TECHDOCS

Manual

# WAGO-I/O-SYSTEM 750



# 750-8xxx(/xxx-xxx)

## PFC100/200

## Cyber Security for Controller PFC100/PFC200

Version 1.2.1

© 2025 WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

**WAGO Kontakttechnik GmbH & Co. KG**

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: www.wago.com

**Technical Support**

Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.

# Table of Contents

# 1 Notes about this Documentation

**Note**

**Always retain this documentation!**
This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

This documentation applies to the standard versions and all variants of the PFC100/PFC200 controller.

## 1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

## 1.2 Symbols

**⚠ DANGER**

**Personal Injury!**
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

**⚠ DANGER**

**Personal Injury Caused by Electric Current!**
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

**⚠ WARNING**

**Personal Injury!**
Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

**⚠ CAUTION**

**Personal Injury!**
Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

**NOTICE**

**Damage to Property!**
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

**NOTICE**

**Damage to Property Caused by Electrostatic Discharge (ESD)!**
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

**Note**

**Important Note!**
Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

**Information**

**Additional Information:**
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

## 1.3 Number Notation

Table 1: Number Notation

| Number Code | Example | Note |
|---|---|---|
| Decimal | 100 | Normal notation |
| Hexadecimal | 0x64 | C notation |
| Binary | '100'<br>'0110.0100' | In quotation marks, nibble separated with dots (.) |

## 1.4 Font Conventions

Table 2: Font Conventions

| Font Type | Indicates |
|---|---|
| *italic* | Names of paths and data files are marked in italic-type.<br>e.g.: *C:\Program Files\WAGO Software* |
| **Menu** | Menu items are marked in bold letters.<br>e.g.: **Save** |
| **>** | A greater-than sign between two names means the selection of a menu item from a menu.<br>e.g.: **File** > **New** |
| **Input** | Designation of input or optional fields are marked in bold letters,<br>e.g.: **Start of measurement range** |
| “Value” | Input or selective values are marked in inverted commas.<br>e.g.: Enter the value “4 mA” under **Start of measurement range**. |
| **[Button]** | Pushbuttons in dialog boxes are marked with bold letters in square brackets.<br>e.g.: **[Input]** |
| **[Key]** | Keys are marked with bold letters in square brackets.<br>e.g.: **[F5]** |

# 2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

## 2.1 Legal Bases

### 2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

### 2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

### 2.1.3 Use of the 750 Series in Compliance with Underlying Provisions

Fieldbus couplers, controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler or controller.

Appropriate housing (per 2014/34/EU) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

The implementation of safety functions such as EMERGENCY STOP or safety door monitoring must only be performed by the F-I/O modules within the modular WAGO-I/O-SYSTEM 750. Only these safe F-I/O modules ensure functional safety in accordance with the latest international standards. WAGO's interference-free output modules can be controlled by the safety function.

### 2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. These modules contain no parts that can be serviced or repaired by the user. The following actions will result in the exclusion of liability on the part of WAGO Kontakttechnik GmbH & Co. KG:

- Repairs,
- Changes to the hardware or software that are not described in the operating instructions,
- Improper use of the components.

Further details are given in the contractual agreements. Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

# 3 General

Networking industrial systems via the Internet has made control systems like the WAGO-I/O-SYSTEM more vulnerable to cyber attacks. In order to minimize security threats and prevent economic damage, there are three essential security criteria a system must meet:

- **Availability**:
  The data and functions of a systems should be available in a timely fashion as needed.
- **Integrity**
  The correctness and completeness of data that requires protection and the specific functionality of the system should be ensured.
- **Confidentiality**
  Data and information that require protection should only be accessible to people who are authorized to access them.

This documentation describes potential security threats with the goal of fending off these security threats with appropriate, effective measures.

The principles of secure system design include:

- **Minimum privileges/minimum need-to-know principle:**
  User and system components have only the minimum privileges and access rights necessary to carry out a particular action.
- **Multiple security levels/“defense-in-depth principle”:**
  Security threats are mitigated, not through single countermeasures, but rather through multiple, tiered, complementary security measures.
- **Redundancy principle:**
  Systems are designed in such a way that a fault in one component will not adversely affect the functions of the security systems. The likelihood and severity of the problems (e.g., denial-of-service attacks) caused by excessive use of system resources must be minimized accordingly.

WAGO’s ETHERNET-based products are intended for operation on a closed industrial communication network. If the devices are not used on closed, industrial networks, the integrator and operator must take further safety measures for optimal use of the products.

If the industrial networks are publicly accessible (e.g., through freely accessible network interfaces within the closed industrial network) or publicly available (e.g., via data connections through public data traffic [Internet]), then organizational and technical security measures must be taken to protect the internal network and ensure that the security criteria are met. The security measures to take depend on the anticipated risk caused by potential external influences.

The PFC100 and PFC200 controllers offer extensive security functions such as TLS, SSH, VPN and a host-based firewall. Integrated password protection and secured communication protect against access to functions, program contents and the introduction of malware.

These and other security measures recommended in this manual will help you to minimize the risk of attack due to particular threats to machines and systems and come closer to meeting the security criteria described above. In addition to the recommendations, WAGO investigates, assesses and eliminates reported potential security flaws insofar as eliminating them does not impair the the general functionality of the product. The information in this manual will be continually updated as needed.

## 3.1 Abbreviations

Table 3: Abbreviations

| Abbreviation | Explanation | Description |
|---|---|---|
| AH | Authentication Header | Within IPsec (VPN), the AH protocol ensures the authenticity of the transferred data and the authentication of the sender. The AH ensures the integrity and validity of the data. However, the user data is not encrypted and is thus readable by anyone. |
| BSI | Bundesamt für Sicherheit in der Informationstechnik | The BSI is an independent, neutral body for issues related to IT security in the information society. |
| ESP | Encapsulating Security Payload | The ESP protocol is responsible for ensuring the authenticity, integrity and confidentiality of the transferred IP packets. Unlike the AH protocol, the ESP protocol also encrypts the user data. |
| IKE | Internet key exchange | The IKE is a key protocol for exchange of “Internet Protocol Security” keys (see the section “IPsec”). |
| IPsec | Internet Protocol Security | IPsec is an extension of the Internet Protocol (IP). With extended encryption and authentication mechanisms, IP packets can be transported in cryptographically secure way over insecure public networks. |
| PSK | Pre-Shared Key | Pre-Shared Key means that the participants exchange the keys for authentication and encryption in advance. |
| ROOT-CA | Root Certificate Authority | The root CA signs its own certificate itself. Thus the root certificate forms the shared trust anchor of all certificates subordinate to it. |
| SA | Security Association | A security association is the underlying foundation of every IPsec connection. It describes how two parties communicating with each other on computer networks can do so securely.<br>These security associations are formed separately for both the “Authentication Header” (AH) and “Encapsulated Security Payload” (ESP). |
| SCEP | Simple Certificate Enrollment Process | SCEP simplifies requesting and issuing certificates on trusted internal networks. The idea is that every standard network user can retrieve its digital certificate electronically on its own. |

Table 3: Abbreviations

| Abbreviation | Explanation | Description |
|---|---|---|
| SPI | Security Parameter Index | The SPI is a number that, together with the IP destination address and a security protocol, identifies a particular security association (SA). |
| VPN | Virtual Private Network | A virtual private network is a closed logical network within which the participants are spatially separated from each other and connected via a VPN tunnel. |
| WBM | Web-Based-Management | WAGO devices can be configured and administered via Web-Based Management. |

# 4 Standard Settings

The standard configuration of the controllers is presented below. Possible security risks due to network-based or physical attacks are described in the section “Threat Scenarios.” Resulting measures necessary in order to eliminate these security risks are described in the section “Hardening.”

## 4.1 Physical Interfaces

The controllers have the physical interfaces listed in the following table.

| Device | Explanation |
|---|---|
| Memory card slot | Slot for SD memory card |
| RS-232/-485 | Communication port |
| RJ-45 (ETHERNET) | ETHERNET network connection |
| Service Interface | Serial connection for service activities on WAGO devices |
| Reset Button | Key switch with functions that differ depending on the position of the mode selector switch |
| Mode Selector Switch | Different functions (RUN, STOP, RESET) depending on the state of the device |
| X1/X2 ETHERNET interfaces | Network connections |
| 3G modem | Internal wireless modem (only with 750-8207) |

The following fieldbus systems are supported:

- CANopen,
- PROFIBUS DP,
- Modbus TCP/UPD/RTU

**Note**

**You can find more information in our product manuals:**
You can find more information on the physical interfaces and fieldbuses in the corresponding manuals of the controllers in question.

## 4.2 Network Services

The protocols/services of the controllers that are supported by default are listed below. In addition to these services, the section “Device-specific Services” lists the services that are only supported by particular devices.

**Note**

**Active ports can be displayed:**
You can view the currently open active ports by executing the command “netstat -tulp” as “root” on the Linux® console.

Table 4: Basic Server Configuration

| Port | Protocol | Description | Program |
|---|---|---|---|
| 20/TCP<br>21/TCP | FTP (File Transfer Protocol) | Protocol for file transfer[2] | pure-ftpd |
| 20/TCP<br>21/TCP | FTPS (File Transfer Protocol over SSL) | Encrypted file transfer protocol[2)] | pure-ftpd |
| 22/TCP | SSH (Secure Shell) | Encrypted network protocol for remote access[1)] | dropbear |
| 22/TCP | SFTP (Secure File Transfer Protocol) | Encrypted file transfer protocol[1)] | sftp-server |
| 23/TCP | Telnet | Network protocol for remote access[2)] | busybox |
| 53/TCP<br>53/UDP | DNS (Domain Name System) | Name resolution protocol[2)] | dnsmasq |
| 67/UDP | DHCP (Dynamic Host Configuration Protocol) | Device parameterization protocol[2)] | dnsmasq |
| 69/UDP | TFTP (Trivial File Transfer Protocol) | Protocol for file transfer[2)] | tftpd |
| 80/TCP | HTTP (Hyper Text Transfer Protocol) | Protocol for loading websites[1)] | lighttpd |
| 161/UDP<br>162/UDP (Trap) | SNMP v1 (Simple Network Management Protocol v1) | Protokoll für die Steuerung und Überwachung von Netzwerkelementen[2)] | net-smnp |
| 161/UDP<br>162/UDP (Trap) | SNMP v2c (Simple Network Management Protocol v2c) | Protocol for control and monitoring of network elements[2)] | net-smnp |
| 161/UDP<br>162/UDP (Trap) | SNMP v3 (Simple Network Management Protocol v3) | Protocol for control and monitoring of network elements[2)] | net-smnp |
| 443/TCP | HTTPS (Hyper Text Transfer Protocol over SSL) | Protocol for secure transfer of websites[1)] | lighttpd |
| 4500/UDP | IPsec (Internet Protocol Security) | Protocol for connecting two trusted devices/networks over an untrusted network, e.g. the Internet | ipsec |
| 500/UDP | IKEv2 (Internet-Key-Exchange-Protocol) | Protocol for automatic key exchange for IPsec | charon |
| 502/TCP<br>502/UDP | Modbus | Protocol for process data exchange (***e!****RUNTIME*)[1)] | codesys3 |
| 502/TCP<br>502/UDP | Modbus | Protocol for process data exchange (CODESYS V2)[1,5)] | plclinux_rt |
| 514/UDP | Syslog | Protocol for transfer of log messages[1,4)] | syslog-ng |

Table 4: Basic Server Configuration

| Port | Protocol | Description | Program |
|---|---|---|---|
| 1194/UDP | OpenVPN | Protocol for connecting two devices/networks over an untrusted network, e.g. the Internet. | openvpn |
| 1740/UDP | PLC Handler | ***e!****RUNTIME* runtime environment[2)] | codesys3 |
| 2159/TCP | GDB remote serial protocol | Protocol for debugging remote targets | gdbserver |
| 2455/TCP | PLC Handler | CODESYS runtime environment[3,5)] | plclinux_rt |
| 4840/TCP | OPC UA (OPC Unified Architecture) | Data exchange protocol[3)] | codesys3 |
| 6626/TCP | I/O-Check | Proprietary protocol for WAGO device parameterization[1,4)] | iocheckd |
| 8080/TCP | HTTP (Hyper Text Transfer Protocol) | ***e!****RUNTIME*-Webserver[2,3)] | codesys3 |
| 8080/TCP | HTTP (Hyper Text Transfer Protocol) | CODESYS Web server[2,3,5)] | plclinux_rt |
| 11740/TCP | PLC Handler | ***e!****RUNTIME* runtime environment[3)] | codesys3 |
| UDP | | Port opened by ***e!****RUNTIME* with no function[3)] | codesys3 |

[1)] Service is active by default when delivered.
[2)] The service must be enabled by the user, or is enabled as soon as a (CODESYS/***e!****RUNTIME*) application is launched on the device.
[3)] Service/port depends on the runtime environment used.
[4)] Service/port is linked to the local host and not accessible from outside.
[5)] Only available with PFC200.

Table 5: Basic Client Configuration

| Port | Protocol | Description | Program |
|---|---|---|---|
| 52/TCP<br>52/UDP | DNS (Domain Name System | Name resolution protocol | - |
| 68/UDP | DHCP (Dynamic Host Configuration Protocol) | Device parameterization protocol | busybox |
| 69/UDP | TFTP (Trivial File Transfer Protocol) | [2)] | busybox |
| 123/UDP | SNTP/NTP (Network Time Protocol) | Protocol for time synchronization on a network | ntpclient |
| 1883/TCP<br>8883/TCP | MQTT (Message Queue Telemetry Transport) | Protocol for machine-to-machine communication | dataagent |
| 4500/UDP | IPsec (Internet Protocol Security) | Protocol for connecting two trusted devices/networks over an untrusted network, e.g. the Internet | Ipsec |
| 514/UDP | Syslog | Protocol for transfer of log messages[1,4)] | syslog-ng |
| 1194/UDP | OpenVPN | Protocol for connecting two devices/networks over an untrusted network, e.g. the Internet | openvpn |

[1)] Client is active by default when delivered.
[2)] The client must be enabled by the user, or is enabled as soon as a (CODESYS/***e!****RUNTIME*-) application is launched on the device.
[3)] The client depends on the runtime environment used.
[4)] The client is linked to the local host and not accessible from outside.

### 4.2.1 Device-Specific Services

Table 6: Applications for PFC100/PFC200

| Device | Port | Protocol | Description | Program |
|---|---|---|---|---|
| PFCx00 | 102/TCP | IEC 61850 | Protocol for transfer between control systems and remote control terminals (telecontrol technology) | CODESYS V2.3 |
| PFCx00 | 2404/TCP<br>2404/UDP | IEC 60870-5-104 | | |
| PFCx00 | 20000/TCP<br>20000/UDP | DNP3 (Distributed Network Protocol) | | |

## 4.3 Users and Passwords

The following chapters describe the preconfigured users of the various services of the controller.

### 4.3.1 WBM Users

The WBM has its own user administration system. The users in this system are isolated from the other user groups in the system for security reasons. No new users can be created; the existing users are permanently stored in the application. You can find instructions for changing the passwords in the section “Hardening” > … > “Change Passwords in the Web-Based Management.”

Table 7: WBM Users

| User | Rights | Default Password |
|---|---|---|
| admin | All (administrator) | wago |
| user | Limited | user |
| guest | Display only | Login not possible; used if no login has occurred. |

### 4.3.2 Linux® Users

The Linux® user group includes the users of the operating system. Services offered by the device are each executed under their own user, which is locked for login. Additional users can be added.

**Note**

**Passwords for service users**
Users of services such as www, messagebus, rpcuser, nobody or opc are locked for login. Please do not change these passwords.

| User | Special Feature | Default Password |
|---|---|---|
| root | Administrator | wago |
| admin | CODESYS runtime user | wago |
| user | User | user |

You can find instructions for changing the passwords in the section “Hardening” > … > “Change Linux® Passwords via the Linux® Console.”

### 4.3.3 SNMP Users

The SNMP service manages its own users. When first delivered, no users are stored in the system.

**Information**

**You can find additional information in the product manual.**
You can find more information on the SNMP services in the manual of the corresponding controller at www.wago.com.

### 4.3.4 CODESYS and *e!RUNTIME* Web Visualization

For the CODESYS and ***e!****RUNTIME* Web visualization, a password can be set for each project for different work groups.

**Information**

**Users and passwords are not created automatically.**
The users and passwords for the CODESYS and e!RUNTIME Web visualization must be created by the end user.

# 5 Threats to Industrial Control Systems

This section describes potential threat scenarios that can arise when linking to a public network, e.g. the Internet. Solution approaches (defense in depth) and specific measures for the controllers are also recommended on the basis of the scenarios described.

The following list from the BSI provides an overview of the threats to industrial control systems that are critical from their point of view. This section focuses on the direct threats to the controller.

- Social engineering and phishing
- Introduction of malware via removable media and external hardware*
- Infection with malware via Internet and intranet*
- Incursion via remote maintenance access
- Human misconduct and sabotage
- Control components connected to the Internet*
- Technical misconduct and force majeure
- Compromised extranet and cloud components
- (D)DoS attacks*
- Compromised smartphones in the production environment

* Direct threats to the controllers

In order to achieve a level of protection that can counter the majority of the threats, it is necessary to pursue an integrated approach according to the defense-in-depth principle.

**Note**

**See the BSI recommendation for further information!**
For further information, see the “IT in Production” recommendation of the German Federal Office for Information Security > “Industrial Control System Security: 2016 Top 10 Threats and Countermeasures.”

## 5.1 Defense-in-Depth Principle

The operator of an infrastructure must take an integrated view of the threats to industrial control systems listed by the BSI. The infrastructure is only secure if multiple tiers of complementary measures, both organizational and technical, are implemented. This is meant to ensure that defeating one security measure does not allow an entire system/facility to be compromised. The defense-in-depth principle can be applied both to an operator's entire architecture and to an individual controller.

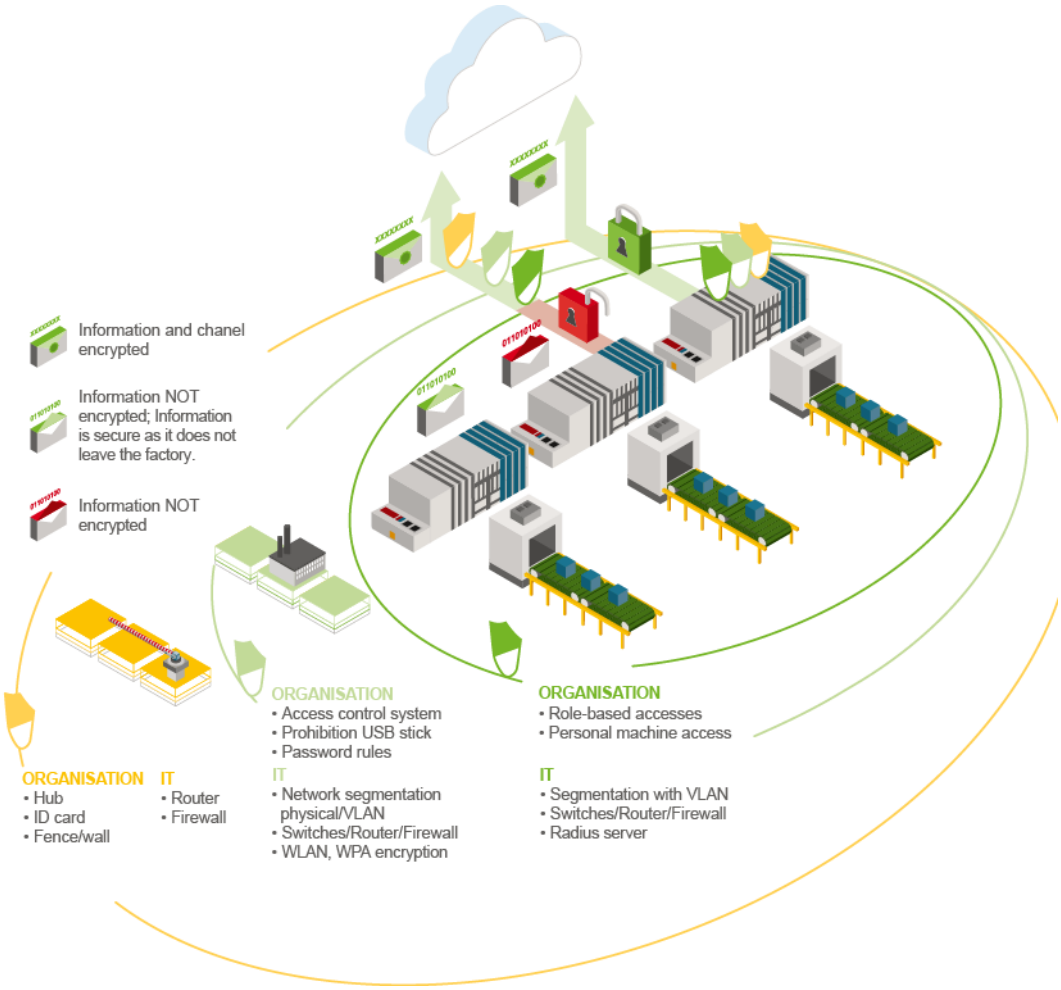


Figure 1: Onion Model

The figure "Onion Model" shows an example of an infrastructure protected by multi-tier security measures. The outer layer (yellow) represents physical access control governing entry to premises. Physical access control ensures that unauthorized persons cannot readily enter critical facilities. The middle layer (light green) represents guidelines and processes in combination with technical measures for network security. These measures represent additional barriers for a potential attacker. However, if an attacker nonetheless makes it through to the controllers, the risk that the controllers will be compromised can be further reduced through security measures on the controller level (dark green layer). Only threats and potential security measures on the controller level are considered below.

**Note**

**Further information on cybersecurity in production facilities:**
The white paper “IT Security in Production Facilities” describes detailed measures for implementing cybersecurity in production. You can request this white paper from https://www.wago.com/ downloads.

## 5.2 Specific Threats on the Example of a Reference Architecture

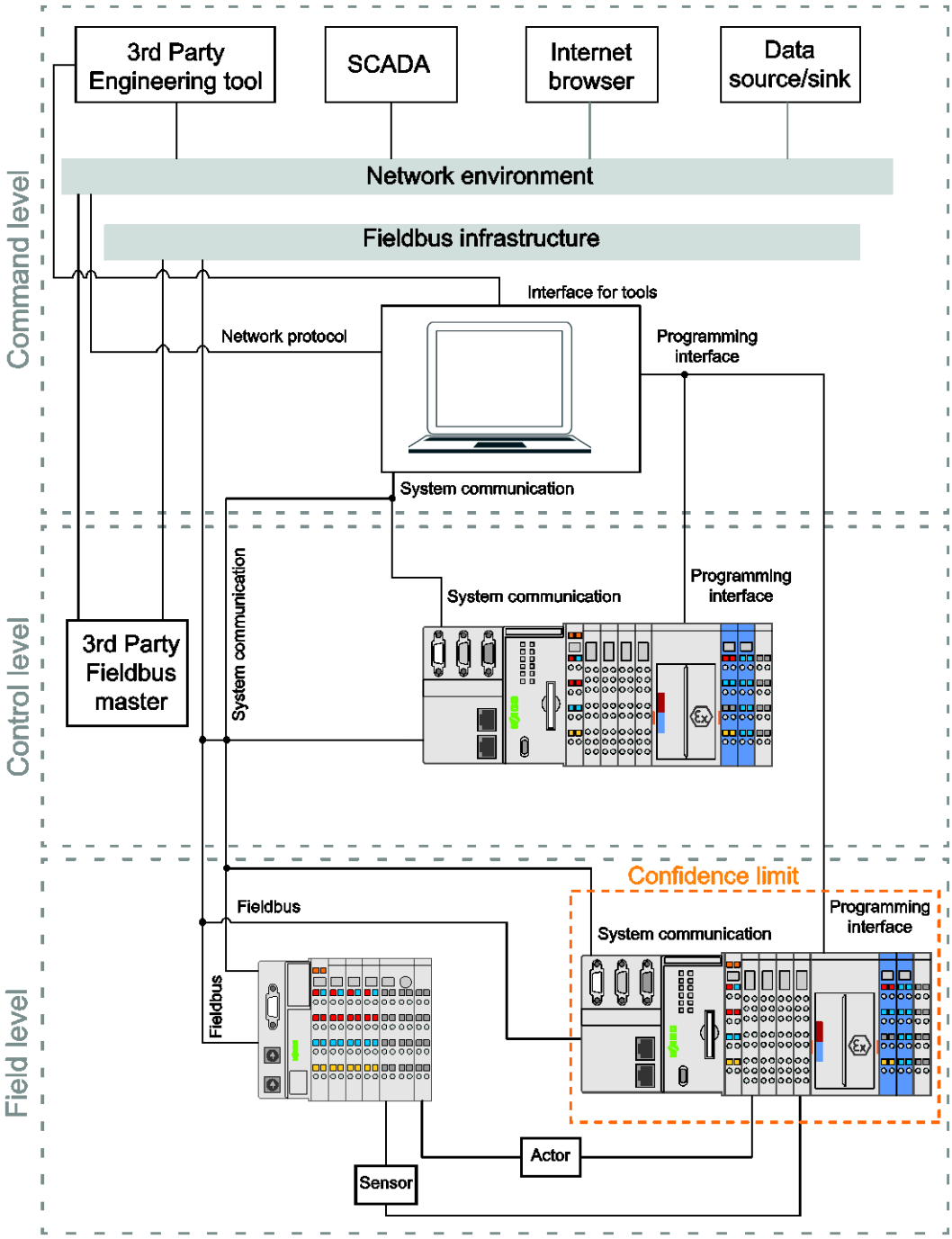


Figure 2: Reference Architecture

The reference architecture represents a classic application environment in which WAGO control systems are used.

The so-called trust boundary represents the transition from a trusted area to an untrusted area, separating the different security levels. The trust boundary is especially significant, since possible attack vectors can arise in the transition from or to another area. The various interfaces that the controller provides for communication with other systems represent potential threats to the controller.

Possible cyber attacks that can occur through both physical access and network-based access to the controller are described below. If an attacker has physical access to the controller, interaction can occur via additional connected input devices through the interfaces.

