconfig.atv

# create a small configuration-script that installs the
# configuration fetched from ${server} cat >/bootstrap/preconfig.sh <<EOF
#!/bin/sh

modprobe param_dev 2>/dev/null
gaiconfig --silent --set-all < /bootstrap/preconfig.atv EOF

# Make it executable. It will be executed after all packets
# are installed completely.

chmod 755 /bootstrap/preconfig.sh
```

2.17.6.2 Dynamic TFTP (individual configuration for every mGuard device)

A sample *rollout.sh* script is documented below. This downloads a device-specific configuration file from the TFTP server via *tftp*. The name of the configuration file defined in the script is *<serialnumber>.atv*.

```
#!/bin/sh -ex

# The IP address of the DHCP/TFTP server
# is supplied by install.p7s server=$1

export PATH=/bin:/bootstrap

mount -t proc none /proc || : mount -t sysfs sysfs /sys || :

if test -f /proc/sys/mguard/parameter/oem_serial ; then SERIAL=`cat
/proc/sys/mguard/parameter/oem_serial`
else
SERIAL=`sysmguard param oem_serial`
fi

# This is the filename of the user supplied static configuration file
# on the host in the TFTP-server directory cfg_name=${SERIAL}.atv

# fetch the static configuration-file "preconfig.atv"
tftp -g -l /bootstrap/preconfig.atv -r $cfg_name ${server}

# create a small configuration-script that installs the
# configuration fetched from ${server} cat >/bootstrap/preconfig.sh <<EOF
#!/bin/sh
modprobe param_dev 2>/dev/null || :
gaiconfig --silent --set-all < /bootstrap/preconfig.atv EOF

# Make it executable. It will be executed after all packets
# are installed completely.
chmod 755 /bootstrap/preconfig.sh

umount /proc umount /sys || :
```

2.18 Setting up mGuard firmware update repositories

If you have questions, please contact Support at your local PHOENIX CONTACT subsidiary.

To update your mGuard devices, you can use your own update server (Unix or Windows server). You can download the required update files on the device-specific product pages in the Phoenix Contact Web Shop.

Download file:

– **FL MGUARD CENTERPORT**
Unix and Windows Server: *mguard-firmware-repositories-8.8.0_x86.zip*

– **Other FL/TC MGUARD devices**
Unix and Windows Server: *mguard-firmware-repositories-8.8.0_mpc.zip*

To operate an update server, proceed as follows:

1. Open the website of the Phoenix Contact Web Shop in a web browser at: phoenixcontact.com/products.
2. Search for the device's product name (e.g. FL MGUARD RS 4000).
3. Open the desired product page.
4. Select the *Downloads* tab and the *Firmware update* category.
5. Download the desired **Download file**:
mguard-firmware-repositories-8.8.0_mpc.zip
6. Copy the contents of the ZIP folder onto your update server.
7. Enter the update server on the mGuard web interface under **Management >> Update >> Update** (see “Online Update” on page 18).
8. You can now carry out **Online Updates** or **Automatic Updates** from your update server.

NOTE: Online or Automatic Updates from the installed source firmware version **7.6.8** can lead to an error when the update server is operated with newer versions of the Apache Web Server (e.g. 2.4.18).

This problem will not occur if the Phoenix Contact update server which has been preset ex-works (https://update.innominate.com) is used.

To avoid the problem, an update server such as *nginx* or *fnord* can be used instead of an Apache Web Server.

3 Upgrade FL MGUARD DM to version 1.11.x

Document-ID: 107821_en_05
Document-Description: AH EN MDM UPGRADE

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

Contents of this document

This document explains how to upgrade *mGuard device manager* (mdm) to version 1.11.x.

3.1 Introduction

Since mdm versions 1.5.2 (Windows) and 1.7.0 (Windows and Ubuntu) *mGuard device manager* as well as mdm and third-party components can be installed and upgraded automatically via the included *mdm Installer for Windows* or Ubuntu‘s package management tools.

If your system does not fulfill the system requirements demanded in Table 3-1 and Table 3-2, *mdm server* and *mdm CA server databases* must either be dumped and imported to a newly installed mdm 1.11.x or the installed mdm version must be upgraded stepwise to mdm 1.11.x.

The following chapters describe

– how to upgrade your mdm installation and third-party components and
– how to migrate your existing mdm databases on Windows and Linux systems using batch files and shell scripts (see Chapter 3.6).

For a detailed view on the installation, configuration and usage of mGuard device manager (mdm) 1.11.x, please refer to the *mdm User Manual*, available online or as a PDF version in the PHOENIX CONTACT Webshop (phoenixcontact.net/product/2981974).

3.2 General notes

NOTE: Backup important files and databases

Keep backup copies of the following files and databases to avoid data loss during the upgrade process of mdm:

- current mdm server and mdm CA server databases
- *preferences.xml* and *ca-preferences.xml*

  These files usually contain individual parameters that are to be taken over again after the upgrade.
- *mdm license file*

  You need the license file to use the mdm to its full extent.

NOTE: Incompatibility of PostgreSQL databases

To upgrade from an older version to mdm 1.11.x, it is necessary to make irreversible changes to the backing PostgreSQL database. Once these changes have been made, the database can no longer be accessed with an older version.

NOTE: Java Runtime Environment (JRE) will be uninstalled

As of version mdm 1.11.0, mdm uses the Java platform *OpenJDK* and no longer the *Java Runtime Environment* (JRE).

The *mdm 1.11.x Installer for Windows* automatically installs the required version of *OpenJDK* and **uninstalls** existing versions of the *Java Runtime Environment*.

Migrating mdm installations with the provided batch files/shell scripts restores only the database(s) dumped and imported. Any other installation data (e.g. pull server certificate and config files) must be manually copied to the new installation as explained below.

The provided batch files/shell scripts will only work in case of standard mdm installations (check default database names, ports, and user names in the provided *preferences.xml* and *ca-preferences.xml* files).

mdm server (and CA server) will be stopped and restarted during the dump generation process.

3.3 Known issues

3.3.1 CA database migration using equal CA certificate attributes

Issue

If the mdm CA database migration (via provided database export/import scripts) to a newly installed mdm version 1.11.x on Windows

- from a different operating system **or**
- from an installed mdm version < 1.10.0

is done using the same CA certificate attributes on the new mdm 1.11.x installation, the mdm CA server will fail to start.

Solution

During the installation of mdm 1.11.x do not provide all certificate attributes of the CA identical to the ones of the older installation (e.g. add a suffix to the *Common Name*).

3.3.2 Different HTTP Server Directory Structure and Password Protection in Ubuntu and Windows

Issue

The HTTP server directory structures created by the *mdm Installer for Window*s and Ubuntu‘s package managing tools are different:

- In Windows, the server access is password protected, and three different directories are used: “atv”, “crl”, and “fw”, where "fw" is defined as the root directory.
- In Ubuntu, the server access **is not** password protected, and the server root directory is used to store pull configuration files, firmware upgrade packages and CRL files.

Solution

To enable password protection in Ubuntu, proceed as follows:

A) Edit the file “*/var/www/mdm/.htaccess*” and uncomment and edit the existing lines:

AuthType Basic
AuthName "username"
AuthUserFile /etc/mdm/mdm-webbase/.htpasswd
Require valid-user

Where "*username*" must be replaced with the username you want to grant the access to.

B) Use (as sudo) the Apache tool “*htpasswd*” to create the desired user password configuration in the file “*/etc/mdm/mdmwebbase/.htpasswd*”:

```
sudo htpasswd -c /etc/mdm/mdm-webbase/.htpasswd username
```

Where "*username*" must be replaced with the username you want to grant the access to. You will be asked to introduce the desired password.

To use the same directory structure in Ubuntu and Windows, proceed as follows:

A) Edit the file “*/etc/apache2/sites-available/mdm-webbase-ssl.conf*”:

Define the aliases “***atv***”, and “***crl***”. E.g.:

Alias "/atv/" "/var/www/mdm/"
Alias "/atv" "/var/www/mdm/"
Alias "/crl/" "/etc/mdm/security/crl/"
Alias "/crl" "/etc/mdm/security/crl/"
<Directory /etc/mdm/security/crl/>
Options +Indexes -FollowSymLinks +Multiviews
AllowOverride All

Require all granted
</Directory>

This will not change the real directory structure in the system, but will make it possible for already configured mGuards which expect the directories **atv**, and **crl** to download pull configurations, and CRL files successfully.

If you additionally want to have the same directory structure in the system, you have to create the corresponding directories and define the access permissions of each in ***mdm-webbase-ssl.conf***.

3.4 Operating system: Microsoft Windows

mdm 1.11.x can only be installed on supported Microsoft Windows systems if the required preconditions are fulfilled (see Table 3-1 on page 84).

NOTE: Incorrectly installed *Microsoft Visual C++ 2017 Redistributable Package (x64)* may break current mdm installation

Prior to the **installation or update** of mdm, *Microsoft Visual C++ 2017 Redistributable Package (x64)* (Version: 14.16.x) must have been **successfully** installed on the Windows system.

Download: https://aka.ms/vs/15/release/VC_redist.x64.exe

NOTE: It is possible that the specified link is no longer valid. In any case, make sure that the correct version is used! (Recommended version: 14.16.27027)

Precondition: All current Windows Update Packages must have been installed first.

Make sure that the package has been installed without warnings or error messages.

If the package has been installed unsuccessfully or incomplete, the mdm installation may fail and break existing mdm installations.

NOTE: All current Windows Update Packages must have been installed

Prior to the **installation or update** of mdm or Windows components, all available update packages for the Windows operating system must have been **successfully** installed.

Caution: It might be necessary to re-check several times that all necessary packages have been installed. Sometimes some of the packages will not be installed during the first or even second Windows Update session.

If your system does not fulfill the system requirements demanded in Table 3-1 *mdm server* and *mdm CA server databases* must either be dumped and imported to a newly installed mdm 1.11.x or the installed mdm version must be upgraded stepwise to mdm 1.11.x.

System requirements

Table 3-1 System requirements (Microsoft Windows systems)

	mdm Client	mdm Server	mdm CA
Supported operating system	– Windows Server 2016 – Windows Server 2012 R2 – Windows Server 2008 R2 SP1 – Windows 10 (mdm client only) – Windows 7 (mdm client only)		
Hardware	– A minimum of 512 MB RAM – 500 MB free hard disk space – Color monitor with at least 1280 x 1024 resolution	– A minimum of 4 GB RAM – 100 GB free hard disk space	– A minimum of 512 MB RAM – 5 GB free hard disk space
Software Components	– Third-party components (*PostgreSQL 10.7, Apache Webserver 2.4.38, OpenJDK 11 and OpenSSL 1.1.1b*) will automatically be installed via the *mdm 1.11.x Installer for Windows*. – *Apache Web Server* requires *Microsoft Visual C++ 2017 Redistributable Package (x64)* (Version: 14.16.x) to be installed. Download: https://aka.ms/vs/15/release/VC_redist.x64.exe NOTE: It is possible that the specified link is no longer valid. In any case, make sure that the correct version is used! (Recommended version: 14.16.27027) – mdm clients, independently run on systems other than the “server system“, require the Java platform *OpenJDK 11* to be installed.		

Precondition	– If not installed via *mdm Installer for Windows*:
	OpenJDK 11
		– mdm **not installed** (or mdm 1.10.0 or later installed).
		– *PostgreSQL* **not installed** (or installed by previous mdm installations).
		– *Apache Web Server* **not installed**
		  – (or installed and listening to a port other than 443),
		  – (or installed by previous mdm installations).
		– *Microsoft Visual C++ 2017 Redistributable Package (x64)* (Version: 14.16.x) **installed**.	

3.4.1 Upgrade on supported Microsoft Windows systems

Upgrade on supported Microsoft Windows systems	
From mdm 1.10.0 or later	To upgrade mdm version 1.10.x or later on supported Microsoft Windows systems, use the *mdm 1.11.x Installer for Windows* (see *mdm User Manual*, available online or as a PDF version in the PHOENIX CONTACT Webshop (phoenixcontact.net/product/2981974). It is not necessary to uninstall mdm version 1.10.x and components. The *mdm Installer for Windows* will automatically create database dumps of the current mdm installation 1.10.x.
From mdm < 1.10.0	To upgrade installed mdm version 1.5.2 or later on supported Microsoft Windows systems, there are two options: 1. **mdm Installer for Windows**: Upgrade the current mdm installation stepwise to the next minor version, using the corresponding *mdm Installer for Windows,* until mdm 1.11.x is installed (e.g. from mdm 1.5.2 >> 1.6.2 >> 1.7.0 >> 1.8.0 >> 1.9.x >> 1.10.x to mdm 1.11.x) **or** 2. **Database dumps:** create, export and import database dumps of the *mdm sever* and *CA server databases* as described below: – dump and backup the databases, – remove the complete mdm installation, – install mdm 1.11.x via the *mdm 1.11.x Installer for Windows*, – import the dumped databases. **NOTE: Irreversible data loss!** Data of your current mdm server and CA server database will be deleted. Keep a backup copy of your current databases in a secure place. **NOTE: Incompatibility of mdm databases!** To upgrade from an older version to mdm 1.11.x, it is necessary to make irreversible changes to the backing PostgreSQL database. Once these changes have been made, the database can no longer be accessed with an older version. Keep a backup copy of your current databases in a secure place. **To dump and backup the databases, proceed as follows:** **A) Make database dumps of the mdm and mdm CA server database** 1. Copy the required batch files to the Windows system where mdm is installed. 2. Execute (as administrator) the batch file ***export_mdm_server.bat*** . 3. Provide the path where the database dump shall be saved (default: *C:\Users\username\Documents\mdm-server.sql*). 4. Provide the path to your current mdm installation directory (default: *C:\Program Files\mGuard device manager*). 5. Provide the password of the database user *innomms* if required. 6. Press *any key* to close the command prompt when the database dump generation has finished. 7. (If necessary)) Repeat 1–6 **but** execute the batch file ***export_mdm_ca.bat*** to dump the *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*).

Upgrade on supported Microsoft Windows systems

B) Keep a backup copy of the database dumps

1. Open the directory where the database dumps have been saved.
2. Copy the database dump(s) created at step (A) to a secure place (e.g. a secure backup folder at another company server).

C) (If necessary)) Backup the pull server configuration

1. To backup the web server configuration file, copy the following file to a secure place:
   *<path to mdm installation>\apache\conf\extra**httpd-mdm.conf***
   – Search for the following entries (your entries may differ from the default settings of the mdm Installer given below) and write down the aliases of the pull config server:
   # Verzeichnis für ATV-Profile (wie in preferences.xml eingestellt).
   # Alias /atv/ /var/apache-data/atv/
   # <Directory /var/apache-data/atv/>
   ***Alias /atv/** "C:/Program Files/mGuard device manager/apache-data/atv/"*
   ***Alias /atv** "C:/Program Files/mGuard device manager/apache-data/atv/"*
   – If you have configured a service to send the pull feedback to mdm, search for and write down the following entries (CustomLog) as well:
   # Pull Config-Feedback an den mdm-Server (derzeit auskommentiert).
   # CustomLog "\| /bin/nc -u -i1 127.0.0.1 7514" common
   <your feedback configuration>
2. To backup the certificate and private key, copy the following files to a secure place:
   *<path to mdm installation>\apache\conf**server.crt***
   *<path to mdm installation>\apache\conf**server.key***

D) Remove the complete mdm installation from the Windows system

1. Remove the mdm installation by using its own unistaller or Microsoft Window's standard uninstall procedures (e.g. *Control Panel\Programs\Programs and Features*).

E) Install mdm 1.11.x and desired components via the mdm 1.11.x Installer for Windows.

1. Use the *mdm 1.11.x Installer for Windows* as described in the mdm User Manual.

Do not provide all certificate attributes of the CA identical to the ones of the older installation (e.g. add a suffix to the *Common Name*).

Upgrade on supported Microsoft Windows systems

F) Import the dumped databases (mdm server and mdm CA server)

1. Make the dumped databases available on the system where mdm 1.11.x has been installed.
2. Execute (as administrator) the batch file ***import_mdm_server.bat*** .
3. Provide the path to the database dump (default: *C:\Users\username\Documents\mdm-server.sql*).
4. Provide the installation path of mdm 1.11.x (default: *C:\Program Files\mGuard device manager*).
5. Provide the password of the database user *innomms* if required.
6. Press *any key* to close the command prompt when the database import has finished.
7. (If necessary)) Repeat 1–6 **but** execute the batch file ***import_mdm_ca.bat*** to import the dumped *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*).
8. mdm will restart automatically and connect to the imported databases.

G) (If necessary)) Update the firmware upgrade/pull server configuration

1. Open *Apache HTTP Server Monitor* (included in the mdm installation) and **stop** the service ***ApacheMDM***.
2. Compare the web server configuration file, with the backup file copied and stored in step (C):

   *<path to mdm installation>\apache\conf\extra**httpd-mdm.conf***
   - Compare the aliases of the pull config server. If your former settings differ from the default settings of the mdm Installer (given below), replace the default settings accordingly to your settings (e.g. ***Alias /my_company_atv/***):
     Alias /atv/ *"C:/Program Files/mGuard device manager/apache-data/atv/"*
     Alias /atv *"C:/Program Files/mGuard device manager/apache-data/atv/"*
   - Update the pull configuration feedback to the mdm server, if it must be configured:
     # Pull Config-Feedback an den mdm-Server (derzeit auskommentiert).
     # CustomLog "\| /bin/nc -u -i1 127.0.0.1 7514" common
     <your feedback configuration>
3. Copy the certificate and private key, backed up and stored in step (C) to
   *<path to mdm installation>\apache\conf**server.crt***
   *<path to mdm installation>\apache\conf**server.key***
4. Open *Apache HTTP Server Monitor* and **start** the service ***ApacheMDM***.

3.4.2 Upgrade from unsupported Microsoft Windows systems

Upgrade from unsupported Microsoft Windows systems	
All mdm versions	mdm versions installed on unsupported Windows systems cannot be upgraded to mdm 1.11.x. To reuse the databases of these mdm versions on supported Windows systems, you have to: – dump and backup the databases, – install mdm 1.11.x via the *mdm 1.11.x Installer for Windows* on a supported system (see mdm User Manual), – import the dumped databases. To dump and import the *mdm server database* and *mdm CA server database*, proceed as described above (“From mdm < 1.10.0” on page 85).

3.4.3 Upgrade from Linux systems

Upgrade from Linux systems

All mdm versions

To reuse the databases of mdm versions, installed on Linux systems, on supported Windows systems, you have to:

– dump and backup the databases,
– install mdm 1.11.x via the *mdm 1.11.x Installer for Windows* on a supported system (see mdm User Manual),
– import the dumped databases.

NOTE: Incompatibility of mdm databases!

To upgrade from an older version to mdm 1.11.x, it is necessary to make irreversible changes to the backing PostgreSQL database. Once these changes have been made, the database can no longer be accessed with an older version. Keep a backup copy of your current databases in a secure place.

Proceed as follows:

A) Make database dumps of the mdm and mdm CA server database

1. Copy the required shell script files to the Linux system where mdm is installed.
2. Execute (as sudo) the shell script ***export_mdm_server.sh*** .
3. Provide the path where the database dump shall be saved (default: */tmp/mdm-server.sql*).
4. Provide the password of the database user *innomms* if required.
5. (If necessary)) Repeat 1–4 **but** execute the shell script ***export_mdm_ca.sh*** to dump the *mdm CA server database* (default: *mdm_ca_server.sql,* database user = *mdmca*).

B) Keep a backup copy of the database dumps

1. Open the directory where the database dumps have been saved.
2. Copy the database dump(s) created at step (A) to a secure place (e.g. a secure backup folder at another company server).

C) (If necessary)) Backup the pull server configuration

1. To backup the web server configuration file, copy the following server configuration file to a secure place. E.g. if you are using Apache 2.x, your configuration file may be stored in: */etc/apache2/sites-available/your-server.conf* .
   – If your server configuration defines any aliases for the pull configuration directory, write them down. They may look like:
     Alias /atv/ *"/var/www/mdm-pull/"*
     Alias /atv *"/var/www/mdm-pull/"*
2. To backup the certificate and private key, proceed as follows:
   – Check the configuration file from step (1.) (e.g. */etc/apache2/sites-available/your-server.conf*) and look for the certificates used by your server. E.g. if you are using Apache 2.x, the entries may look like:
     SSLCertificateFile /etc/mdm/mdm-pull-server/cert.pem
     SSLCertificateKeyFile /etc/mdm/mdm-pull-server/key
   – Copy those files to a secure place, using the following file names:
     *<path to certificate>/****server.crt***
     *<path to certificate>/****server.key***

Upgrade from Linux systems

D) Install mdm 1.11.x and desired components via the mdm 1.11.x Installer for Windows.

1. Use the *mdm 1.11.x Installer for Windows* as described in the mdm User Manual.

> Do not provide all certificate attributes of the CA identical to the ones of the older installation (e.g. add a suffix to the *Common Name*).

E) Import the dumped databases (mdm server and mdm CA server)

1. Make the dumped databases available on the system where mdm 1.11.x has been installed.
2. Execute (as administrator) the batch file ***import_mdm_server.bat*** .
3. Provide the path to the database dump (default: *C:\Users\username\Documents\mdm-server.sql*).
4. Provide the installation path of mdm 1.11.x (default: *C:\Program Files\mGuard device manager*).
5. Provide the password of the database user *innomms* if required.
6. Press *any key* to close the command prompt when the database import has finished.
7. (If necessary)) Repeat 1–6 **but** execute the batch file ***import_mdm_ca.bat*** to import the dumped *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*).
8. mdm will restart automatically and connect to the imported databases.

F) (If necessary)) Update the firmware upgrade/pull server configuration

1. Open *Apache HTTP Server Monitor* (included in the mdm installation) and **stop** the service ***ApacheMDM***.
2. Compare the web server configuration file, with the backup file copied and stored in step (C):

   *<path to mdm installation>\apache\conf\extra**httpd-mdm.conf***

   – Compare the aliases of the pull config server. If your former settings differ from the default settings of the mdm Installer (given below), replace the default settings accordingly to your settings (e.g. ***Alias /my_company_atv/***):
   Alias /atv/ *"C:/Program Files/mGuard device manager/apache-data/atv/"*
   Alias /atv *"C:/Program Files/mGuard device manager/apache-data/atv/"*
   – Update the pull configuration feedback to the mdm server, if it must be configured:
   # Pull Config-Feedback an den mdm-Server (derzeit auskommentiert).
   # CustomLog "\| /bin/nc -u -i1 127.0.0.1 7514" common
   <your feedback configuration>
3. Copy the certificate and private key, backed up and stored in step (C) to
   *<path to mdm installation>\apache\conf**server.crt***
   *<path to mdm installation>\apache\conf**server.key***
4. Open *Apache HTTP Server Monitor* and **start** the service ***ApacheMDM***.

3.5 Operating system: Ubuntu Linux

Version **mdm 1.11.x** can only be installed on **Ubuntu (Server) 18.04 LTS**.

To upgrade an older version of mdm to mdm 1.11.x, you must first upgrade Ubuntu 16.04 LTS to Ubuntu 18.04 LTS via Ubuntu‘s package management tools.

See also:

“Quick Guide: Upgrade Ubuntu 16.04 to 18.04” on page 96

“Quick Guide (Ubuntu): Upgrade mdm 1.10.x to 1.11.x” on page 97

If your system does not fulfill the system requirements demanded in Table 3-2, *mdm server* and *mdm CA server databases* must either be dumped and imported to a newly installed mdm 1.11.x or the installed mdm version must be upgraded stepwise to mdm 1.11.x.

Privacy notice: Access to the *mdm software repository* server is logged to ensure the security and stability of the service. Only anonymized data is retained for statistical analysis.

System requirements

Table 3-2 System requirements (Ubuntu Linux)

	mdm Client	mdm Server	mdm CA
Operating system	– Ubuntu Desktop 16.04 LTS	– Ubuntu (Server) 16.04 LTS	
Hardware	– A minimum of 512 MB RAM – 500 MB free hard disk space – Color monitor with at least 1280 x 1024 resolution	– A minimum of 4 GB RAM – 100 GB free hard disk space	– A minimum of 512 MB RAM – 5 GB free hard disk space
Software components	– Third-party components (*PostgreSQL 10, Apache Webserver 2.4, OpenJDK 11 and OpenSSL 1.1.x*) will automatically be installed via the package management of Ubuntu.		
Precondition	– If not installed via Ubuntu‘s package managing tools: *OpenJDK 11*	– mdm **not installed** (or mdm 1.10.0 or later installed). – All components of previous mdm installations < 1.7.0 must have been removed.	

3.5.1 Upgrade on supported and unsupported Linux systems

Upgrade on supported and unsupported Linux systems	
From mdm 1.10.0 or later (installed on Ubuntu 16.04 LTS)	To upgrade mdm versions 1.10.0 or later, installed on Ubuntu Server 16.04 LTS, you have to: – upgrade Ubuntu Server 16.04 LTS to Ubuntu Server 18.04 LTS: **see** “Quick Guide: Upgrade Ubuntu 16.04 to 18.04” on page 96. – upgrade mdm to mdm 1.11.x via Ubuntu‘s package management tools: **see** “Quick Guide (Ubuntu): Upgrade mdm 1.10.x to 1.11.x” on page 97.
From mdm 1.7.0 or later (installed on Ubuntu 16.04 LTS)	To upgrade mdm version 1.7.0 or later, installed on Ubuntu Server 16.04 LTS, you have to proceed stepwise: 1. Upgrade the installed mdm version in several steps via the package management tools of Ubuntu 16.04 LTS to the next possible version (mdm 1.7.x >> 1.8.x >> 1.9.x >> 1.10.x). 2. Upgrade mdm 1.10.x to mdm 1.11.x as described above (*From mdm 1.10.0 or later*). For further information see mdm User Manual.
From mdm < 1.7.0	mdm versions < 1.7.0, installed on supported and unsupported Linux systems, cannot be upgraded to mdm 1.11.x. To reuse the databases of these mdm versions on supported Linux systems, you have to: – dump and backup the databases, – install Ubuntu Server 18.04 LTS, – install mdm 1.11.x via Ubuntu‘s package management tools on Ubuntu Server 18.04 LTS (see Section 3.7), – import the dumped databases. **NOTE: Incompatibility of mdm databases!** To upgrade from an older version to mdm 1.11.x, it is necessary to make irreversible changes to the backing PostgreSQL database. Once these changes have been made, the database can no longer be accessed with an older version. Keep a backup copy of your current databases in a secure place. **Proceed as follows:** **A) Make database dumps of the mdm and mdm CA server database** 1. Copy the required shell script files to the Linux system where mdm is installed. 2. Execute (as sudo) the shell script ***export_mdm_server.sh*** . 3. Provide the path where the database dump shall be saved (default: */tmp/mdm-server.sql*). 4. Provide the password of the database user *innomms* if required. 5. (If necessary)) Repeat 1–4 **but** execute the shell script ***export_mdm_ca.sh*** to dump the *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*). **B) Keep a backup copy of the database dumps** 1. Open the directory where the database dumps have been saved. 2. Copy the database dump(s) created at step (A) to a secure place (e.g. a secure backup folder at another company server).

Upgrade on supported and unsupported Linux systems

C) (If necessary)) Backup the pull server configuration

1. To backup the web server configuration file, copy the following server configuration file to a secure place. E.g. if you are using Apache 2.x, your configuration file may in:
   /etc/apache2/sites-available/your-server.conf .
   – If your server configuration defines any aliases for the pull configuration directory, write them down. They may look like:
     Alias */atv/ "/var/www/mdm-pull/"*
     Alias */atv "/var/www/mdm-pull/"*
2. To backup the certificate and private key, proceed as follows:
   – Check the configuration file from step (1.) (e.g. */etc/apache2/sites-available/your-server.conf*) and look for the certificates used by your server. E.g. if you are using Apache 2.x, the entries may look like:
     SSLCertificateFile /etc/mdm/mdm-pull-server/cert.pem
     SSLCertificateKeyFile /etc/mdm/mdm-pull-server/key
   – Copy those files to a secure place, using the following file names:
     *<path to certificate>/**cert.pem***
     *<path to certificate>/**key***

D) Install mdm 1.11.x and desired components via Ubuntu‘s package management tools

1. Use Ubuntu‘s package management tools as described in Section 3.7.

Do not provide all certificate attributes of the CA identical to the ones of the older installation (e.g. add a suffix to the *Common Name*).

E) Import the dumped databases (mdm server and mdm CA server)

1. Make the dumped databases available on the system where mdm 1.11.x has been installed.
2. Execute (as sudo) the shell script ***import_mdm_server.sh*** .
3. Provide the path to the database dump (default: */tmp/mdm-server.sql*).
4. (If necessary)) Repeat 1–3 **but** execute the shell script ***import_mdm_ca.sh*** to import the dumped *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*).
5. mdm will restart automatically and connect to the imported databases.

F) (If necessary)) Update the firmware upgrade/pull server configuration

1. Stop (as sudo) the Apache Web Server: `service apache2 stop`
2. If your previous configuration defined aliases for the pull configuration directory (check files backup up and stored in step (C)), edit the Apache configuration file of your new mdm installation:
   /etc/mdm/mdm-webbase/30-configpull.conf
   – Add the aliases of your previous configuration (do not change the real export directory: */var/www/mdm/*). E.g.:
     Alias */atv/ "/var/www/mdm/"*
     Alias */atv "/var/www/mdm/"*
3. Copy the certificate and private key, backed up and stored in step (C) to
   */etc/mdm/mdm-webbase/**cert.pem***
   */etc/mdm/mdm-webbase/**key***
4. Start (as sudo) the Apache Web Server: `service apache2 start`

3.5.2 Upgrade from Microsoft Windows systems

Upgrade from Microsoft Windows systems	
All mdm versions	To reuse the databases of mdm versions, installed on Windows systems, on Ubuntu Server 18.04 ITS, you have to: – dump and backup the databases, – install Ubuntu Server 18.04 LTS, – install mdm 1.11.x via Ubuntu‘s package management tools on Ubuntu Server 18.04 LTS (see Section 3.7), – import the dumped databases. **NOTE: Incompatibility of mdm databases!** To upgrade from an older version to mdm 1.11.x, it is necessary to make irreversible changes to the backing PostgreSQL database. Once these changes have been made, the database can no longer be accessed with an older version. Keep a backup copy of your current databases in a secure place. **Proceed as follows:** **A) Make database dumps of the mdm and mdm CA server database** 1. Copy the required batch files to the Windows system where mdm is installed. 2. Execute (as administrator) the batch file ***export_mdm_server.bat*** . 3. Provide the path where the database dump shall be saved (default: *C:\Users\username\Documents\mdm-server.sql*). 4. Provide the path to your current mdm installation directory (default: *C:\Program Files\mGuard device manager*). 5. Provide the password of the database user *innomms* if required. 6. Press *any key* to close the command prompt when the database dump generation has finished. 7. (If necessary)) Repeat 1–6 **but** execute the batch file ***export_mdm_ca.bat*** to dump the *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*). **B) Keep a backup copy of the database dumps** 1. Open the directory where the database dumps have been saved. 2. Copy the database dump(s) created at step (A) to a secure place (e.g. a secure backup folder at another company server).

Upgrade from Microsoft Windows systems

C) (If necessary)) Backup the pull server configuration

1. To backup the web server configuration file, copy the following file to a secure place:
   *<path to mdm installation>\apache\conf\extra**httpd-mdm.conf***
   – Search for the following entries (your entries may differ from the default settings of the mdm Installer given below) and write down the aliases of the pull config server:
     # Verzeichnis für ATV-Profile (wie in preferences.xml eingestellt).
     # Alias /atv/ /var/apache-data/atv/
     # <Directory /var/apache-data/atv/>
     ***Alias /atv/** "C:/Program Files/mGuard device manager/apache-data/atv/"*
     ***Alias /atv** "C:/Program Files/mGuard device manager/apache-data/atv/"*
2. To backup the certificate and private key, copy the following files to a secure place:
   *<path to mdm installation>\apache\conf**server.crt***
   *<path to mdm installation>\apache\conf**server.key***

D) Install mdm 1.11.x and desired components via Ubuntu's package management tools

1. Use Ubuntu's package management tools as described in Section 3.7.

Do not provide all certificate attributes of the CA identical to the ones of the older installation (e.g. add a suffix to the *Common Name*).

E) Import the dumped databases (mdm server and mdm CA server)

1. Make the dumped databases available on the system where mdm 1.11.x has been installed.
2. Execute (as sudo) the shell script ***import_mdm_server.sh*** .
3. Provide the path to the database dump (default: */tmp/mdm-server.sql*).
4. (If necessary)) Repeat 1–3 **but** execute the shell script ***import_mdm_ca.sh*** to import the dumped mdm CA server database (default: *mdm_ca_server.sql*, database user = *mdmca*).
5. mdm will restart automatically and connect to the imported databases.

F) (If necessary)) Update the firmware upgrade/pull server configuration

1. Stop (as sudo) the Apache Web Server: `service apache2 stop`
2. If your previous configuration defined aliases for the pull configuration directory (check files backup up and stored in step (C)), edit the Apache configuration file of your new mdm installation:
   /etc/mdm/mdm-webbase/30-configpull.conf
   – Add the aliases of your previous configuration (do not change the real export directory: */var/www/mdm/*). E.g.:
     ***Alias** /atv/ "/var/www/mdm/"*
     ***Alias** /atv "/var/www/mdm/"*
3. Copy the certificate and private key, backed up and stored in step (C) to
   */etc/mdm/mdm-webbase/**cert.pem***
   */etc/mdm/mdm-webbase/**key***
4. Start (as sudo) the Apache Web Server: `service apache2 start`

3.5.3 Quick Guide: Upgrade Ubuntu 16.04 to 18.04

Version **mdm 1.11.x** can only be installed on **Ubuntu (Server) 18.04 LTS**.

NOTE: Data loss during the upgrade process
Backup your files before you upgrade the system.

Backup *mdm* and *mdm CA server databases* of the current mdm installation

1. Copy the required shell script files to the Linux system where mdm is installed.
2. Execute the shell script ***export_mdm_server.sh*** (as sudo/adminstrator).
3. Provide the path where the database dump shall be saved (default: */tmp/mdm-server.sql*).
4. Provide the password of the database user *innomms* if required.
5. (If necessary) Repeat 1–4 **but** execute the shell script ***export_mdm_ca.sh*** to dump the *mdm CA server database* (default: *mdm_ca_server.sql*, database user = *mdmca*).
6. Copy the created database dumps to a secure location (such as a secure backup directory on another server in the organization).

Upgrade Ubuntu 16.04 LTS to Ubuntu 18.04 LTS

1. Reload the package information:
   ```
   sudo apt update
   ```
2. Update the packages installed under Ubuntu 16.04 LTS:
   ```
   sudo apt upgrade
   ```
3. Start the upgrade to Ubuntu 18.04 LTS:
   ```
   sudo do-release-upgrade
   ```
4. Follow the on-screen instructions or press Enter to continue the upgrade if necessary.

3.5.4 Quick Guide (Ubuntu): Upgrade mdm 1.10.x to 1.11.x

Do not update your mdm installation until the upgrade from Ubuntu 16.04 LTS to Ubuntu 18.04 LTS has successfully been completed (see “Quick Guide: Upgrade Ubuntu 16.04 to 18.04” on page 96).

If mdm 1.11.x is installed via the command line, the variable *DEBIAN_FRONTEND* must be used with the value *readline*. This is obligatory to display and accept the *Software License Terms* (SLT).

1. Use a text editor to change the *mdm software repository* from 1.10.x/ to 1.11.x/ in Ubuntu‘s */etc/apt/sources.list*:

   ```
   sudo nano /etc/apt/sources.list
   ```

2. Alternatively you can use the following command to add the *mdm software repository of version 1.11.x* to your package management tools:

   ```
   sudo apt-add-repository
   „deb http://repositories.mguard.com/mdm 1.11.x/“
   ```

3. Reload the package information:

   ```
   sudo apt update
   ```

4. Start the upgrade to mdm 1.11.x:

   ```
   sudo DEBIAN_FRONTEND=readline apt upgrade
   ```

5. Agree to the *Software License Terms* (SLT).

3.5.5 Update PostgreSQL databases (cluster)

After an upgrade from Ubuntu 16.04 LTS to Ubuntu 18.04 LTS the *PostgreSQL 9.5* version remains installed on the system. *PostgreSQL 10* will be installed, when the installed mdm version is upgraded to mdm 1.11.x.

The *mdm server* and *CA server* databases, created with earlier mdm versions, were created with *PostgreSQL 9.5*. However, this is not a problem as *PostgreSQL 10* is backward compatible and supports databases created with older versions.

An adaptation of the existing *mdm server* and *CA server* databases is therefore not necessary for the operation of mdm 1.11.x.

3.6 Batch files and shell scripts

Dump and import of the databases can be executed using batch files (Windows) and shell scripts (Linux) provided by Phoenix Contact available in the PHOENIX CONTACT Webshop (phoenixcontact.net/product/2981974).

Migrating mdm installations with the provided batch files/shell scripts restores only the database(s) dumped and imported. Any other installation data (e.g. pull server certificate and config files) must be manually copied to the new installation as explained below.

The provided batch files/shell scripts will only work in case of standard mdm installations (check default database names, ports, and user names in the provided *preferences.xml* and *ca-preferences.xml* files).

If mdm 1.11.x has been successfully installed via the *mdm 1.11.x Installer for Windows* or Ubuntu's package management tools, the batch files/shell scripts have been installed automatically in the following system folders:

Microsoft Windows

mdm server and mdm CA server: *<path to mdm installation>\data\db_migration*

Ubuntu Linux

mdm server: */usr/share/mdm-server/db_migration/*

mdm CA server: */usr/share/mdm-ca/db_migration/*

3.6.1 Batch files and shell scripts

Table 3-3 Windows batch files

Name	Description
export_mdm_server.bat	Windows batch file to dump the *mdm server database*
export_mdm_ca.bat	Windows batch file to dump the *mdm CA server database*
import_mdm_server.bat	Windows batch file to import the dumped *mdm server database*
import_mdm_ca.bat	Windows batch file to import the dumped *mdm CA server database*

Table 3-4 Linux shell script files

Name	Description
export_mdm_server.sh	Linux shell script file to dump the *mdm server database*
export_mdm_ca.sh	Linux shell script file to dump the *mdm CA server database*
import_mdm_server.sh	Linux shell script file to import the dumped *mdm server database*
import_mdm_ca.sh	Linux shell script file to import the dumped *mdm CA server database*

3.7 Ubuntu‘s package management tools

For the installation of the following mdm components on Ubuntu Linux, the automatic installation via Ubuntu‘s package management tools and the *mdm software repository* can be used.

Privacy notice: Access to the *mdm software repository* server is logged to ensure the security and stability of the service. Only anonymized data is retained for statistical analysis.

Table 3-5 Installable packages from the *mdm software repository*

Package	Description
mdm-all-server	Meta package to install all mdm server components.
mdm-common	Contains basic components required to install mdm.
mdm-server	Contains the server component of mdm. Starts as ***systemd*** service.
mdm-client	Contains the client components of mdm. Starts as ***systemd*** service.
mdm-ca	Contains the CA components of mdm (CA server).
mdm-configpull	Sets up the apache2 server to provide the mdm configuration pull feature (see mdm User Manual).
mdm-clientdownload	Sets up the apache2 server to allow the mdm client download.
mdm-webbase	Configures apache2 for mdm and allows it to be used as firmware server (see mdm User Manual).

PostgreSQL database server and *OpenSSL* may be automatically installed from Ubuntu‘s standard repositories.

Preconditions

The following system requirements and preconditions must be fulfilled (see Table 3-2 on page 91).

Installation of the license file

Save the license file as */etc/mdm/mdm-server/mdmlic.lic*. The path of the license file can be configured in the *preferences.xml* file afterwards (see mdm User Manual). If you do not specify a path for the license file in the *preferences.xml* file, mdm assumes the license file to be in the same directory as the mdm server.

Install the license file prior to the installation of the mdm server package into */etc/mdm/mdm-server/mdmlic.lic*, creating the path as needed, or restart the server manually after you have installed a new license.

3.7.1 Full mdm installation

To install **mdm and components** on Ubuntu (Server) 18.04 LTS using Ubuntu's package management tools, proceed as follows:

You need administrator rights to install mdm and components.

Make sure, that the correct repository has been successfully added to the file */etc/apt/sources.list*. Check the integrity of the provided repository key.

If mdm 1.11.x is installed via the command line, the variable *DEBIAN_FRONTEND* must be used with the value *readline* to install the package *mdm-common*. This is obligatory to display and accept the *Software License Terms* (SLT).

Copy your mdm license file *mdmlic.lic* to the directory */etc/mdm/mdm-server/* (default setting in *preferences.xml*) before you install mdm.

1. Download the public key of the repository (*pubkey.gpg*):
   ```
   wget http://repositories.mguard.com/pubkey.gpg
   ```
2. Check the fingerprint of the public key:
   ```
   gpg -finger pubkey.gpg
   ```
   The fingerprint **must match** the following fingerprint:
   AD3E B1F9 473D 5CC7 2ED4 2D4C 0571 79A3 CC0F FA55
3. Add the public key of the repository (*pubkey.gpg*) to the GPG public keyring (*trusted.gpg*)*:*
   ```
   sudo apt-key add pubkey.gpg && apt-key list
   ```
4. Add the *mdm software repository* to your package management tool:
   ```
   sudo apt-add-repository
   „deb http://repositories.mguard.com/mdm 1.11.x/“
   ```
5. Reload the package information:
   ```
   sudo apt update
   ```
6. Display the available mdm packages by searching for the term *mdm*:
   ```
   sudo apt search mdm
   ```
7. Install and agree to the *Software License Terms* (SLT) before installing mdm:
   ```
   sudo DEBIAN_FRONTEND=readline apt install mdm-common
   ```
8. Install mdm and server components:
   ```
   sudo apt install mdm-all-server
   ```
   Follow the on-screen instructions and enter mandatory and optional parameters (e.g. for CA component and configuration pull server).
9. Install mdm client using the package management tools:
   ```
   sudo apt install mdm-client
   ```

3.7.2 Quick mdm server and client installation (full installation)

```
wget http://repositories.mguard.com/pubkey.gpg

sudo apt-key add pubkey.gpg

sudo apt-add-repository "deb http://repositories.mguard.com/mdm
1.11.x/"

sudo apt update

sudo DEBIAN_FRONTEND=readline apt install mdm-common

sudo apt install mdm-all-server mdm-client
```

3.7.3 Analyzing server log files

mdm server

Output of the complete log entries of the mdm server:

```
journalctl -u mdm-server.service
```

Output of the log entries of the mdm server since the last reboot:

```
journalctl -b -u mdm-server.service
```

mdm CA Server

Output of the log entries of the mdm CA server:

```
journalctl -u mdm-ca.service
```

Output of the log entries of the mdm CA server since the last restart:

```
journalctl -b -u mdm-ca.service
```

The mdm CA server log files are also saved by Ubuntu to the file: */var/log/mdm-ca.log* .

4 Create X.509 certificates with OpenSSL

Document-ID: 108395_en_01
Document-Description: AH EN X.509 CERT OPENSSL

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

Contents of this document

This section explains briefly how to create X. 509 certificates using the tool *OpenSSL*.

4.1 Introduction

The enrollment of certificates requires a certification authority (CA) which issues public key certificates for a specific period of time. A CA can be a private (in-house) CA, run by your own organization, or a public CA. A public CA is operated by a third party that you trust to validate the identity of each client or server to which it issues a certificate.

There are several tools available for creating and managing certificates, as for example *Microsoft Certification Authority (CA) Server*, *OpenSSL* and *XCA*.

This application note explains how to create X.509 certificates with the tools **OpenSSL** and **XCA** for setting up a VPN connection using X.509 certificates as authentication method.

The scope of this document is not to be a complete user's guide for the described tools. It shall help you getting familiar with them and to create the required certificates in a short term.

4.1.1 Introduction OpenSSL

OpenSSL is available for several platforms (Linux, UNIX, Windows) and can be downloaded from the Internet. We have used *OpenSSL 1.1.0e* on a *Windows 7* platform. Please refer to http://www.openssl.org for getting further information about OpenSSL and the supported command line options.

OpenSSL provides various ways for specifying the required options. You can enter them at the command line, specify them in a configuration file or you'll be prompted to enter them when the *openssl* command is executed. When using configuration files, you can either

specify all required parameters in one single file or use different ones, depending on which kind of certificate you want to create. The OpenSSL configuration file, which comes with OpenSSL, is called *openssl.cnf*.

Please note that Windows hides the file extension *.cnf*, even if you have configured the *Windows Explorer* not to do so. Therefore we use the extension *.conf*.

In the following chapters we will explain how to setup OpenSSL to act as certification authority (CA). A certificate request must be signed by the CA to become a valid certificate.

Basically you can use the examples of the following chapters for creating the certificates. You only need to follow the instructions and adjust the parameters in the section *req_dn* of the OpenSSL configuration file *openssl.conf* (see chapter “Modifying the OpenSSL configuration file” on page 108) to your company needs.

Here is a small legend with **file extensions** we will use for the created files and their meaning.

File extension	Explanation
key	Private key Restrictive permissions should be set on these files.
csr	Certificate Request The request will be signed by the CA in order to create the certificate. After doing this, the file is not needed anymore and can be deleted.
crt	Certificate This certificate can be publicly distributed.
p12	PKCS#12 export of the certificate, containing its private and public key. The export file is secured by a password to protect the private key against unauthorized usage. This certificate may not be distributed publicly.

4.2 Preparing the CA environment

First of all we will create a directory structure where all certificate stuff will be kept. In the following examples we use **C:\CA** as root directory. The following subdirectories need to be created:

Subdirectory	Purpose
.\certs	Directory where the certificates will be placed.
.\newcerts	Directory where OpenSSL puts the created certificates in PEM format as *<cert serial number>.pem* (e.g. *07.pem*). OpenSSL requires this directory.
.\private	Directory for storing the private keys. Ensure that you set restrictive permissions to this directory so that they can be read only by user with the appropriate privileges.

Apart of the directory tree, the following two files (*index.txt* and *serial*) need to be created:

- **index.txt**: This file is used as certificate “database” by OpenSSL. To create this file, proceed as follows:
  - Open a DOS prompt.
  - Switch to the CA root directory (in our example *C:\CA*).
  - Execute the command: *copy NUL: index.txt*
    This command creates the empty file *index.txt*.
- **serial**: This file contains the certificate serial number counter. This counter will be incremented automatically by OpenSSL when its value has been used for creating a certificate. To create this file, proceed as follows:
  - Open a DOS prompt.
  - Switch to the CA root directory (in our example *C:\CA*).
  - Execute the command: *echo 0001 > serial*
    This command creates the file *serial* with the initial serial number 0001.

4.3 Modifying the OpenSSL configuration file

We have named the OpenSSL configuration file *openssl.conf* and placed it into the CA root directory (in our example *C:\CA*). The OpenSSL configuration file has multiple sections. Each section is used for a different purpose. The sections include the following positions:

– **ca, CA_default**: Defines certification authority configuration.
– **policy_any**: Defines request policies.
– **req, req_dn**: Defines request defaults.

In our examples the configuration file (*openssl.conf*) has the following entries:

```
[ req ]
prompt                          = yes
default_bits                    = 4096
distinguished_name              = req_dn
x509_extensions                 = req_ext
string_mask                     = utf8only

[ ca ]
default_ca                      = CA_default

[ CA_default ]
dir                             = C:/CA
certs                           = $dir/certs
database                        = $dir/index.txt
new_certs_dir                   = $dir/newcerts

certificate                     = $dir/certs/ca.crt
serial                          = $dir/serial
private_key                     = $dir/private/ca.key

default_md                      = sha256
default_days                    = 365

x509_extensions                 = req_ext
policy                          = policy_any

[ req_dn ]
countryName                     = Country Name (2 letter code)
countryName_default             = DE

organizationName                = Organization Name (company)
organizationName_default        = PHOENIX CONTACT Cyber Security AG

organizationalUnitName          = Organizational Unit Name (department, division)
organizationalUnitName_default  = Support

commonName                      = Common Name (hostname, IP, or your name)

# Not used in our example
#emailAddress                   = Email Address
#localityName                   = Locality Name (city, district)
#stateOrProvinceName            = State or Province Name (full name)