### 5.2.1 Physical Interfaces on the WAGO Controller

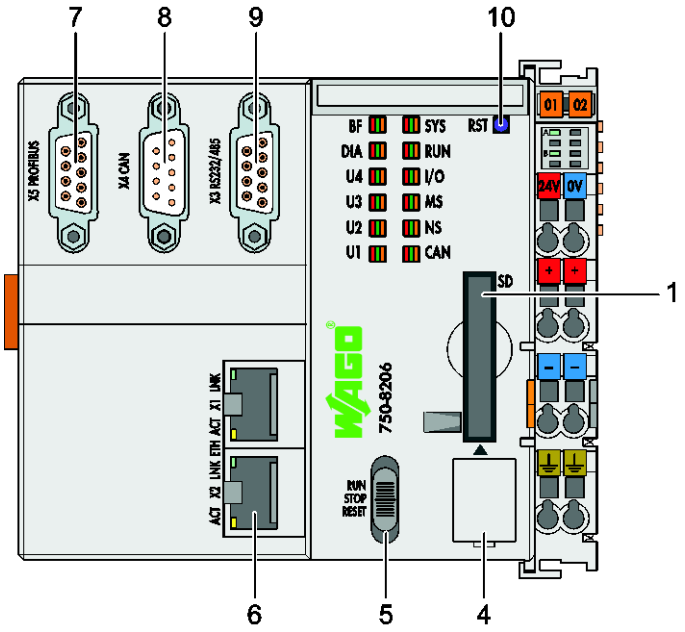


Figure 3: Physical Interfaces on the WAGO Controller

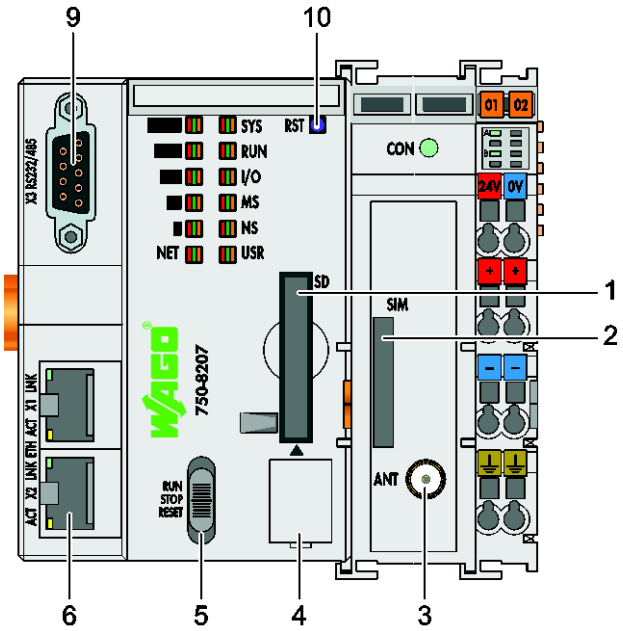


Figure 4: Physical Interfaces on the WAGO Controller with GSM/3G Modem Interface

1 Memory card slot
2 SIM card slot
3 Antenna
4 Service interface
5 Mode selector switch
6 ETHERNET interfaces
7 Fieldbus connection – PROFIBUS DP
8 Fieldbus connection – CANopen
9 Serial interface (RS-232)
10 Reset button

**Note**

**WAGO controller product manuals!**
You can find detailed information on the controllers in the corresponding product manuals at www.wago.com.

#### 5.2.1.1 Reset Button

Different functions can be executed using the Reset button (10) depending on the position of the mode selector switch. To prevent the device from being reset to the default settings, rendering the password obsolete, access to the Reset button should be restricted to the group of authorized persons.

**Note**

**Restrict access to the control cabinet!**
Install the controller in a control cabinet and ensure that only a restricted group of persons has access to the control cabinet!

Network settings and the following passwords are reset:
- “admin” (Linux),
- “admin” (WBM)
- “user” (WBM)

The following passwords are not reset:
- “root” (Linux)
- “user” (Linux)

#### 5.2.1.2 Mode Selector Switch

The mode selector switch (5) can be used to switch between STOP, RUN and RESET. In order to prevent a denial-of-service (DoS) attack being carried out with this against a running CODESYS or e!RUNTIME application, access to the mode selector switch should be restricted to the authorized group of persons.

**Note**

**Restrict access to the control cabinet:**
Install the controller in a control cabinet and ensure that only a restricted group of persons has access to the control cabinet.

#### 5.2.1.3 ETHERNET Interfaces (X1/X2)

The two ETHERNET interfaces (6) of the controller can be operated in either switch mode or separated mode. Switch mode is activated by default upon delivery. You can find an overview of the network services in the section “Standard Settings” > “Network Services.”

If the ETHERNET interfaces are not used, you can restrict their function in the configuration; see the section “Hardening” > ... > “Restrict Access via Open Network Interfaces.”

**Note**

**Note the security information for ETHERNET-based devices:**
Note the security information for WAGO ETHERNET-based devices at https://www.wago.com/de/automatisierungstechnik/security.

#### 5.2.1.4 Service Interface

The service interface (4) is provided for the use of the WAGO service protocol. The system can be configured through it, e.g. with the “WAGO Ethernet Settings,” “WAGO-*I/O-CHECK*” or ***e!****COCKPIT* software.

If the service interface is not used, it should be disabled for security reasons in order to reduce the possibilities for attack; see “Hardening” > ... “Disable Service Interface.”

#### 5.2.1.5 Serial Interface (RS-232)

The RS-232 communication interface (9) is unassigned by default and intended for use of the CODESYS or ***e!****RUNTIME* runtime environment.

If the serial interface is not used, it should be unassigned; see the section “Hardening” > … > “Disable the Linux® Console on the Serial Interface.”

**Note**

**RS-232 communication interface is not always available.**
Note that the RS-232 interface is not found on all devices!

#### 5.2.1.6 Memory Card Slot

The controllers have a memory card slot (1). A potential attacker could start the system from a prepared SD card. As a general rule, as soon as an SD card is installed, the controller boots from the SD card. This allows the data in the internal flash memory to be manipulated or the control application to be interfered with directly. Manipulation is difficult or impossible to detect.

**Note**

**Restrict access to the control cabinet:**
Install the controller in a control cabinet and ensure that only a restricted group of persons has access to the control cabinet.

#### 5.2.1.7 GSM/3G Modem Interface

Certain controllers have an additional modem module with an SIM card slot (2) and an SMA jack for the antenna (3) for use of the wireless functionality. The SIM card should be protected against unauthorized access with PIN protection.

**Note**

**Restrict access to the control cabinet:**
Install the controller in a control cabinet and ensure that only a restricted group of persons has access to the control cabinet.

**Note**

**The antenna signal may be too weak in the control cabinet:**
If you store the device in the control cabinet, ensure that the antenna signal is sufficient. If the antenna signal is too weak, there is no access to the mobile network.

## 5.2.2 Access via the Network

This section describes potential network-based cyber attacks, which can occur via local networks, for example. Due to the increasing complexity and interconnectedness of the communication elements on multi-layer networks (see the figure “Reference Architecture”), cyber criminals can endanger system security in many different ways. Since industrial controllers are increasing linked to corporate networks, they represent an additional attack vector.

### 5.2.2.1 Software Components

Vulnerabilities in the software used represent a potential threat, since they may provide opportunities to introduce malicious code or carry out DoS attacks.

To counteract these threats, it is advisable to keep the controller software up-to-date (e.g., through a firmware update). Update you system regularly with patches provided by WAGO Kontakttechnik GmbH & Co. KG. In addition, take hardening measures that make the controller more secure.

To get an up-to-date list of your controller’s Linux® packages, you can execute the following command on the Linux® console:

```
„ipkg list“.
```

### 5.2.2.2 „Man-in-the-Middle Attacks“

Man-in-the-middle attacks are attacks against the communication channel between two communication partners. The attacker masquerades as a trusted source, so the two communication partners do not realize they are communicating with the attacker. This allows the attacker to read and manipulate all the information transferred.

For optimal controller security, it is advisable to change the TLS configuration from “Standard” to “Strong”; see the section “Hardening” > “TLS Encryption.” It is also advisable to replace the generic TLS certificate; see the section “Hardening” > ... > Generate and Replace Certificates.

### 5.2.2.3 Network Interfaces and Protocols

Cyber criminals can use port scanners to scan other computers for opportunities for access via open ports. Every open port represents a potential threat, since the system can be attacked via network services that are not needed.

In order to block specific network interfaces/protocols that are not needed in a specific application environment, a firewall can be used. For more information, see the section “Hardening” > ... > “Configure Firewall.”

You can find an overview of the network services that WAGO uses by default in the section “Standard Settings” > “Network Services.”

**Note**

**Note the security instructions for network-based WAGO controllers:**
You can find security instructions for network-based WAGO controllers at https://www.wago.com/de/automatisierungstechnik/security

#### 5.2.2.4 Firewalls

With a firewall, you can set up a protective measures against insecure and/or harmful connections. A firewall rule should always be configured restrictively in order to restrict access to a particular network interface. Access to the network interface should be restricted to individual computers or subnets that access the service. You can find more information on firewall rules in the section “Hardening” > “Configure Firewall.”

### 5.2.3 Access via Users and Passwords

**Note**

**Change Passwords**
The default passwords for all users provided upon delivery are documented in these operating instructions and do not offer sufficient protection! Change the passwords to meet your particular needs during initial startup.

Password protection with default passwords or low complexity passwords does not offer sufficient protection. A potential attacker can easily circumvent such password protection and get access to the user account in question with the corresponding permissions.

Each of the services listed below has its own user administration for user accounts:

- Web-Based Management (WBM)
- Linux®
- SNMP
- CODESYS Web visualization
- ***e!****RUNTIME* Web visualization

**Recommendations for secure passwords:**

- Change your password regularly.
- Use at least eight characters.
- Do not store you password on your hard drive as unencrypted text.
- Use as many different characters, upper and lower case letters, special characters and numbers as possible.
- You password should not contain any reference to your identity such as names or birthdays.

**Information**

**You can get more information from the “National Institute of Standards and Technology” (NIST).**
In “NIST Special Publication 800-63B,” the section “Authenticator and Verifier Requirements” provides instructions for secure passwords!
(https://pages.nist.gov/800-63-3/sp800-63b.html)

# 6 Hardening

Hardening means increasing the security of a system with a series of measures to better protect it against threats; see the section “Threats to Industrial Controller Systems.”

The WAGO controllers are Linux®-based PCs. The Linux® operating system offers numerous network services that should not be accessible to every system and every user. Only the necessary processes should be active, and only with minimal rights. Some measures that help pare your system down to a minimum of security-sensitive aspects are described below.

**Note**

**This document makes no claim to completeness!**
Make sure to review the security technology of your application with reference to your requirements.

## 6.1 Restrict Physical Access

### 6.1.1 Disable Service Interface

The service interface is used for communication with the WAGO-*I/O-CHECK* and WAGO ETHERNET Settings software, among other things. If the service interface is not used continuously, it should be disabled; also see the section “Threats to Industrial Control Systems” > “Specific Threats on the Example of a Reference Architecture.”

**Note**

**Only administrator can disable service interface:**
You need admin permissions to disable the service interface.

1. In the WBM, select the menu item Administration > Service Interface to disable the service interface.


Figure 5: Disable Service Interface

2. In the section “Assign Owner of Service Interface,” select the Unassigned option.
   Specify that the serial interface is not to be assigned to any specific application and is available, so that a CODESYS program, for example, can access it via function blocks.

3. Click the **[Change Owner]** button.

4. Restart the controller in order to apply the change.

### 6.1.2 Disable Linux® Console on the Serial Interface

The controller has a serial RS-232 interface that can be configured for various functions. It is unassigned by default upon delivery and can be used by applications such as CODESYS. Alternatively, the serial interface can also be assigned to Linux® so a Linux® command line can be provided. With this setting, the serial interface can be used for communication with the Linux® console.

If the serial interface is used for other applications, it is blocked for access to a console. If the serial interface is not supposed to be used regularly for access to the console, it is advisable to disable the link between the serial port and the console.

**Note**

**Only administrator can disable Linux® console:**
You need admin permissions to disable the Linux® console.

1. In the WBM, select the menu item Administration > **Serial Interface** to disable access to the Linux® console.

2. Select the **Unassigned** option.
This ensures that the service interface is not assigned to a command line.


Figure 6: Disable Linux® Console

3. Click the [**Change Owner**] button to apply the change.

# 6.2 Secure Network Access Points

## 6.2.1 Encrypted Communication

### 6.2.1.1 Web Protocols for WBM Access

The HTTP and HTTPS web protocols can be used to access the WBM pages for the controller. HTTPS is preferred because it uses the SSL/TLS protocol. The SSL/TLS protocol ensures secure communication through encryption and authentication

The default setting for the controller allows strong encryption, but uses only simple authentication methods. As authentication for any secure communication channel plays a central role, it is strongly recommended that you use secure authentication. The security certificate saved on the controller is the basis for authentication. The default location for the security certificate is: /etc/lighttpd/https-cert.pem

As delivered, the controller uses a generic security certificate based on x509. To allow secure authentication, you must replace the generic security certificate with a security certificate specific for the individual device.

### 6.2.1.2 TLS Encryption

When an HTTPS connection is established, the Web browser and Webserver negotiate what TLS version and what cryptographic method are to be used.

The “TLS Configuration” group of the WBM page “Security” can be used to switch the cryptographic methods allowed for HTTPS and the TLS versions that can be used.

The settings “Strong” and “Standard” are possible.
If “Strong” is set, the Webserver only allows TLS Version 1.2 and strong algorithms.
Older software and older operating systems may not support TLS 1.2 and encryption algorithms.
If “Standard” is set, TLS 1.0, TLS 1.1 and TLS 1.2 are allowed, as well as cryptographic methods that are no longer considered secure.

**Information**

**BSI Technical Guidelines TR-02102**
The rules for the “Strong” setting are based on technical guidelines TR-02102 of the German Federal Office for Information Security.
You can find the guidelines on the Internet at: https://www.bsi.bund.de > “Publications” > “Technical Guidelines.”

**Information**

**BSI Guidelines on Migration to TLS 1.2**

The German Federal Office for Information Security guidelines on migration to TLS 1.2 contain “compatibility matrices” that show what software is comparable with TLS 1.2.

You can find the guidelines on the Internet at: https://www.bsi.bund.de > “Topics” > “Standards and Criteria” > “Minimum Standards“.

For optimal controller security, it is advisable to change the TLS configuration in the Web-Based Management from “Standard” to “Strong.”

1. In the WBM, navigate to the menu **Security** > **TLS Configuration**.

2. Check the **Strong** box


Figure 7: TLS Configuration

3. Click the [**Submit**] button to apply the changes.

#### 6.2.1.3 Generate Diffie–Hellman Parameters

The Diffie–Hellman method is a procedure for agreeing on a shared digital key. What is transferred is not the secret session key, but only the result of a computational operation. This allows two communication participants to communicate securely over an open network. They use an encryption method of their choice, which uses the shared key agreed through the Diffie–Hellman method.

You can generate Diffie–Hellman parameters with the XCA key management software.

1. Open the XCA software; from the **Extra** menu, select the **Generate DH Parameters** submenu.

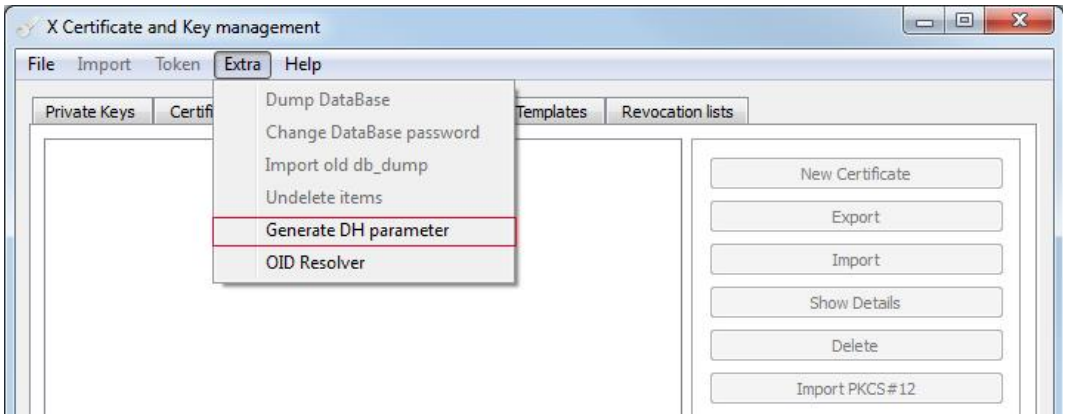


Figure 8: Generate Diffie–Hellmann Parameters

2. Select a key length of at least 2000 bits.

**Note**

**Note the key length requirements!**
If use after 2022 is planned, the key length should be at least 3000 bits; see BSI TR 02102-1, page 56/57, “7.2.1. Diffie–Hellman.”

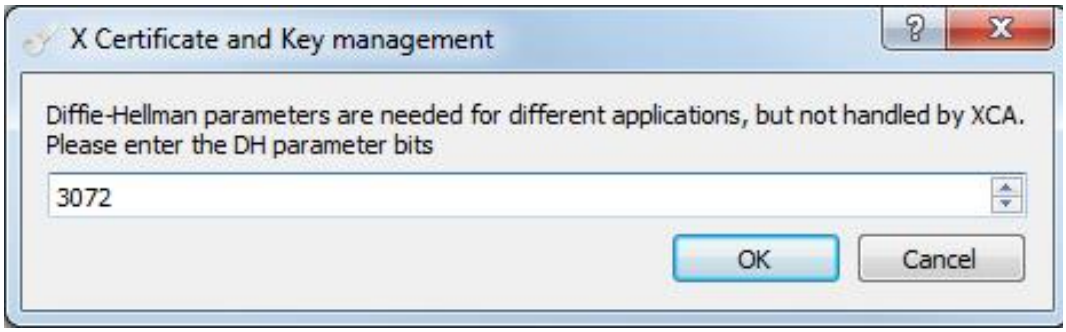


Figure 9: Key Length, DH Parameters

**Note**

**Generating the parameter may take a significant amount of time.**
Depending on the key size selected, generating the DH parameters may take a very long time.

The parameters “p” and “g” are generated in the background. When this concludes, a dialog opens for saving the parameters. The parameters “p” and “g” do not have to remain secret and can be transferred via an insecure connection. During key exchange with the Diffie–Hellman method, the two communication partners also select a secret number in addition to the parameters “p” and “q.” A new number is calculated from the public and private numbers. The new numbers are exchanged again in order to generate a new secret key “k,” which is not accessible to third parties.

Diffie–Hellman key exchange is used for SSL/TLS connections; see the section “OpenVPN,” and for the Web server, also see the following section.

#### 6.2.1.3.1 Replace Diffie–Hellman Parameters for the Web Server

You can replace the generated Diffie–Hellman parameters (see the section “Generate Diffie–Hellman Parameters”) for the Web server with your own:

1. Load the generated parameters into the following folder on the controller via SCP/FTPS/SFTP:
   /etc/lighttpd/

2. You must indicate the parameter file in the configuration files “tls-strong.conf” and “tls-standard.conf” in the key “ssl.dh-file”:
   ssl.dh-file = “/etc/lighttpd/<name of your DH parameter file>”

3. Finally, restart the Web server:
   /etc/init.d/lighttpd stop
   /etc/init.d/lighttpd start

### 6.2.1.4 “Harden” SSH Access

Besides authentication via usernames and passwords, SSH also supports authentication based on a key pair (private/public). Generate the keys with the free Windows program “Putty Key Generator” (Putty v0.68 or current version, steps 1–10), for example. Furthermore, “root” login should be blocked (see the section “‘Harden’ SSH Access” > “Refuse Root Login”), and the default port should be changed (see the section “Change Default Network Port”).

1. Download PuTTYgen from the website
   https://the.earth.li/~sgtatham/putty/latest/w64/puttygen.exe.

2. Start the PuTTY utility program PuTTYgen:

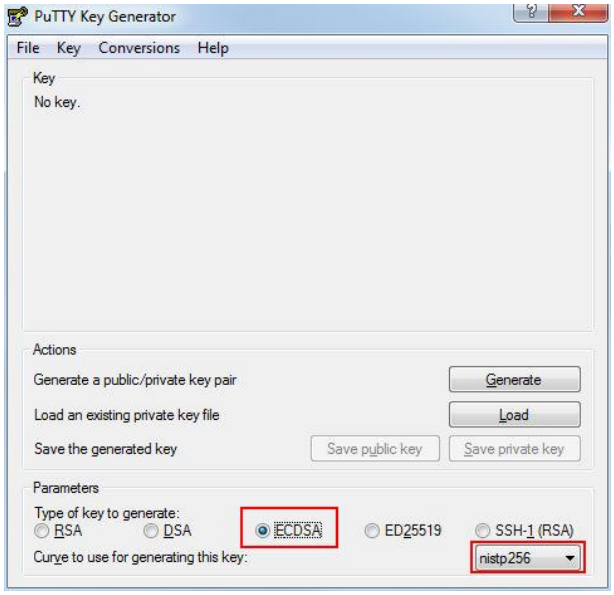


Figure 10: Start PuTTYgen

3. Select the type of key to generate (ECDSA) and the elliptical curve (nistp 256).

**Note**

**Note the recommendations for the cryptographic method!**
According to the BSI TR-02102-4 technical guidelines (version 2017-01), a key length of at least 250 bits is required for ECDSA.

4. Then click the [**Generate**] button to start key generation.

5. Move the mouse randomly around the window during key generation until the progress bar has reached the end. PuTTYgen generates the random numbers necessary for key generation from the mouse cursor movements, among other things.

When generation is complete, the key data is shown in the window:

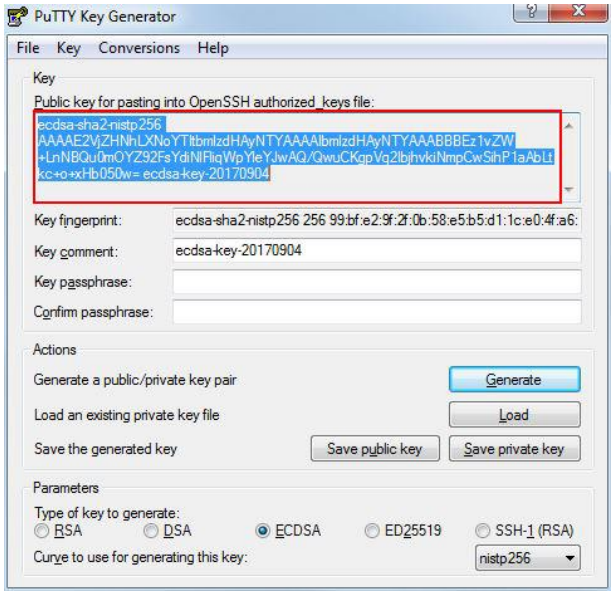


Figure 11: PuTTYgen Key Generation

6. Click the buttons [**Save public key**] and [**Save private key**] to save your key pair.

7. Specify another password for the private key in the **Key passphrase** input field.

8. Confirm the password in the **Confirm passphrase** input field.

9. Save the private key in a secure location to prevent unauthorized parties from using the key for authentication on the device.

The public key must be stored on the controller in the home directory of the user who is to use the key for authentication (e.g., /home/user). A subfolder “.ssh” and the file “authorized_keys” must be created there first; see the description below.

**Note**

**Note the file permissions:**
The ".ssh" directory must have the Linux® permissions **rwx------ (700)**, and the "authorized_keys" file must have the Linux® permissions **rw------- (600)**. The key will not be accepted otherwise.

10. Create the ".ssh" directory:
```
user@PFC200-40ED7D:~ pwd
/home/user
user@PFC200-40ED7D:~mkdir .ssh && chmod 700 .ssh
```

11. Copy the public key from the PuTTY key generator; see the figure "PuTTYgen Key Generation."

12. Add the key to the file "authorized_keys":
```
user@PFC200-40ED7D:~ pwd
/home/user
user@PFC200-40ED7D:~ cat << 'EOF' >.ssh/authorized_keys
  > Public key (ecdsa-sha2-nistp256AAAAE2V … ecdsa-key-
20170904)
  >EOF
user@PFC200-40ED7D:~ chmod 600 .ssh/authorized_keys
user@PFC200-40ED7D:~ ls -l .ssh/authorized_keys
-rw-------    1 user     user             261 Jan 24 08:39
.ssh/authorized_keys
```

**Note**

**Note the syntax when entering the keys:**
Each key must be written on one line in the "authorized_keys" file.

13. Test access via your private key before disabling login via password entry.

**PuTTY Configuration**

To use the key for authentication, you must inform your SSH client of the key. An example for PuTTY is described below (other clients should be configured analogously):

1. Start the PuTTY utility program. The "PuTTY Configuration" dialog opens.

2. Navigate through the directory structure to the menu **Connection > SSH > AUTH**. An SSH authentication options dialog opens.

3. Select your private key in the "Authentication parameters" section.

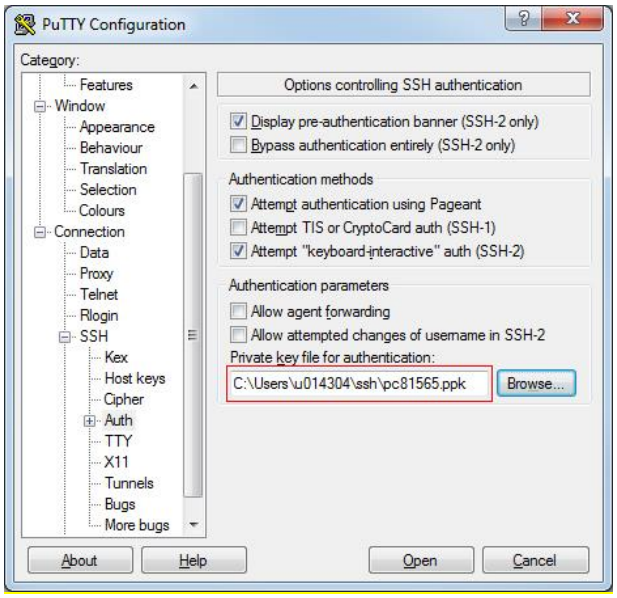


Figure 12: PuTTY Configuration

4. Switch to the **Session** menu in the directory structure.

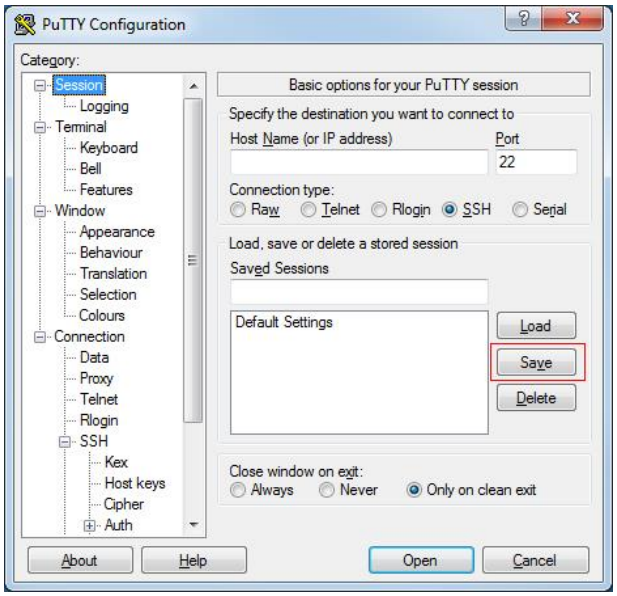


Figure 13: Save PuTTY Configuration

5. Save the configuration with the [**Save**] button.

#### 6.2.1.4.1 Disable Login via Password Entry

Two options are available:

**Configuration file**:

1. Set "`PASSWORD_LOGIN`" to "false" (default = true). You can find the configuration file at /etc/dropbear/dropbear.conf.

2. Restart the service.

**WBM**:

1. In the WBM, navigate to the menu **Ports and Services** > **SSH**.
2. Uncheck the **Allow password login** box.


Figure 14: Disable Login via Password Entry

3. Save the setting with the [**Submit**] button.

When changes are made via the WBM, the service restarts automatically.

#### 6.2.1.4.2 Refuse Root Login

**Note**

**Be careful not to lock yourself out of the system!**
Set up a user account in advance with same access rights as the root/superuser account. The commands `su` and `sudo` allow you to give individual users administrator rights.

Two options are available:

**Configuration file:**

1. Set "`ROOT_LOGIN`" to "false" (default = true). You can find the configuration file at /etc/dropbear/dropbear.conf.
2. Restart the service.

**WBM**:

1. In the WBM, navigate to the menu **Ports and Services** > **SSH**.
2. Uncheck the **Allow root login** box.


Figure 15: Refuse Root Login

3. Save the setting with the [**Submit**] button.

When changes are made via the WBM, the service restarts automatically.

#### 6.2.1.4.3 Changing the Server Key

In the delivery state, a generic key is set on the controller and must be replaced with an individual key.

The devices 750-8202 to 750-8207 are not equipped with a hardware random number generator; thus, they must be initialized with random numbers from an external source. To set up all other devices, you can start directly with Item 3.

To enter the new key on the device, follow these steps.

1. Establish a connection to the Linux® control panel via SSH.

2. Copy the random numbers from your host to the controller:

**Linux via SSH:**

```
openssl rand 1024 | ssh root@192.168.1.72 'cat >
/dev/urandom'
```

**Windows via PuTTYgen**:

- Generate the random numbers as described in Section "'Hardening' SSH Access" (Items 1 to 5).
- Copy the key data displayed in the window to the clipboard.
- Establish a connection to the Linux® control panel via SSH.
- Start the command "cat > /dev/urandom" on the controller.
- Paste in the key data from the clipboard.
- To finish the entry, first press ENTER; then CTRL+D.

3. Log on to the controller:
   ```
   ssh root@192.168.1.17
   ```

4. Generate the new key on the controller:
   ```
   cd /etc/dropbear
   rm -f dropbear_ecdsa_host_key && dropbearkey -t ecdsa -s 521
   -f dropbear_ecdsa_host_key
   rm -f dropbear_dss_host_key && dropbearkey -t dss -s 1024 -f
   dropbear_dss_host_key
   rm -f dropbear_rsa_host_key && dropbearkey -t rsa -s 2048 -f
   dropbear_rsa_host_key
   ```

5. Save the key and restart the controller:
   ```
   sync && reboot
   ```

**Linux via SSH:**

After the restart, a message in the command line indicates that the changed server key may point to a Man-in-the-Middle attack; e.g.:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:CSD8j+LxPxE4mtLIPzgyTbKozoCPYZNBvPEQVfrpVws.
```

6. Delete the key stated in the error message from the subfolder ".ssh/known_host" in the home directory on your host.

**Windows via PuTTY**:

The following warning message appears after the restart:

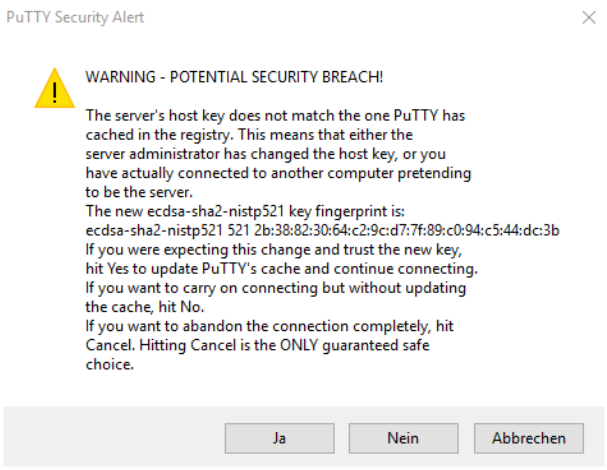


Figure 16: Neustart Putty

6. Click [**Ja**] to update the cache and continue with the connection.

You can find more information in section “Hardening” > ... > “Generate and Replace Certificates.”

### 6.2.1.5 Generate and Replace Certificates

A certificate allows a secure connection for network communication and is used for authenticating the remote host. The green lock symbol in the browser indicates that this website has a valid, trusted certificate and that the connection is secure.

It is advisable to replace the WAGO certificates provided by default with your own certificates, since the private key is identical for all devices, customers and firmware and thus cannot be considered secret. Certificates you create yourself must be signed by a certificate authority (the so-called root CA). The root certificate forms the shared trust anchor for all certificates subordinate to it and must be stored in the local trust store of the browser or client.

The following sections describe an example of creating keys and certificates with the XCA key management software. This free software allows you to create certificates yourself. The certificates/keys are stored in a local database file. The database, which contains private keys among other things, is protected with a password.

#### 6.2.1.5.1 Create a Template for the Certificates

1. Open the XCA software; from the **File** menu, select the **New Database** submenu.

2. Select a storage location and appropriate name for the database.

3. Enter a password to protect the database. The newly created database then opens automatically:

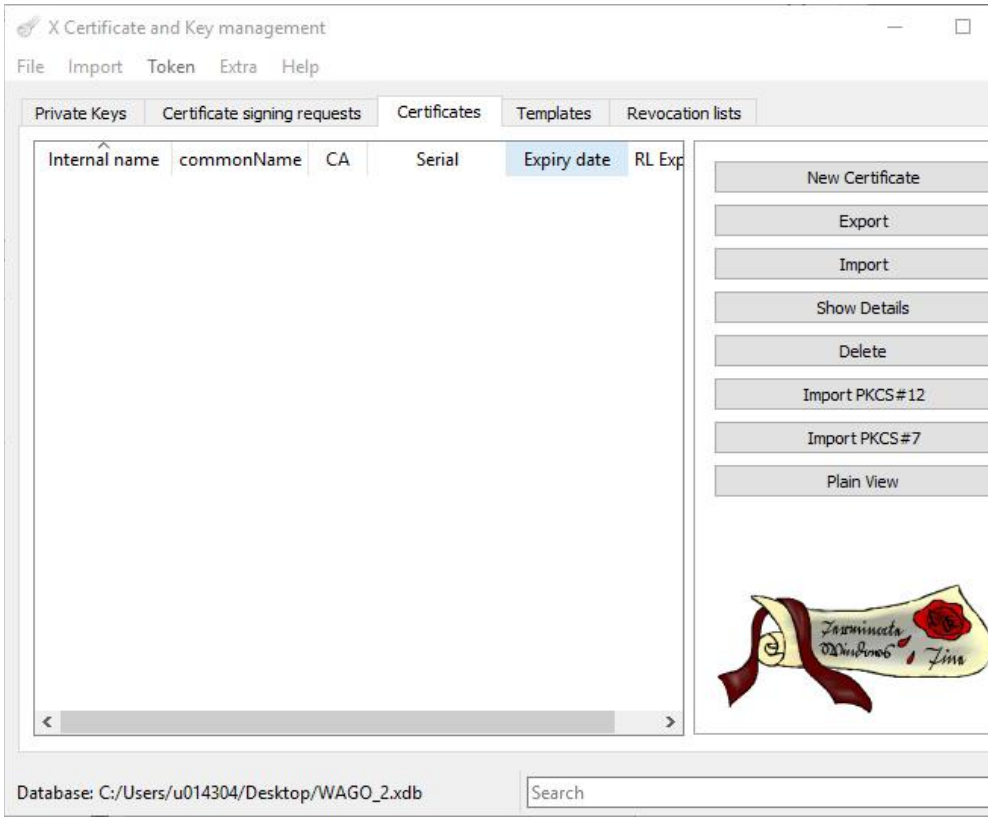


Figure 17: XCA Database

4. In the “Templates” tab, select the [**New Template**] button**.**

5. Select the setting “[default] empty template” in the dialog window “Preset Template Values”.

6. Confirm the selection with [**OK**].

7. In the “Edit XCA template” dialog, change to the “Subject” tab.

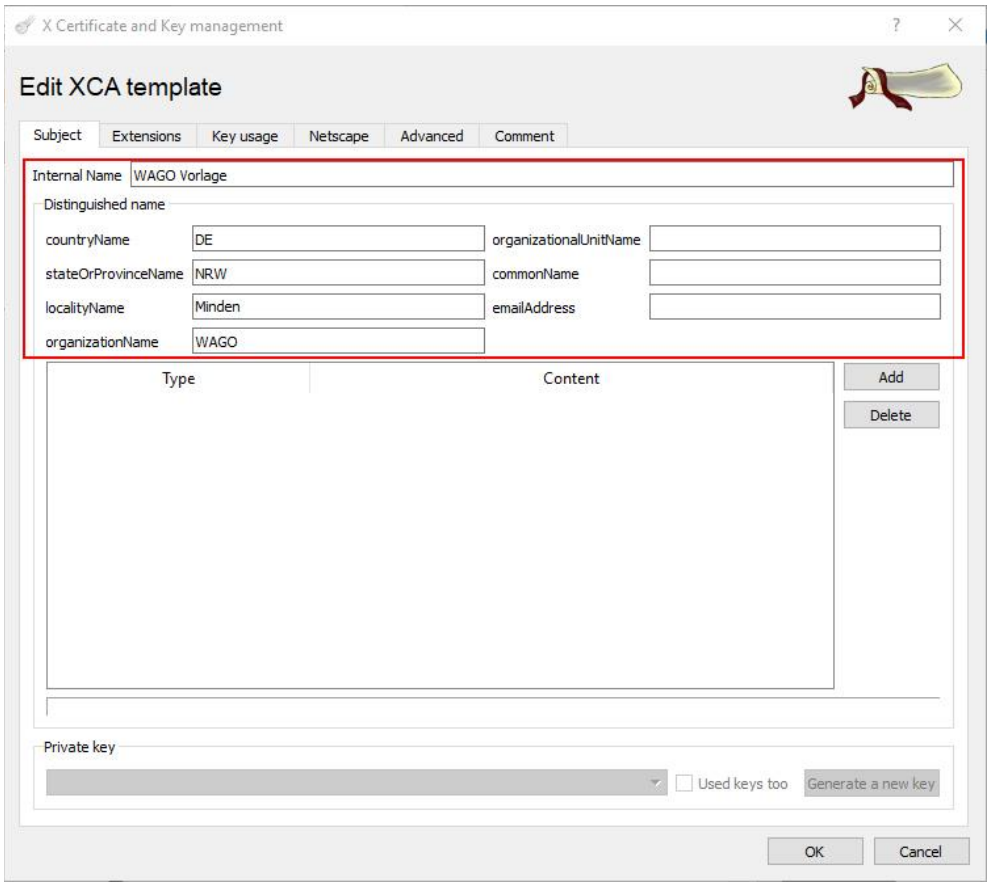


Figure 18: Create Template, “Subject” Tab

Table 8: “Subject” Tab

| Input Field | Explanation |
|---|---|
| Internal name | The value in this field serves as an internal reference and should identify the certificate uniquely. |
| countryName | Country code (e.g., DE for Germany) |
| stateOrProvinceName | State or province (e.g., NRW for North Rhine-Westphalia) |
| localityName | Place where certificate issued |
| organizationName | Name of the organization that issued the certificate |
| organizationUnitName | Department identifier |
| commonName | A general identifier can be stored here. |
| emailAddress | An email address can be stored here. |

8. Fill in the marked input fields in the upper section. The field "commonName" is left empty in the template and filled out later.

9. Confirm your entries with [**OK**].

After the template has been created, it will be displayed in the window.

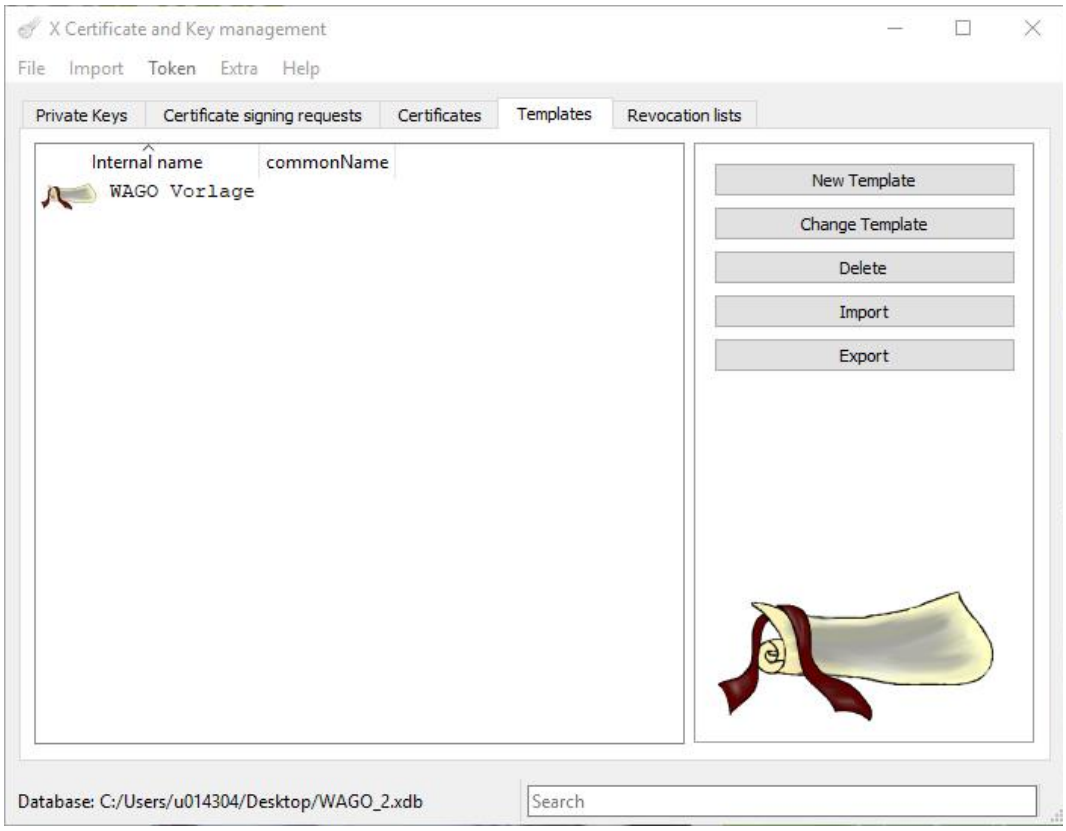


Figure 19: Template created

#### 6.2.1.5.2 Create Root CA Certificate

1. Switch to the “Certificates” tab to create the root CA certificate.
2. Click the [**New Certificate**] button. The following dialog appears:

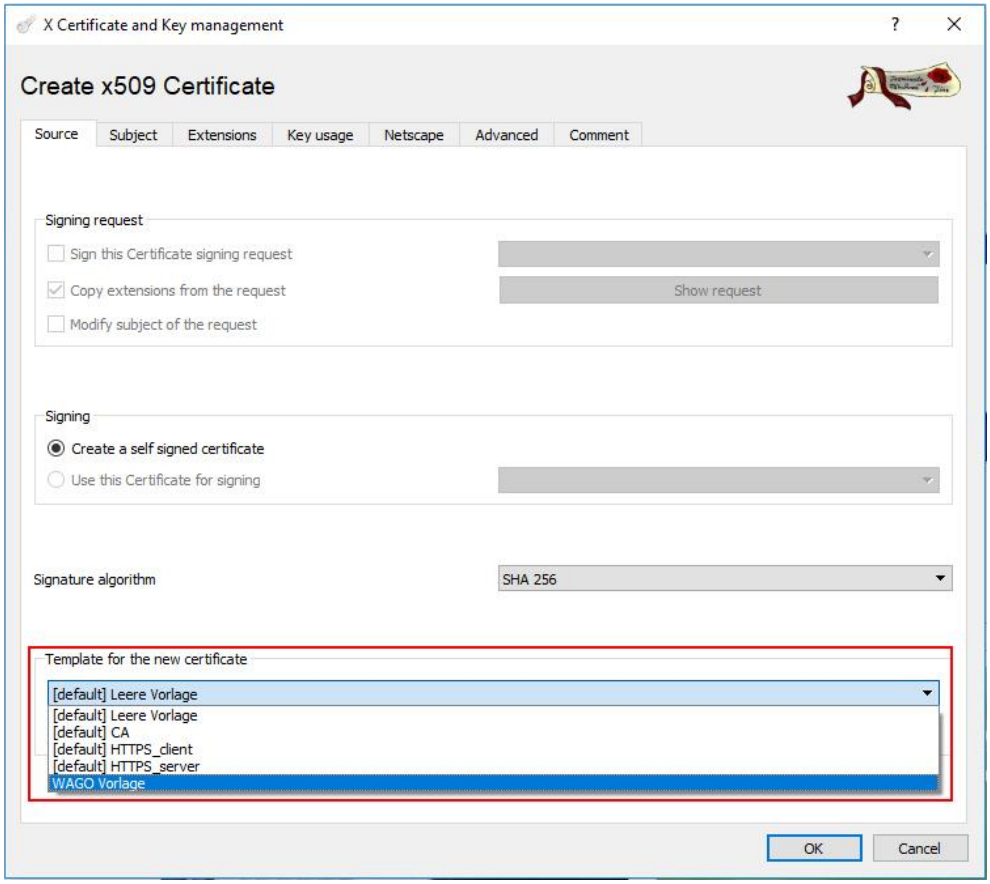


Figure 20: Create Root CA Certificate

3. Select your created template from the “Template for the new certificate” selection field.
4. Select the [**Apply subject**] button.
5. Select the template “[default] CA” from the “Template for the new certificate” selection field.
6. Select the [**Apply extensions**] button.
7. Switch to the “Subject” tab.
8. Enter an identifier in the input field “CommonName”, e.g. “Root-CA”.
9. Select the [**Create new key**] button.


Figure 21: Create New Key

10. Adjust key type and key size for the root CA. The name is default.
The assignment depends on whether the key is generated for the root CA or the controller (for recommended key lengths, see the BSI TR-02102-2 technical guidelines).

11. Click the [**Create**] button to create the key.

12. Close the dialog by clicking [**OK],** after the message about successful key creation.

The created certificate is displayed in the Certificates tab:

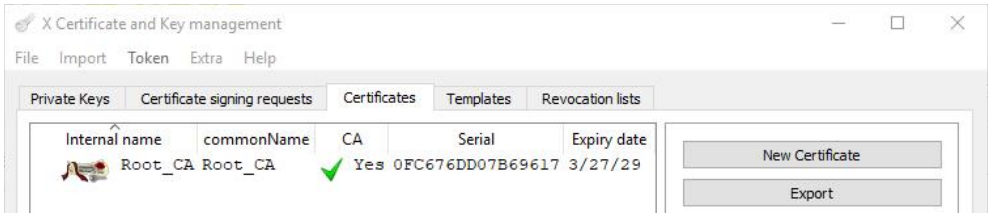


Figure 22: New Certificate Created

#### 6.2.1.5.3 Create Device Certificate

1. Switch to the “Certificates” tab to create the certificate.

2. Click the [**New Certificate**] button. The following dialog appears:

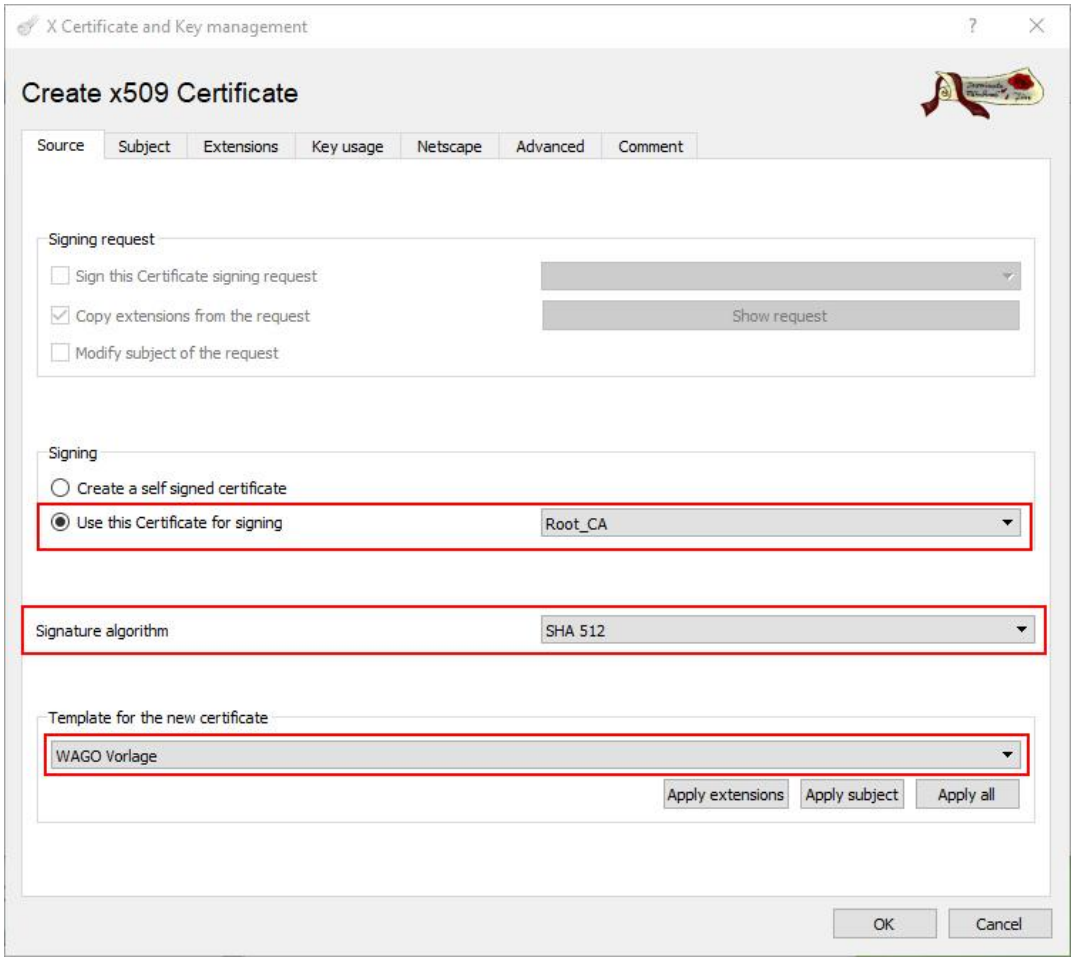


Figure 23: Sign Certificate Request

3. Check the box **Use This Certificate for signing** and select the root CA certificate that has been created.

4. In the **Signature algorithm** selection field, select the value “SHA 512” (see the BSI TR-02102-2 technical guidelines).

5. In the **Template for the new certificate** selection field, select your created template.

6. Select the [**Apply subject**] button.

7. Select the template “[default] HTTPS_server” from the “Template for the new certificate” selection field.

8. Select the [**Apply extensions**].

9. Switch to the “Subject” tab.

10. Enter the IP address in the input field “CommonName”.

11. Select the [**Create new key**] button.


Figure 24: Create New Key

12. Adjust key type and key size for the root CA. The name is default.
The assignment depends on whether the key is generated for the root CA or the controller (for recommended key lengths, see the BSI TR-02102-2 technical guidelines).

13. Select the [**Create**] button to create the key.

14. Switch to the “Extensions” tab.

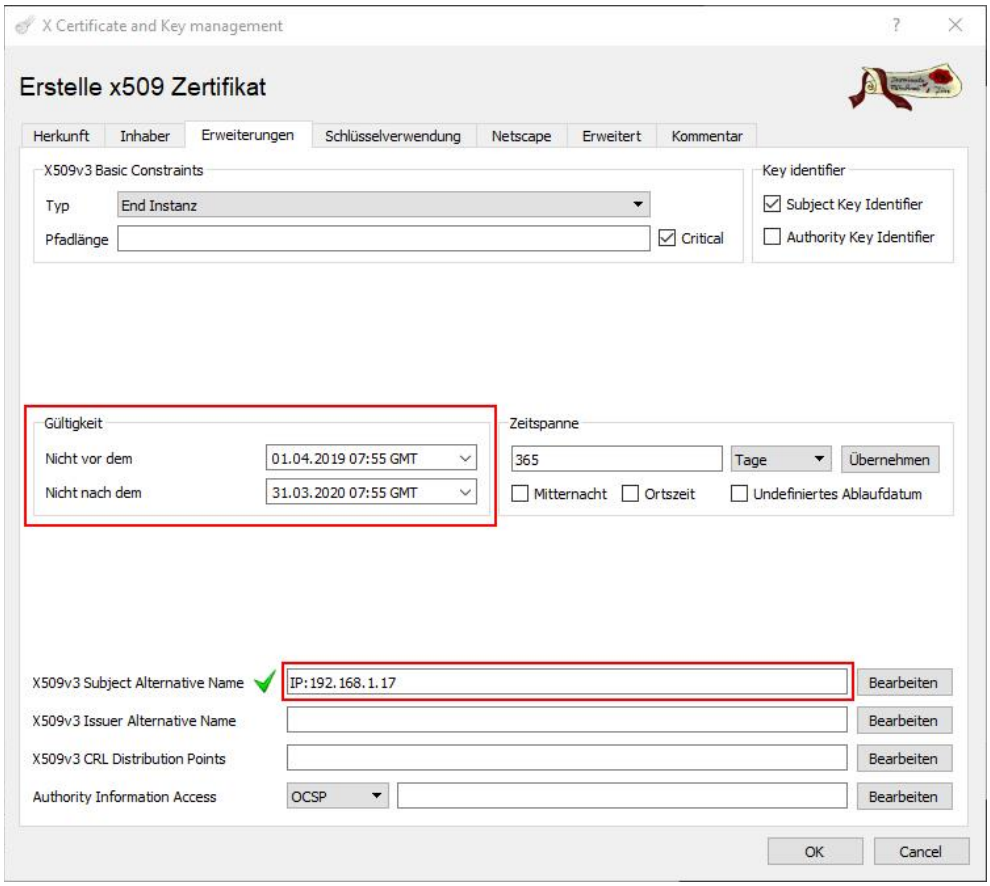


Figure 25: Extensions Tab

15. Set the validity of the device certificate. Observe the recommendations of the "technical guidelines of the BSI TR-02102-2".

16. Add the IP address and/or hostname in the **X509v3 Subject Alternative Name** address field.

**Note**

**The value in the “X509v3 Subject Alternative Name” input field must be identical to the address bar:**
The IP address/host name is used by browsers to determine the identity. If the value entered in the “X509v3 Subject Alternative Name” input field differs from the value in the address bar, the certificate is recognized as invalid.

17. Click the [**Edit**] button. The following input window opens:

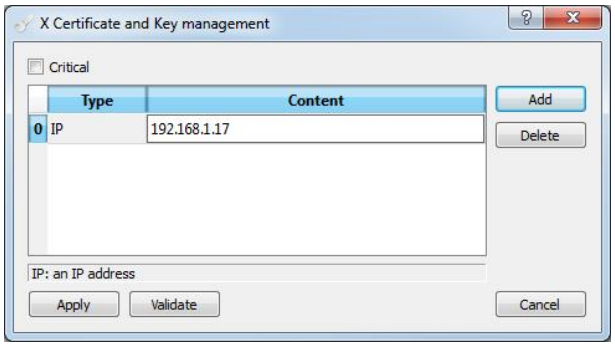


Figure 26: X509v3 Subject Alternative Name, Enter IP Address

18. Click the [**Add**] button.

19. In the **Type** selection field, select either “IP” for the IP address or “DNS” for the hostname.

20. Enter the corresponding value in the **Content** input field.

21. Switch back to the “Key usage” tab to restrict the use of the certificates.

22. Enter the values marked in the figure.

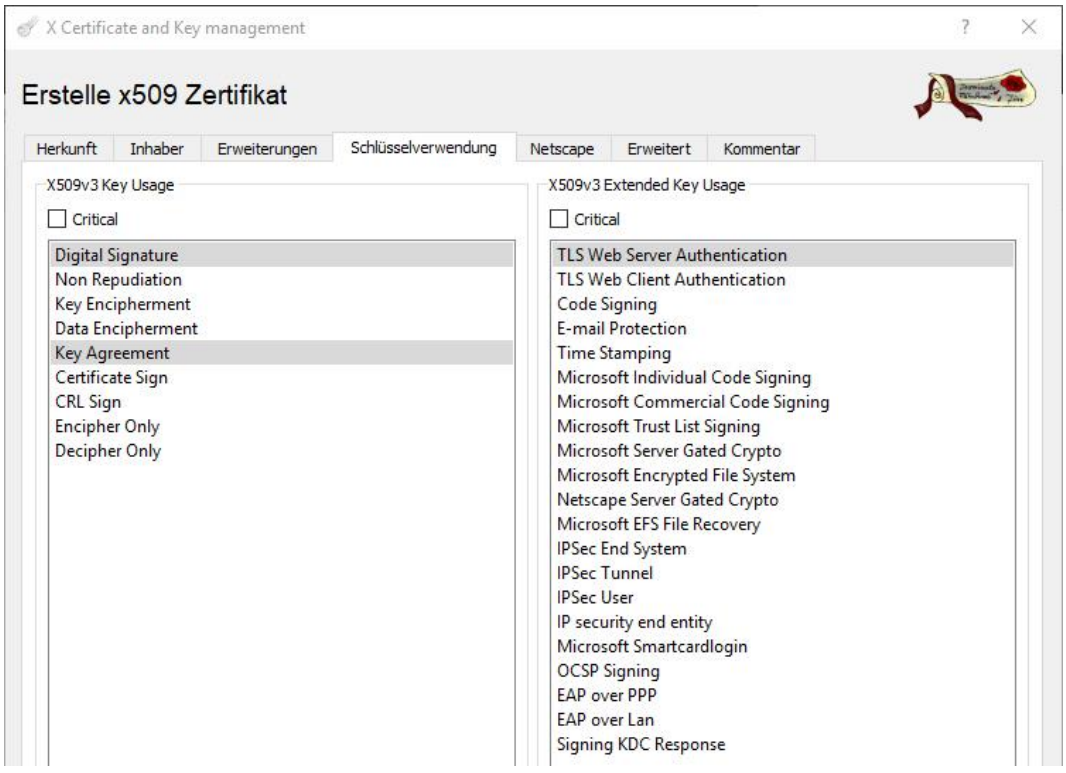


Figure 27: New Certificate Request, Client Key Use

**Note**

**Enter different values for server authentication!**
Note that for server authentication, the entry “TLS Web Server Authentication” is entered in field on the right. In the field on the left, the values “Digital Signature” and “Key Encipherment,” or, alternatively, “Key Agreement,” are entered. The certificate generation procedure is otherwise identical for server and client.

23. Confirm your entries with [**OK**].
The new certificate is shown below the root CA certificate on the “Certificates” tab.

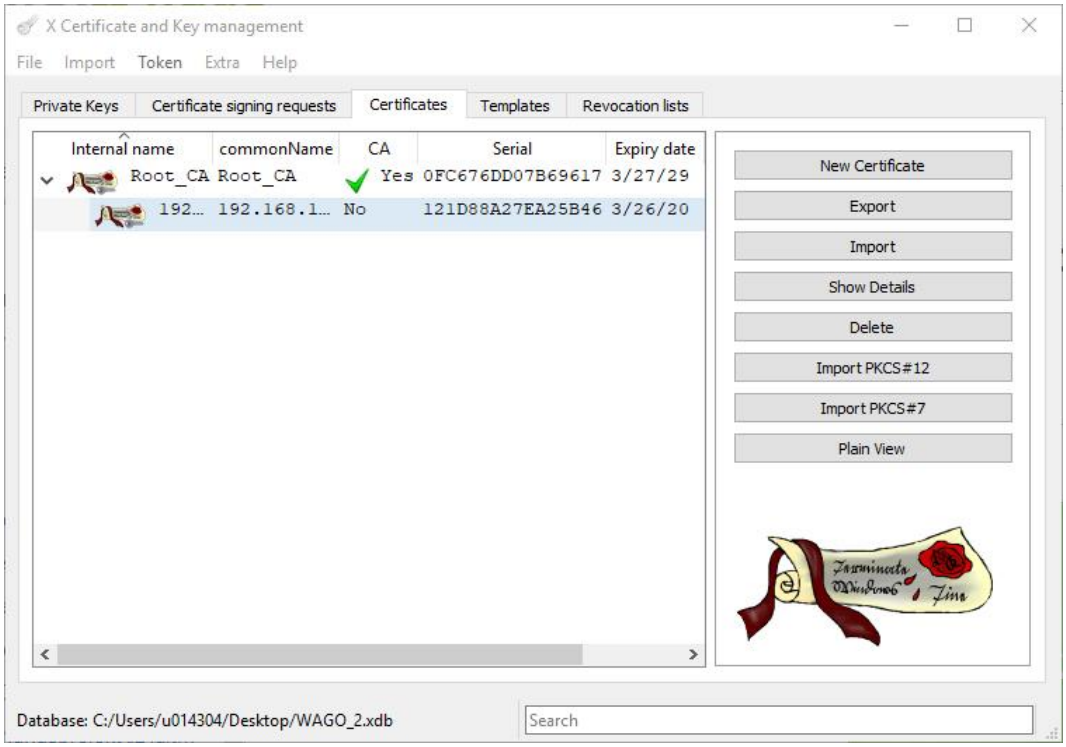


Figure 28: Result Device Certificate

#### 6.2.1.5.4 Export Certificates

1. In the main window, switch to the “Certificates” tab and expand the tree structure completely.
2. Mark your root CA certificate and open the context menu by right-clicking.
3. Select **Export** > **File**.

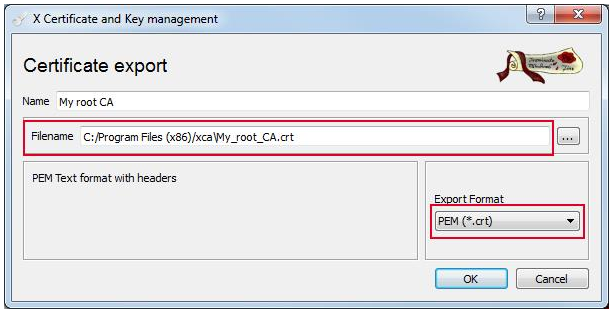


Figure 29: Export Root CA Certificate

4. Select the storage location with the [**...**] button.
5. In the **Export Format** selection list, select the entry “PEM without Key.”
6. Click [**OK**] to confirm.
7. Mark your controller certificate and open the context menu by right-clicking.
8. Select **Export > File**.

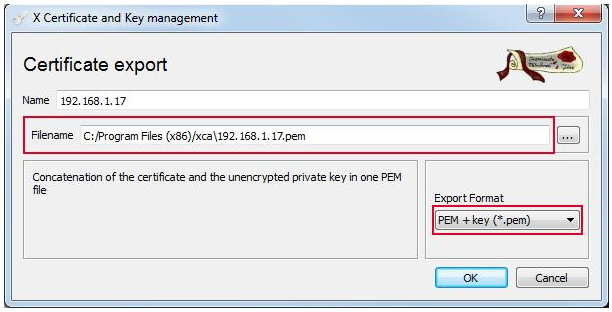


Figure 30: Controller-Zertifikat exportieren

5. Select a storage location with the [**...**] button.
10. In the **Export Format** selection list, select the entry “PEM with Key.”
11. Click [**OK**] to confirm.

#### 6.2.1.5.5 Install Certificates on the Client and Device

**Note**

**New device certificate necessary if IP address/host name changes!**
If the IP address or host name has changed, the certificate for the controller must be recreated with the corresponding IP address or host name (see the section “Create Device Certificate”).

1. Import your root CA certificate into the browser.
   The process depends on the browser used.
2. Transfer the certificates to the controller via FTPS/SCP.
3. Store the PFC certificate in “/etc/lighttpd/” and rename it “https-cert.pem.”
4. Transfer the root CA certificate to the controller via FTPS/SCP.
5. Store the root CA certificate in “/etc/lighttpd/” and rename it “root-ca.pem”.
6. Open the file “tls.conf” (/etc/lighttpd/tls.conf).
7. Add the line “ssl.ca-file” and enter the path: “/etc/lighttpd/root-ca.pem”.


Figure 31: Path “/etc/lighttpd/root-ca.pem”

8. Restart the Web server with the tool “/etc/config-tools/restart_webserver.”
   Alternatively, the device can be restarted.

As soon as a green lock symbol appears to the left or right (depending on the browser) of your Web address, the action has been successful, and your connection is secure from now on. The browsers often indicate how trusted a connection is in the address bar. For example, Firefox displays a green lock if the certificate is signed by a trusted root CA.


Figure 32: Green Lock in the Browser (Firefox)

#### 6.2.1.5.6 Create Certificate Revocation List

**Note**

**Certificate revocation list only with OpenVPN and Ipsec:**
On the controllers, certificate revocation lists are currently used exclusively with OpenVPN and IPsec connections.

A certificate revocation list (CRL) contains certificates that are locked, invalid or fake or have been revoked within the validity period. This makes sense for cases where a private key is lost or trust has to be withdrawn from a client. For example, when employees leave the company, their certificates should normally be declared temporarily invalid to prevent access to the company network.

An entry for a revoked certificate can be made temporarily. The certificate revocation list is stored on the server.

**Note**

**You can create an empty list initially:**
You can create an empty certificate revocation list initially so a VPN service does not have to restart in the event of an update. An empty list can also be updated during ongoing operation.

You can generate a certificate revocation list with the XCA key management software.

1. Open the XCA software and select the “Certificates” tab.
2. Mark the certificate you want to add to the revocation list.
3. Select the **Revoke** menu with the right mouse button.

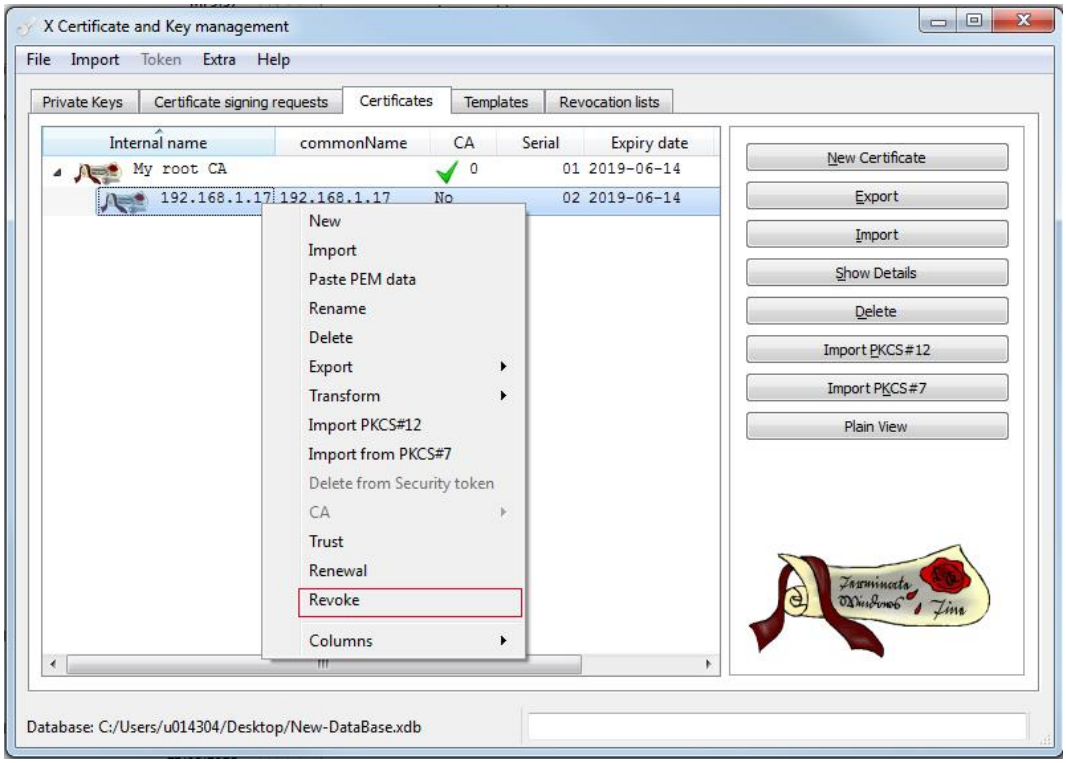


Figure 33: Create Certificate Revocation List

4. In the window that opens, you can indicate a reason for withdrawing trust and the date when the certificate becomes invalid.

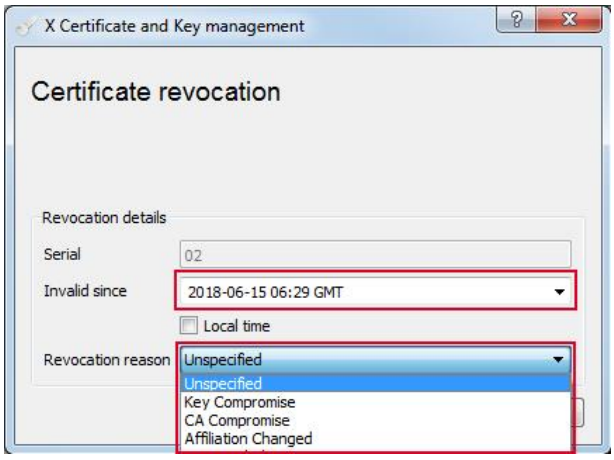


Figure 34: Certificate Revocation

5. Repeat steps 2 to 4 as needed.

6. Select the “Revocation lists” tab.

7. Select the **New** menu with the right mouse button.

In the window that opens, you do not need to make any further settings; XCA automatically adds the revoked certificates to the list. You can optionally set the update interval and adjust the signature algorithm.

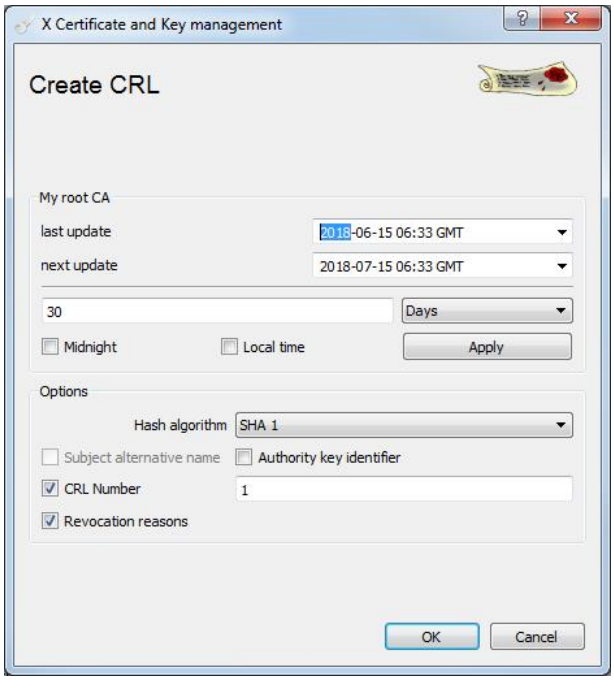


Figure 35: Create CRL

6. Click the [**OK**] button. The certificate revocation list is then shown on the “Revocation lists” tab.

7. Select the certificate revocation list and click the [**Export**] button.

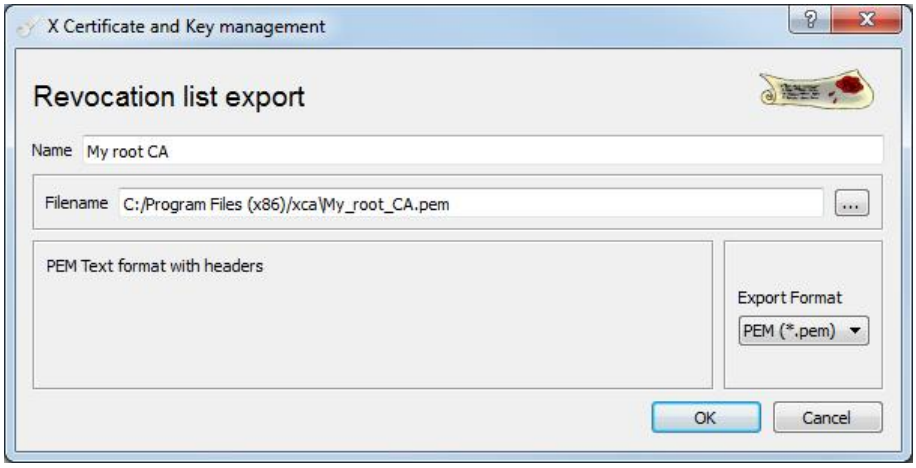


Figure 36: Export Revocation List

8. Save the certificate revocation list.

9. Click the [**OK**] button.

**Note**

**Note the directory structure:**
When you set up an OpenVPN network, the certificate revocation list must be stored in the directory specified for OpenVPN (see the section “OpenVPN” > “Create Configuration File,” “crl-verify” server configuration).

**Note**

**Note the storage location for certificate revocation lists for OpenVPN:**
With OpenVPN, the configuration file for OpenVPN must be created in advance, since the storage location is specified there; see the section “OpenVPN” > “Create Configuration File.”

**Note**

**The VPN service must be able to access the certificate revocation list:**
Ensure that the VPN service can access the certificate revocation list even if it does not have root permissions. To ensure this, either transfer file ownership to the OpenVPN user:

```
chown openvpn:openvpn <file>
```

or create the file in such a way that it is “world-readable”:

```
chmod 644 <file>
```

(The file contains no secret information.)

## 6.2.2 Restrict Access via Open Network Interfaces

All services/network interfaces that are not used or not required for the current application can present an unnecessary potential risk. Therefore, it is necessary to review in each individual case which services and network interfaces are required and which can be disabled. However, for productive systems, it is necessary to test in advance what effects disabling them has on the system.

The following sections describe how these services, which are active by default, can be disabled. Furthermore, access to a service can also be restricted to a particular interface. This method is recommended if a service is required and thus cannot be switched off. This restriction can further reduce the target for a cyber attack. You can find details in the section “Hardening” > “Configure Firewall.”

**Note**

**Always use secure protocols!**
Always use secure protocols, e.g. HTTPS instead of HTTP and SNMPv3 instead of SNMPv1 etc.

### 6.2.2.1 Disable WAGO Service Communication

If WAGO service communication is not used, or if the WAGO-I/*O-CHECK*, ETHERNET-Settings and ***e!****COCKPIT* development tools are not needed, they should be disabled.

1. In the WBM, select the menu item **Ports and Services > Network Services** to disable WAGO service communication.

2. Uncheck the **Service active** box in the “I/O Check” section.


Figure 37: Disable WAGO Service Communication

3. Click the [**Submit**] button to apply the changes.

### 6.2.2.2 Change Default Network Ports

The majority of automated login attacks against network services like SSH occur on the default network port 22. A simple and effective method to protect against this consists in changing the SSH port. The default network ports used for CODESYS and SSH can be changed via the Web-Based Management.

1. Navigate to the menu **Ports and Services** > **PLC Runtime Services**.

2. Preferably enter a port from the “Dynamic Port Numbers” section in the **Communication Port Number** input field in the “CODESYS” section.

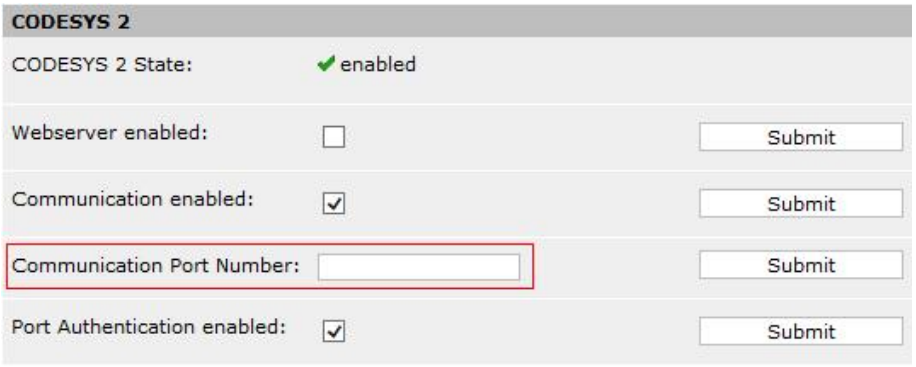


Figure 38: Change Default Network Ports

Each network port can only be used once on a system. Therefore, make sure that the port is not used by another application; otherwise, connection problems may occur. No reserved ports of other services should be used either.

**Note**

**Dynamic port numbers**
Use of dynamic port numbers is only a recommendation to prevent collisions with ports that are already in use.

The dynamic port numbers are also called private port numbers. These are port numbers that any application can use for communication with any other application via the TCP or UDP Internet protocol. More information: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

3. Click the [**Submit**] button to apply the change.

4. Repeat steps 2 and 3 in the **Ports and Services** > **SSH** menu in the “SSH Server” section to change the port there.

**Note**

**External firewalls must be configured correspondingly:**
If you use an external firewall, you may have to enable the ports used!

### 6.2.2.3 Block Unencrypted Access to the WBM

Port TCP/80 is used by default for the WBM and, if applicable, for Web visualization when the controller is delivered. During access via HTTP, the browser is automatically redirected to the encrypted connection. It is advisable to switch this port off via the WBM.

1. Open the WBM and navigate to **Ports and Services** > **Network Services**.

2. Uncheck the **Service active** box in the “http” section**.**


Figure 39: Block Unencrypted Access to the WBM

3. Click the [**Submit**] button. The adjusted settings are applied.

#### 6.2.2.4 Disable Access to the CODESYS Runtime Environment

**Note**

***e!****RUNTIME* **runtime environment cannot be disabled.**
Disabling access to the runtime environment is only possible for CODESYS, not for ***e!****RUNTIME*.

The controller has a CODESYS runtime environment through which it can be programmed.

The CODESYS program is normally downloaded via the ETHERNET interfaces. The controller's two ETHERNET interfaces can be configured for various functions. Once the programming or initial startup is completed, CODESYS access to the device can be disabled to prevent unwanted access.

1. In the WBM, select the menu time **Ports and Service** > **PLC Runtime Services** to disable CODESYS access.

2. In the "CODESYS" section, uncheck the box **Communication enabled**.


Figure 40: Disable Access to the CODESYS Runtime Environment

2. Click the [**Submit**] button to apply the changes.

#### 6.2.2.5 Block Direct Access to the CODESYS Web Visualization

The CODESYS Web visualization can be reached via a Web server through the following ports:

- Port TCP/80 und/oder TCP/443
- Port TCP/8080

It is advisable to block direct access via port TCP/8080 in the firewall settings, since only an unencrypted connection is possible here.

1. In the WBM, select the menu item **Firewall** > **General Configuration**.

2. In the section “Firewall Parameter Interface X1/X2 > Service enabled,” uncheck the box **PLC WebVisu – direct link (port 8080)**.

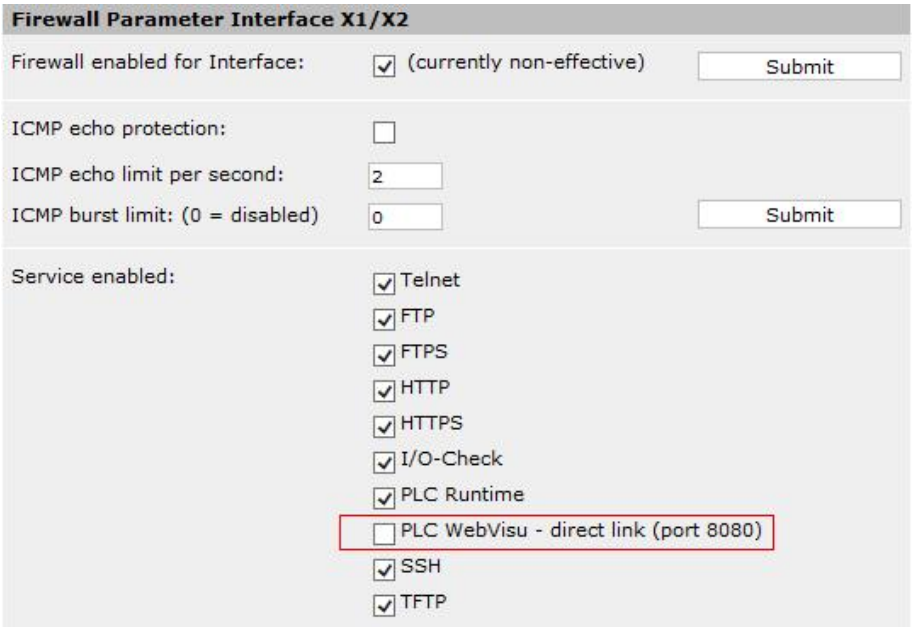


Figure 41: Block Direct Access to the CODESYS Web Visualization

2. Click the [**Submit**] button to apply the changes.

4. Repeat steps 2 and 3 in the menu **Firewall** > **General Configuration** > **Firewall Parameter Interface VPN** if you use a virtual private network.

**Note**

**X1 and X2 can run as separate interfaces.**
If you run the two ETHERNET interfaces X1 and X2 as separate interfaces, steps 1 through 3 above must be performed for both interfaces X1 and X2.

#### 6.2.2.6 Block Access to the *e!RUNTIME* Runtime Environment

Enable the firewall if it has not yet been enabled; see the section “Configure Firewall” > “Configure the Firewall in the Web-Based Management.”

1. In the WBM, select the menu item **Firewall** > **General Configuration**.

2. Uncheck the **PLC Runtime** box in the “Firewall Parameter Interface X1/X2” and **“**Firewall Parameter Interface VPN” sections.

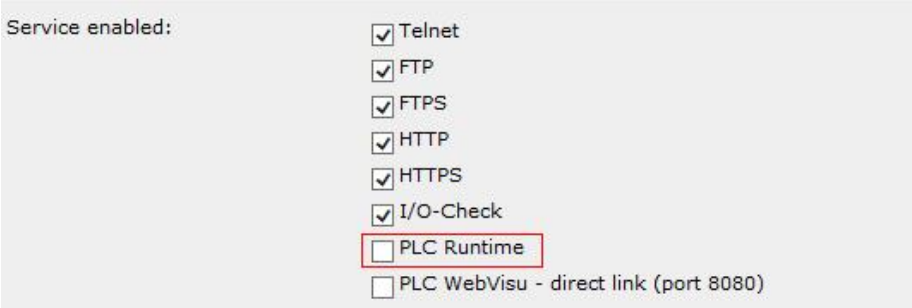


Figure 42: Block Access to the ***e!****RUNTIME* Runtime Environment

3. Click the [**Submit**] button to apply the changes.

This blocks access to either port 11740/TCP or port 2455/TCP, depending on which runtime environment is used. Alternatively, you can create a user filter for port 11740/TCP (e!Runtime) or port 2455/TCP (CODESYS). For details, see the section “Configure Firewall in Web-Based Management" > “Create Whitelist for Specific IP Addresses.”

**Close port 1740/UDP (*e!****Runtime*)**

1. Delete the following link in the file system:
/usr/lib/cds3-custom-components/libCmpBlkDrvUdp.so
2. Restart ***e!****RUNTIME*.