[ policy_any ]
countryName                     = supplied
organizationName                = supplied
organizationalUnitName          = optional
commonName                      = supplied
# Not used in our example
#emailAddress                   = optional
#localityName                   = optional
#stateOrProvinceName            = optional

[ req_ext ]
basicConstraints                = critical, CA:false

[ ca_ext ]
basicConstraints                = critical, CA:true, pathlen:0
keyUsage                        = critical, cRLSign, keyCertSign
```

Section	Option	Description
[req]	This section is called when requesting a certificate by calling the *openssl* command wth the option **req**.	
	prompt	If set to the value **no** this disables prompting of certificate fields and just takes values from the configuration file directly. You should enable this option for being able to enter the *common name* or to modify the default values of the certificate's distinguished name for each requested certificate.
	default_bits	This specifies the default key size in bits. If not specified then 512 is used.
	distinguished_name	This specifies the section containing the distinguished name fields to prompt for when generating a certificate or certificate request. In our example this section is called **[req_dn]**.
	x509_extensions	This specifies the configuration file section containing a list of extensions to add to certificate generated when the **-x509** switch is used. It can be overridden by the **-extensions** command line switch.
	string_mask	This option masks out the use of certain string types in certain fields. If the **utf8only** option is used then only UTF8Strings will be used: this is the PKIX recommendation in RFC2459 after 2003.
[ca]	This section is called when signing certificate requests by calling the *openssl* command with the option **ca**.	
	default_ca	If the **-name** command line option is used, then it names the section to be used. Otherwise the section to be used must be named in the **default_ca** option of the **ca** section of the configuration file, in our example **[CA_default]**.

[CA_default]	This section is called when signing certificate requests by calling the *openssl* command with the option **ca**, referenced by the **default_ca** option of the **ca** section.	
	dir	Root directory of the CA environment. If the configuration file is located in this directory and if you execute all *openssl* commands from this directory, you simply can specify “dir = .”.
	certs	Certificates output directory.
	database	The text database file to use (mandatory parameter). This file must be present though initially it will be empty.
	new_certs_dir	It specifies the directory where new certificates will be placed. Mandatory.
	certificate	Location and filename of the CA certificate.
	serial	A text file containing the next serial number to use in hex. Mandatory. This file must be present and contain a valid serial number.
	private_key	Location and filename of the file which contains the CA’s private key.
	default_md	This option specifies the digest algorithm to use. Any digest supported by the OpenSSL *dgst* command can be used.
	default_days	The default number of days the certificate will be valid. This default value can be overridden by the **-days** command line switch.
	x509_extensions	This specifies the configuration file section containing a list of extensions to add to certificate generated when the **-x509** switch is used. It can be overridden by the **-extensions** command line switch.
[req_dn]	This specifies the parameters containing the distinguished name fields to prompt for when generating a certificate or certificate request, referenced by the **distinguished_name** option of the **req** section. If the **prompt** option in the **req** section is absent or set to **yes** then the section contains field prompting information. <fieldname> is the field name being used, for example commonName (or CN).	
	<fieldname> = “prompt”	The "prompt" string is used to ask the user to enter the relevant details.
	<fieldname>_default =”default field value”	If the user enters nothing then the default value is used if no default value is present then the field is omitted.

[policy_any]	This option defines the CA "policy" to use and needs to be specified by the –policy command line switch. This is a section in the configuration file which decides which fields should be mandatory or match the CA certificate. The policy section consists of a set of variables corresponding to certificate DN fields. If the value is **match** then the field value must match the same field in the CA certificate. If the value is **supplied** then it must be present. If the value is **optional** then it may be present. Any fields not mentioned in the policy section are silently deleted.	
[…_ext]	Those sections specify the X.509 extensions and are referenced by the **x509_extensions** option within the configuration file (section **[req]** and **[CA_default]**). It can be overridden by the **-extensions** command line switch.	
	basicConstraints	This flag is used to determine whether the certificate can be used as a CA certificate.

4.4 Create the CA Certificate and Key

Now, that all initial configuration is done, we may create a self signed certificate, that will be used as our CA certificate. In other words, we will use this to sign other certificate requests.

Switch to the CA root directory. From this directory we can issue all **openssl commands** because our OpenSSL configuration file (*openssl.conf*) is located here.

Syntax to create the CA certificate and private key:

```
openssl req -new  -config <filename> -x509 -extensions <section> -keyout
<filename> -out <filename> -days <nn>
```

Option	Description
req	The *req* command primarily creates and processes certificate requests. It can instead create self signed certificates when the option **–x509** is specified.
-new	This option generates a new certificate request.
-config <filename>	This allows an alternative configuration file to be specified.
-x509	This option outputs a self signed certificate instead of a certificate request.
-extensions <section>	Specifies the section in the openssl configuration file (specified by **-config <filename>**) where the X.509 certificate extensions are defined.
-keyout <filename>	Filename of the CA's private key. Although it is protected with a pass phrase you should restrict access to it, so that only authorized users can read it.

Example:

```
C:\CA>openssl req -new -config openssl.conf -x509 -extensions ca_ext -keyout
private/ca.key -out certs/ca.crt -days 3640
Generating a 4096 bit RSA private key
.....................................................++
.....................................++
writing new private key to 'private/ca.key'
Enter PEM pass phrase: - enter a strong pass phrase to use for this key
Verifying - Enter PEM pass phrase: - reenter the pass phrase again for verification
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]: - we have kept the default value
Organization Name (company) [PHOENIX CONTACT Cyber Security AG]: - we have kept
the default value
Organizational Unit Name (department, division) [Support]: - we have kept the default
value
Common Name (hostname, IP, or your name) []:CA – we have entered the common name
for the CA certificate


C:\CA>
```

Two files are created:

– **certs/ca.crt**: This is the CA's certificate and can be publicly available and of course world readable.
– **private/ca.key**: This is the CA's private key. Although it is protected with a pass phrase you should restrict access to it, so that only authorized users may have access to it.

4.5 Create a Certificate Request for the mGuard

For obtaining a valid mGuard certificate you need to create a certificate request first and then sign it with the CA certificate (explained in chapter “Sign the mGuard’s Certificate Request with the CA” on page 116).

Syntax for creating a certificate request for the mGuard:

```
openssl req –new -config <filename> -keyout <filename> -out <filename> -days <nn>
```

Option	Description
req	The *req* command primarily creates and processes certificate requests.
-new	This option generates a new certificate request.
-config <filename>	This allows an alternative configuration file to be specified.
-keyout <filename>	Filename of the mGuard private key. Although it is protected with a pass phrase you should restrict access to it, so that only authorized users can read it.
-out <filename>	Filename of the mGuard certificate.
-days <nn>	The number of days the certificate should be valid.

Example:

```
C:\CA>openssl req -new -config openssl.conf -keyout private/mGuard.key -out
mGuard.csr -days 364
Generating a 4096 bit RSA private key
.....++
.....
+
writing new private key to 'private/mGuard.key'
Enter PEM pass phrase: - enter a strong pass phrase to use for this key
Verifying - Enter PEM pass phrase: - reenter the pass phrase again for verification
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]: - we have kept the default value
Organization Name (company) [PHOENIX CONTACT Cyber Security AG]: - we have kept
the default value
Organizational Unit Name (department, division) [Support]: - we have kept the default
value
Common Name (hostname, IP, or your name) []:mGuard – enter the common name for
the mGuard certificate

C:\CA>
```

Two files are created:

– **mGuard.csr**: This is the certificate request which needs to be signed by the CA certificate.
– **private/mGuard.key**: This is the private key, which is not protected with a pass phrase.

4.6 Sign the mGuard's Certificate Request with the CA

The mGuard's certificate request needs to be signed by the CA to become a valid certificate.

Syntax for signing the mGuard's certificate request with the CA:

```
openssl ca –config <filename> -out <filename> -infiles <filename>
```

Option	Description
ca	The *ca* command is a minimal CA application. It can be used to sign certificate requests in a variety of forms and generate CRLs it also maintains a text database of issued certificates and their status.
-config <filename>	This allows an alternative configuration file to be specified.
-out <filename>	Filename of the signed mGuard certificate.
-infiles <filename>	Filename of the mGuard's certificate request. This must be the last option.

Example:

```
C:\CA>openssl ca -config openssl.conf -out certs/mGuard.crt -infiles mGuard.csr
Using configuration from openssl.conf
Enter pass phrase for C:/CA/private/ca.key: - enter the pass phrase of the CA's private key
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
organizationName      :ASN.1 12:'PHOENIX CONTACT Cyber Security AG'
organizationalUnitName:ASN.1 12:'Support'
commonName            :ASN.1 12:'mGuard'
Certificate is to be certified until Jul 7 09:02:23 2018 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

C:\CA>
```

After all this is done two new files are created:

- **certs/mGuard.crt**: This is the mGuards's certificate, which can be made available publicly.
- **newcerts/01.pem**: This is exactly the same certificate, but with the certificate's serial number (hex number) as filename. For subsequent requests the number is incremented. This file is not needed anymore and can be removed.

Now you can delete the mGuard's certificate request (*mGuard.csr*). It's no longer needed.

4.7 Creating the mGuard's PKCS#12 file (Machine Certificate)

This file combines the private and public key and is the mGuard's machine certificate which needs to be imported through the mGuard menu **Authentication >> Certificates >> Machine Certificates**. You'll be prompted to enter a password which protects the PKCS#12 export of the certificate against unauthorized usage.

Following is the syntax to create the mGuard machine certificate:

```
openssl pkcs12 -export -in <filename> -inkey <filename> -out <filename>
```

Option	Description
pkcs12	The *pkcs12* command allows PKCS#12 files to be created and parsed.
-export	This option specifies that a PKCS#12 file will be created rather than parsed.
-in <filename>	The filename to read the certificate from. The format of the file must be PEM. This is the mGuard's certificate you have created in the previous step.
-inkey <filename>	File to read private key from. This is the file which contains the private key of the mGuard's certificate.
-out <filename>	The filename to write certificates and private keys to. They are all written in PEM format.

Example:

```
C:\CA>openssl pkcs12 -export -in certs/mGuard.crt -inkey private/mGuard.key -out
certs/mGuard.p12

Enter pass phrase for private/mGuard.key: - enter the password of the mGuard's private
key

Enter Export Password: - enter a strong pass phrase to use for this export

Verifying - Enter Export Password: - reenter the pass phrase again for verification


C:\CA>
```

This command will create a file called **certs/mGuard.p12**, containing the mGuard certificate public and private key. The file is protected by the entered password.

4.8 Example: VPN connection between two mGuard devices

We assume that you already have setup the CA environment, configured the OpenSSL's configuration file (*openssl.conf*) and created the CA certificate and key. (As described in the previous chapters.)

Step 1: Create a certificate request for each mGuard

mGuard 1

```
openssl req -new -config openssl.conf -keyout private/mGuard1.key -out
mGuard1.csr -days 364
```

mGuard 2

```
openssl req -new -config openssl.conf -keyout private/mGuard2.key -out
mGuard2.csr -days 364
```

Step 2: Sign each certificate request with the CA

mGuard 1

```
openssl ca -config openssl.conf -out certs/mGuard1.crt -infiles mGuard1.csr
```

mGuard 2

```
openssl ca -config openssl.conf -out certs/mGuard2.crt -infiles mGuard2.csr
```

The two certificates **certs/mGuard1.crt** and **certs/mGuard2.crt** are created. **mGuard1.crt** needs to be imported on mGuard **2** as connection certificate through the menu **IPsec VPN >> Connections >> Authentication**. **mGuard2.crt** on **mGuard 1** correspondingly.

Step 3: Obtain the machine certificate for each mGuard

mGuard 1

```
openssl pkcs12 -export -in certs/mGuard1.crt -inkey private/mGuard1.key -out
certs/mGuard1.p12
```

mGuard 2