```
root@PFC200-40ED7D:~ rm /usr/lib/cds3-custom-components/libCmpBlkDrvUdp.so
root@PFC200-40ED7D:~ /etc/init.d/runtime stop
Terminate eRUNTIME...done
root@PFC200-40ED7D:~ /etc/init.d/runtime start
Starting eRUNTIME...done.
```

## 6.3 Change Passwords

**Note**

**Change default passwords:**
The default passwords are documented in these instructions and therefore do not offer adequate protection! Change the passwords to meet your particular needs; however, you need administrator permissions to do so.

### 6.3.1 Change Passwords in the Web-Based Management

To open the Web-Based Management, enter the IP address or hostname of your controller in the address bar of your browser. The required settings are available in the manual for the respective controller..

**Note**

**Login required:**
To change default parameters, you must first log in. Enter the following access information:
**Username:** admin
**Password:** wago

1. In the WBM, select the menu item Administration > Users to change a password.

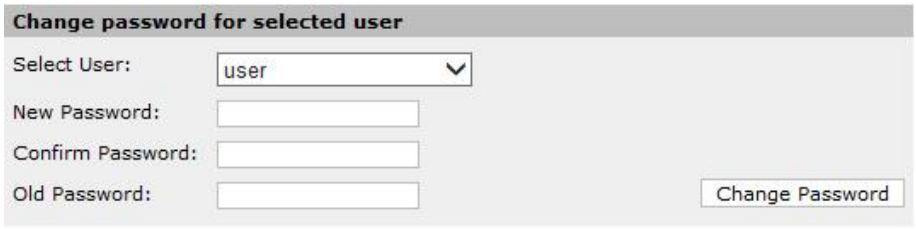


Figure 43: Change Passwords in the Web-Based Management

2. Select the user (“user” or “admin”) to whom you want to assign a new password.
3. Enter the new password in the **New Password** input field.
4. Confirm the new password in the **Confirm Password** input field.
5. Click the [**Change Password**] button to apply the change.

**Note**

**Note the permitted characters for passwords:**
The permitted characters for the password are the following ASCII characters: a … z, A … Z, 0 … 9, blank spaces and the following special characters:
]!"#$%&'()*+,./:;<=>?@[\^_`{|}~-

If WBM passwords with invalid characters are set outside the WBM system (e.g. via CBM), then accessing the WBM pages is no longer possible.

### 6.3.2 Change Linux® Passwords via the Linux® Console

A connection to the Linux® console can be established through various access channels:

- With the console, via the RS-232 interface

- Via SSH over ETHERNET

WAGO controllers are supplied with various usernames and default passwords:

- root/wago
- admin/admin
- user/user

When entering new passwords, please follow the recommendations for secure passwords in the section “Threat Scenarios” > ... > “Access via Users and Passwords.” Each user can change his or her own password; the “root” user can change the passwords for all other users.

1. Establish a connection to the Linux® console, either with the console via the RS-232 interface, or via SSH over the ETHERNET port.
2. Change the passwords with the Linux® “passwd” tool:

```
root@PFC200-40ED7D:~ passwd <Benutzer>
Changing password for <Benutzer>
New password: <Neues Passwort eingeben>
Retype password: <Neues Passwort wiederholen>
Password for <Benutzer> changed by root
```

**Note**

**For the current user, the username can be omitted.**
If the password only needs to be changed for the current user, it is not necessary to indicate the username.

Alternatively, you can change the password for the user “admin” via the WBM.

1. In the WBM, select the menu item **Ports and Services** > **PLC Runtime Services** > **General Configuration**.


Figure 44: Change the Password for the User “admin”

2. Enter your password in the **Port Authentication Password** input field.
3. Confirm your password in the **Confirm Password** input field.
4. Click the [**Submit**] button to apply the password change.

## 6.4 Configure Firewall

To protect your network against attacks from the outside, it is important to configure your firewall with a rule set that corresponds to your application.

**NOTICE**

**Provide emergency access to the system!**
Before configuring the firewall, always set up emergency access, e.g., a connection via a serial connection. This ensures that you cannot lock yourself out of your system by mistake.

The controller has a built-in host-based firewall based on the Linux® “iptables” program. The firewall works as a whitelist filter, i.e. packets that are not explicitly allowed by the whitelist are blocked by the firewall. The filters are processed in so-called chains; the order within the chains determines the processing order.

The controller supports creating your own filter rules with the following actions:

Table 9: Actions for Filter Rules

| Action | Description |
|---|---|
| ACCEPT | The packet is accepted. |
| DROP | The packet is not accepted; the sender is not notified. |

You can create your own filters in the WBM; see the section “Configure the Firewall in the Web-Based Management.”

No further rules are processed after an ACCEPT or DROP action.
If no rule applies to a packet, a pre-defined rule is executed for the packet. The controller uses a DROP action as the pre-defined rule for incoming network traffic, with the exception of pre-existing connections.
New filters created by the user take effect immediately and are processed before the pre-defined rules. You can find information on the pre-defined rules in the manual of the corresponding controller at www.wago.com. Packets that are accepted by a user filter (ACCEPT action) are forwarded directly to the corresponding service without passing through the pre-defined rules.

## 6.4.1 Configure the Firewall in the Web-Based Management

You can configure the firewall settings under the menu item **Firewall** > **General Configuration**.

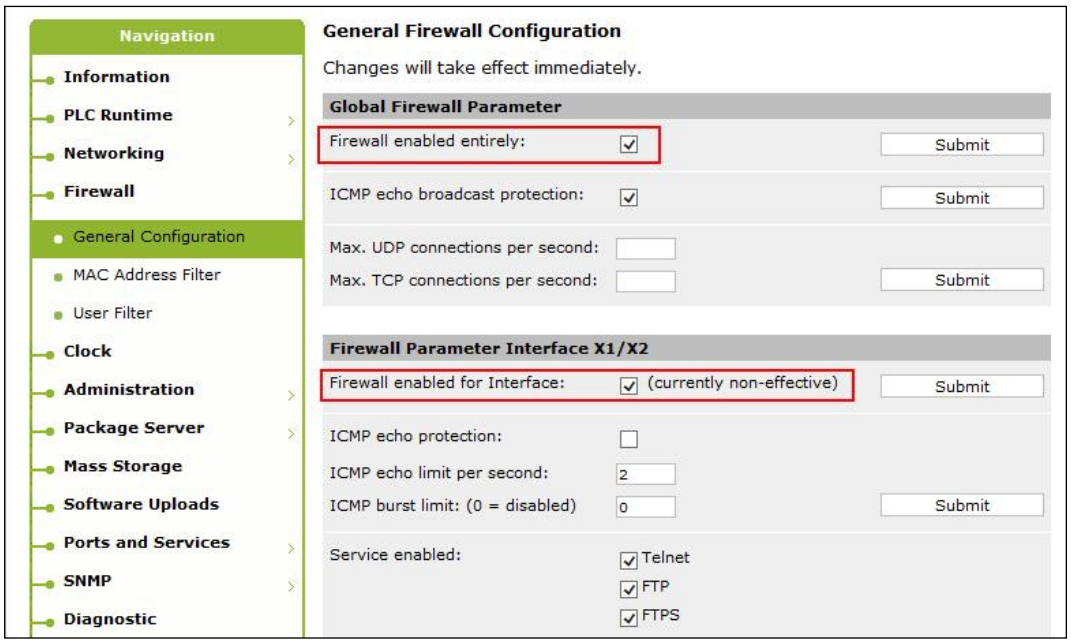


Figure 45: Firewall Configuration in the WBM

In order to enable the firewall, the following settings must be made:

1. Check the **Firewall enabled entirely** box (unchecked by default).
2. Check the **[Submit]** button. The adjusted settings are applied.
3. Click the box **Firewall enabled for Interface**.
4. Check the **[Submit]** button. The adjusted settings are applied.

**Note**

**Allow access only from trusted networks!**
The firewall can also be enabled for only one selected interface. Please pay attention to the network structure and security concept for your application. Allow access to your device only from trusted networks.

**Note**

**The firewall rules apply to both IPsec and OpenVPN:**
Configuration of the firewall rules for a VPN via the WBM (**Firewall Parameter Interface VPN**) in independent of the VPN technology used (IPsec or OpenVPN).
IPsec default rules for the modem interface (wwan) and standard rules for OpenVPN (tun+ and tap+) are defined upon delivery. You can display the pre-defined rules via the Linux® console with the “iptables-save” command.

#### 6.4.1.1 Create Whitelist for Specific IP Addresses

You can create a whitelist and use it to allow specified IP addresses to access services your system offers.

**NOTICE**

**Always create a DROP action AFTER an ACCEPT action!**
Always create an ACCEPT rule for your own IP address first; otherwise, you risk locking yourself out.

**Note**

**User-defined filter rules take precedence.**
Please note that the user-defined filter rules are executed before the rules pre-defined in the controller. After an ACCEPT action, the pre-defined rules are no longer applied. Therefore, you should always indicate a port under “Destination Port” to avoid accidentally granting full access to all ports on your system.

To create a whitelist, all IP addresses that are allowed to access the system’s service are first added to the whitelist. A filter that blocks all other access is then created with a DROP action.

In the following example, the IP address 192.168.147.1 is enabled for access to SSH (see steps 1 to 8). All other IP addresses are blocked for access (see steps 9 to 16).

**ACCEPT-Action**

1. Switch to the menu **Firewall** > **User Filter**.
2. In the “Add new user filter > Policy” area, check the **Accept** box.
3. In the **Source IP address** input field, enter the IP address for which you want to allow access.
4. In the **Source netmask** input field, enter the netmask “255.255.255.255” if you want to allow access **exclusively** to the IP address indicated.

5. In the **Destination port** input field, enter the port of the application to enable.

6. Check the **TCP** or **UDP** box, depending on which protocol you want to enable.

7. Check the **TCP** or **UDP** box, depending on which protocol you want to enable.

**Note**

**The X2 interface is not available in switch mode.**
The two ETHERNET interfaces X1 and X2 can be operated either in switch mode or as separate network interfaces. If you use both network interfaces, you have to copy the rule for the X2 interface.

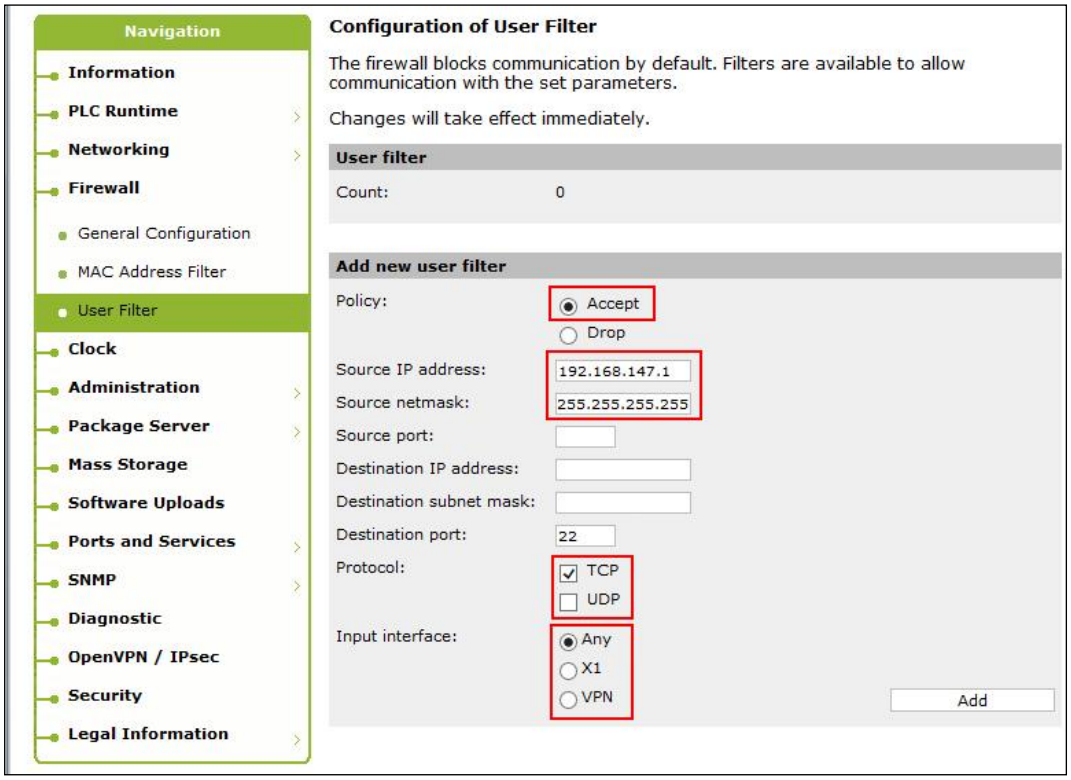


Figure 46: User Filter: Create Whitelist

8. Click the [**Add**] button. The rule is applied.

**Drop-Action**

9. Switch to the menu **Firewall > User Filter**.

10. In the “Add new user filter > Policy” section, check the **Drop** box.

11. Leave the **Source IP address** input field empty.

12. Leave the **Source netmask** input field empty.

13. In the **Destination port** input field, enter the port of the application to enable, e.g. “22” for SSH.

14. Check the **TCP** or **UDP** box, depending on which protocol you want to enable.

15. Check the **Any** box in order to apply the rule to each interface.

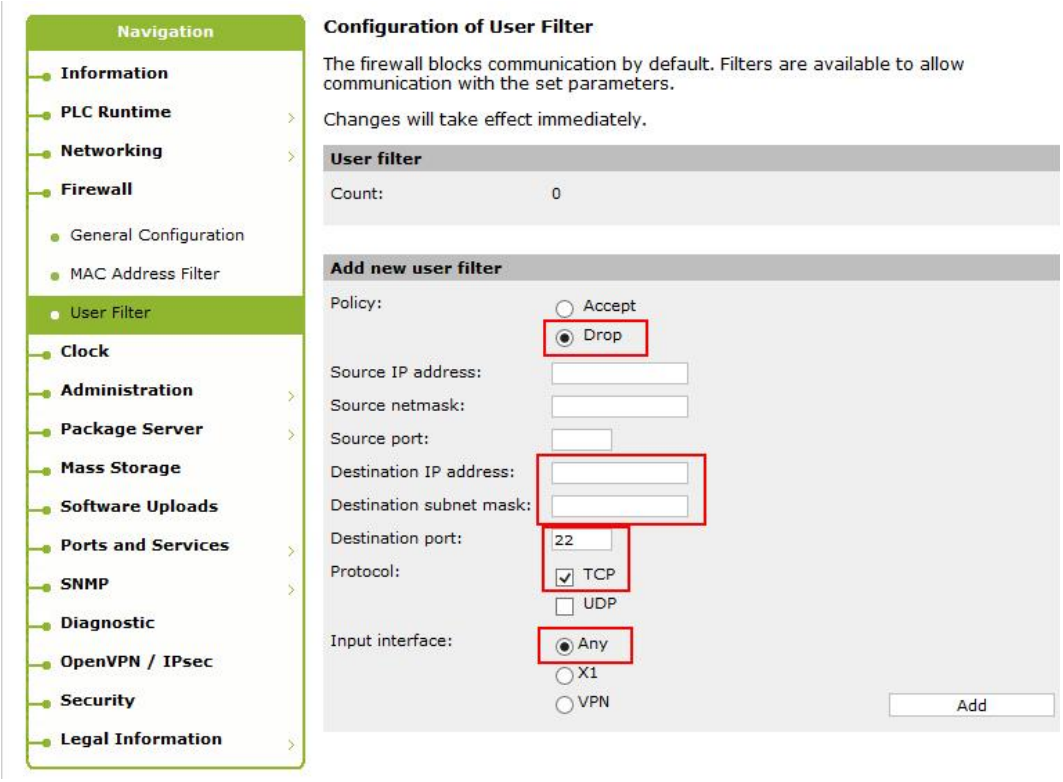


Figure 47: Create a Blacklist for All Access

16. Click the **[Add]** button. The rule is applied.

Finally, the filters are executed in the order shown:

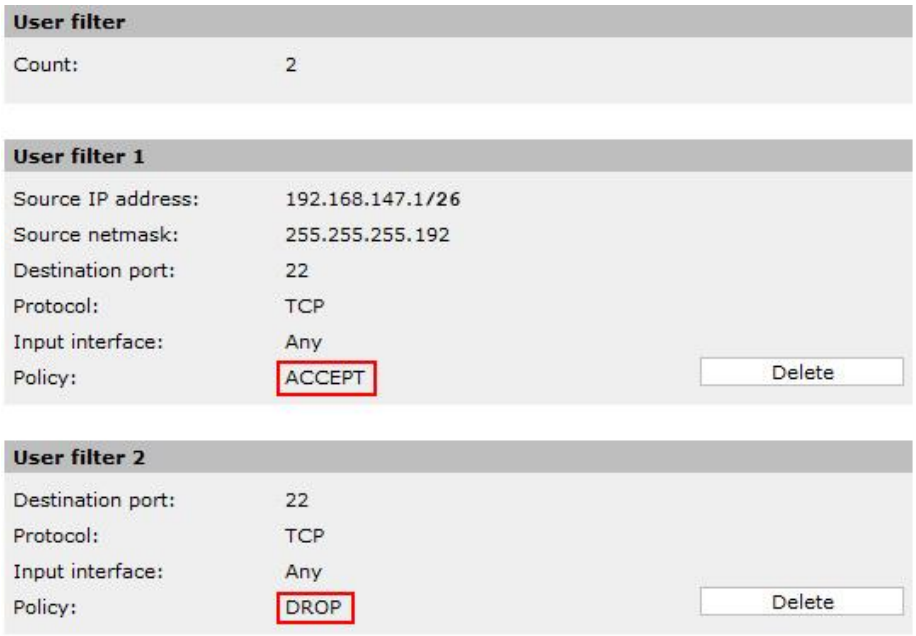


Figure 48: Order of the Filter Rules

### 6.4.1.2 Create Whitelist for Networks

If a whitelist needs to be created for one or more networks, they must first be enabled with an ACCEPT action. Access must then be blocked for all other networks with a DROP action, analogously to the procedure for enabling an individual IP address. This overrides the filter rules you enabled under “Firewall” > “General Configuration.” Therefore, specify the port for the service you want to enable.

**NOTICE**

**Always create a DROP action AFTER an ACCEPT action!**
Before you grant access to specific IP addresses or networks, you must always create the whitelist first; otherwise they may no longer be applied under some circumstances. This could cause access to the network to be blocked accidentally.

In the following example, the network 192.168.147.1/26 is enabled for access to SSH (see steps 1 to 8). In the examples presented, all access is blocked for other network elements (see steps 9 to 16).

**ACCEPT-Action**

1. Switch to the menu **Firewall > User Filter**.
2. In the “Add new user filter > Policy” area, check the **Accept** box.
3. In the **Source IP address** input field, enter the IP address for which you want to allow access.
4. In the **Source netmask** input field, enter the netmask “255.255.255.192.”
5. In the **Destination port** input field, enter the port of the application to enable.
6. Check the **TCP** or **UDP** box, depending on which protocol you want to enable.
7. Check the **Any, X1**, or **VPN** box in order to apply the rule to the respective interface.

**Note**

**The X2 interface is not available in switch mode.**
The two ETHERNET interfaces X1 and X2 can be operated either in switch mode or as separate network interfaces. If you use both network interfaces, you have to copy the rule for the X2 interface.

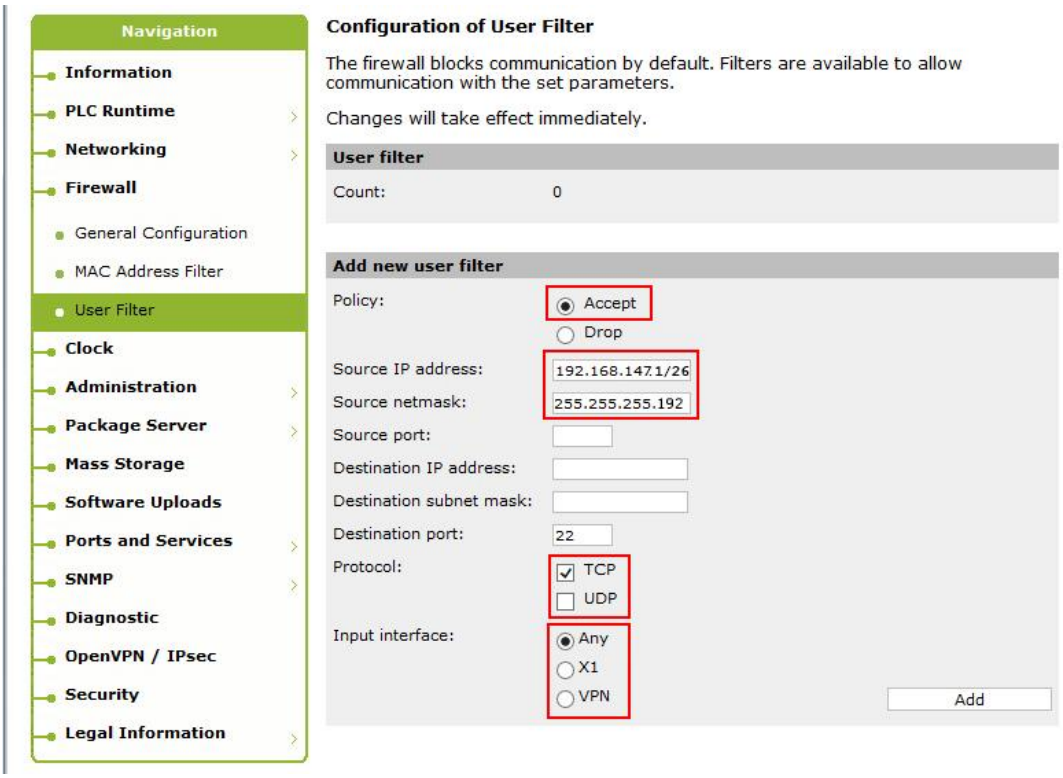


Figure 49: User Filter: Create Whitelist for Networks

8. Click the [**Add**] button. The rule is applied.

### Drop-Action

9. Switch to the menu **Firewall > User Filter**.

10. In the “Add new user filter > Policy” section, check the **Drop** box.

11. Leave the **Source IP address** input field empty.

12. Leave the **Source netmask** input field empty.

13. In the **Destination port** input field, enter the port of the application to enable, e.g. “22” for SSH.

14. Check the **TCP** or **UDP** box, depending on which protocol you want to enable.

15. Check the **Any** box in order to apply the rule to each interface.

16. Click the **[Add]** button. The rule is applied.

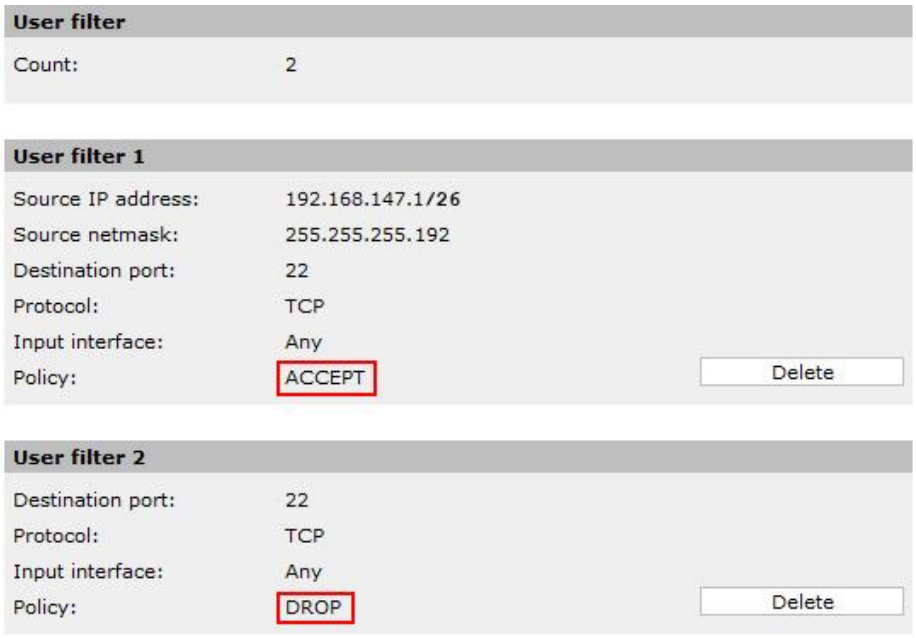


Figure 50: Enabling Specified Networks

In the example shown, the network 192.168.147.1/26 is enabled for access to SSH. This contains the IP addresses 192.168.147.1 to 192.168.147.62 and the broadcast address 192.168.147.1.63.

## 6.4.2 MAC Address Filter

MAC addresses of a device can be easily spoofed. Therefore, filtering MAC addresses is insufficient as the sole security measure. MAC addresses are used on the ETHERNET level and require physical access to the local network (Local Area Network – LAN). Therefore, access from outside, e.g. via a router, cannot be prevented. If the system is accessed via a router, the controller only sees the MAC address of the router. Therefore, it is advisable to combine MAC addresses with further protection mechanisms.

When the controller is delivered, a “whitelisting” is entered, but not active, for all WAGO MAC addresses, based on the manufacturer OUI (Organizationally Unique Identifier). This filter should be deleted and replaced by filters that suit the circumstances of your network.

1. You should always prefer MAC addresses of other trusted network elements to the manufacturer identifier. All MAC addresses of a manufacturer are enabled through the manufacturer identifier; in many cases, that is not desirable.

2. Before enabling the MAC address filter, enable the MAC address of your PC or of the access channel to the device (e.g. router, proxy, etc.) to avoid locking yourself out.

### 6.4.2.1 Configure MAC Addresses in the Web-Based Management

Under the menu item **Firewall** > **MAC Address Filter**, you can enable and configure MAC address filters:

The MAC addresses of all devices that are allowed to communicate with the device are entered first.

1. In the **MAC address** field in the **MAC address filter whitelist** section, enter the MAC address to enable.

2. In the **MAC mask** input field in the “MAC address filter whitelist” section, enter the value “ff:ff:ff:ff:ff:ff.”

**Note**

**MAC mask for the new list entry**
The entry in the MAC mask input field determines the bits that are checked when a specific MAC address is to be enabled.

3. Check the **Filter enabled** box.


Figure 51: Enter MAC Addresses

4. Click the **[Add]** button. The MAC address entered is enabled.

**Note**

**The number of list entries is limited:**
At most 10 filters for MAC addresses can be entered.

Once all MAC addresses have been entered, the MAC address filter must be enabled.

5. In the “Global MAC address filter state” section, check the **Filter enabled** box to enable the global MAC address filter.

**6.** Click the **[Submit]** button. The adjusted settings are applied.

7. In the “MAC address filter state X1/X2” section, check the **Filter enabled** box to enable the MAC address filter for the respective interface.

**8.** Click the **[Submit]** button. The adjusted settings are applied.

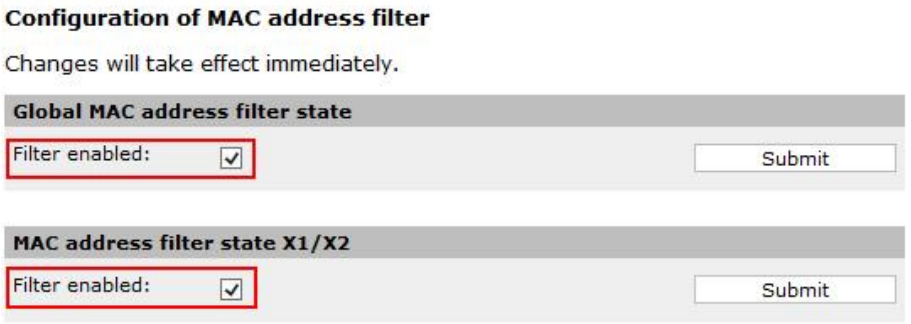


Figure 52: Enable MAC Address Filter

# 7 Extended Security Measures

This section describes additional security measures that can be implemented with the PFC100/200 and contribute to increasing the security of your system.

## 7.1 VPN – Virtual Private Network

### 7.1.1 General Information

The term “Virtual Private Network” stands for a series of technologies (e.g., IPsec) that can be used to create a private network within a publicly accessible network. Within this closed VPN, only the participants that are connected and authorized can communicate securely with each other.

Essentially, two methods are available for authentication: a certificate-based one and one using a pre-installed static key (pre-shared key).

- **Certificate-based:** In certificate-based authentication, the identity of one VPN endpoint is checked/confirmed with a digital certificate. A digital certificate is based on an individual key pair composed of a private and public key. If the digital certificate of one VPN endpoint is compromised, the certificate must be revoked and replaced on the device in question; see the section “Hardening” > ... > “Create Certificate Revocation List.”
- **Pre-shared key:** In the pre-shared key procedure, a shared static key is used for all VPN endpoints. If the shared key is compromised, the key must be replaced manually for all VPN endpoints.

**Note**

**Note the recommendations for the cryptographic method:**
Safeguarding the data packets through encryption in addition causes a delay and longer packet delivery times. Pay attention to the choice of the key lengths for the cryptographic procedure corresponding to the BSI TR-02102-4 technical guidelines (version 2017-01).

The following typical basic scenarios for setting up a VPN are described below:

### Site-to-Site-VPN


Figure 53: Site-to-Site-VPN

In a site-to-site VPN, two or more local networks are connected to each other via the Internet through a virtual logical network; see the figure “Site-to-Site VPN.” The secure communication takes place between the two gateways (e.g., VPN gateway or router).

### Host-to-Site-VPN

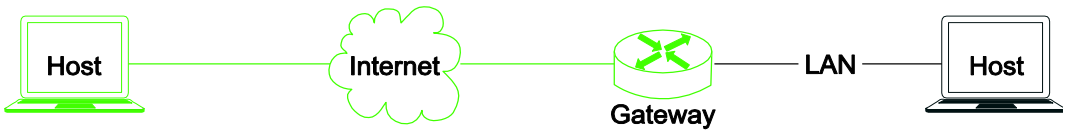


Figure 54: Host-to-Site-VPN

The host-to-site VPN allows specific users (e.g., in a home office or on a mobile device) to access a remote network securely; see the figure “Host-to-Site VPN.” The secure communication proceeds from the user’s end system (host) to the gateway of the remote network (e.g., VPN gateway or router).

### Host-to-Host-VPN


Figure 55: Host-to-Host-VPN bzw. Remote-Desktop-VPN

The host-to-host VPN allows a secure VPN connection between two end systems (end-to-end protection); see the figure “Host-to-Host VPN.” This secures the entire communication between the two end systems involved.

The following chapters describe the “site-to-site” and “host-to-host” scenarios for the controllers as examples.

### 7.1.2 Generate Certificates

For reciprocal certificate-based authentication, the VPN endpoints each need a certificate that has been signed by a trusted certificate authority (CA). When creating the certificates, pay attention to the notes at the end of this section.

You can find instructions for creating certificates and keys in the section “Hardening” > ... > “Generate and Replace Certificates.”

**Note**

**With TLS-based VPNs (OpenVPN), a different template is required for a client certificate than for a server certificate.**
When creating a request for a client certificate, make sure to select the entry **[default] HTTPS_client** in the **Template for the New Certificate** selection field on the **Origin** tab. See the section “Hardening” > ... > “Create Request for Device Certificate.” In the description, the **[default] HTTPS_client** template is selected.

**Note**

**Make sure the time is the same on all systems:**
If you use certificates, the time must be identical on all systems. Otherwise, problems can arise, for instance if certificates from one system are considered not yet valid or already expired.

### 7.1.3 Enable “IP Forwarding”

For a site-to-site VPN, the endpoints of the VPN must be configured as routers. For this purpose, the “IP Forwarding” function must be enabled in the controller. This setting is disabled by default so incoming and outgoing data packets are forwarded to and from a network behind it.
Enable “IP Forwarding” on the controller as follows:

1. In the WBM, select the menu item **Networking** > **Routing** in order to enable the “IP Forwarding” function.

2. Check the **Routing enabled entirely** box in the **“**General Routing Configuration” section.


Figure 56: Enable “IP Forwarding”

3. Click the [**Submit**] button to save the settings.

Alternatively, you can enable the “IP Forwarding” function by entering the following line in the “etc/sysctl.conf” file, or, if one already exists, modifying it:

```
net.ipv4.ip_forward = 1
```

## 7.1.4 OpenVPN

With the free OpenVPN software, a “Virtual Private Network” (VPN) can be set up via a TLS connection. Two operating modes are supported:

- **Routing mode**: This creates an encrypted tunnel in which only IP packets (OSI layer 3) are forwarded.
- **Bridging mode**: This allows complete tunneling of ETHERNET frames (OSI layer 2) so any network protocol can be used.

The following sections describe the configuration of an OpenVPN connection step by step.

### 7.1.4.1 Set up the User and Group for the OpenVPN Service

A dedicated user and a group must first be created for the OpenVPN service:

1. Log onto the controller via SSH.
2. Create a group “openvpn”:

```
addgroup -S openvpn
```

3. Create a user “openvpn” and add this user to the group “openvpn” that was already created:

```
adduser -G openvpn -S -D -H openvpn
```

**Note**

**Write the user that has been created to the OpenVPN configuration file.**
OpenVPN must be configured in such a way that, after initialization, the user switches to the unprivileged user created above; see the section “Host-to-Host VPN,” subsection “Minimum Rights.”

### 7.1.4.2 Configure Firewall

**Note**

**In the firewall, OpenVPN must be enabled on the server.**
You must create an exception rule on the server for the OpenVPN connection, since the firewall does not automatically enable the connection for the VPN.

1. In the WBM, select the menu item **Firewall** > **User Filter** to create an exception rule for OpenVPN.

2. Fill out the fields in the **“**Add new user filter” area as shown in the figure:

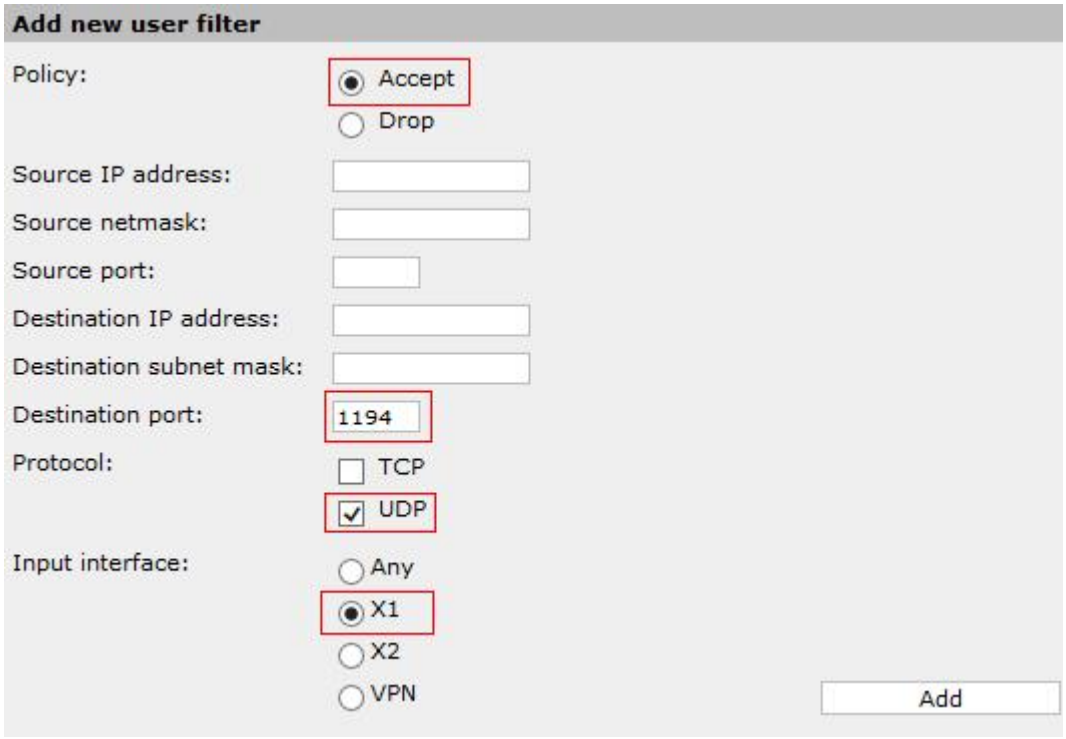


Figure 57: Firewall Configuration – OpenVPN

**Note**

**Make sure the port entry is correct:**
Make sure that the entry under “Destination port” is identical to your configured port.

3. Click the **[Add]** button to apply the filter.

Alternatively, you can configure the firewall via the Linux® console with the following command:

```
firewall iptables --add-filter on X1 udp - - - - - 1194 accept --apply
```

### 7.1.4.3 Configure Routing

Routing allows communication beyond network boundaries. If two hosts are located on different networks, the data must be forwarded through a router that connects the two networks to each other. For large networks like the Internet, several routers may be involved.

Routes specify the logical path within a network to the destination host. When routes are created, only the nearest host on the route to the destination host is indicated, not the entire route. If the data is transported via multiple routes, routes to the following router must be created on the routers in between as well. On large networks like the Internet, this is controlled automatically until the destination host is reached (e.g., via the “Border Gateway Protocol” – BGP).

Creating routes on the controller is described below. For other systems, please consult the manual for your device and/or operating system.

For the routing configuration described below, the following network topology is used as an example:

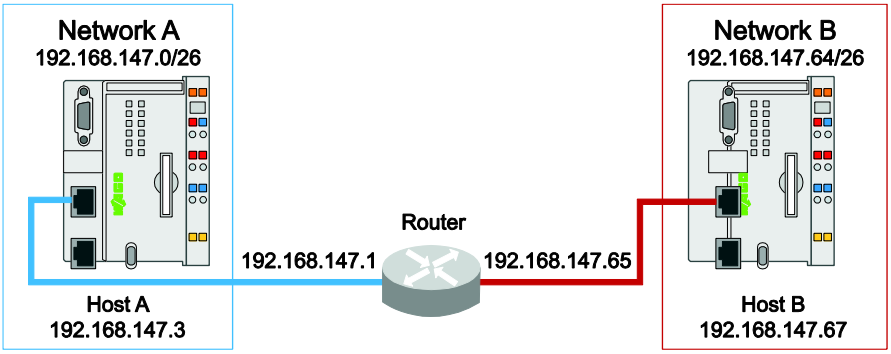


Figure 58:Network Topology, Routing

Host A needs to exchange data with host B. For this purpose, the data from host A must be routed through the router that forwards the data from network A to network B. To obtain the desired behavior, host A is first informed via a route that network B, in which host B is located, is accessible via the router (192.168.147.1).

You can open this route with config tools:

```
/etc/config-tools/config_routing -a static state=enabled dest=192.168.147.64 dest-
mask=255.255.255.192 gw=192.168.147.1 metric=20
```

Alternatively, you can add the route to a controller via the WBM:

1. In the WBM, select the menu item **Networking** > **Routing**.

2. Alternatively, check the **Routing enabled entirely** box in the “General Routing Configuration” section.


Figure 59: Routing Enabled

3. Check the **Enabled** box in the “Static Routes” section.

4. Enter the destination address of network B in the **Destination Address** input field.

5. Enter the destination mask of network B in the **Destination Mask** input field.

6. Enter the IP address of the router in the **Gateway Address** input field.

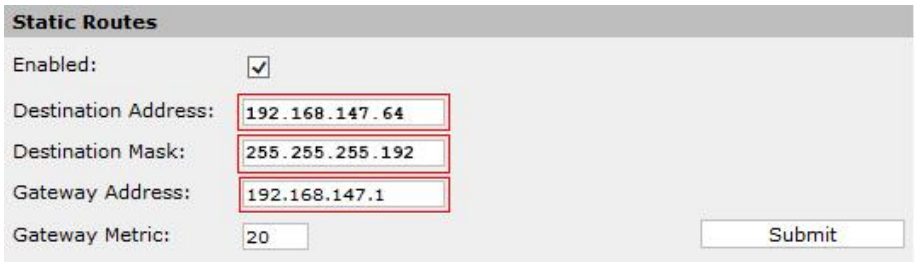


Figure 60: Static Routes

7. Save the setting with the [**Submit**] button.

8. Repeat the routing for host B.

**Note**

**Routes must be created in both directions:**
A route must also be created to host B to inform host B that network A is accessible via the router (192.168.147.65). If the route is only created in one direction, data from host A, for example, could arrive at host B, but host B could not send any data back.

#### 7.1.4.4 Create Configuration Files

Two configuration files, for a host-to-host and a site-to-site VPN, are described below. You can copy the example configurations one-to-one and apply them to your VPN configuration. Only your specific values have to be adapted.

**Prerequisites:**

- Certificates and keys have been generated; see the sections “Generate and Replace Certificates” and “Generate Diffie–Hellman Parameters.”
- The certificates are stored in the /etc/certificates/ folder. The private key is stored in the /etc/certificates/keys/ folder. You must also indicate both paths in the OpenVPN configuration file; see “Specification of the Storage Locations for Certificates and Keys” in the server configuration. You can find information on this in the section “Transfer the Configuration to the Controller.”

##### 7.1.4.4.1 Host-to-Host-VPN

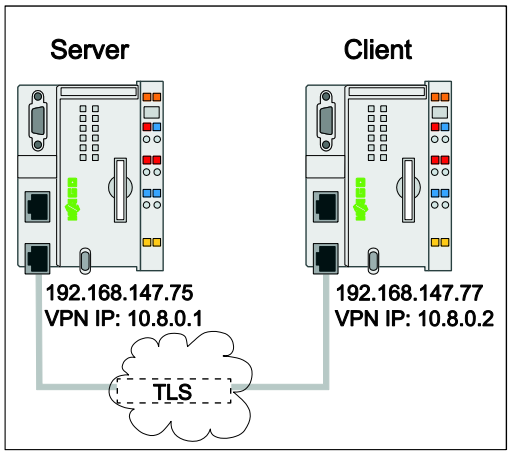


Figure 61: Host-to-Host Connection

The following example configuration assumes that the server is accessible at the IP address 192.168.147.75 and the client at the IP address 192.168.147.77. Change these values according to your specific circumstances.

Create the following configurations for the client and server for a host-to-host VPN.

### Server Configuration

```
###################
#    General    #
###################

# Multi-client Server Configuration
mode server

###################
#    Network    #
###################

# On what interface should the OpenVPN service be offered?
# Specify the IP address of the corresponding interface here, or leave # the entry
out if the service is to be offered on all interfaces.

local 192.168.147.75

# On what port should the OpenVPN service be offered?
# The default port is 1194, but you should avoid this in order to avoid automatic
# scans on the Internet.
port 1194

# Use UDP as transfer protocol
proto udp

# We use tunnel mode; only OSI layer 3 protocols are transferred.
dev tun

# What topology do we use within the VPN network?
topology subnet

# Clients are assigned an IP address from the following address pool.
# You are not allowed to use this address range in your network; # if you do,
problems arise and the VPN cannot be set up.

server 10.8.0.0 255.255.255.0

# OpenVPN regularly sends keep-alive packets; you can use the key “keepalive” # to
set the frequency and the number of seconds after which a client/server should #
be considered no longer accessible.
keepalive 10 120

#######################
#    Cryptography    #
#######################

# Specification of the storage locations for certificates and keys
# Keep the private key secret!
ca /etc/certificates/my_root-ca.crt
cert /etc/certificates/my_openvpn_server.crt
key /etc/certificates/keys/my_openvpn_server.key

# Storage location of the CRL
crl-verify /etc/certificates/my_root-ca_crl.pem

# Diffie-Hellman Parameters
dh /etc/certificates/dh2048.pem

# In addition to the certificate, the client needs a static key. This
# is a precaution against DoS attacks.
# You can generate the key with the following command:
#      openvpn --genkey --secret /etc/openvpn/static.key
tls-auth static.key 0

# Cipher that should be used by default – in this case, AES 256 # in CBC (Cipher
Block Chaining) mode
cipher AES-256-CBC
```

```
#########################
#    Minimum Rights     #
#########################

# The service should run with minimum rights; the group and the user # used for
execution are configured here. This user may need to be # created in advance. See
<Create User for OpenVPN>.
user openvpn
group openvpn

# Special mode so the connection can be reestablished after a restart # without
“root” permissions
persist-key
persist-tun

#################
#    Logging    #
#################

# File to which the current status is written at one-minute intervals
status /var/log/openvpn-status.log

# Verbosity of the server; for troubleshooting, a higher value can be set here #
so more messages are saved.
verb 4
```

### Client Configuration

```
###################
#    General    #
###################

# Client configuration
client

##################
#    Network    #
##################

# The same values must be given here as for the server – in this case,
# “tun,” since OpenVPN should be operated in tunnel mode.

dev tun

# Identical to the server configuration; alternatively, TCP is also possible.
proto udp

# The OpenVPN server and port are specified here. The port must be # identical to
the one configured on the server.
remote 192.168.147.75 1194

# The client constantly attempts to connect to the server. There is # no timeout
after which the connection attempts cease.
resolv-retry infinite

# No bind to a network interface is necessary.
nobind

#########################
#    Minimum Rights     #
#########################

# The user and group with which the OpenVPN client should run
user openvpn
group openvpn
```

```
# Special mode so that after a restart of the OpenVPN service, a connection can be
established even with minimum rights.
persist-key
persist-tun

######################
#    Cryptography    #
######################

# Specification of the storage locations for certificates and keys
# Keep the private key secret!
ca /etc/certificates/my_root-ca.crt
cert /etc/certificates/my_openvpn_client.crt
key /etc/certificates/keys/my_openvpn_client.key

# To prevent MITM attacks through client certificates, the client # is instructed
to check the purpose of use of the certificates. # Therefore, when creating
certificates, make sure to enter the correct use.
remote-cert-tls server

# In addition to the certificate, the client needs an additional key.
# This is identical on all systems and only serves to prevent DoS attacks.
tls-auth static.key 1

# Cipher used by default – in this case, AES 256 in CBC ("Cipher # Block
Chaining") mode. With the following command, you can view other # supported
ciphers:
#    openvpn --show-ciphers
cipher AES-256-CBC

#################
#    Logging    #
#################

# File to which the current status is written at one-minute intervals
status /var/log/openvpn-status.log

# Storage location for the log file; this is always newly created. If you # want
continuous saving, replace the key "log" # with "log-append."
log /var/log/openvpn.log

# Verbosity of the server; for troubleshooting, a higher value can be set here #
so more messages are saved.
verb 4
```

### 7.1.4.4.2 Site-to-Site VPN

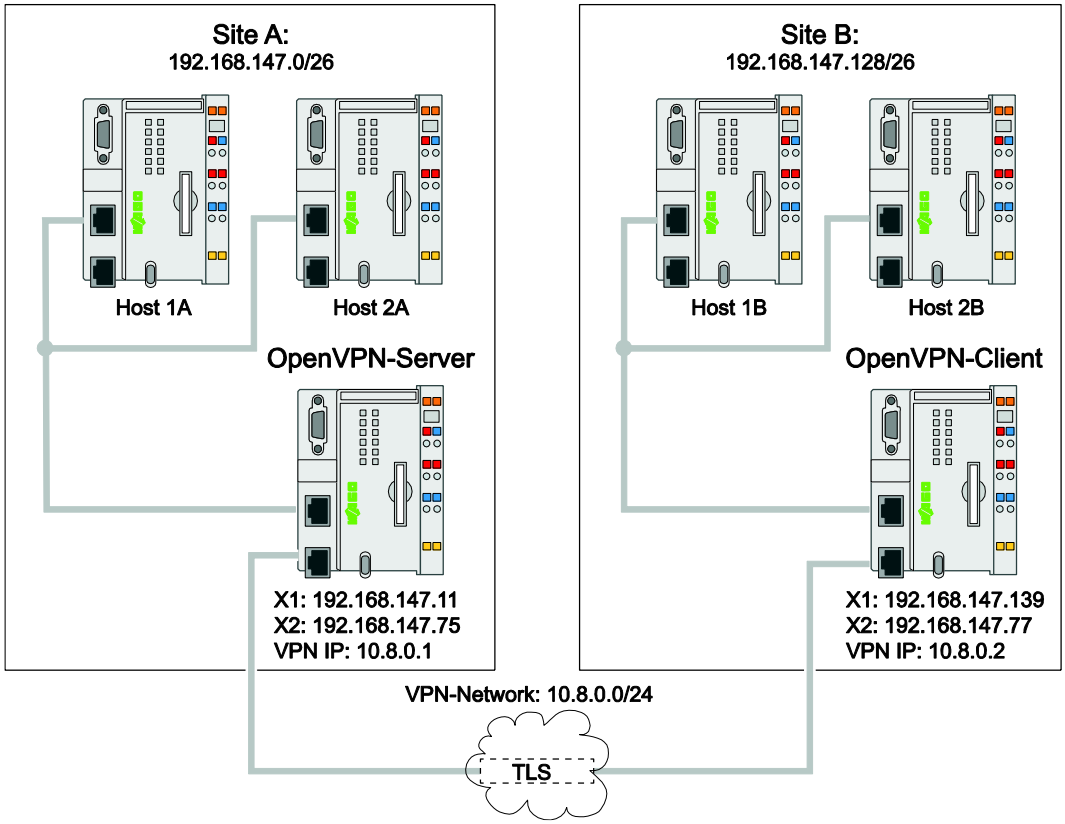


Figure 62: Site-to-Site VPN

Site-to-site means that two or more networks can be connected to each other and access a host behind the connection partner (OpenVPN server or client). For a site-to-site VPN, add the settings described below to the host-to-host configuration given above. Only the differences are listed here; the entire configuration is not repeated.

#### OpenVPN Server Configuration

1. Enable “IP Forwarding” on the OpenVPN server so you can access the network behind the OpenVPN server from the OpenVPN clients; see the section “Enable IP Forwarding.”

The OpenVPN server serves as a router for the network (site A) and forwards the network packets to the hosts.

2. Create a route for the VPN network on the hosts in the network behind the OpenVPN server (site A – see the section “Configure Routing”).

The VPN network 10.8.0.0/24 is accessible via the host 192.168.147.11.

3. Add the following section to the server configuration under the network settings to allow access to the network behind the OpenVPN server:

```
# Create a directory below the OpenVPN # configuration directory for the client
configuration(s)  client-config-dir ccd
```

4. Create a configuration file for each client below the directory created before (see “client-config-dir” server configuration).

The file must be named exactly like the “commonName” in the client certificate; see the section “Create Request for Device Certificate.”

**Note**

**The OpenVPN user or group must have access to the file.**
Make sure that the “openvpn” user and/or the “openvpn” group you created in the server configuration for a host-to-host VPN has read access to the file.