```
openssl pkcs12 -export -in certs/mGuard2.crt -inkey private/mGuard2.key -out
certs/mGuard2.p12
```

The two exports **certs/mGuard1.p12** and **certs/mGuard2.p12** are created.

mGuard1.p12 needs to be imported on mGuard **1** as machine certificate through the menu **Authentication >> Certificates >> Machine Certificates**. **mGuard2.p12** on mGuard **2** correspondingly.

5 Create X.509 certificates with XCA

Document-ID: 108396_en_01
Document-Description: AH EN X.509 CERT XCA

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

Contents of this document

This section explains briefly how to create X.509 certificates using the tool *XCA*.

XCA provides much more functionality than explained in this document. Please refer to the XCA documentation for further information (http://xca.sourceforge.net/xca.html – 15.09.2017). You can download XCA from http://xca.sourceforge.net. The screenshots and descriptions in this chapter are related to XCA v1.3.2.

5.1 Introduction

The enrollment of certificates requires a certification authority (CA) which issues public key certificates for a specific period of time. A CA can be a private (in-house) CA, run by your own organization, or a public CA. A public CA is operated by a third party that you trust to validate the identity of each client or server to which it issues a certificate.

There are several tools available for creating and managing certificates, as for example *Microsoft Certification Authority (CA) Server*, *OpenSSL* and *XCA*.

This application note explains how to create X.509 certificates with the tools **OpenSSL** and **XCA** for setting up a VPN connection using X.509 certificates as authentication method.

The scope of this document is not to be a complete user's guide for the described tools. It shall help you getting familiar with them and to create the required certificates in a short term.

5.1.1 XCA - X Certificate and key management

XCA is intended for the creation and management of X. 509 certificates, certificate requests, RSA, DSA and EC private keys, smart cards and CRLs. Everything that is required for a CA is implemented. All CAs can sign sub-CAs recursively.

For enterprise-wide use, templates are available that can be used and adapted to generate certificates or certificate request. All crypto data is stored in an endian-agnostic file format portable across operating systems.

5.2 Create an XCA database

To create X.509 certificates and keys using XCA you need to create a database first. Proceed as follows:

1. Click **File >> New DataBase**.
2. Specify the filename and the storage location of the database.
3. Click **Save**.
4. Enter a password which protects the database against unauthorized usage. The password will be requested every time you open the XCA database.

5.2.1 Open an XCA database

When restarting XCA, you need to reconnect to a database first. To open an already created database, proceed as follows:

1. Click **File >> Open DataBase**.
2. Select the desired database (file **.xdb*).
3. Click **Open**.

5.2.2 Set default hash algorithm

NOTE: Phoenix Contact recommends using secure and up to date encryption and hash algorithms, as stated in the mGuard Software Reference Manual, available at phoenixcontact.net/products (search for "UM EN MGUARD", choose a product and select the manual in the download area).

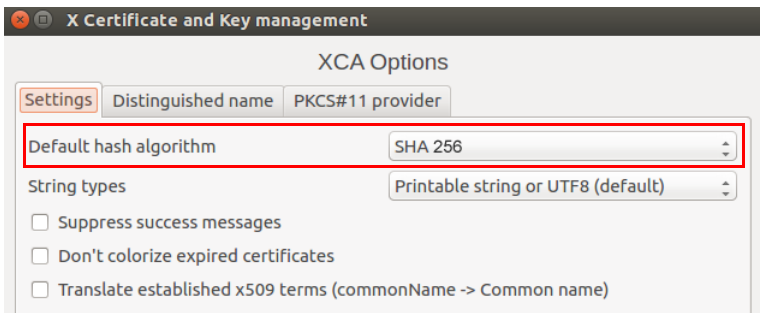
Before you start creating certificates, you should set the default hash algorithm to **SHA 256**. If you don't set the default hash algorithm to SHA 256 you will need to do it every time creating a new certificate.

NOTE: Not all appliances support the functionality of the SHA 2 family

If you are unsure, if all of your appliances support the functionality of the SHA 2 family, the less secure SHA 1 algorithm might be used instead (not recommended by PHOENIX CONTACT and not in accordance with ANSSI-CSPN-2016-09).

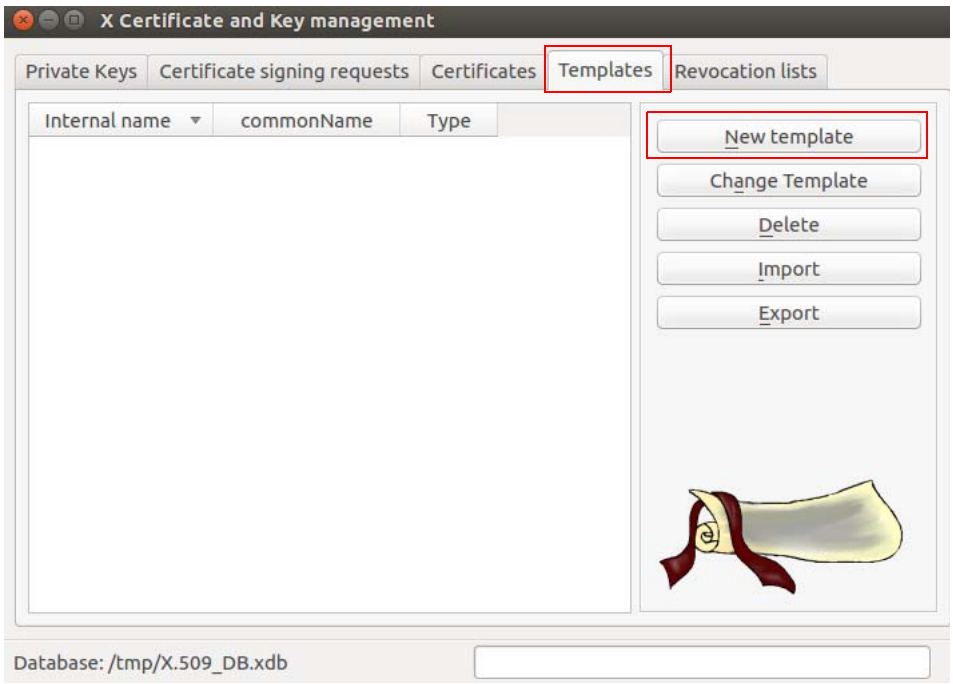
Proceed as follows:

- Click **File >> Options** and set the default hash algorithm to SHA 256 (or the algorithm you will use in your setup).



5.3 Create a certificate template

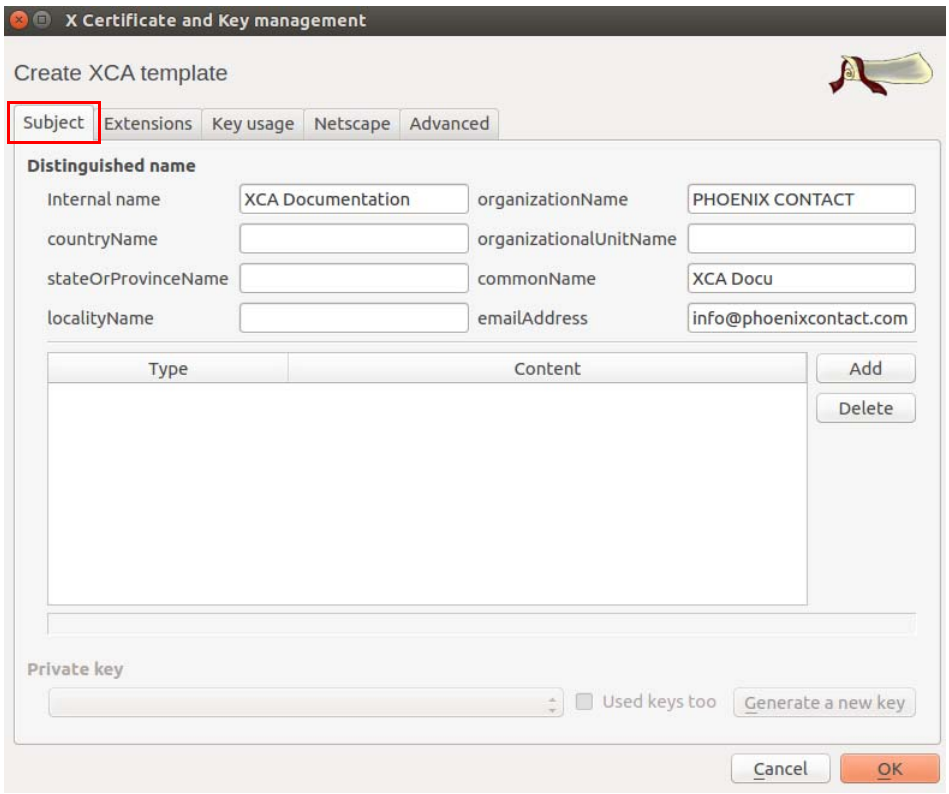
If you need to create more than one certificate it is useful to define a template for consistency reasons and less typing. This template can be used when creating the certificates.



Proceed as follows:

1. Move to the tab **Templates**.
2. Click **New template**.
3. Select the **Preset Template Values** and click **OK**.

5.3.1 Create XCA template >> Tab: Subject

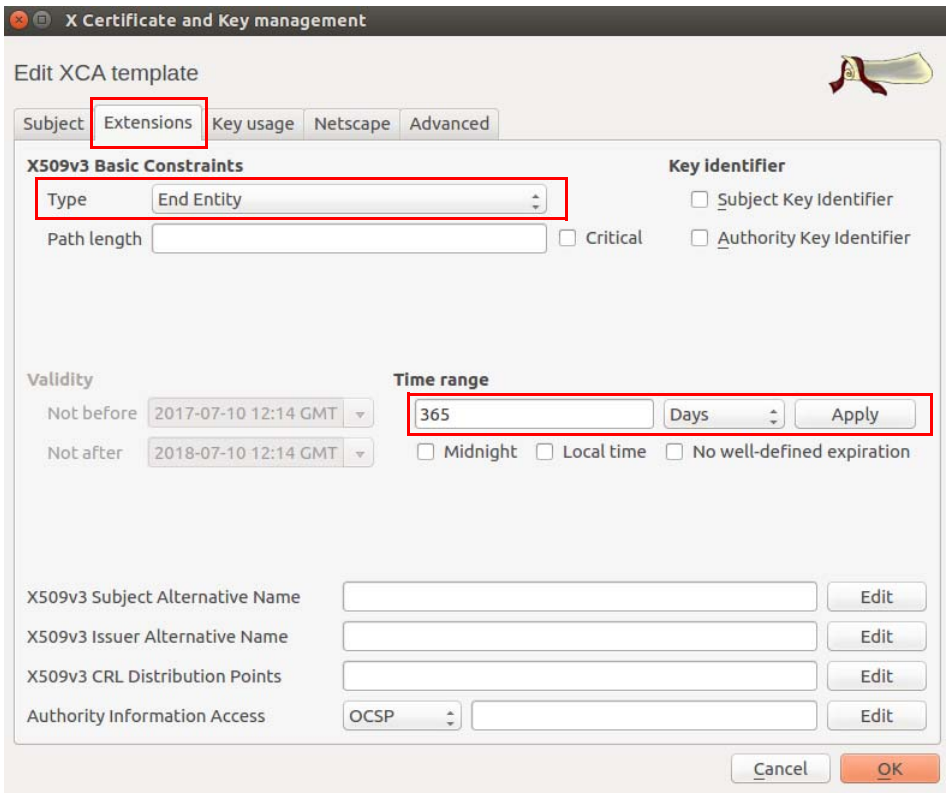


Proceed as follows:

1. Move to the tab **Subject**
2. Use the entry fields from **Internal name** to **emailAddress** for entering the identifying parameters that shall be common for all certificates.

   The template will be stored in *XCA* under the **Internal name**.
3. Move to the tab **Extensions**.

5.3.2 Create XCA template >> Tab: Extensions

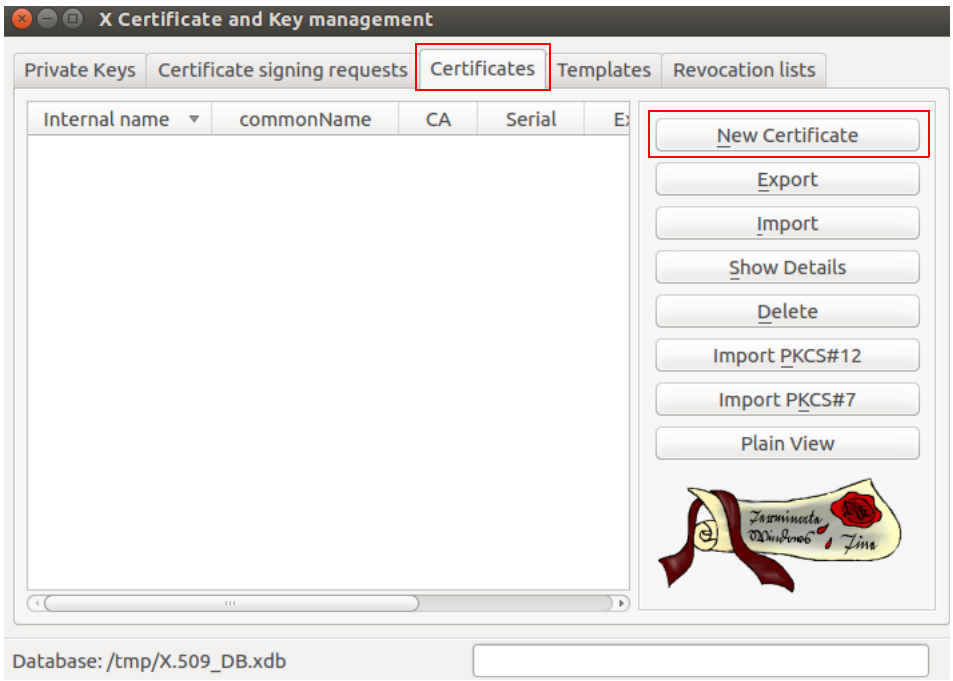


Proceed as follows:

1. In Section **X509v3 Basic Constraints**:
   - Set the **Type** to *End Entity* if you want to use the template for creating client certificates.
   - Set the **Type** to *Certification Authority* if the template should be used for creating CA certificates.
2. In Section **Time Range**:
   - Set the default lifetime of the certificates and click **Apply**.
3. Click **OK** to create the template.

5.4 Create a CA Certificate

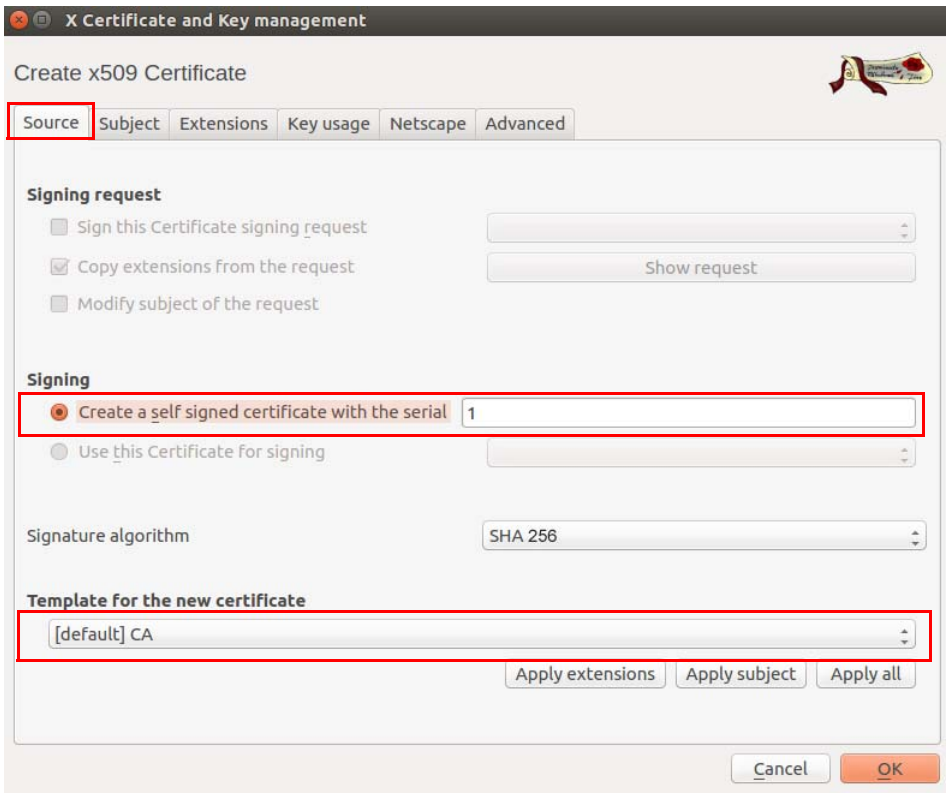
If you don't use self signed client certificates, a client certificate must be signed by the CA certificate to become a valid certificate. Therefore you need to create the CA certificate first before creating the client certificates. The CA certificate is a self signed certificate.



Proceed as follows:

1. Move to the tab **Certificates**.
2. Click **New Certificate**.

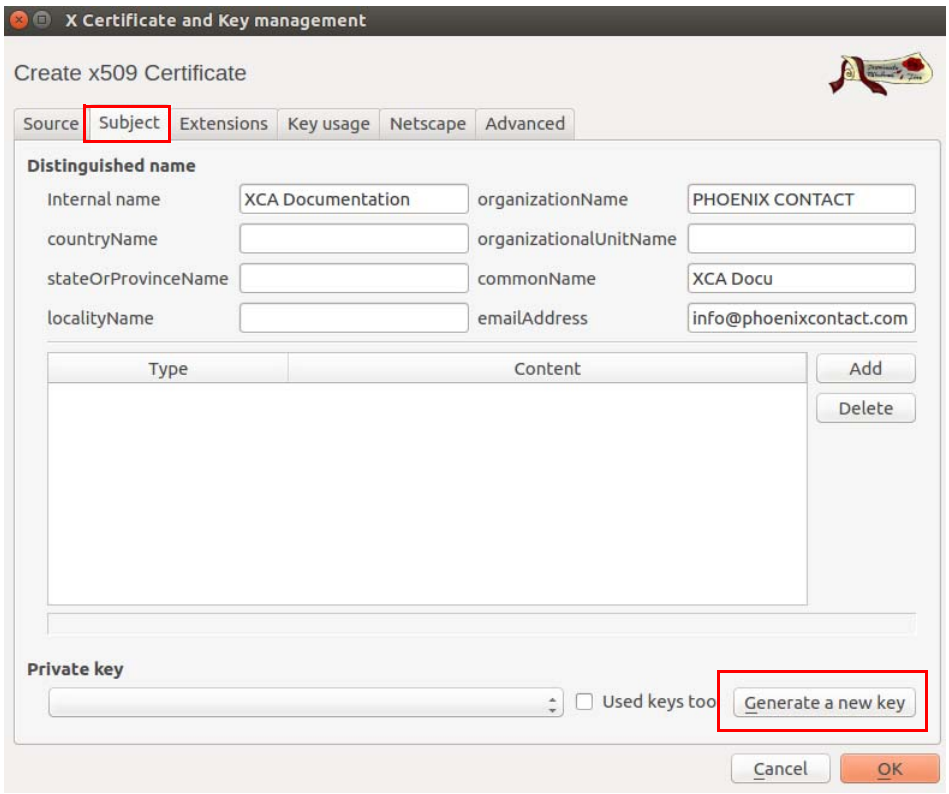
5.4.1 Create x509 (CA) Certificate >> Tab: Source



Proceed as follows:

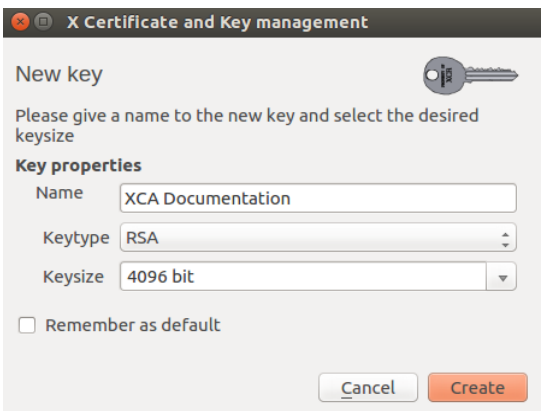
1. Move to the tab **Source**.
2. In Section **Signing**: Ensure that **Create a self signed certificate with the serial** is selected.
3. You may enter a serial number for the certificate or leave the default value.
4. In Section **Template for the new certificate**: If you have created a template for creating CA certificates, you may select it and click **Apply**.
5. Move to the tab **Subject**.

5.4.2 Create x509 (CA) Certificate >> Tab: Subject



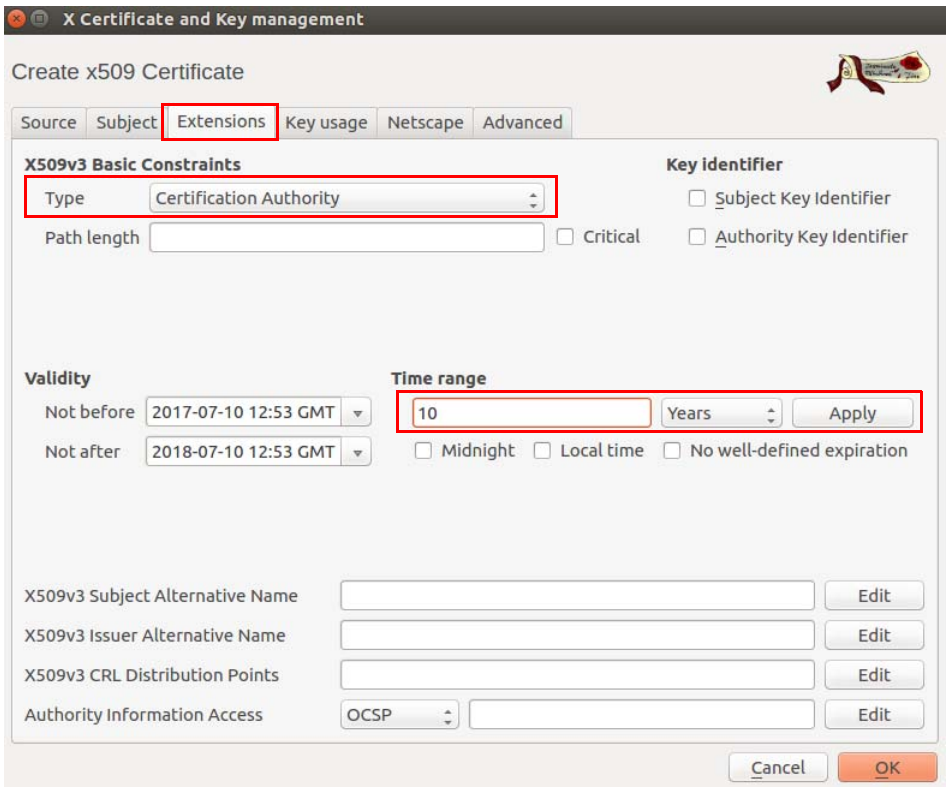
Proceed as follows:

1. In Section **Distinguished name**: Use the entry fields from **Internal name** to **emailAddress** for entering the identifying parameters of the CA.
2. In Section **Private key**: Click **Generate a new key** for creating the private RSA key for the CA.



3. Enter a **Name** for the key, specify the desired **Keytype** and **Keysize** and click **Create**.
4. Move to the tab **Extensions**.

5.4.3 Create x509 (CA) Certificate >> Tab: Extensions



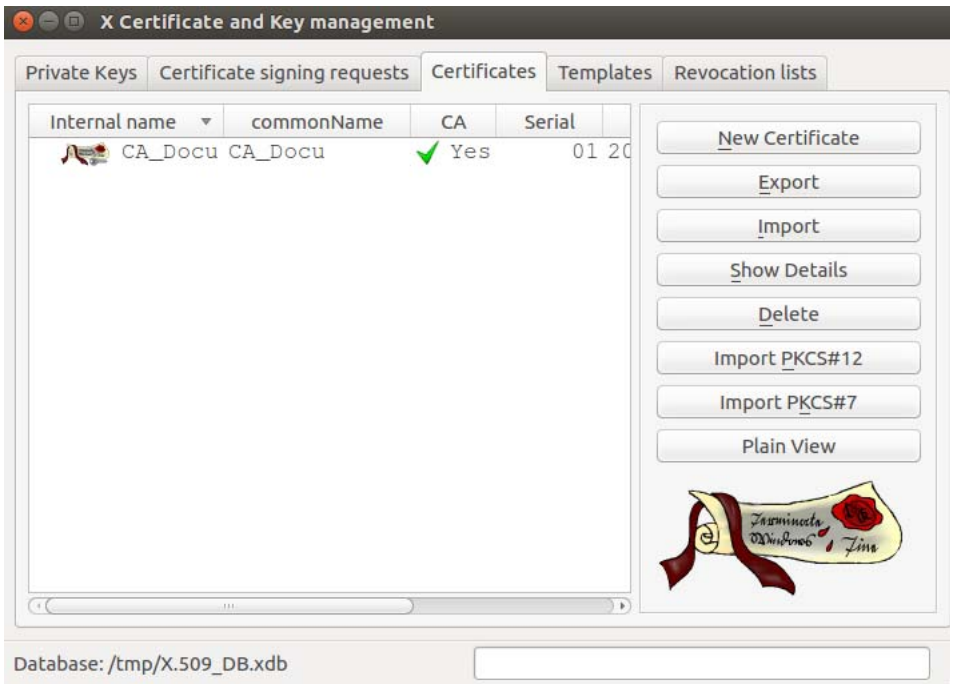
Proceed as follows:

5. In Section **X509v3 Basic Constraints**: Set the **Type** to *Certification Authority*.
6. In Section **Time Range**: Set the default lifetime of the certificates and click **Apply**. For a CA certificate you may want it to last longer than the client certificates so that you do not have to reissue the certificates so often. A lifetime of 10 years might be a good value.
7. Click **Apply**.
8. Click **OK** to create the certificate.
   The CA certificate is displayed in the tab **Certificates**.

5.5 Create a Client Certificate

If you want to create client certificates, you have to create or import a CA certificate first, which will be used to sign the client certificate. By signing the client certificate with the CA certificate, it becomes valid.

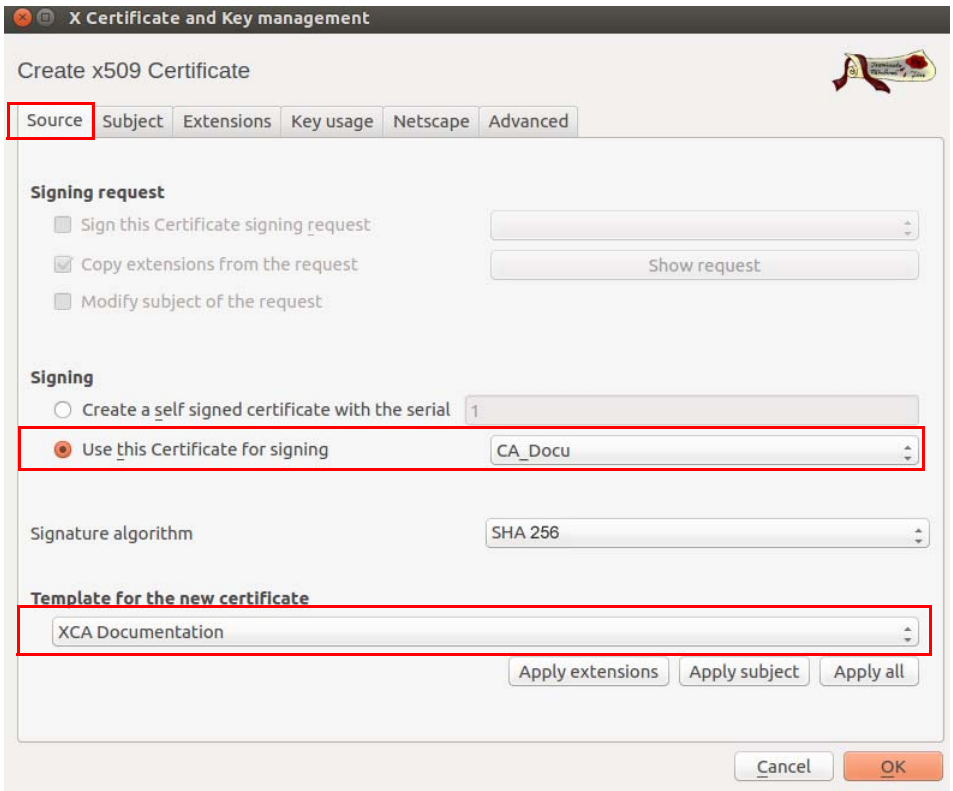
A CA certificate to sign the client certificate must be available in the *XCA* database. If it is not available it has to be created first (see “Create a CA Certificate” on page 127).



Proceed as follows:

1. Move to the tab **Certificates**.
2. Click **New Certificate**.

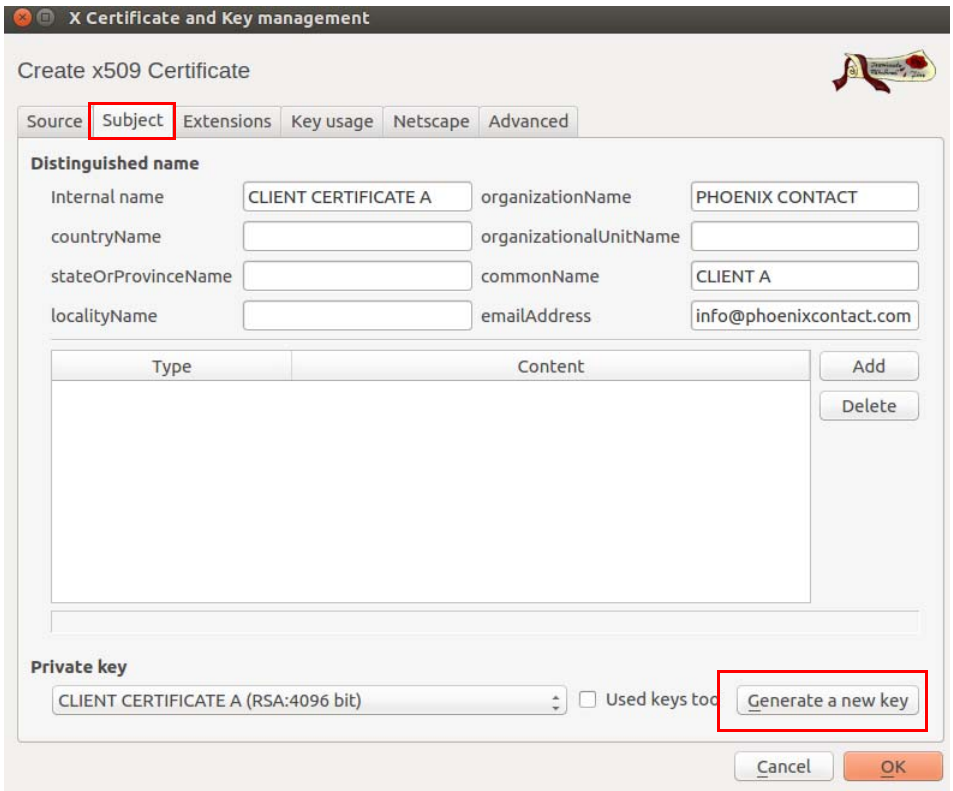
5.5.1 Create x509 (Client) Certificate >> Tab: Source



Proceed as follows:

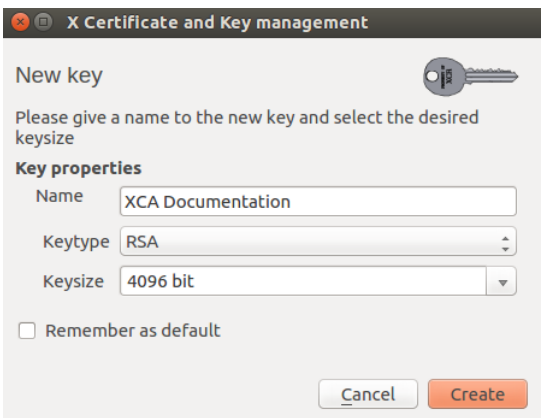
1. Move to the tab **Source**.
2. In Section **Signing**: Ensure that the correct CA is selected in the field **Use this certificate for signing**.
3. In Section **Template for the new certificate**: If you have created a template for creating client certificates, you may select it and click **Apply**.
4. Move to the tab **Subject**.

5.5.2 Create x509 (Client) Certificate >> Tab: Subject



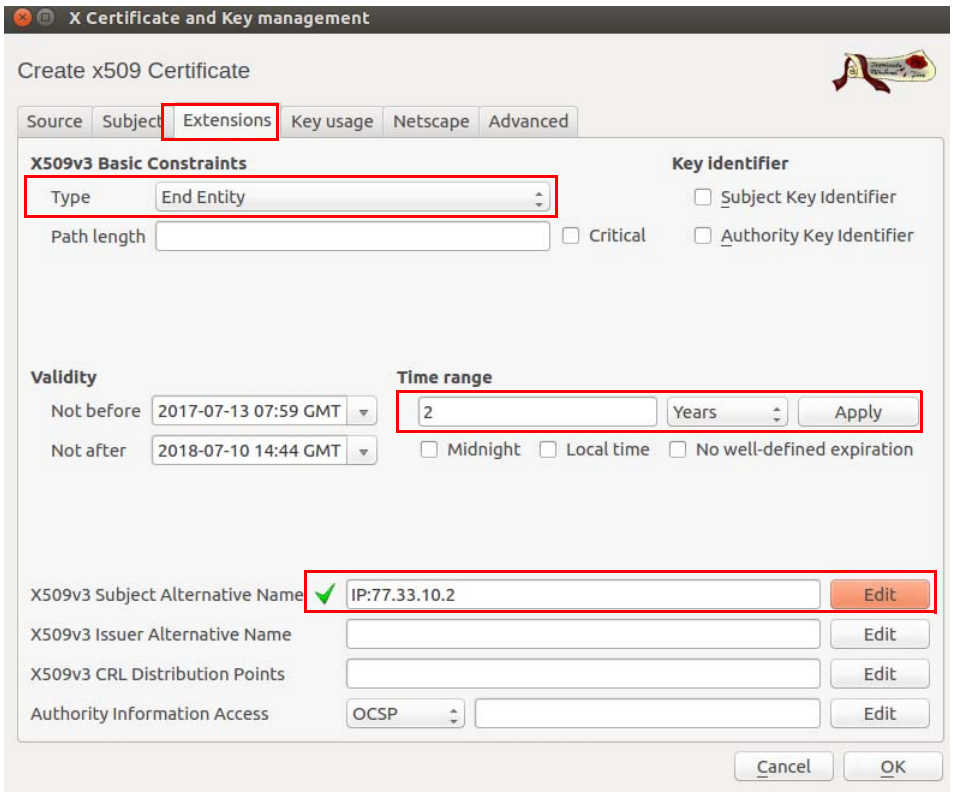
Proceed as follows:

1. In Section **Distinguished name**: Use the entry fields from **Internal name** to **emailAddress** for entering the identifying parameters of the client certificate.
2. In Section **Private key**: Click **Generate a new key** for creating the private RSA key for the certificate.



3. Enter a **Name** for the key, specify the desired **Keytype** and **Keysize** and click **Create**.
4. Move to the tab **Extensions**.

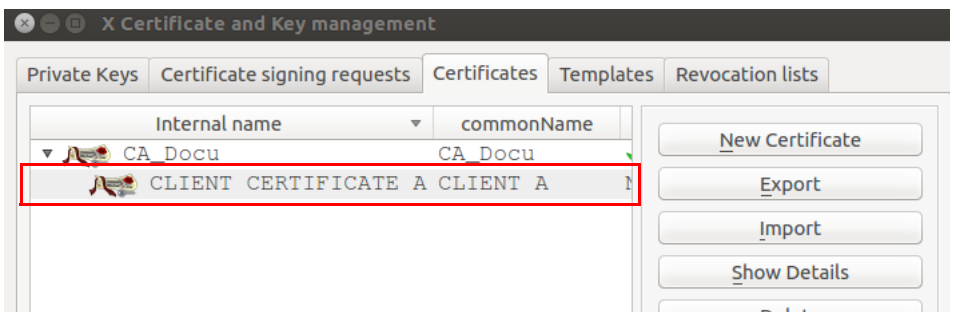
5.5.3 Create x509 (Client) Certificate >> Tab: Extensions



1. In Section **X509v3 Basic Constraints**: Set the **Type** to *End Entity*.
2. In Section **Time Range**: Set the default lifetime of the certificates and click **Apply**.
3. The mGuard uses as default VPN identifier the subject name of the certificate. If you want to use another VPN identifier (e. g. email address, hostname or IP address), this identifier must be present in the certificate as **subject alternative name**.

   To add another identifier, click **Edit** in the line **X509v3 Subject Alternative Name**, select the identifier type (email, DNS or IP), enter its value, click **Add** and then **Apply**.
4. Click **OK** to create the certificate.

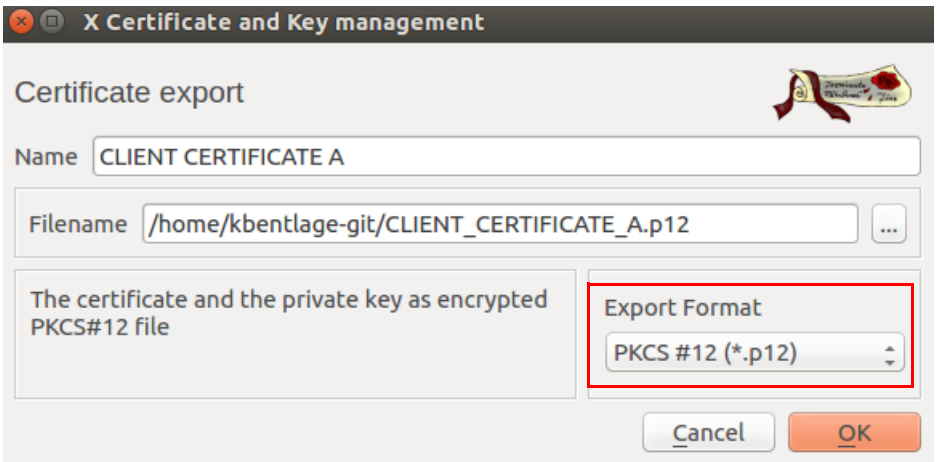
   The client certificate will be displayed in the tab **Certificates** beneath the CA certificate.



5.6 Export a certificate

To export a certificate created with XCA, proceed as follows:

1. Move to the tab **Certificates**.
2. Highlight the certificate that shall be exported.
3. Click **Export**.



4. Select the **Export Format** (PEM or PKCS#12 – see info box below).
5. Specify the desired **Filename** and the location where the export should be stored.
6. Click **OK**.
7. If you export the certificate as PKCS#12 then you'll be prompted to enter a password which protects the export against unauthorized usage. Enter the Password and click **OK**.

PKCS (Public Key Cryptography Standards)

PKCS #12: Personal Information Exchange Syntax v1.1 (defined in **RFC 7292**)

PKCS #12 v1.1 describes a transfer syntax for personal identity information, including private keys, certificates, miscellaneous secrets, and extensions. Machines, applications, browsers, Internet kiosks, and so on, that support this standard will allow a user to import, export, and exercise a single set of personal identity information. This standard supports direct transfer of personal information under several privacy and integrity modes (RFC 7292).

PEM (privacy-enhanced mai) (defined in RFC's 1421 through 1424)

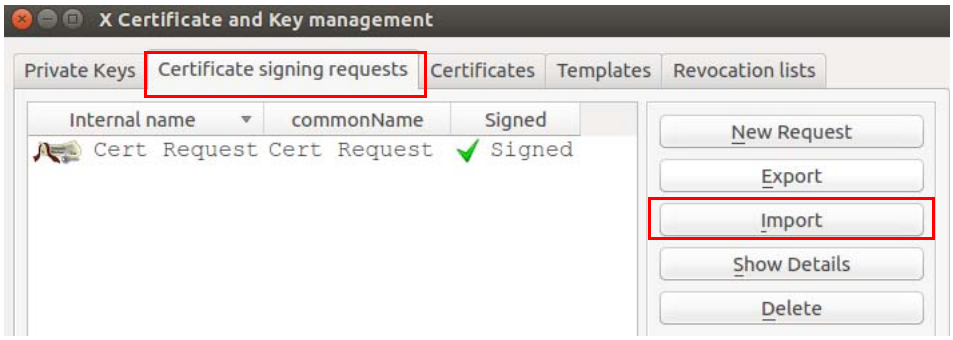
A PEM container may include just the public certificate or an entire certificate chain (including public key, private key, and root certificates).

PEM data is commonly stored in files with a "**.pem**" or "**.cer**" suffix or a "**.crt**" suffix (for certificates), or a "**.key**" suffix (for public or private keys).

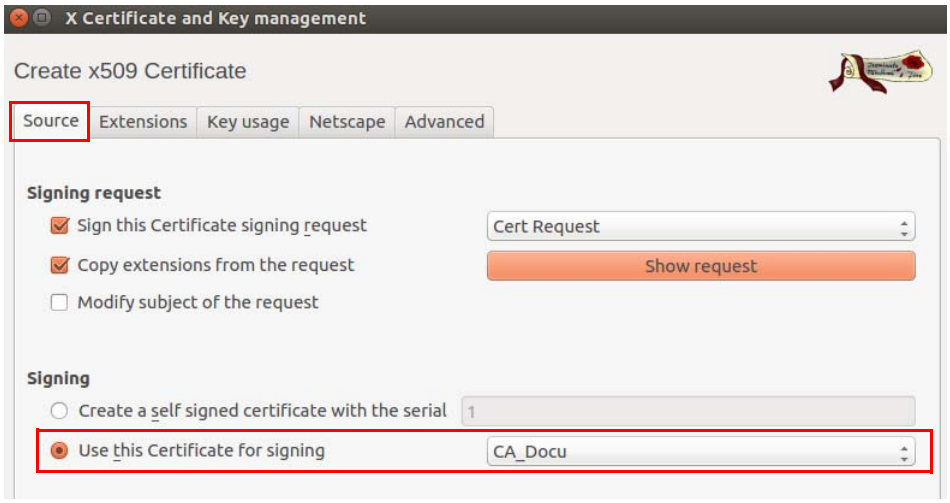
5.7 Sign a Certificate Request with the CA

To sign a certificate request, proceed as follows:

1. Move to the tab **Certificate signing requests**.
2. Click **Import**.
3. Select a certificate request (PKCS#10 file) which should be signed by the CA and click **Open**.
4. The imported certificate request is displayed in the tab **Certificate signing requests**.



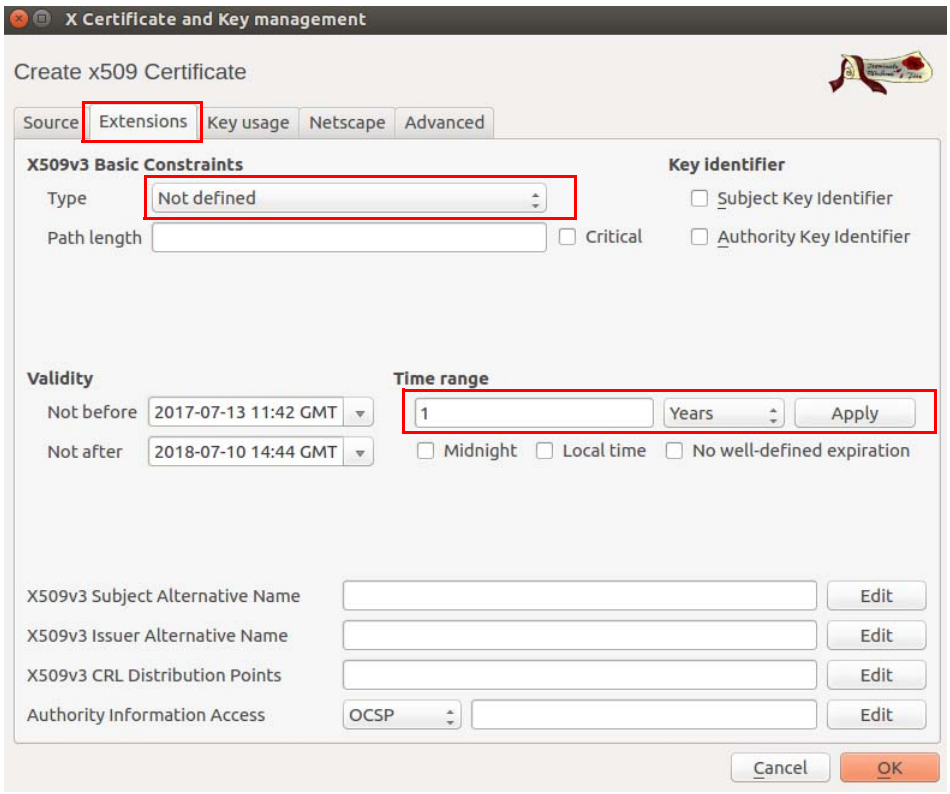
5.7.1 X Certificate and Key Management >> Tab: Source



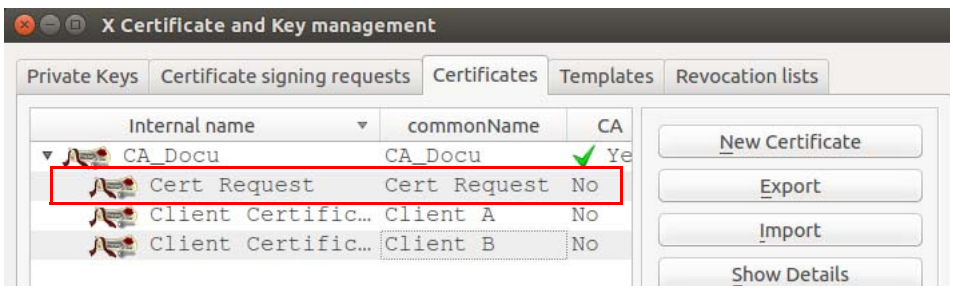
To sign the certificate request, proceed as follows:

1. Move to the tab **Certificate signing requests**.
2. Right click the certificate request and select **Sign** from the context menu.
3. In Section **Signing**: Ensure that the correct CA certificate is selected in the field **Use this certificate for signing**.
4. Move to the tab **Extensions**.

5.7.2 X Certificate and Key Management >> Tab: Extensions



1. In Section **X509v3 Basic Constraints**: Leave **Type** as *Not defined*. Otherwise XCA would copy the certificate extensions twice into the signed certificate.
2. In Section **Time Range**: Set the default lifetime for the new certificate and click **Apply**.
3. Click **OK**.
4. The signed certificate request is displayed in the tab **Certificates** beneath the CA certificate.



5.8 Using a Certificate Revocation List (CRL)

5.8.1 Revoke a certificate

1. Move to the tab **Certificates**.
2. Right click the client certificate that should be revoked and select **Revoke** from the context menu.
3. Edit the parameters and click **OK**.
4. The revoked certificate is marked with a cross icon  and the **Trust state** is *Not trusted*.

5.8.2 Specify the CRL renewal period

1. Move to the tab **Certificates**.
2. Right click the CA and select **CA >> Properties** from the context menu.
3. Enter the desired renewal period into the field **Days until next CRL issuing**.
4. Click **OK**.

5.8.3 Create the CRL

1. Move to the tab **Certificates**.
2. Right click the CA and select **CA >> Generate CRL** from the context menu.
3. Edit the parameters and click **OK**.
4. The CRL is displayed in the tab **Revocation lists**.