```
# Every OpenVPN client gets a fixed IP address from the IP address pool configured
previously # for the VPN ifconfig-push 10.8.0.2 255.255.255.0

# The OpenVPN client must be informed of the route for the network behind the
server.
push "route 192.168.147.0 255.255.255.192 10.8.0.1"
```

With this configuration, OpenVPN clients can access the network behind the OpenVPN server.

**Note**

**This setting is client-specific.**
Note that an individual configuration must be created for OpenVPN, analogous to the description above. In it, replace the IP address for the respective client.

**OpenVPN Client Configuration**

1. Enable “IP Forwarding” on the OpenVPN client so you can access the network behind the OpenVPN client; see the section “Enable IP Forwarding.”

For the hosts behind the OpenVPN server to be able to access the hosts behind the OpenVPN client, the OpenVPN server must be informed of the network behind the OpenVPN client.

2. Add a route by adding the following settings to the server-side client configuration:

```
# Inform the OpenVPN server of the network behind the client.
iroute 192.168.147.128 255.255.255.192
```

**Note**

**This setting is client-specific.**
The route is specific to each host and the corresponding network behind the OpenVPN client. You must adjust the route separately for each host.

3. Create a route on the hosts in the network behind the OpenVPN client (site B – see the section “Configure Routing”).

With the help of this route, network packets from network A (192.168.147.0/26) should be forwarded via the OpenVPN client (192.168.147.139) to hosts from network B.

4. In addition, create a route on the hosts in the network behind the OpenVPN server (site A – see the section “Configure Routing”).

This route should allow communication with the hosts from network B (192.168.147.128/26) via the OpenVPN server (192.168.147.11).

**Multi-Client Configuration**

If multiple clients are connected to the OpenVPN servers, or multiple network (site C, D etc.) are connected to each other, the clients must be able to communicate among themselves. In this case, you must add the settings described below to your OpenVPN configuration.

1. Add the following console command to your OpenVPN configuration:

```
# Allow client-to-client communication.
client-to-client
```

In this configuration, the clients can communicate among themselves. Likewise, for the hosts from the connected networks to be able to communicate with each other, you must create corresponding routes.

2. Add a route to the server-side client configuration:

```
# Inform the OpenVPN server of the network.
iroute 192.168.147.128 255.255.255.192
```

3. Create routes for the network you want to access on the hosts in the network behind both the OpenVPN client and OpenVPN servers (see the section “Configure Routing”).

With these routes, the hosts from the network can send the data packets via the OpenVPN client/server.

### 7.1.4.5 Transfer the Configuration to the Controller

You can configure the controller if you have:

- Set up the OpenVPN service on the controller
- Generated certificates and Diffie–Hellman keys
- Created OpenVPN configuration files

1. Open the WBM and log in as administrator (“admin”).
2. Navigate to the **OpenVPN / IPsec** menu.
3. In the “OpenVPN” section, select the configuration files you want to transfer to the device.


Figure 63: WBM, Select Configuration File

**Note**

**Naming convention and storage location of the configuration file:**
The file does not need to be named openvpn.conf, but it is renamed “openvpn.conf” and stored in the /etc/openvpn/ folder after the transfer.

4. In the **New Certificate** input field in the “Certificate Upload” section, select the corresponding certificates (root CA certificate and certificate for client or server). For the server, you can also load the Diffie–Hellman parameters that the server provides.
5. In the **New Private Key** input field in the “Certificate Upload” section, select your private key.

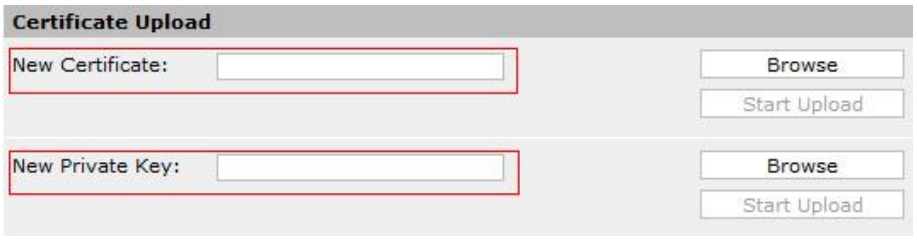


Figure 64: WBM, Select Certificates

**Note**

**Storage locations of the certificates and keys:**
The certificates are stored in the /etc/certificates/ folder after transfer. The private key is stored in the /etc/certificates/keys/ folder. You must indicate both paths in the OpenVPN configuration.

6. In the “OpenVPN” section, check the **OpenVPN enabled** box so that the OpenVPN service will be available after a restart


Figure 65: Enable OpenVPN Service

6. Start the OpenVPN service with config-tools:

```
/etc/config-tools/vpncfg ovpn --start
```

Alternatively, you can restart the device.

## 7.1.5 IPsec

IPsec is an extension of the IP protocol that has been supplemented with security objectives, confidentiality, authentication and integrity. This makes it possible to protect IP packets cryptographically, which implements secure communication over insecure networks. The packets are protected on layer 3 (see OSI model, network layer). IPsec distinguishes the following transfer modes:

- **Tunnel mode:** In tunnel mode, the entire IP packet (including the IP header) is encapsulated and given an additional new IP header. The advantage of this mode over transport mode is that it hides the source/target address. Thus the identity of the actual partners remains hidden. Tunnel mode can be use in the following basic scenarios: “host-to-host,” “host-to-site” and “site-to-site.”
- **Transport mode:** In transport mode, no new IP header is added, so the necessary information for the transfer of the network packets from the original IP header can be used. In this transfer mode, it is not possible to couple different networks to each other. This mode can only be used for the “host-to-host” scenario.

### 7.1.5.1 Security Protocols

IPsec provides the two safety protocols Authentication Header (AH) and Encapsulating Security Payload (ESP):

- **Authentication Header (AH):** he “Authentication Header” (AH) ensures the integrity and authenticity of the transferred data. AH offers no protection of confidentiality; all data is transferred as unencrypted text.
- **Encapsulating Security Payload (ESP):** “Encapsulating Security Payload” (ESP) ensures integrity and authenticity, analogously to the AH security protocol. Unlike AH, it also guarantees confidentiality through encryption of the transferred data.

The ESP and AH security protocols can be used separately or together, depending on the security requirements. The encryption and authentication methods can be configured accordingly; see the section “Extended Security Measures” > ... > “Create Configuration Files.”

#### 7.1.5.2 Internet Key Exchange Protocol (IKE)

The IKE protocol is responsible for the exchange of connection parameters that are needed in order to establish a secure communication channel between the IPsec endpoints. The following parameters are exchanged, among other things:

- Type of secure transfer
- Encryption algorithm
- Cryptographic keys
- Duration of validity of the cryptographic keys

The test scenarios in this documentation consider IKEv2 exclusively (see the sections “Host-to-Host VPN” and “Site-to-Site VPN”).

#### 7.1.5.3 Security Policy Database (SPD)

The “Security Policy Database” defines rule sets (security policies) that specify how incoming and outgoing data packets are handled. Three basic functions are distinguished:

- Packet is rejected immediately (DISCARD).
- Packet is forwarded without change (BYPASS).
- Packet is processed by IPsec (PROTECT).

The data packets are handled through specific selection criteria (selectors), which are listed in the “ipsec.conf” configuration file. Examples include:

- Source or destination IP address
- Transport layer protocol: TCP/UDP
- Identity name of the certificate

#### 7.1.5.4 Security Association (SA) and Security Parameter Index (SPI)

The necessary information/parameters must be provided so the IPsec endpoints can process the cryptographically protected network packets according to the security policy (decryption/integrity check). The necessary information is provided to the respective IPsec endpoints in a database in order to keep the additional data overhead per network packet as small as possible.

Unique association with the cryptographic parameters and algorithms occurs within the database through

- The “Security Parameter Index” (SPI), which is transferred in addition for each IPsec packet,
- The transferring IP destination address and
- The security protocol used (ESP/AH).

This association is referred to as the “Security Association” (SA). It governs the communication between the IPsec endpoints.

Since the IPsec endpoints can both send and receive, one SA per communication direction is needed for each IPsec endpoint. The negotiated SAs are managed in the “Security Association Database” (SAD), which lists all SAs. The SAs are negotiated through the Internet “Key Exchange Protocol” (IKE).

The following authentication methods, among others, are provided for checking the authenticity of the IPsec endpoints during the setup of an SA:

- PSK – Pre Shared Keys
- X.509 certificates

For the identification of the VPN remote stations with a certificate, additional information in the form of an “identifier” is necessary. The identifier can take the form of an IP address, a DNS name (FQDN) or an email address (FQUN).

**Note**

**With strongSwan, indicate the identifier in the “Subject Alternative Name” and/or the Common Name (CN).**
It is advisable to indicate the identifier (e.g., DNS name or IP address) in the “Subject Alternative Name” field of the certificate. For more information on certificates in strongSwan, see https://wiki.strongswan.org/projects/strongswan/wiki/SimpleCA

In order to establish a successful connection via IPsec, the participants must have the following information:

- IP address of the remote station
- Subnet mask of the network
- Tunnel name
- Authentication method
- Encryption and authentication method used
- Keys of the cryptographic method

### 7.1.5.5 Create Configuration Files

Two configuration files, for a host-to-host and a site-to-site connection, are described below. You can copy the example configurations one-to-one and apply them to your VPN configuration. Only your specific values have to be adapted. It is possible to specify the ESP/AH method explicitly within the configuration files.

**Note**

**Cipher suites supported with the strongSwan IPsec application:**
You can find an overview of the cipher suites supported with the strongSwan IPsec application at the following link:
https://wiki.strongswan.org/projects/strongswan/wiki/IKEv2CipherSuites.

#### 7.1.5.5.1 Host-to-Host VPN

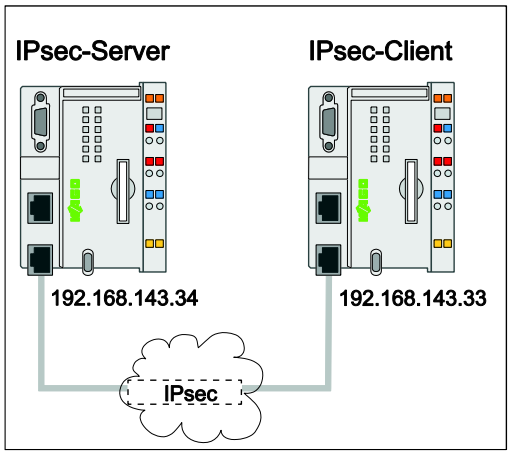


Figure 66: Host-to-Host Connection, IPsec

The following example configuration assumes that the IPsec server is accessible at the IP address 192.168.143.34 and the IPsec client at the IP address 192.168.143.33. Change these values according to your specific circumstances.

In addition to the “ipsec.conf” configuration file, a file “ipsec.secrets” must also be created that specifies the authentication secret. You can find both files in the following sample configurations. The private key is specified for certificate-based authentication. For the PSK procedure, the “shared secret” key is specified.

### Transport Mode with X.509 Certificates

#### Configuration for IPsec Client

```
# /etc/ipsec.conf - strongSwan IPsec configuration file

config setup

conn %default
   ikelifetime=60m
   keylife=20m
   rekeymargin=3m
   keyingtries=1
   mobike=no
   keyexchange=ikev2
   ike=aes128-sha256-modp2048
   esp=aes128-sha256-modp2048

conn host-host
   left=192.168.143.33
   leftcert=PFC1Cert.pem
   leftid=@PFC1.wago.org
   leftfirewall=yes
   right=192.168.143.34
   rightid=@PFC2.wago.org
   type=transport
   auto=start

# /etc/ipsec.secrets - strongSwan IPsec secrets file
: RSA PFC1Cert.pem
```

#### Konfiguration für IPsec-Server

```
# /etc/ipsec.conf - strongSwan IPsec configuration file

config setup

conn %default
   ikelifetime=60m
   keylife=20m
   rekeymargin=3m
   keyingtries=1
   mobike=no
   keyexchange=ikev2
   ike=aes128-sha256-modp2048
   esp=aes128-sha256-modp2048

conn host-host
   left=192.168.143.34
   leftcert=PFC2Cert.pem
   leftid=@PFC2.wago.org
   leftfirewall=yes
   right=192.168.143.33
   rightid=@PFC1.wago.org
   type=transport
   auto=start

# /etc/ipsec.secrets - strongSwan IPsec secrets file
: RSA PFC2Cert.pem
```

**Note**

**Note the information on the configuration parameters.**
The description of the individual configuration parameters can be viewed on the strongSwan homepage:
https://wiki.strongswan.org/projects/strongswan/wiki/IpsecConf

**Tunnel Mode with X.509 Certificates**

**Note**

**Only minimal changes are necessary for tunnel mode.**
For tunnel mode, you must assign the value “tunnel” to the parameter “type” in the “ipsec.conf” configuration file. Alternatively, the parameter “type” can be removed, since the default mode corresponds to the value “tunnel.”

### 7.1.5.5.2 Site-to-Site VPN

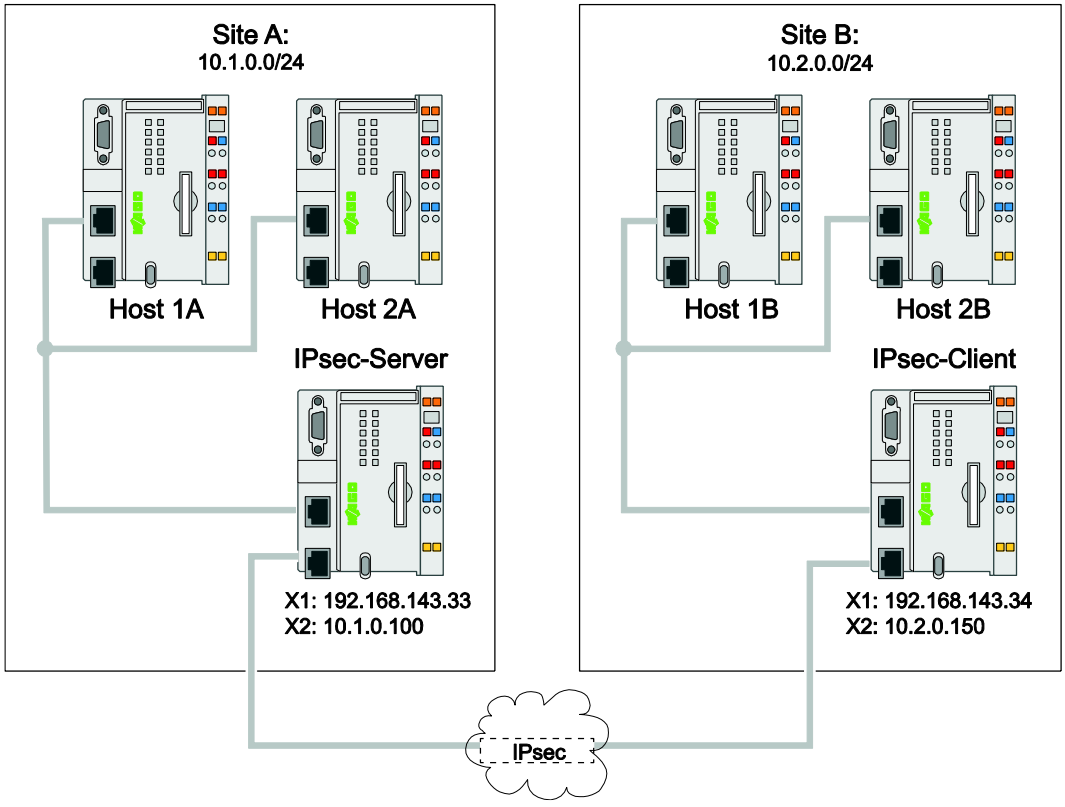


Figure 67: Site-to-Site VPN, IPsec

Site-to-site VPN means that systems behind the connection partner (IPsec server or IPsec client) can also be accessed. For a site-to-site VPN, you can use the following sample configuration and adapt your specific values. The “ipsec.secret” file can be taken over from the previous example.

### Configuration for IPsec Client

```
# /etc/ipsec.conf - strongSwan IPsec configuration file

config setup

conn %default
   ikelifetime=60m
   keylife=20m
   rekeymargin=3m
   keyingtries=1
   keyexchange=ikev2
   mobike=no
   ike=aes128-sha256-modp2048
   esp=aes128-sha256-modp2048
conn net-net
   left=192.168.143.33
   leftcert=PFC1Cert.pem
   leftid=@PFC1.wago.org
   leftsubnet=10.1.0.0/24
   leftfirewall=yes
   right=192.168.143.34
   rightid=@PFC2.wago.org
   rightsubnet=10.2.0.0/24
   auto=start
```

### Configuration for IPsec Server

```
  # /etc/ipsec.conf - strongSwan IPsec configuration file

config setup

conn %default
   ikelifetime=60m
   keylife=20m
   rekeymargin=3m
   keyingtries=1
   keyexchange=ikev2
   mobike=no
   ike=aes128-sha256-modp2048
   esp=aes128-sha256-modp2048

conn net-net
   left=192.168.143.34
   leftcert=PFC2Cert.pem
   leftid=@PFC2.wago.org
   leftsubnet=10.2.0.0/24
   leftfirewall=yes
   right=192.168.143.33
   rightid=@PFC2.wago.org
   rightsubnet=10.1.0.0/24
   auto=add
```

**Note**

**Note the information on the configuration parameters.**
The description of the individual configuration parameters can be viewed on the strongSwan homepage:
https://wiki.strongswan.org/projects/strongswan/wiki/IpsecConf

**Access of the Clients (Site A) to a Network behind the IPsec Client (Site B)**

1. Enable “IP Forwarding” on the IPsec server and IPsec client to allow access to the network located behind it. See the section “Extended Security Measures” > ... > “Enable IP Forwarding.”

In addition, a route must be crated on the systems in the network behind the IPsec server so the packets for the network clients (site A) can be forwarded to the IPsec server. The IPsec server then forwards the packets to the corresponding IPsec client.

2. In the WBM, select the menu item **Networking** > **Routing**.

3. Enter the IP address of your server in the **Gateway Address** input field in the “Static Routes” section.

4. Save the setting with the [**Submit**] button.

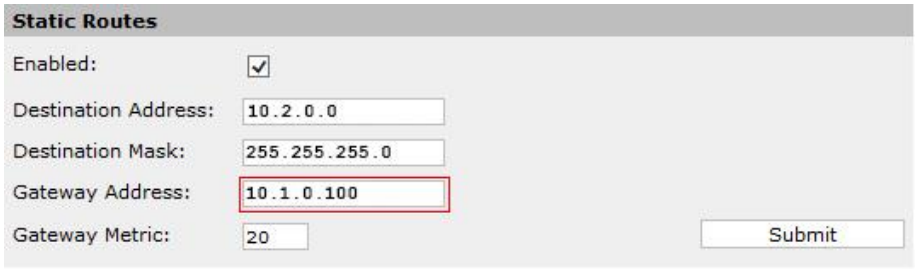


Figure 68: Static Routes, Access of the Clients to a Network behind the IPsec Client

### Access of the Clients (Site B) to a Network behind the IPsec server (Site A)

A route must be created on the systems in the network behind the IPsec client so the packets can be forwarded to the upstream IPsec client. The IPsec client then forwards the packets to the IPsec server as a router.

You can add the route via the WBM:

5. In the WBM, select the menu item **Networking** > **Routing**.

6. Enter the IP address of your server in the **Gateway Address** input field in the “Static Routes” section.

7. Save the setting with the [**Submit**] button.

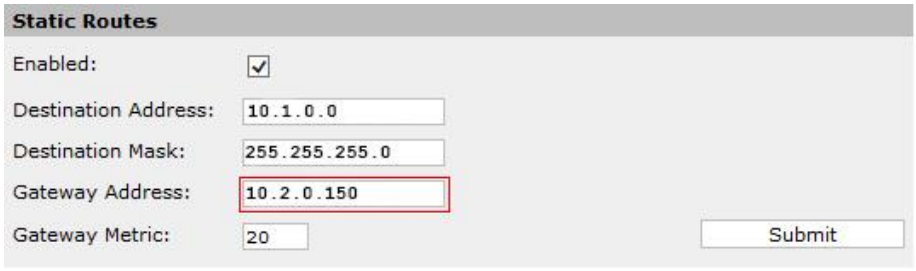


Figure 69: Static Routes, Access of the Clients to a Network behind the IPsec Server

### 7.1.5.6 Configure Firewall

**Note**

**In the firewall, IPsec must be enabled explicitly for X1 and X2.**
You must create exception rules for both IPsec remote stations for the IPsec connection, since the firewall does not automatically enable the IPsec VPN through the X1/X2 interface. Only pre-defined IPsec exception rules for the modem interface (wwan) exist.

The IPsec exception rules for the X1 interface are added through the following steps (and analogously for the X2 interface as well):

1. Connect to the Linux® console of the controller via SSH or the serial interface.

2. Edit the “params.xml” file under the path /etc/firewall/, e.g. with the Linux® application “nano”:

```
nano /etc/firewall/params.xml
```

In the following figure, you see the modified “params.xml” file in the form necessary for establishing an IPsec connection via the X1 interface:

```
<?xml version="1.0" encoding="utf-8"?>
<firewall xmlns="http://www.wago.com/security/firewall"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.wago.com/security/firewall params.xsd">

    <parameters>
        <interfaces>
            <!-- In case any names ('name' and 'rname' tags) should be changed
                 please amend validate_if.sh script accordingly! -->
            <interface name="X1"  rname="br0"   ethernet="yes" ipsec_srv="yes"/>
            <interface name="X2"  rname="br1"   ethernet="yes"/>
            <interface name="WAN" rname="wwan0" ethernet="yes" ipsec_srv="yes"/>
            <interface name="VPN" rname="wwan0" ethernet="no"  ipsec="yes"/>
            <interface name="VPN" rname="tun+"  ethernet="no" />
            <interface name="VPN" rname="tap+"  ethernet="yes"/>
            <interface name="VPN"  rname="br0"   ethernet="no" ipsec="yes"/>
        </interfaces>
    </parameters>

</firewall>
```

#### Description of the “ipsec_srv="yes"” Parameter

With the additional “ipsec_srv="yes"” parameter, the following persistent exception rules are automatically added for the X1 interface (br0) when the firewall restarts:

- Enabling ports 500/UDP (IKE) and 4500/UDP (NAT)
- Enabling the IPsec security protocol ESP

You can display these rules with the Linux® command “iptables-save.”