5.8.4 Obtain information about a CRL

1. Move to the tab **Revocation lists**.
2. Highlight the CRL and click **Show Details**.

5.8.5 Export of the CRL

1. Move to the tab **Revocation lists**.
2. Highlight the CRL.
3. Click **Export**.
4. Specify the filename and location for storing the CRL.
5. Chose the export format (DER or PEM).
6. Click **OK**.

5.9 Example: VPN connection between two mGuard devices

To create and import the required certificates for a VPN connection between two mGuard devices, proceed as follows:

CA Certificate

- Create a CA certificate as described in chapter “Create a CA Certificate” on page 127.

Client Certificate

- Create a client certificate for **mGuard #1** and a client certificate for **mGuard #2** as described in chapter “Create a Client Certificate” on page 131.

Export certificates

- Export the certificates as described in chapter “Export a certificate” on page 135.

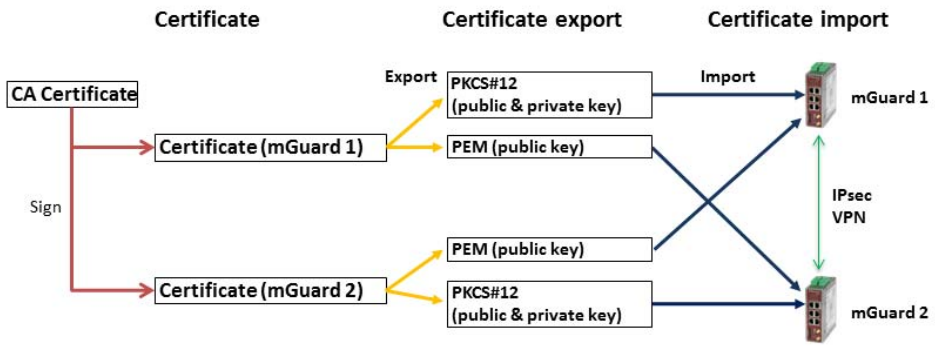
The following exports are required:

– **mGuard #1** as PKCS#12: This export needs to be imported on **mGuard #1** as a *Machine Certificate* (menu: Authentication >> Certificates, tab *Machine Certificates*).

– **mGuard #2** as PKCS#12: This export needs to be imported on **mGuard #2** as a *Machine Certificate* (menu: Authentication >> Certificates, tab *Machine Certificates*).

– **mGuard #1** as PEM: This export needs to be imported on **mGuard #2** as connection certificate (menu: IPsec VPN >> Connections >> (Edit), tab *Authentication*).

– **mGuard #2** as PEM: This export needs to be imported on **mGuard #1** as connection certificate (menu: IPsec VPN >> Connections >> (Edit), tab *Authentication*).



6 Establish an IPsec VPN connection between iOS client and mGuard device

Document-ID: 108393_en_01

Document-Description: AH DE MGUARD IOS SUPPORT

Make sure you always use the latest documentation.

It can be downloaded using the following link phoenixcontact.net/products.

Contents of this document

This document describes the required steps to configure a VPN connection between the mGuard server and an iOS client (iPad or iPhone with iOS version 8.0 or later).

6.1 Introduction

The iOS device acts as a remote client that initiates the IPsec VPN connection. The mGuard acts as the local server and configures and provides the local network for the clients via the XAuth/Mode Config extension.

The VPN connections require the installation of X.509 certificates and keys both on the iOS client and the mGuard device.

For general information on how to configure VPN connections, please refer to the “Software Reference Manual – mGuard Firmware”, available online or in the PHOENIX CONTACT Webshop at: phoenixcontact.net/products. For further information regarding the iOS client, please refer to the corresponding manufacturer's web page.

6.1.1 Requirements

- mGuard device with installed firmware 8.5 or later
- iOS device with installed firmware version 8.0 or later
- All required and signed certificates

How to obtain X.509 certificates?

For further information about certificate management please refer to the application note X.509 CERTIFICATES, available in the PHOENIX CONTACT Webshop at: phoenixcontact.net/products.

6.2 Manage certificates

To establish an IPsec VPN connection between an iOS client and the mGuard server, the devices need to authenticate each other via X.509 certificates.

Table 6-1 Required certificates

Device	Required certificate	Format
mGuard	CA Certificate	**PEM / CER**
	mGuard Machine Certificate (signed by CA)	**PKCS#12**
iOS client	CA Certificate	**PEM / CER**
	iOS Client Certificate (signed by CA)	**PKCS#12**

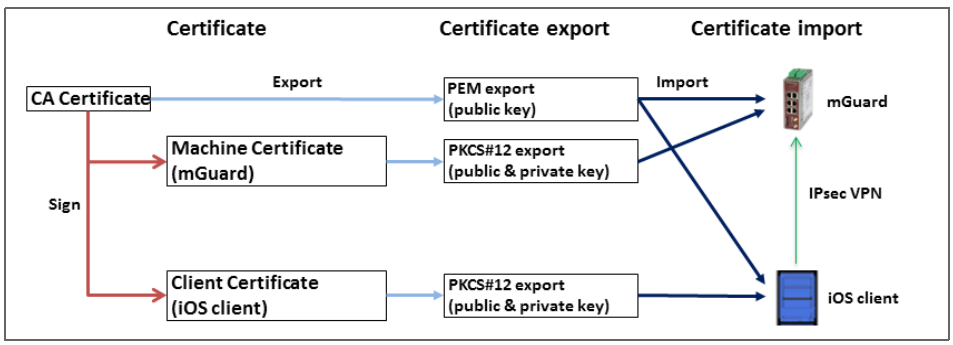


Figure 6-1 Certificate handling for connections initiated by iOS clients

The terms “Machine Certificate” and “Client Certificate” signify an X.509 certificate and it‘s corresponding private key by which the machine/client identifies itself to it‘s peers.

6.2.1 Required certificates on the mGuard device

The following certificates need to be installed on the mGuard device.

1. **CA Certificate (PEM / CER)**

The mGuard verifies the iOS client on the basis of the iOS Client Certificate signed by the CA Certificate.

2. **mGuard Machine Certificate (PKCS#12)**

The iOS client verifies the mGuard on the basis of the mGuard Machine Certificate signed by the CA Certificate. The CA Certificate must therefore be installed on the iOS client.

NOTE: Only for connections from iOS clients: the mGuard Machine Certificate must use the external server IP address or DNS name of the mGuard as the Common Name (CN) of the certificate (see Figure 6-2 and Figure 6-3).

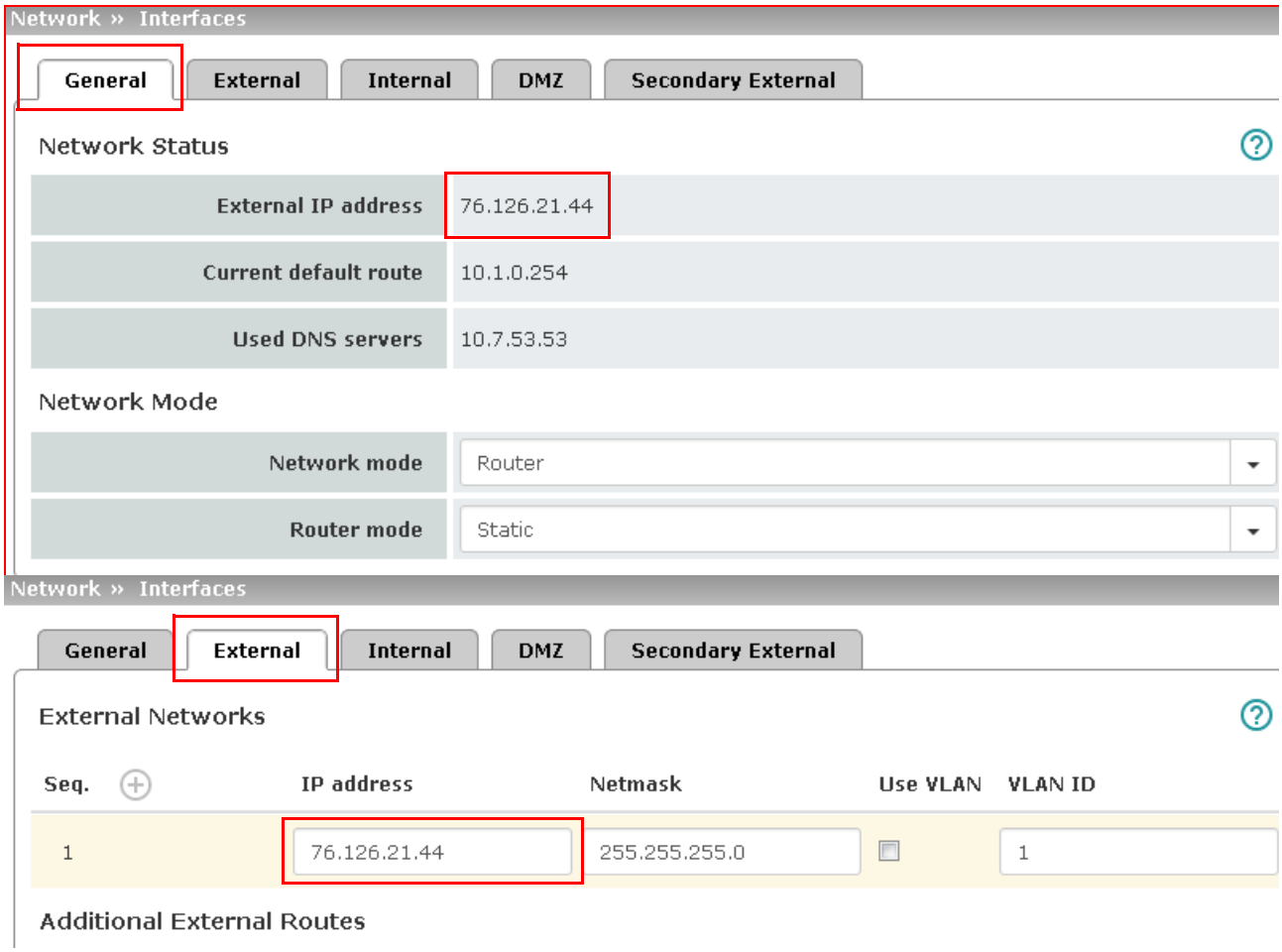


Figure 6-2 Network settings on the mGuard: external IP address highlighted

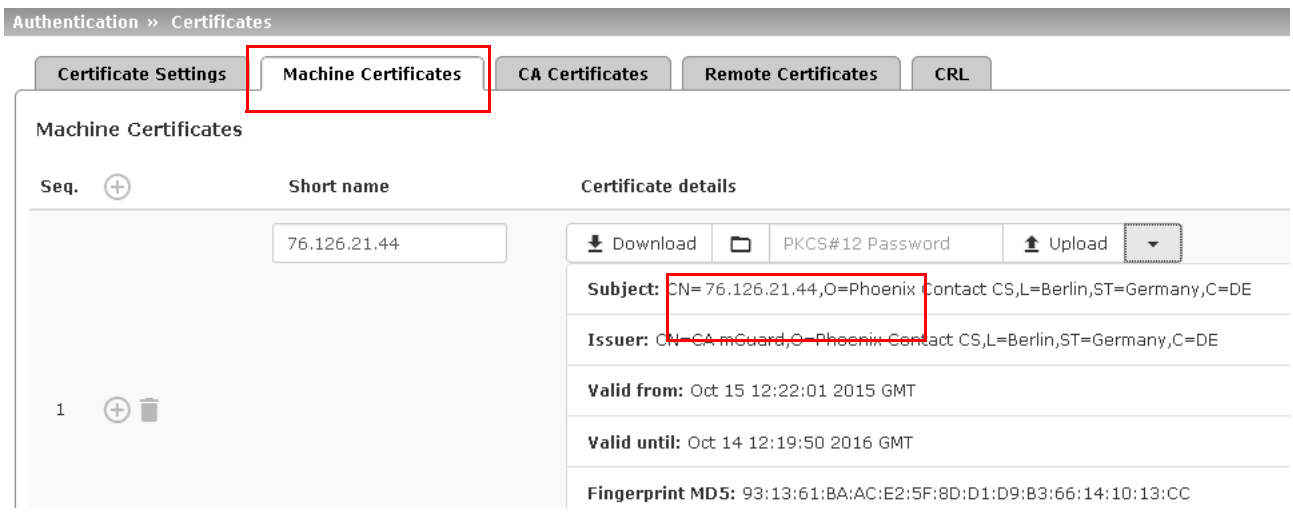


Figure 6-3 Machine Certificate: CN = mGuard's external IP address or DNS name

6.2.2 Required certificates on the iOS client

The following certificates need to be installed on the iOS device (see page 142).

1. **CA Certificate (PEM/CER)**

The iOS client verifies the mGuard server on the basis of the mGuard Machine Certificate signed by the CA.

2. **iOS Client Certificate (PKCS#12)**

The mGuard verifies the iOS client on the basis of the iOS Client Certificate signed by the CA. The signing CA Certificate must therefore be installed on the mGuard.

Because the iOS client ignores the keychain of the PKCS#12 file, the signing CA Certificate must therefore be separately installed on the mGuard.

6.2.3 Install certificates on the mGuard device

Machine Certificate

To upload the mGuard Machine Certificate to the mGuard, proceed as follows:

1. Select the Menu “Authentication >> Certificate” (Tab “Machine Certificates”)
2. Click the icon ⊕ to create a new table row.
3. Click the icon 🗀 .
4. Choose the Machine Certificate (PKCS#12 file) and click “Open”.
5. Enter the password, that has been used to protect the private key of the certificate.
6. Click the button “Upload”.
   ► The uploaded certificate appears in the certificates list.
7. Click “Apply” to save the settings.
   ► The mGuard Machine Certificate has been uploaded and can be used for authentication towards the iOS client (see “Configure mGuard” , Tab “Authentication”).

CA Certificate

To upload the CA Certificate to the mGuard, proceed as follows:

1. Select the menu “Authentication >> Certificate” (Tab “CA Certificates”).
2. Click the icon ⊕ to create a new table row.
3. Click the icon 🗀 .
4. Choose the CA Certificate (PEM or CER file) and click “Open”.
5. Click the button “Upload”.
   ► The uploaded certificate appears in the certificates list.
6. Click “Apply” to save the settings.
   ► The CA Certificate has been uploaded and can be used to authenticate the iOS client certificate (see “Configure mGuard” , Tab “Authentication”).

6.2.4 Install certificates on the iOS client

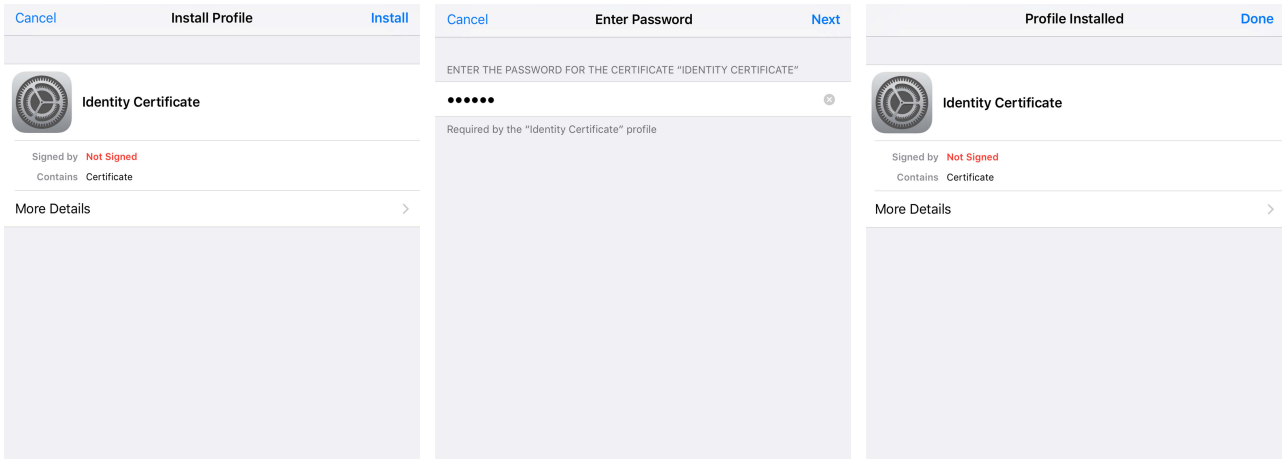


Figure 6-4 Installation of client certificates

Figure 6-5 Installed certificates in the certificate list

To install the iOS Client Certificate or the CA Certificate on the iOS client, proceed as follows:

1. Make the certificate file available on the iOS client.
2. Open the file.
   ► The screen “Install Profile” appears.
3. Click twice on “Install”.
   ► If the certificate has been secured with a secret key (PKCS#12 files), the screen “Enter Password” appears.
4. In this case, enter the password.
5. Click “Next”.
   ► The screen “Profile Installed” appears.
6. Click “Done” to finish the installation of the certificate.
   ► The installed certificate appears in the certificate list.

6.3 Configure VPN connections

6.3.1 Configure mGuard

The IPsec VPN connection between the iOS client and the mGuard will be established using the XAuth/Mode Config extension. The configuration of the iOS client will be configured by the mGuard and communicated to the iOS client.

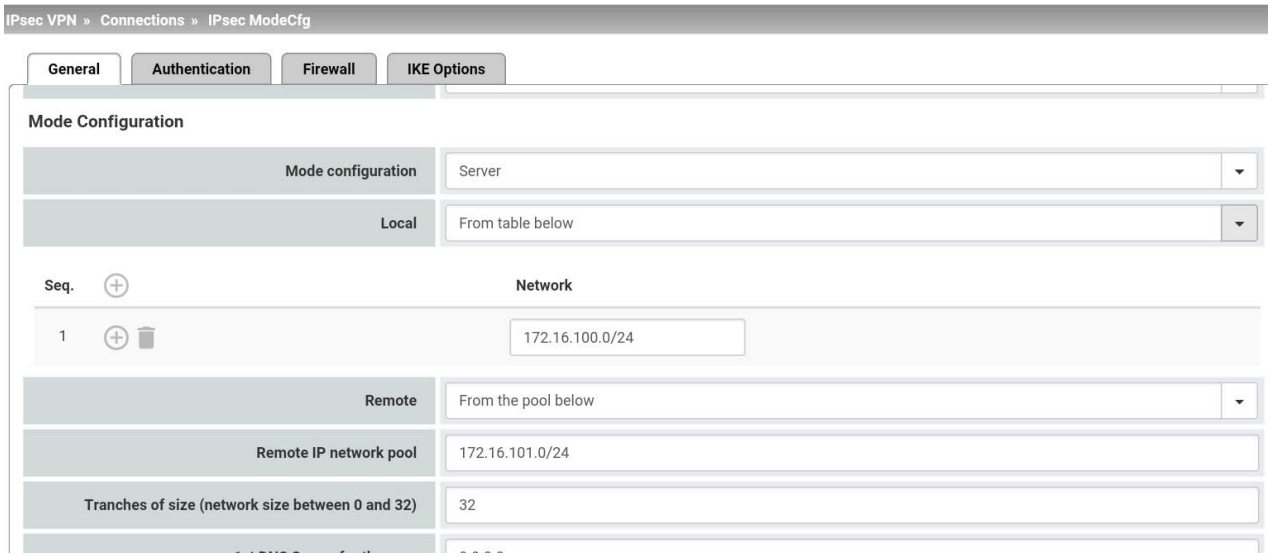


Figure 6-6 mGuard VPN configuration – Mode Configuration

6.3.1.1 Tab “General”

To configure a VPN connection to an iOS client on the mGuard, proceed as follows:

1. Select the menu “IPsec VPN >> Connections”.
2. Click the icon ⊕ to create a new table row.
3. Click the icon ✎ “Edit row”.
   ► The tab “General” appears.
4. Enter a descriptive name for the connection and change further settings optionally.

   Verify that the input field “Address of the remote site’s VPN gateway” contains the value “**%any**” and “Connection startup” is set to “**Wait**” (default values).

5. In section **Mode Configuration** select Mode configuration **Server**.
6. **Local**: Enter the local network(s) on the server side (mGuard) that shall be accessible by the iOS client via VPN connection.
   – **Fixed**: The *Local IP network* must be set to 0.0.0.0/0. In this case, all traffic from the iOS client will be sent over the VPN connection.
   – **From table below**: Only traffic to the *Networks* listed in the *table below* will be send over the VPN connection. On iOS clients, traffic to networks not listed in the *table below* will bypass the VPN connection.
7. **Remote**: Define the network pool (**From the pool below**) from which the mGuard allocates a variable tranche (**Tranches of size**) to be used by the remote client‘s network.

6.3.1.2 Tab “Authentication”

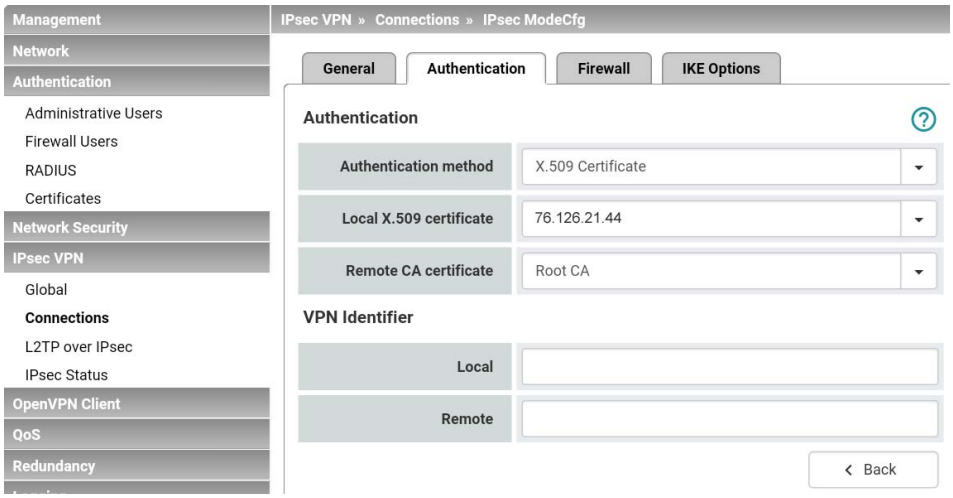


Figure 6-7 mGuard VPN configuration – Authentication

The VPN connection between an iOS client and the mGuard must be authorized by X.509 certificates, that have to be installed on the corresponding devices (see “Manage certificates” on page 142).

To assign the required certificates to a VPN connection, proceed as follows:

1. Select the menu “IPsec VPN >> Connections”.
2. Edit the desired VPN connection (Tab “Authentication”).
3. Select the **Authentication method** “X.509 Certificate”.
4. As the *Local X.509 certificate* select the **mGuard Machine Certificate**.

	Only for connections from iOS clients: The CN of the certificate must correspond with the external IP address or DNS name of the mGuard server.

	The certificate must have been signed by the CA Certificate that has been installed on the iOS client.

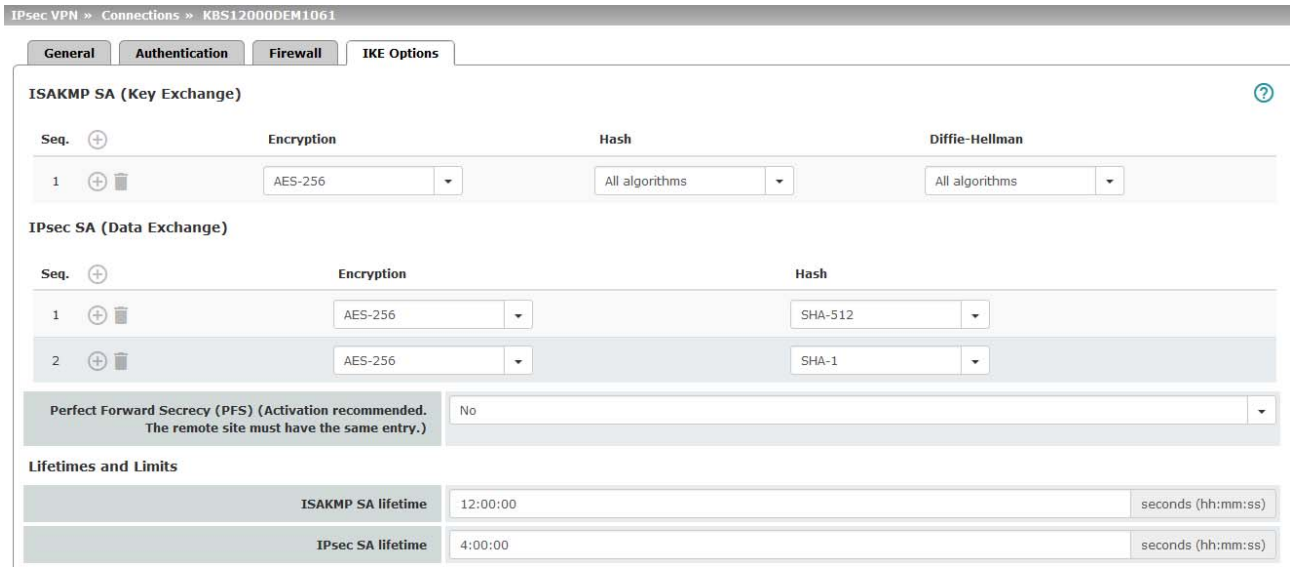
5. As the *Remote CA certificate* select the *CA Certificate* that has been used to sign the **iOS Client Certificate**.
6. Click “Apply” to save the settings.
   ► The VPN connection will be established after being initiated by the iOS client.

6.3.1.3 Tab“Firewall”

The VPN firewall restricts the access through the VPN tunnel. You may configure the VPN firewall if required.

	By default, **any incoming** and **outgoing** traffic will be accepted**.**

6.3.1.4 Tab “IKE Options”



It is necessary to change the default IKE options:

1. Select the menu “IPsec VPN >> Connections”.
2. Edit the desired VPN connection (Tab “IKE Options”).
3. Configure the following settings and leave all other settings on default.

ISAKMP SA (Key Exchange)

- Encryption: AES-256
- Hash: All algorithms
- Diffie-Hellman: All algorithms

IPsec SA (Data exchange)

- Click the icon ⊕ to create two table rows and use the following settings:
  - (Row 1) Encryption: AES-256 | Hash: SHA-512
  - (Row 2) Encryption: AES-256 | Hash: SHA-1

6.3.2 Configure iOS client

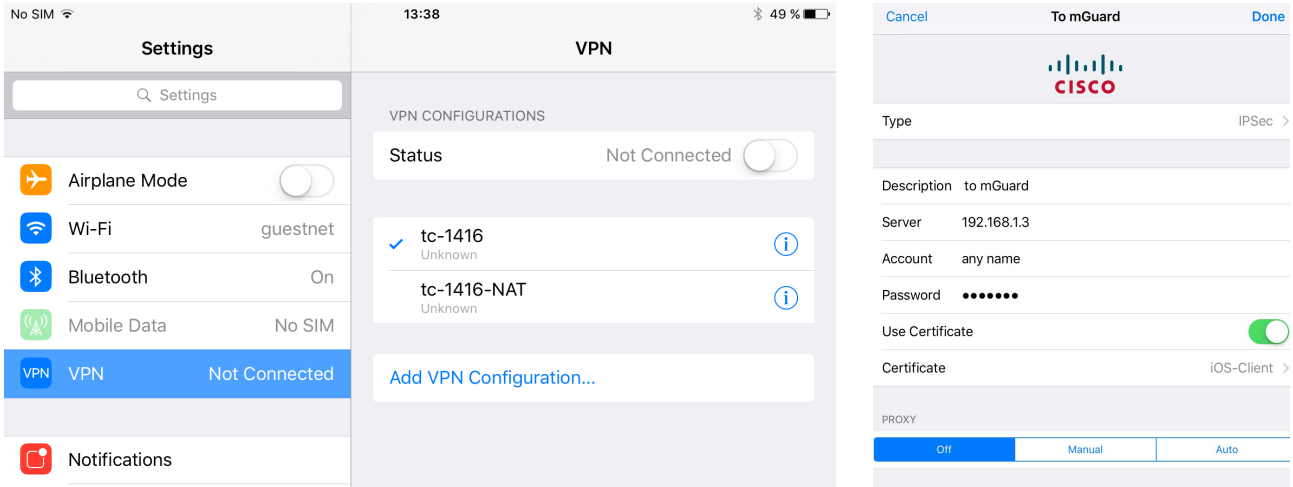


Figure 6-8 iOS client: VPN configuration

To configure an IPsec VPN connection on the iOS client, proceed as follows:

1. Select the menu “Settings >> VPN”.
2. Click “Add VPN Configuration...”.
3. Click “Type”.
4. Select “IPSec” and click “Back”.
5. Fill out the following input fields:
   - Description: A descriptive name for the connection
   - Server: The external IP address or the DNS name of the mGuard server

   This IP address or the DNS name must correspond with the Common Name (CN) of the Machine Certificate of the mGuard server.

   - Account: The Authentication of VPN peers relies on certificates. Thus the account name and password will be **ignored by the mGuard**. To avoid ongoing requests, enter some random text.
   - Password: The password will be **ignored by the mGuard**. Enter random text.
   - Use Certificate: To select a certificate, activate the switch.
6. Click “Certificate”.
   ► A list with all installed certificates appears.
7. Select the appropriate client certificate and click “Back”.
8. Click “Done” to save the configuration.
   ► The VPN configuration has been saved and is ready to be started.

6.4 Start VPN connections on the iOS client

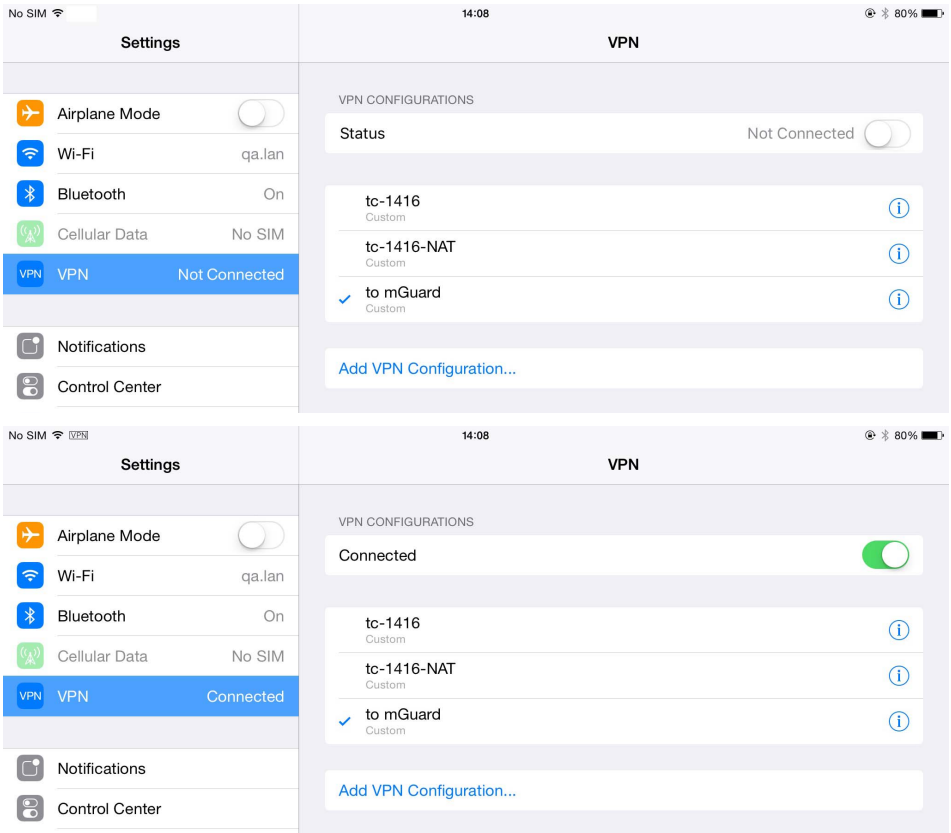


Figure 6-9 Start VPN connection on the iOS client

To start an IPsec VPN connection on the iOS client, proceed as follows:

1. Select the menu “Settings >> VPN”.
2. Click on the name of the appropriate VPN connection.
3. In the area “Status”, click the Button “Not Connected”.
   ► The VPN connection will be established and the status changes from “Not Connected” to “Connected”.

If the connection fails, click the Info icon of the VPN connection to check for errors in the configuration or check your internet connection.

6.5 Check VPN connections on the mGuard

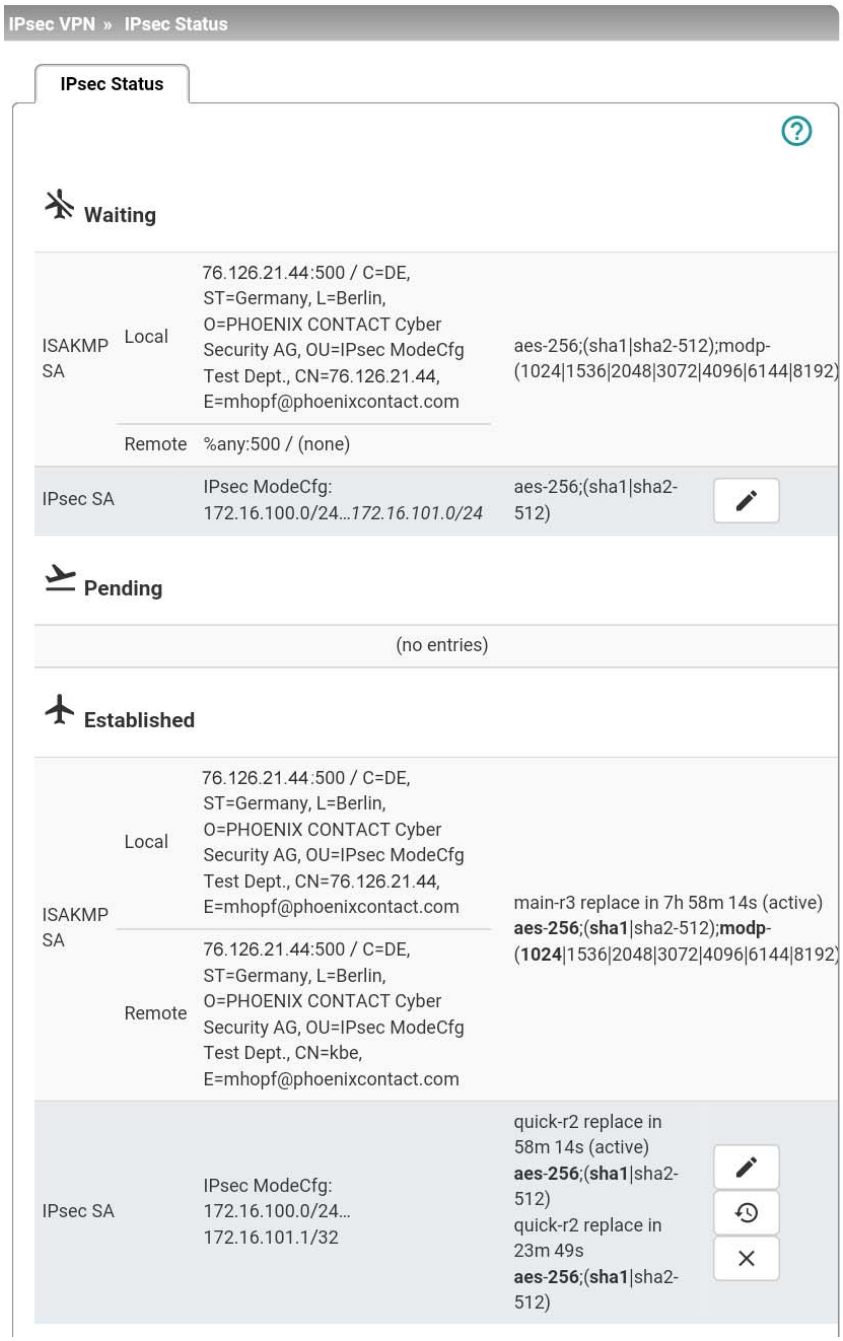


Figure 6-10 IPsec VPN status

To check the status of an IPsec VPN connection, proceed as follows:

- Select the menu “IPsec VPN >> IPsec Status”.
  - ► An established IPsec VPN connection appears in the area “Established”.

7 Establish an IPsec VPN connection between Android client and mGuard device

Document-ID: 108394_en_01

Document-Description: AH DE MGUARD ANDROID SUPPORT

Make sure you always use the latest documentation.

It can be downloaded using the following link phoenixcontact.net/products.

Contents of this document

This document describes the required steps to configure a VPN connection between the mGuard server and an Android client (tablet PC or mobile phone with Android OS version 6.0).

7.1 Introduction

The Android device acts as a remote client that initiates the IPsec VPN connection. The mGuard acts as the local server and configures and provides the local network for the clients via the XAuth/Mode Config extension.

The VPN connections require the installation of X.509 certificates and keys both on the Android client and the mGuard device.

For general information on how to configure VPN connections, please refer to the “Software Reference Manual – mGuard Firmware”, available online or in the PHOENIX CONTACT Webshop at: phoenixcontact.net/products. For further information regarding the Android client, please refer to the corresponding manufacturer's web page.

Settings and user interfaces may look different on different Android devices. They depend on the manufacturer‘s implementation. The present document was created on the basis of the following device: *SAMSUNG SM-T580* with installed Android version 6.0.1.

7.1.1 Requirements

- mGuard device with installed firmware 8.5 or later
- Android device with installed firmware version 6.0
- All required and signed certificates

How to obtain X.509 certificates?

For further information about certificate management please refer to the application note X.509 CERTIFICATES, available in the PHOENIX CONTACT Webshop at: phoenixcontact.net/products.

7.2 Manage certificates

To establish an IPsec VPN connection between an Android client and the mGuard server, the devices need to authenticate each other via X.509 certificates.

Table 7-1 Required certificates

Device	Required certificate	Format
mGuard	CA Certificate	**PEM / CER**
	mGuard Machine Certificate (signed by CA)	**PKCS#12**
Android client	mGuard Machine Certificate (signed by CA)	**PEM / CER**
	Android Client Certificate (signed by CA)	**PKCS#12**

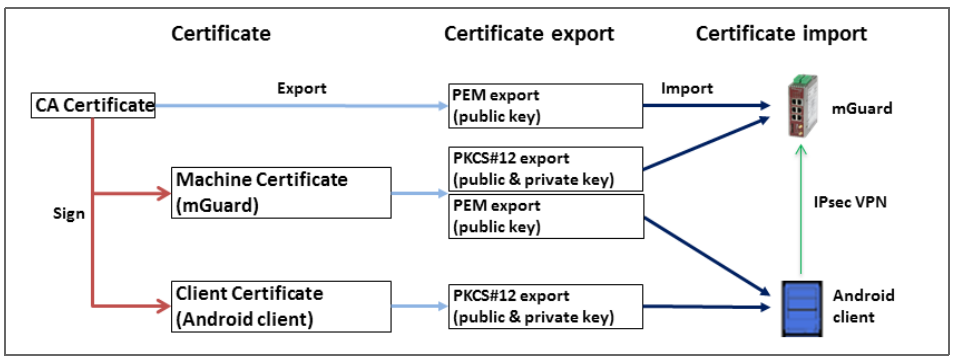


Figure 7-1 Certificate handling for connections initiated by Android clients

The terms “Machine Certificate” and “Client Certificate” signify an X.509 certificate and it‘s corresponding private key by which the machine/client identifies itself to it‘s peers.

7.2.1 Required certificates on the mGuard device

The following certificates need to be installed on the mGuard device.

mGuard Machine Certificate (PKCS#12)

The **Android client** verifies the mGuard on the basis of the mGuard Machine Certificate. The mGuard Machine Certificate must therefore be installed on the Android client.

7.2.2 Required certificates on the Android client

The following certificates need to be installed on the Android device (see page 154).

1. **mGuard Machine Certificate (PEM/CER)**

The Android client verifies the mGuard server on the basis of the mGuard Machine Certificate.

2. **Android Client Certificate (PKCS#12)**

The mGuard verifies the Android client on the basis of the Android Client Certificate signed by the CA. The signing CA Certificate must therefore be installed on the mGuard.

7.2.3 Install certificates on the mGuard device

Machine Certificate

To upload the mGuard Machine Certificate to the mGuard, proceed as follows:

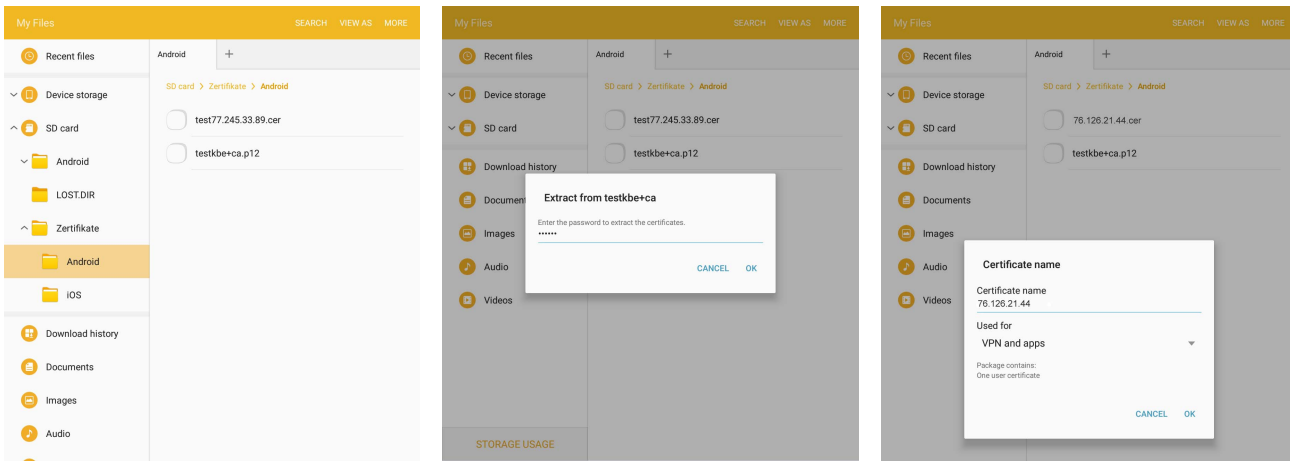
1. Select the menu **Authentication >> Certificates >> Machine Certificates**.
2. Click the icon ⊕ to create a new table row.
3. Click the icon 🗀 .
4. Choose the Machine Certificate (PKCS#12 file) and click “Open”.
5. Enter the password, that has been used to protect the private key of the certificate.
6. Click the button “Upload”.
   ► The uploaded certificate appears in the certificates list.
7. Click “Apply” to save the settings.
   ► The mGuard Machine Certificate has been uploaded and can be used for authentication towards the Android client (see “Configure the mGuard” , Tab “Authentication”).

CA Certificate

To upload the CA Certificate to the mGuard, proceed as follows:

1. Select the menu **Authentication >> Certificates >> CA Certificates**.
2. Click the icon ⊕ to create a new table row.
3. Click the icon 🗀 .
4. Choose the CA Certificate (PEM or CER file) and click “Open”.
5. Click the button “Upload”.
   ► The uploaded certificate appears in the certificates list.
6. Click “Apply” to save the settings.
   ► The CA Certificate has been uploaded and can be used to authenticate the Android client certificate (see “Configure the mGuard” , Tab “Authentication”).

7.2.4 Install certificates on the Android client



To install the **Android Client Certificate** (PKCS#12 file with signing CA certificate) and the **mGuard Machine Certificate** (PEM / CER file) on the Android client, proceed as follows:

1. To use the VPN feature on the Android device, you must set the screen lock type pattern, PIN, or password.
2. Make the certificate files available on the Android client.
3. Open the PKCS#12 file (**.p12)* to extract and install the Android Client and signing CA Certificates.
   ► The screen “Extract from <certificate name>” appears.

   If the screen does not appear and the device displays the content of the file instead, download the file to the storage of your device or make it available via SD card. Open the file from the corresponding directory.
4. Enter the password and click “OK”.
   ► The screen “Certificate name” appears.
5. Optional: Assign a new name to the certificate to easily locate the certificate in the certificate list.
6. Click “OK” to finish the installation of the Android Client and signing CA Certificate.
   ► The installed certificates appear in the user certificates list (Apps >> Settings >> Lock screen and security >> Other security settings >> User certificates).
7. Open the PEM or CER file (**.pem / *.cer*) to install the mGuard Machine Certificate.
   ► The screen “Certificate name” appears.

   If the screen does not appear and the device displays the content of the file instead, download the file to the storage of your device or make it available via SD card. Open the file from the corresponding directory.
8. Click “OK” to finish the installation of the mGuard Machine Certificate.
   ► The installed certificate appears in the user certificates list (Apps >> Settings >> Lock screen and security >> Other security settings >> User certificates).

7.3 Configure VPN connections

7.3.1 Configure the mGuard

The IPsec VPN connection between the Android client and the mGuard will be established using the XAuth/Mode Config extension. The configuration of the iOS client will be configured by the mGuard and communicated to the iOS client.

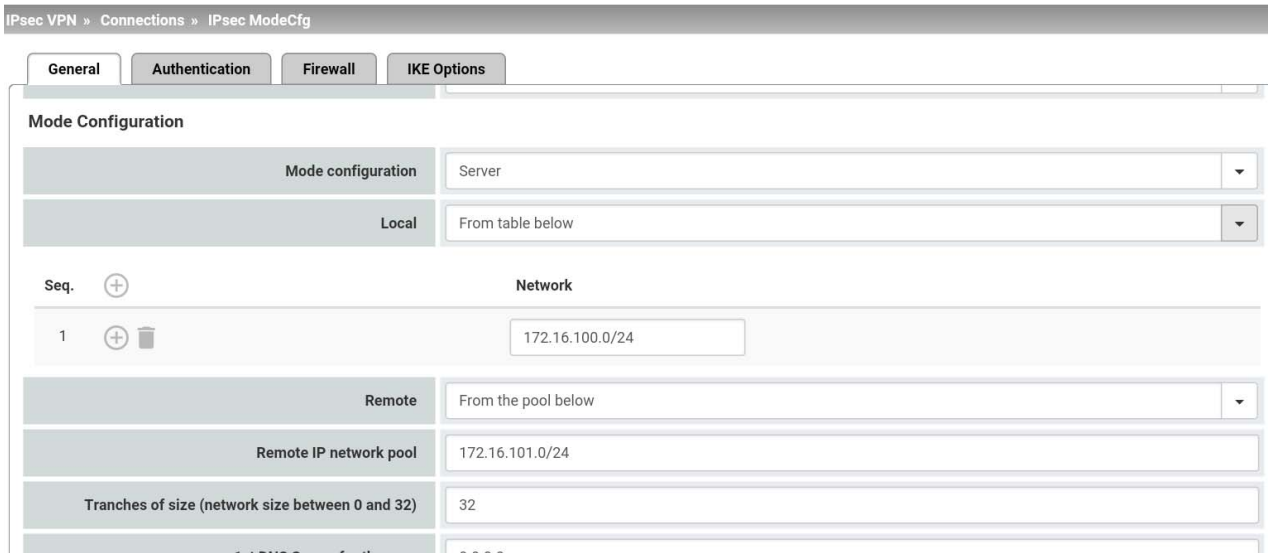


Figure 7-2 mGuard VPN configuration – Mode Configuration

7.3.1.1 Tab “General”

To configure a VPN connection to an Android client on the mGuard, proceed as follows:

1. Select the menu “IPsec VPN >> Connections”.
2. Click the icon ⊕ to create a new table row.
3. Click the icon ✎ “Edit row”.
   ► The tab “General” appears.
4. Enter a descriptive name for the connection and change further settings optionally.

   Verify that the input field “Address of the remote site’s VPN gateway” contains the value “**%any**” and “Connection startup” is set to “**Wait**” (default values).

5. In section **Mode Configuration** select Mode configuration **Server**.
6. **Local**: Enter the local network(s) on the server side (mGuard) that shall be accessible by the Android client via VPN connection.
   – **Fixed**: The *Local IP network* must be set to 0.0.0.0/0. In this case, all traffic from the Android client will be sent over the VPN connection.
   – **From table below**: Only traffic to the *Networks* listed in the *table below* will be send over the VPN connection.

   Android clients do not fully support this feature. Traffic from Android clients to networks not defined in the *table below* **will be blocked!**

7. **Remote**: Define the network pool (**From the pool below**) from which the mGuard allocates a variable tranche (**Tranches of size**) to be used by the remote client‘s network.

7.3.1.2 Tab “Authentication”

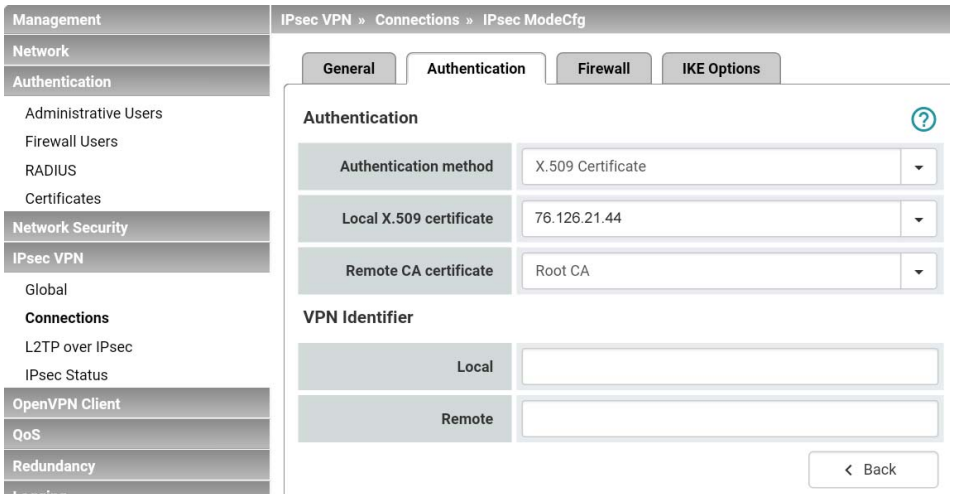


Figure 7-3 mGuard VPN configuration – Authentication

The VPN connection between an Android client and the mGuard must be authorized by X.509 certificates, that have to be installed on the corresponding devices (see “Manage certificates” on page 154).

To assign the required certificates to a VPN connection, proceed as follows:

1. Select the menu “IPsec VPN >> Connections”.
2. Edit the desired VPN connection (Tab “Authentication”).
3. Select the **Authentication method** “X.509 Certificate”.
4. As the *Local X.509 certificate* select the **mGuard Machine Certificate**.

Only for connections from iOS clients: The CN of the certificate must correspond with the external IP address or DNS name of the mGuard server.

The certificate must have been signed by the CA Certificate that has been installed on the Android client.

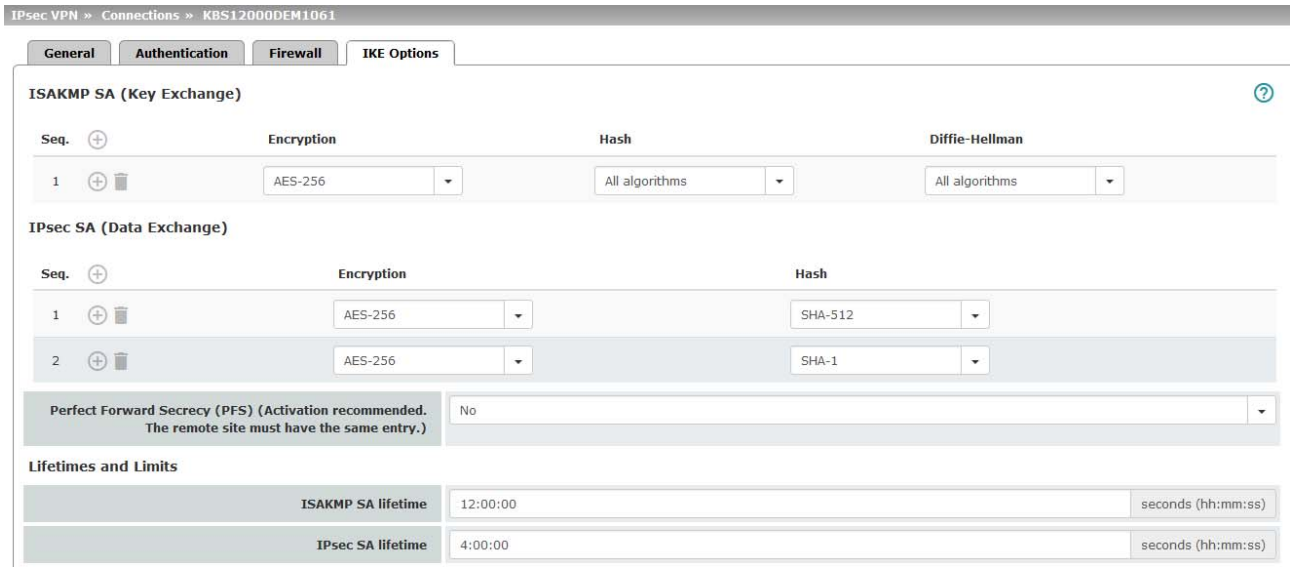
5. As the *Remote CA certificate* select the *CA Certificate* that has been used to sign the **iOS Client Certificate** and the **Android Client Certificate**.
6. Click “Apply” to save the settings.
   ► The VPN connection will be established after being initiated by the Android client.

7.3.1.3 Tab“Firewall”

The VPN firewall restricts the access through the VPN tunnel. You may configure the VPN firewall if required.

By default, **any incoming** and **outgoing** traffic will be accepted**.**

7.3.1.4 Tab “IKE Options”



It is necessary to change the default IKE options:

1. Select the menu “IPsec VPN >> Connections”.
2. Edit the desired VPN connection (Tab “IKE Options”).
3. Configure the following settings and leave all other settings on default.

ISAKMP SA (Key Exchange)

- Encryption: AES-256
- Hash: All algorithms
- Diffie-Hellman: All algorithms

IPsec SA (Data exchange)

- Click the icon ⊕ to create two table rows and use the following settings:
  - (Row 1) Encryption: AES-256 | Hash: SHA-512
  - (Row 2) Encryption: AES-256 | Hash: SHA-1

Perfect Forward Secrecy (PFS)

- The PFS must be set to **No**.
  (Even if set to **No**, iOS clients will still be able to use PFS.)

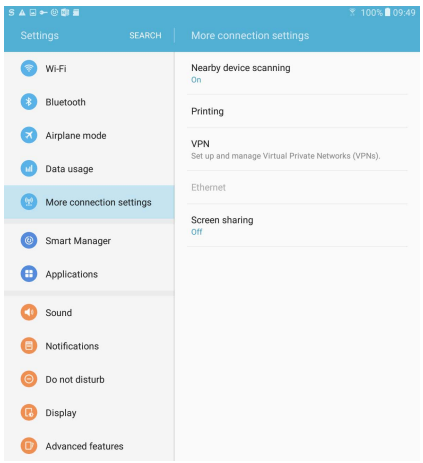
ISAKMP SA lifetime

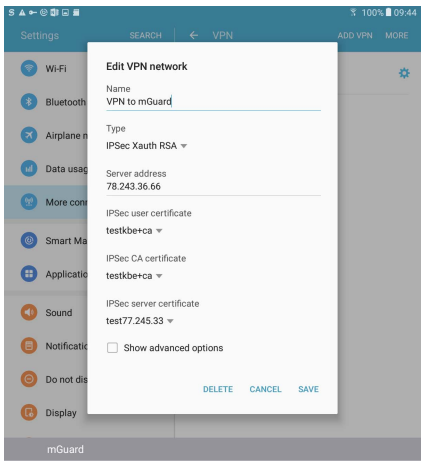
- 12:00:00 (hh:mm:ss)

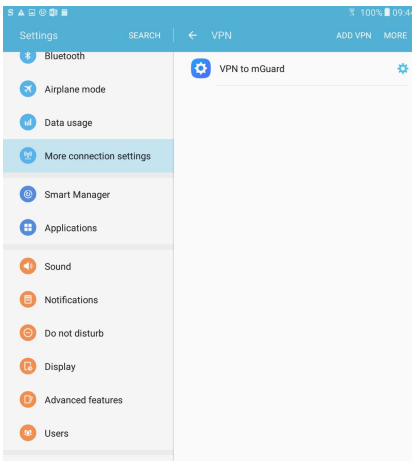
IPsec SA lifetime

- 04:00:00 (hh:mm:ss)

7.3.2 Configure the Android client







To configure an IPsec VPN connection on the Android client, proceed as follows:

1. Select the menu “Settings >> More connection settings >> VPN”.
2. Click “ADD VPN” or “+”.
   ► The screen “Edit VPN network” appears.
3. Configure the following settings:
   – Name: A descriptive name for the connection
   – Type: IPSec Xauth RSA
   – Server address: The external IP address or the DNS name of the mGuard server
   – IPSec user certificate: Select the name you have assigned to the Android Client Certificate from the PKCS#12 file.
   – IPSec CA certificate: Select the name you have assigned to the Android Client Certificate from the PKCS#12 file.
   – IPSec Server certificate: Select the name you have assigned to the mGuard Machine Certificate of the mGuard server (VPN gateway).
4. Click “Save” to save the configuration.
   ► The VPN configuration has been saved and is ready to be started.

7.4 Start VPN connections on the Android client

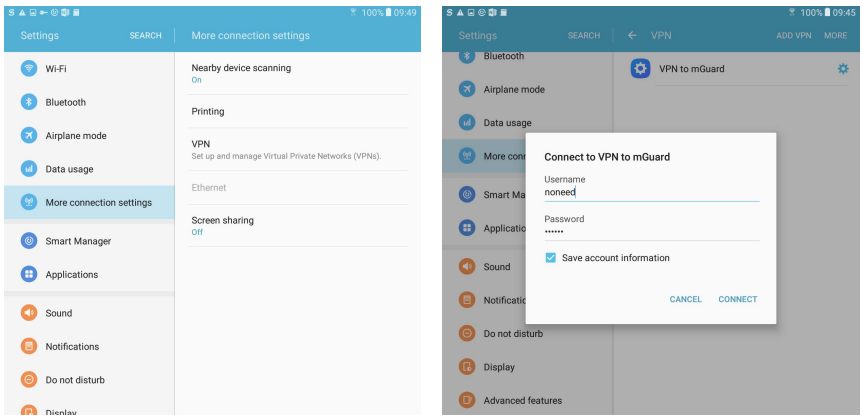


Figure 7-4 Start VPN connection on the Android client

To start an IPsec VPN connection on the Android client, proceed as follows:

1. Select the menu “Apps >> Settings >> More connection settings >> VPN”.
2. Click on the name of the appropriate VPN connection.
   ► The screen “Connect to <connection name>” appears.

The username and password for Xauth will be ignored by the mGuard. Enter some random text and save the account information.

3. Click “CONNECT” to start the connection.
   ► The VPN connection will be established and the status changes from “Not Connected” to “Connecting...” to “Connected”.

If the connection fails, click the “gear” symbol of the VPN connection to check for errors in the configuration or check your internet connection.

7.5 Check VPN connections on the mGuard

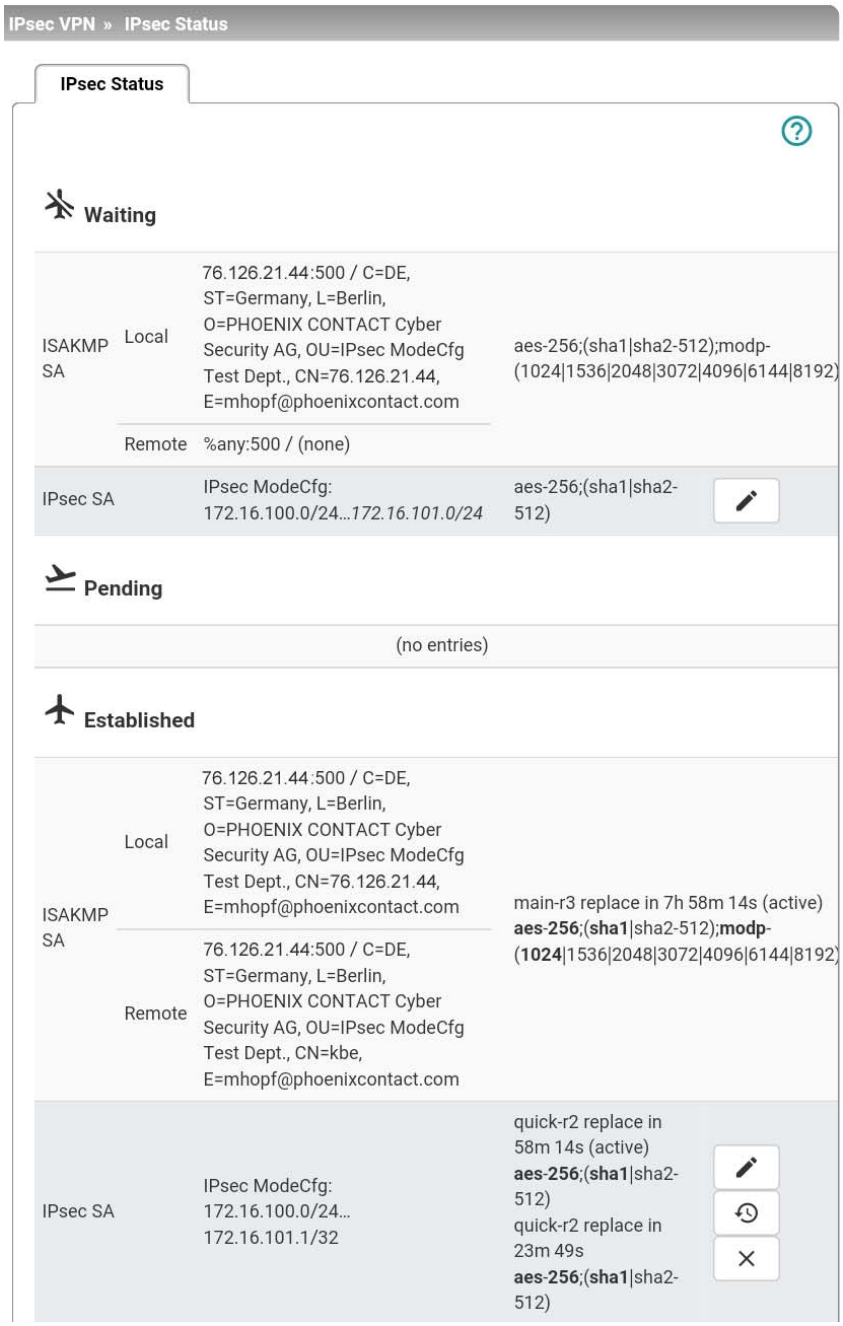


Figure 7-5 IPsec VPN status

To check the status of an IPsec VPN connection, proceed as follows:

- Select the menu “IPsec VPN >> IPsec Status”.
  - ► An established IPsec VPN connection appears in the area “Established”.

8 Update the mGuard configuration using pull configuration

Document-ID: 108398_en_01
Document-Description: AH EN MGUARD PULLCONFIG

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

Contents of this document

This document describes how to perform pull configuration for your mGuard device. It also describes how to obtain pull-config feedback from the server logs.

8.1 Introduction

An mGuard device can automatically “retrieve” new configuration profiles from a configuration pull server (*pull configuration*), provided that the corresponding profiles (with file extension *.atv*) have been stored there.

New configurations can be created and stored on the pull server using the mGuard device manager (mdm / FL MGUARD DM). The intervals at which new configurations are “retrieved” from the pull server can be configured on the mGuard device.

8.2 Configure pull configuration on the mGuard device

Proceed as follows to configure pull configuration on the mGuard device:

1. Log on to the web interface of the mGuard device.
2. Open **Management >> Central Management** (see also mGuard firmware manual).
3. Specify a schedule for the mGuard device to send a request to the pull server (*pull request*).
4. Make other settings, if required.

At the specified intervals, the mGuard device will attempt to “retrieve” new configurations from the pull server.

8.3 Pull configuration using mdm

Pull configuration (*pull configuration*) is one method for updating the configurations or the firmware version of an mGuard device using the mGuard device manager (mdm / FL MGUARD DM).

The configurations created in the mdm are first exported to the pull server and later “retrieved” by the mGuard device or uploaded to the device (see also mdm software manual).

The mGuard device sends the status of its configuration as a HTTP(S) request on every request to the pull server. The pull server then sends a SYSLOG message to the mdm server (*pull feedback*) in order to inform the mdm server about the configuration status of the mGuard device.

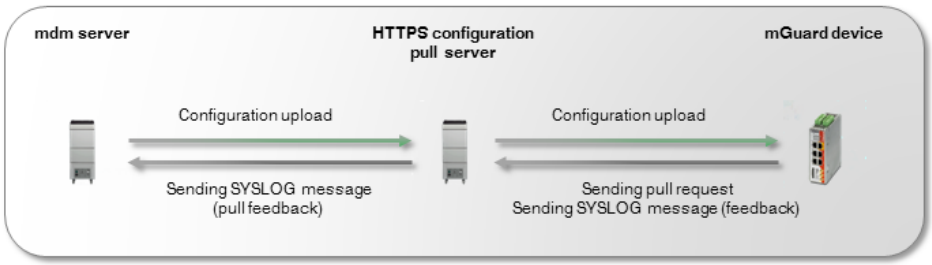


Figure 8-1 Pull configuration using mdm

Configure the mdm server to be able to receive SYSLOG messages from the HTTPS pull server.

Please make sure that neither the network connection between the HTTPS pull server and the mdm server nor the network connection between the HTTPS pull server and the mGuard device is blocked by a firewall or a NAT router.

8.4 Obtaine pull configuration feedback from server logs

In the event that communication from the configuration pull server to the mdm server is blocked due to firewall or NAT settings, the status of a *configuration pull* can also be obtained from the log entries of the pull server.

When an mGuard device retrieves a new configuration from the pull server, the mGuard device returns specific parameters (e.g., update status) as pull configuration feedback (*pull feedback*) in the form of an URL to the pull server (see the following Examples and Table 8-1). The pull server logs can be evaluated to verify whether the configuration pull was successful.

Examples

1. Configuration applied successfully:

"GET
//atv//00000001.atv?**a**=8.6.0.default&**b**=N205414313033131033abebcecfccecefcc&**c**=2031420608&**d**=e2adce0a1edd2c72e1910303f9d86925&**e**=0&**f**=-&**g**=-&**k**=-&**i**=0&**j**=0&**z**=1670 HTTP/1.1"

2. **Invalid configuration** (because of missing license for an activated function)**:**

"GET
//atv//00000001.atv?**a**=8.6.0.default&**b**=N205414313033131033abebcecfccecefcc&**c**=20
31420608&**d**=e2adce0a1edd2c72e1910303f9d86925&**e**=5&**f**=-&**g**=-&**k**=-
&**i**=0&**j**=0&**z**=71de HTTP/1.1"

Table 8-1 List of HTTP(S) request parameters evaluated by the mGuard device manager (mdm)

Parameter	Meaning	Status	Description
a	mGuard firmware version		Firmware version currently installed on the mGuard device
b	mGuard Flash ID		Flash ID of the mGuard device
c	mGuard device serial number		Serial number of the mGuard device
d	md5 hash of mGuard configuration		md5 hash value of the configuration currently used on the mGuard device
e	Update status of mGuard configuration (*configuration pull*)	**0**	The configuration on the mGuard device has been successfully updated.
		1	No update: The configuration on the mGuard device already is up to date.
e		**2**	No update: The new configuration could not be applied on the mGuard device. The previous configuration was restored (*rollback*).
		3	No update: The mGuard blocks the new configuration because it was restored (*rollback*) during a previous application attempt.
		4	No update: It was not possible to buffer the old configuration on the mGuard device for restoring (*rollback*) it later, which might be required.
		5	No update: The configuration that was to be used to update the mGuard device is invalid.
		-	No update: The configuration on the device should not be updated.
f	Status of the mGuard firmware update	**0**	The firmware update on the mGuard device was executed successfully.
		-	No update: A firmware update should not be executed on the device.

Table 8-1 List of HTTP(S) request parameters evaluated by the mGuard device manager (mdm)

		Any other character	No update: Firmware update failed
g	Status of license download	**0**	One or more licenses have been successfully installed on the mGuard device.
		-	A license should not be installed on the device.
		Any other character	Installation of the license failed
k	Status of *key renewal*	**0**	The keys (*ssh* and *https*) on the mGuard device have been successfully renewed.
		1	Key renewal failed
		2	Key renewal has not been executed Renewal is recommended because the current key might not be appropriately secure.
		-	Key renewal has not been executed

Further parameters (currently not guaranteed)

- **h** = Device type information; currently only set for NAT router devices. “h” is not transmitted on other devices.
- **i** = Redundancy: status of the password for *availability check*.
- **j** = Redundancy: status of the password for encryption of the network traffic between synchronized mGuard devices.
- **z** = 4 MSB (*Most Significant Bytes*) of the md5 hash value of meta information – without leading “?” and final “&” – but with linefeed character (0x0A) appended.

9 Installing a new bootloader on mGuard devices

Document-ID: 108042_en_02
Document-Description: AH EN MGUARD BOOTLOADER

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

9.1 Introduction

Due to the hardware structures of memory chips becoming increasingly smaller, it has become commonplace that some memory cells may not be fully functional and other memory cells may lose their ability to function as time passes. This reduction in memory capacity is compensated for by increased production capacity. As a result, the desired capacity is always maintained over the product's service life.

The mGuard devices have routines for handling defective memory cells. These routines are optimized when a new bootloader is installed.

If you do not want to update the firmware version, you can downgrade the device to the desired version after updating the bootloader. The newly installed version of the bootloader is retained after you downgrade the firmware version. However, Phoenix Contact always recommends using the latest firmware.

Devices produced with an mGuard firmware version 8.7.0 or later cannot be flashed to a firmware version < 8.7.0 (downgrade).

An up-to-date firmware version ensures that there is an optimized version of the bootloader on the device. Please observe the notes on firmware updates in the user manual of the device.

9.2 Testing the bootloader

If you have an mGuard device that is no longer booting and you want to check whether the bootloader is the cause, please follow the steps below to update the bootloader.

1. Disconnect the device from the supply voltage.
2. Use a tool such as "Putty" for communicating via the serial interface on your PC.
3. Establish the serial connection between the PC and the mGuard device.
4. Start the mGuard device by applying the supply voltage. The device will attempt to boot.

The bootloader must be updated if the following error message appears in the terminal window of your tool:

U-Boot 2009.11 (Dec 13 2013 - 08:34:06) MPC83XX

New bootloader versions are installed on MGUARD **firmware versions 7.6.8** and **8.1.4** or later.

10 Using the CGI Interface

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

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

Contents of this document

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

10.1 Introduction

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

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

nph-vpn.cgi / nph-diag.cgi

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

nph-action.cgi / nph-status.cgi

The CGI interfaces *nph-action.cgi* and *nph-status.cgi* provide an extended range of features and functionality (see Section 10.6, “Interface nph.action.cgi” and Section 10.7, “Interface nph.status.cgi”).

10.2 Usage

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

10.2.1 Available commands

nph-vpn.cgi / nph-diag.cgi

Table 10-1 Commands provided by the CGI scripts *nph-vpn.cgi* and *nph-diag.cgi*

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

nph-action.cgi / nph-status.cgi

For commands provided by the CGI scripts *nph-action.cgi* and *nph-status.cgi* see Section 10.6, “Interface nph.action.cgi” and Section 10.7, “Interface nph.status.cgi”.

10.2.2 Command syntax

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

Example:

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

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

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

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

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

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

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

Table 10-2 Command syntax

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

Examples

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

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

10.2.3 Access rights

Table 10-3 Access rights

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

10.3 Preconditions and restrictions

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

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

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

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

Example:

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

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

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

10.3.1 Preconditions

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

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

10.3.2 Restrictions

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

10.4 Interface nph-vpn.cgi

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

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

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

Examples:

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

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

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

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

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

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

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

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

10.4.2.1 cmd=status, name=<VPN name>

This command retrieves the status of the specified VPN connection.

Example:

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

Return value
active

10.4.2.2 cmd=status, name=*

This command retrieves the status of all configured VPN connections.

Example:

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

Return value
Service 1: active
Service 2: void

10.4.2.3 cmd=status (without parameter *name*)

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

Example:

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

(Parameter *name* not specified)

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

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

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

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

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

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

10.4.3.1 Response message format

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

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

10.4.3.2 Response code

The response may contain one of the following codes:

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

10.4.3.3 cmd=synup

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

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

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

Response:

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

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

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

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

Response pattern

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

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

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

Progress messages

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

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

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

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

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

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

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

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

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

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

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

The format of this message is as follows:

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

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

P IKEv1 uptime=… newstate=state

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

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

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

Final message

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

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

Please refer to “Response code” on page 179 for an explanation about those codes.

10.4.3.4 cmd=synstat

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

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

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

Response:

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

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

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

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

Please refer to “Response code” on page 179 for an explanation about those codes.

10.4.3.5 cmd=syndown

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

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

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

Response:

```
R OKVDOWN
```

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

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

Please refer to “Response code” on page 179 for an explanation about those codes.

10.4.4 Central VPN gateway commands

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

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

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

If the central VPN gateway has only one single *VPN Tunnel Group* configured, where all remote mGuards connect to, there is no way to determine whether there exists an active connection to an individual remote mGuard. Of course, *cmd=status* can be used without a specified VPN connection name (refer to Section 10.4.2.3) but this command would determine the status of all tunnels which is rather inefficient for querying the state of one single tunnel.

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

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

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

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

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

Response:

active

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

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

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

Example:

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

Response:

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

10.4.5 cmd=sysinfo

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

Example:

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

Response:

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

10.5 Interface nph-diag.cgi

10.5.1 cmd=snapshot

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

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

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

10.5.2 cmd=testpull

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

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

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

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

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

Response:

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

10.6 Interface nph.action.cgi

User “root“ and “admin“

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

Row actions

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

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

Table 10-4 Row actions – Parameters

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

Table 10-5 Row actions – Actions

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

User firewall logout

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

Table 10-6 User firewall logout – Parameters

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

Table 10-7 User firewall logout – Actions

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

Simple commands

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

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

Table 10-8 Simple commands – Actions

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

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

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

Mobile actions (User: mobile / root / admin)

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

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

Table 10-9 Mobile actions – Parameters

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

Table 10-10 Mobile actions – Actions

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

10.7 Interface nph.status.cgi

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

Table 10-11 CGI status

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

Table 10-11 CGI status

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

Table 10-11 CGI status

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

11 LED status indicator and blinking behavior

Document-ID: 108400_en_00
Document-Description: AH EN MGUARD LED SIGNALS

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

Contents of this document

This document describes the lighting and blinking behavior of the LED diodes installed in mGuard devices (FL/TC MGUARD RS2000/RS4000).

11.1 Description of LEDs

With the help of built-in LED diodes, mGuard devices indicate different system states. This can be status, alarm or error messages.

The states are indicated by permanent or temporary lighting or blinking of the LEDs. The displayed LED pattern can also represent a combination of different system states.

NOTE: Since several system states are indicated by the LEDs not clearly, only temporarily or in combination with other system states, the log files of the mGuard device must also be checked!

LED diodes of FL/TC MGUARD (RS200x/RS400x) devices:

P1	Stat	Mod	Info2 (Sig)
●	●	●	●
●	●	●	●
P2	Err	Fault	Info1

P1 / P2

LEDs *P1* and *P2* indicate which of the two power supplies is connected (devices of the FL/TC MGUARD RS2000 series: only *P1* is available).

Info 2 / Info 1 (the LED Sig is not in use)

Active VPN connections or (as of Version 8.1) active firewall rule records can be indicated via the LEDs *Info2* and *Info1*. The activation of the LEDs by a certain VPN connection or a certain firewall rule record is configured on the mGuard interface in the menu item **Management >> Service Contacts**.

The following states will be indicated:

ON	The VPN connection is established / the firewall rule record is set.
Blink	The VPN connection will be established or released or has been stopped/disabled by the remote peer.
OFF	The VPN connection is stopped/disabled on both peers.

Stat / Mod / Err / Fault

The LEDs *Stat*, *Mod*, *Err* and *Fault* indicate system states (status, alarm or error messages) (see Table 11-3).

In addition to the alarm messages, an illuminated **Fault LED** generally also indicates that the device is currently not in operation mode.

LAN / WAN

The LAN/WAN LEDs are located in the LAN/WAN sockets (10/100 and duplex LED).

The LEDs Indicate the ethernet status of the LAN or WAN port. As soon as the device is connected to the relevant network, a continuous light indicates that there is a connection to the network partner in the LAN or WAN. When data packets are transmitted, the LED goes out briefly.

If all LAN/WAN LEDs are illuminated, the system is booting.

Bar graph and SIM1/2 (Mobile)

Table 11-1 LEDs on TC MGUARD RS4000 3G and TC MGUARD RS2000 3G

LED	**State and Meaning**					
Bar graph	LED 3	Top	Off	Off	Off	Green
	LED 2	Middle	Off	Off	Green	Green
	LED 1	Bottom	Off	Yellow	Yellow	Yellow
	Signal strength (dBm)		–113 ... 111	–109 ... 89	–87 ... 67	–65 ... 51
	Network reception		Very poor to none	Sufficient	Good	Very good
SIM 1	Green	On Blinking	SIM card 1 active No PIN or incorrect one entered			
SIM 2	Green	On Blinking	SIM card 2 active No PIN or incorrect one entered			

11.2 LED lighting and blinking behavior

Table 11-2 Description of the lighting and blinking behavior of the LED diodes

Heartbeat	The blinking behavior is similar to a heartbeat, in which two strokes are performed in quick succession, followed by a short break.
Running light	Three lights form a continuously repeating running light from left to right and back again.
Blink 50/1500	Flashing with 1500 ms break (50 ms on, then 1500 ms off)
Blink 50/800	Flashing with 800 ms break (50 ms on, then 800 ms off)
Blink 50/100	Flashing with 100 ms break (50 ms on, then 100 ms off)
Blink 500/500	Constant blinking (500 ms on / 500 ms off)
Morse code (. . . – – – . . .)	The blinking behavior shows the *Morse code* '*SOS*', in which the blinking behavior "3x short, 3x long, 3x short" is repeated continuously.
ON	The diode lights up permanently.
ON (n sec)	The diode lights up permanently for the indicated time (in seconds n)

11.3 Representation of system states

The system states (status, alarm or error messages), which are displayed by the LED's lighting and blinking behavior, are shown in Table 11-3.

Table 11-3 System states of FL/TC MGUARD devices represented by lighting and blinking behavior of the LEDs

STAT	MOD	Info 2 (Sig)	ERR	FAULT	Description of the system state
Heart-beat					The system status is OK.
			ON		A severe error has happened.
ON (12 sec)	ON (3 sec)		ON (12 sec)	ON (12 sec)	The system is booting.
Morse code					The license to operate this firmware is missing.
Morse code			**Morse code**		Bootloader replacement failed due to hardware error.
				ON	A power failure was detected.
				ON	No connectivity on WAN interface (link supervision configurable on device)
				ON	No connectivity on LAN interface (link supervision configurable on device)
				ON	No connectivity on LAN 1–4 interface (link supervision configurable on device)
				ON	No connectivity on DMZ interface (link supervision configurable on device)
				ON	Power supply 1 or 2 failed (alarm configurable on device)
				ON	Temperature too high / low (alarm configurable on device)
				ON	(Redundancy) Connectivity check failed (alarm configurable on device)
				ON	(Modem) Connectivity check failed (alarm configurable on the device)
			ON (3 sec)		ECS: The ECS is incompatible.
			ON (3 sec)		ECS: The capacity of the ECS is exhausted.

Table 11-3 System states of FL/TC MGUARD devices represented by lighting and blinking behavior of the LEDs

STAT	MOD	Info 2 (Sig)	ERR	FAULT	Description of the system state
			ON (3 sec)		ECS: The root password from the ECS does not match.
			ON (3 sec)		ECS: Failed to load the configuration from the ECS.
			ON (3 sec)		ECS: Failed to save the configuration to the ECS.
	ON				PPPD: The internal modem got a connect (set by pppd).
	Blink 50/1500				PPPD: The internal modem is armed and expecting a dial in.
	Blink 500/500				PPPD: The internal modem is dialing.
			ON (2 sec)		RECOVERY: The recovery procedure failed.
ON (2 sec)					RECOVERY: The recovery procedure succeeded.
ON				ON	FLASH PROCEDURE: The flash procedure has been started. Please wait.
Running light	Running light	Running light		ON	FLASH PROCEDURE: The flash procedure is currently executed.
Blink 50/800	Blink 50/800	Blink 50/800		ON	FLASH PROCEDURE: The flash procedure succeeded.
	ON		ON		FLASH PROCEDURE: The flash/production procedure failed.
			Blink 50/100 (5 sec)		FLASH PROCEDURE WARNING: Replacing the rescue system. Do not power off. When the blinking stops, the replacement of the rescue system is over.
			ON		FLASH PROCEDURE: The DHCP/BOOTP requests failed.
			ON		FLASH PROCEDURE: Mounting the data storage device failed.
			ON		FLASH PROCEDURE: The flash procedure failed.
			ON		FLASH PROCEDURE: Erasing the file system partition failed.
			ON		FLASH PROCEDURE: Failed to load the firmware image.
			ON		FLASH PROCEDURE: The signature of the firmware image is not valid.
			ON		FLASH PROCEDURE: Failed to load the install script.
			ON		FLASH PROCEDURE: The signature of the install script is not valid.
			ON		FLASH PROCEDURE: The rollout script failed.

Please observe the following notes

Note on the usage of Application Notes

The provided Application Notes are a free service from Phoenix Contact. The examples and solutions shown are not customer-specific solutions, but general support for typical application scenarios. The Application Notes are not binding and do not claim to be complete.

A quality check of the Application Notes takes place but is not comparable with the quality assurance of commercial products. Errors, functional and performance deficiencies cannot be excluded.

To avoid malfunctions/misconfigurations and associated damage, the proper and safe use of the product/software is the sole responsibility of the customer and must comply with the applicable regulations. The customer must check the function of the examples described and adapt them to the individual, customer-specific requirements of the system or application scenario.

The IP settings in the Application Notes have been chosen as examples. In a real network scenario, these IP settings must always be adjusted to avoid address conflicts.

The information in the Application Notes is checked regularly. If corrections are necessary, they will be included in the subsequent revision. Users will not be notified.

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.

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

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

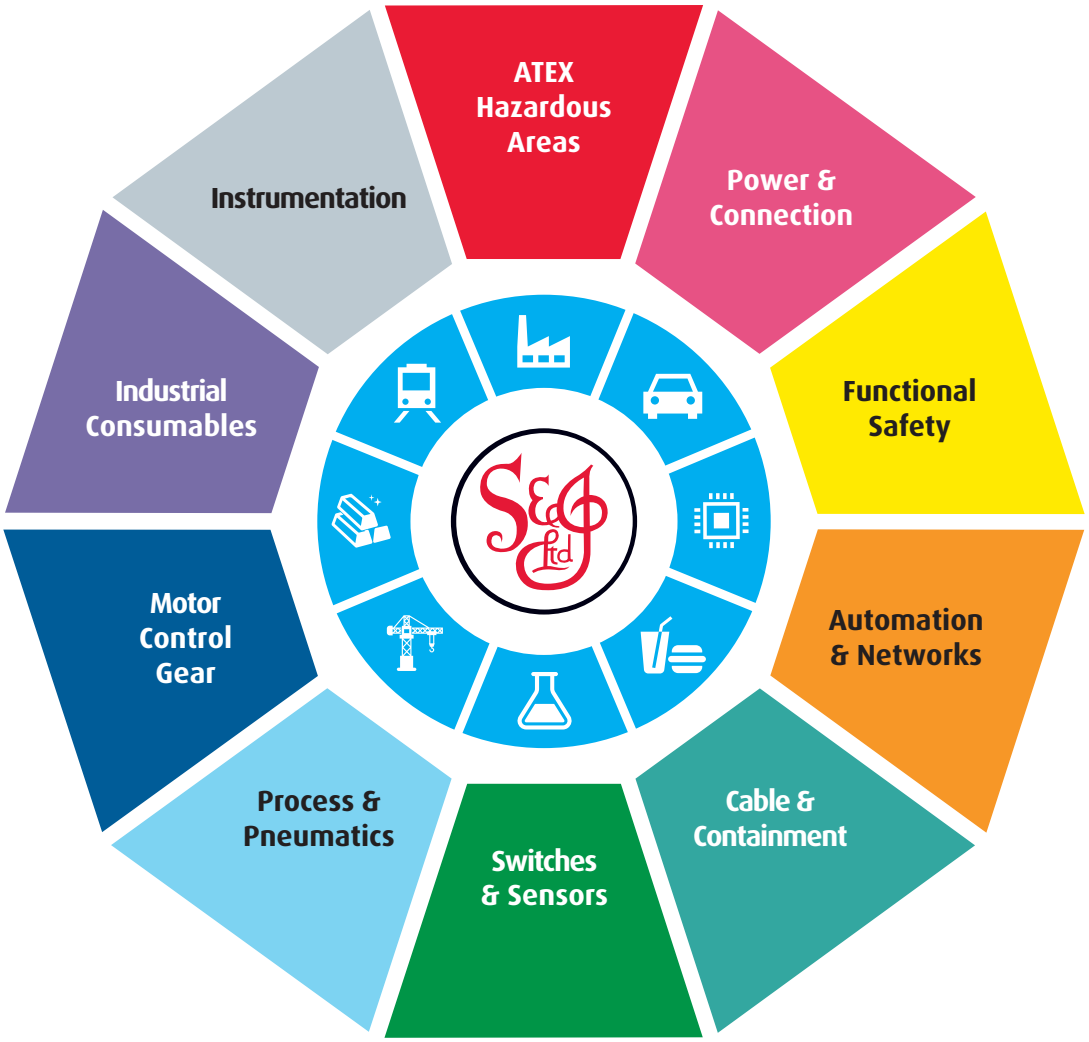
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

www.scatts.co.uk