```
…
-A in_generic -i br0 -p udp -m udp --dport 500 -j ACCEPT
-A in_generic -i br0 -p udp -m udp --dport 4500 -j ACCEPT
-A in_generic -i br0 -p esp -j ACCEPT
…
```

**Description of the XML Tag “<interface name="VPN" rname="br0” ethernet="no" ipsec="yes"/>”**

After the firewall restarts, the additional XML tag “<interface name="VPN" rname="br0" ethernet="no" ipsec="yes"/>” adds exception rules for services that can be reached through the IPsec tunnel. This relates exclusively to the services accessible via the X1 interface (br0) within the IPsec tunnel and not to the services accessible outside the tunnel. These services include:

- FTP/FTPS
- SSH
- HTTPS

You can display these exception rules with the Linux® command “iptables-save.” For example, for the HTTPS service, they have the following form:

```
-A in_https -i br0 -p tcp -m policy --dir in --pol ipsec --proto esp --mode tunnel
-m tcp --dport 443 -j ACCEPT
```

In addition, please note the information on the firewall configuration for the controller; see the section “Hardening” > “Configure Firewall.”

**Note**

**Note the firewall settings for the VPN “site-to-site” scenario:**
In the VPN “site-to-site” scenario when the firewall is enabled, all active services of the downstream client systems can be accessed via the enabled port, 500 or 4500, as long as the tunnel could be successfully.

Take additional measures to prevent access to sensitive client systems. This can be done with additional firewall configurations (see the section “Hardening” > ... > “Create Whitelist for Networks”) or by separating the network, for example. You can find more information on this in the white paper “IT Security in Production Facilities,” which you can request from the download area at https://www.wago.com.

### 7.1.5.7 Transfer the Configuration to the Controller

Configuration of the controller takes place if you have:

- Set up the IPsec service on the controller
- Generated certificates
- Created IPsec configuration files (ipsec.conf and ipsec.secret)

1. Open the WBM and log in as administrator (“admin”).
2. Navigate to the **OpenVPN IPsec** menu.
3. In the “IPsec” section, select the configuration files you want to transfer to the device.


Figure 70: WBM, Select Configuration Files for IPsec

2. In the **New Certificate** input field in the “Certificate Upload” section, select the corresponding certificates (root CA certificate and certificate for client or server).
3. In the **New Private Key** input field, select your private key.

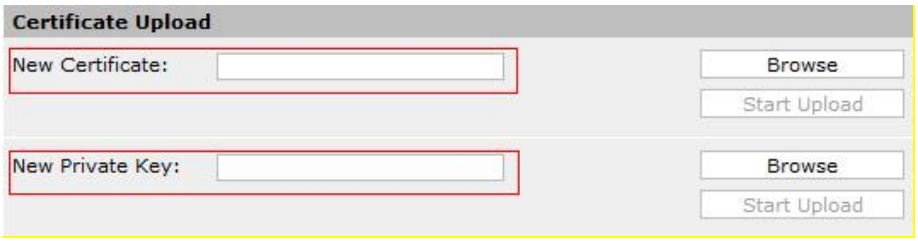


Figure 71: WBM, Select Certificates

**Note**

**Storage locations of the certificates and keys:**
Once they are created, the certificates are stored in the /etc/certificates/ folder. The private key is stored in the /etc/certificates/keys/ folder.

4. In the “IPsec” section, check the **IPsec enabled** box so that the IPsec service will be available after a restart.


Figure 72: Enable IPsec Service

5. Confirm your entry with the [**Submit**] button.

6. Navigate to the **Administrator** > **Reboot** menu.

7. Click the [**Reboot**] button.

The controller then restarts, and the IPsec application starts.

## 7.2 Port Authentication According to IEEE 802.1X

Among other things, the PFCX00 controllers support a mechanism that makes it possible to operate the controller as a supplicant for port authentication according to IEEE 802.1X.

This functionality is provided by the Linux® application “wpa_supplicant.” Two methods are used for authentication of the supplicant:

- **Port authentication via username and password**
  Authentication is done by providing login data in the form of a username and password, which must be specified within the configuration file (see following section). This method is implemented with the EAP-MD5 authentication protocol.
- **Port authentication via certificate**
  Alternatively, it is possible to save a “client certificate” for the supplicant, which is used for authentication. This method is implemented with the extensible EAP-TLS authentication protocol.

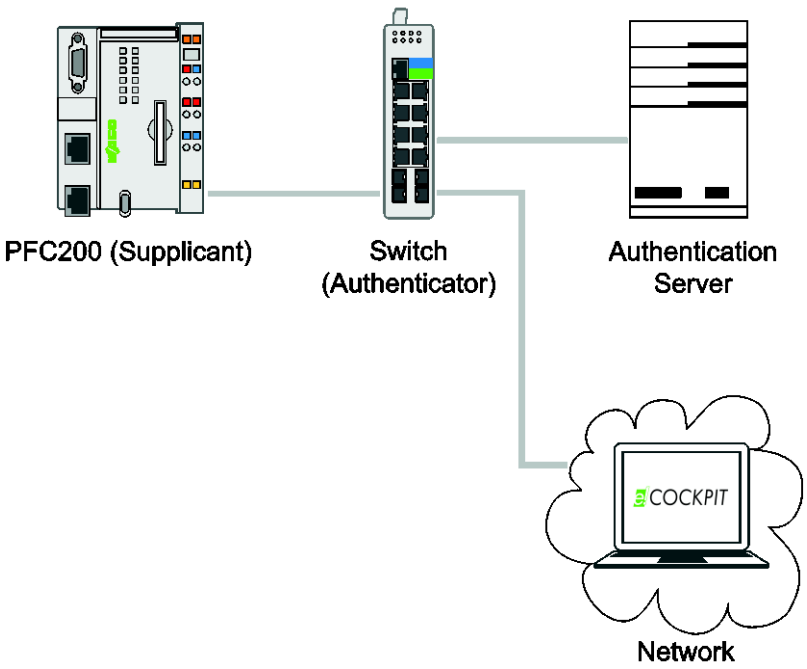


Figure 73: Basic Principle of Port Authentication

The network architecture shown in the figure “Basic Principle of Port Authentication” illustrates basic port authentication according to IEEE 802.1X:

The controller (supplicant) is connected to the switch (authenticator); the port authentication must be switched to active for the switch.

Both the authentication server (e.g., RADIUS servers) and the network from which the controller is to be configured via the ***e!****Cockpit* application are connected via the switch.

**Note**

**No communication with the network is possible without authentication.**
If the controller has not been authenticated to the authentication server, it is not possible to reach the controller through the network. It is not possible for the controller to reach the elements of the network.

Authentication is performed by the switch, which forwards the request of the controller that was not authenticated to the authentication server. If authentication on the authentication server was successful, the switch enables access to the network for the controller; access is denied otherwise.

The switch exchanges the authentication data with the controller via the “Extensible Authentication Protocol over LAN” (EAPOL). For a RADIUS server, the authentication data is exchanged between the switch and authentication server in EAP packets encapsulated in RADIUS packets.

**Note**

**The switch must support the RADIUS protocol.**
The communication between the switch and authentication server takes place via a specific authentication protocol such as RADIUS. The switch converts the EAP packets of the supplicant to the RADIUS protocol and the RADIUS packets from the authentication server to the EAP protocol. A prerequisite for this is that the switch supports this protocol.

## 7.2.1 Port Authentication via Username and Password According to EAP-MD5

The core of this method is the “Challenge Handshake Authentication Protocol” (CHAP) in combination with the MD5 hash algorithm.

The supplicant first establishes an EAPOL connection to the switch. Once the connection is established, the authentication server sends a “challenge request” (random value) to the supplicant. The supplicant then forms an MD5 hash value from the input parameters “challenge” (random value of the authentication server) and “password” of the supplicant. The supplicant sends the calculated hash value back to the authentication server as a “challenge response.”

Since the authentication server knows both the password of the supplicant and also the challenge that was sent, the server also generates an MD5 hash value from the two input parameters and compares its generated hash value to the supplicant’s value. If the two values are identical, the supplicant has been successfully authenticated.

The following figure illustrates the principle of port authentication:

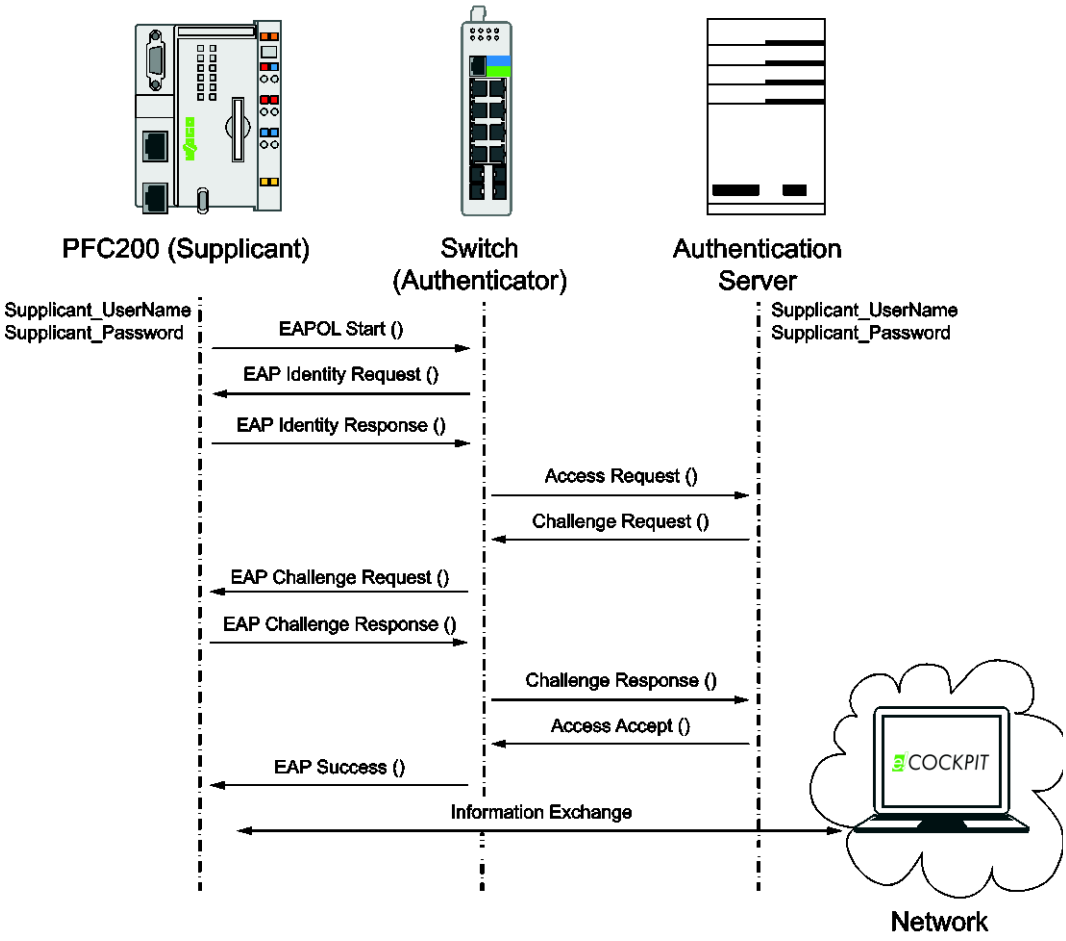


Figure 74: Port Authentication Process According to EAP-MD5

1. As soon as the “wpa_supplikant” application is executed on the controller, the controller sends the initial message “EAPOL Start” to the switch.

2. The switch then sends the “EAP identity request” to the controller, which challenges it to identify itself.
3. The controller then sends its identity (“Supplicant_UserName“) to the authenticator, and this is forwarded to the authentication server.
4. The authentication server checks the identity by comparison to its user database.
5. Once the identity is known, the authentication server sends the “challenge request,” a random number required for authentication, to the server.
6. After the “challenge request” message is received, the controller sends its authentication data back to the authentication server as a “challenge response” (MD5 hash value of the random number of the server and the supplicant password).
7. The “challenge response” message is checked by the authentication server.
8. If the challenge response mechanism was successful, the authentication server sends the message “Access Accept” to the controller.
9. In this way, the switch grants access to the network, and the network element gets confirmation of the successful authentication.
10. Thus the controller can access the network and be accessed by the elements of the network.

#### 7.2.1.1 Set up EAP-MD5 Port Authentication

1. Edit the /etc/wpa_supplicant.conf configuration file of the controller:

```
network={
          key_mgmt=IEEE8021X
          eap=MD5
          identity="Supplicant_Name"
          password="Supplicant_Password"
          eapol_flags=0
}
```

2. Configure the switch and authentication server (e.g., RADIUS server).
3. Use the following command to start/test the EAP-TLS authentication on the controller:

```
wpa_supplicant -dd -Dwired -ibr0 -c/etc/wpa_supplicant.conf
```

Table 10: Description of the Parameters

| Parameter | Explanation |
|---|---|
| -dd | Debug mode |
| -D | Driver to use (wired) |
| -i | Interface of the devices (br0: ETHERNET interface X1; br1: ETHERNET interface X2) |
| -c | Path to the WPA supplicant configuration file (wpa_supplicant.conf) |

For further information on the parameters/configuration, see https://linux.die.net/man/8/wpa_supplicant

**Information**

**Port authentication with a certificate is more secure.**
If possible, use certificate-based authentication (EAP-TLS), since it can ensure both the authenticity of the client and server and the integrity of the communication through modern cryptographic methods:

- Mutual authentication,
- Negotiation of encryption methods with integrity protection,
- Secure key exchange between two endpoints.

## 7.2.2 Port Authentication via Certificates (EAP-TLS)

For certificate-based port authentication via EAP-TLS, both the authentication server and the supplicant need a valid, trusted digital certificate (X.509) from a certificate authority (CA). A “Public Key Infrastructure” (PKI) is a prerequisite for this. The CA certificate forms the “trust anchor” for the authentication, so the trustworthiness (authenticity) of the communication participants can be ensured. The digital certificates of the certificate authority (CA) issued for the supplicant and authentication server is used for mutual authentication. The communication is protected with the established TLS protocol. Access to the network is granted if both the server and the supplicant have been mutually authenticated. Providing a password as in the EAP-MD5 method is no longer necessary in this case (see the section “Port Authentication via Username and Password According to EAP-MD5”).

**Note**

**Certificates and keys must be created first.**
Certificates and keys for the PFC and authentication server must be created first. You can find instructions for creating and setting up certificates and keys in the section “Hardening” > ... > “Generate and Replace Certificates.”

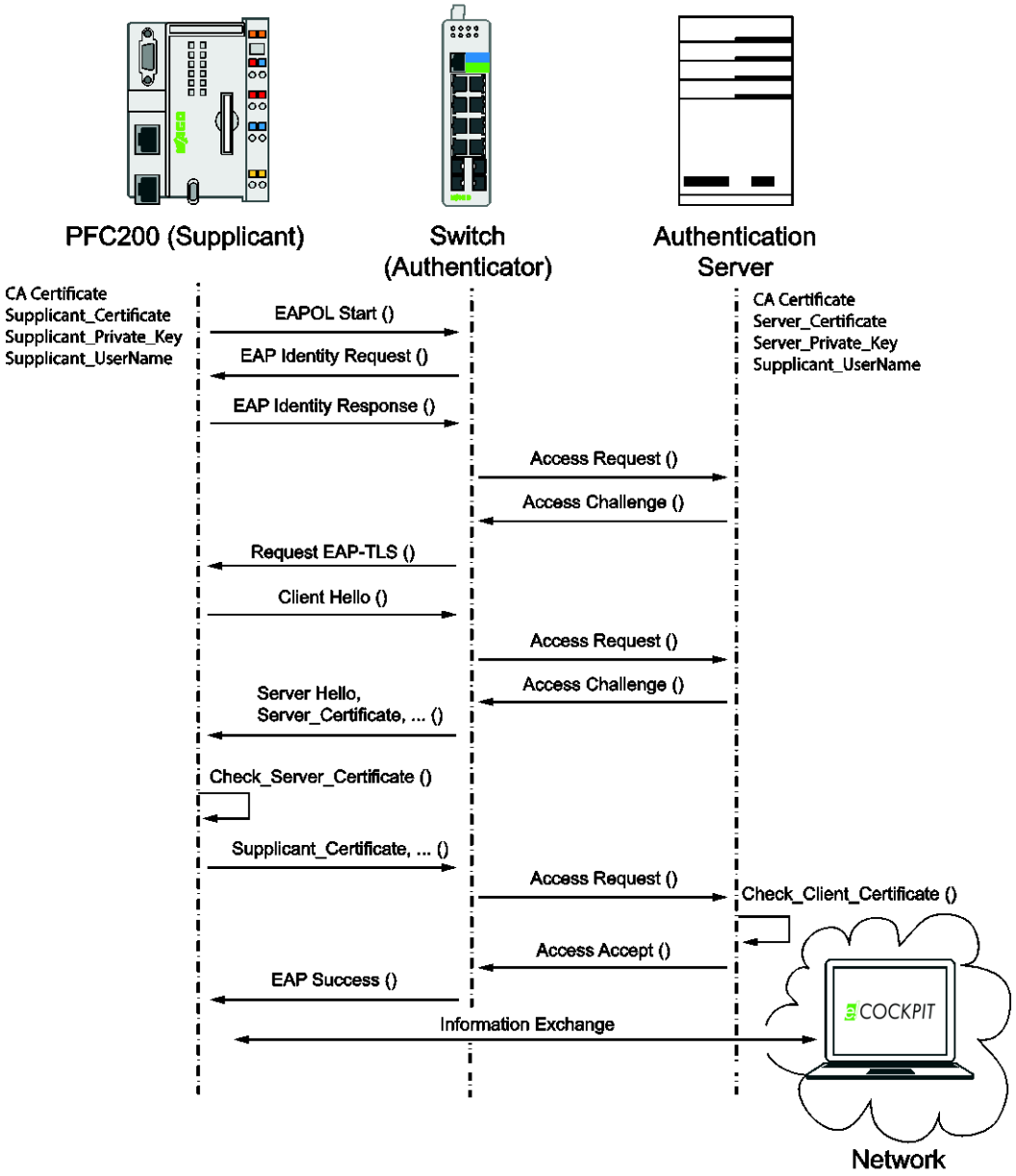


Figure 75: Port Authentication According to IEEE 802.1X, Certificate

1. As soon as the “wpa_supplikant” application is executed on the controller, the controller is challenged to identify itself and sends it identity (Supplicant_UserName) to the authentication server.
2. The authentication server checks whether the identity exists in the identity database.
3. If the identity exists, the message “Request EAP-TLS” is sent to the controller, which prompts the controller to execute a TLS handshake.
4. The controller starts the TLS handshake with the message “Client Hello,” which lists the supported cipher suites, among other things.
5. The authentication server then responds with the message “Server Hello” and sends its server certificate to the controller.
6. After the server certificate is received, the CA certificate is used to check whether the certificate is trusted and valid.
7. If the controller trusts the server certificate and its validity has been verified, the controller sends its certificate to the authentication server.

8. The process for the controller certificate is analogous to step 6.
9. If the authentication server trusts the controller certificate and the certificate is valid, access to the network is granted to the controller with the message “Access Accept” or “EAP-Success.”

**Note**

**No access to the network if authentication unsuccessful**
If the authentication fails, either the server and controller have no certificate, or an existing certificate has been rejected by the remote station, so no access to the network is possible.

#### 7.2.2.1 Set up EAP-TLS Port Authentication

1. Create certificates and keys for port authentication; see the section “Hardening” > ... > “Generate and Replace Certificates.”
2. Edit the /etc/wpa_supplicant.conf configuration file of the controller as follows:

```
network={
          key_mgmt=IEEE8021X
          eap=TLS
          identity="Supplicant_Name"
          ca_cert="/etc/certificates/CA.crt"
          client_cert="/etc/certificates/Supplicant.pem"
          private_key="/etc/certificates/keys/Supplicant_Key.pem"
          eapol_flags=0
}
```

3. Configure the switch and authentication server (e.g., RADIUS server).
4. Use the following command to start/test the EAP-TLS authentication on the controller:

```
wpa_supplicant -dd -Dwired -ibr0 -c/etc/wpa_supplicant.conf
```

Table 11: Description of the Parameters

| Parameter | Explanation |
|---|---|
| -dd | Debug mode |
| -D | Driver to use (wired) |
| -i | Interface of the devices (br0: ETHERNET interface X1; br1: ETHERNET interface X2) |
| -c | Path to the WPA supplicant configuration file (wpa_supplicant.conf) |

For further information on the parameters/configuration, see https://linux.die.net/man/8/wpa_supplicant

### 7.2.3 Automatic Port Authentication during the Boot Process

You can create a start script so you do not have to execute the “wpa_supplicant” application manually (see the section “Port Authentication via Username and Password According to EAP-MD5”). The example start script in the following figure allows you to perform the port authentication automatically while the controller starts. A prerequisite is that you have created a corresponding configuration in the “/etc/wpa_supplicant.conf” configuration file (e.g. see the section “Set up EAP-MD5 Port Authentication.”) After the script starts, the “wpa_supplicant” application runs as a background process.

To set up the start script, the following steps are necessary:

1. Create a file “wpa_supplicant” with the following content:

```
#!/bin/sh

#
# wpa_supplicant
#
PATH=/usr/bin:/usr/sbin:/bin:/sbin

PREFIX="wpa_supplicant: "
WPA="/sbin/wpa_supplicant"
WPA_CONF="/etc/wpa_supplicant.conf"
WPA_IF="br0"
WPA_DRIVER="wired"
WPA_DAEMON_OPT="-B"
WPA_OPTIONS="-D$WPA_DRIVER -i$WPA_IF -c$WPA_CONF $WPA_DAEMON_OPT"

case $1 in

        start)
                echo "${PREFIX}starting"
                if start-stop-daemon --start --quiet --oknodo --exec ${WPA} --
      ${WPA_OPTIONS}; then
                        echo "${PREFIX}done"
                else
                        echo "${PREFIX}error, could not start wpa_supplicant"
                fi
                ;;
        stop)
                echo "${PREFIX}stoppping"
                if start-stop-daemon --stop --quiet --oknodo --exec ${WPA}; then
                        echo "${PREFIX}done"
                else
                        echo "${PREFIX}error, could not stop wpa_supplicant "
                fi
                ;;
        *)
                echo "${PREFIX}usage: ${0} [start|stop]"
                exit 1
                ;;
esac
```

2. Transfer the “wpa_supplicant” file to the /etc/init.d/ folder of the controller.

3. Connect to the Linux® console (e.g., via SSH or the serial console).

4. Create a symbolic link for the execution of the script during the boot process. Linux® command:

```
ln -s /etc/init.d/wpa_supplicant /etc/rc.d/S97_wpa_supplicant
```

After the restart, the “wpa_supplicant” application starts as a background process and attempts to authenticate itself to a remote station (e.g., the authentication server).

**Note**

**Pay attention to the interface information:**
In the example file “wpa_supplicant,” authentication takes place via interface X1 (br0). Please adapt the “WPA_IF” interface variable in your start script according to your configuration.

## 7.3 Simple Certificate Enrollement Protocol (SCEP)

The HTTP-based “Simple Certificate Enrollment Protocol” (SCEP) allows centralized distribution and management of device certificates on any number of controllers in a network. An SCEP server handles providing and managing the certificates. The respective device (controller) automatically generates an RSA key pair for itself and requests a certificate. The SCEP server checks the request and generates a signed X.509 certificate which can then be retrieved via the SCEP protocol and installed locally; see the figure “Simple Certificate Enrollment Protocol” (SCEP). A manual mode and an automatic mode are distinguished. Both modes are described in the following sections.

To ensure integrity and confidentiality, the transferred data is packaged in PKCS#7 formats.

**Note**

**Configure the necessary infrastructure first.**
To create and provide certificates via the SCEP protocol, you need a corresponding infrastructure. The server can be implemented as a Windows 2003 server CA with a special plug-in (mscep.dll), for example.

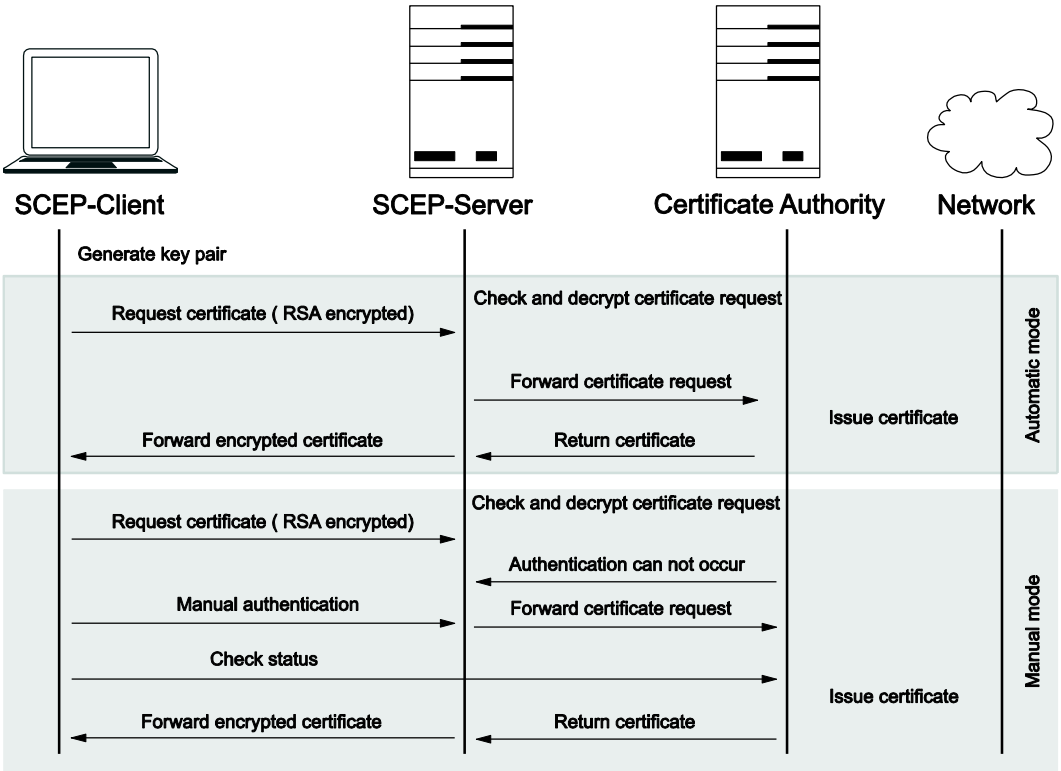


Figure 76: Simple Certificate Enrollement Protocol (SCEP)

### 7.3.1 Automatic Request Processing

In automatic processing, the authenticity of the requester must be guaranteed through a security query. If the security query in the certificate request matches the current valid value on the server, a device certificate can be issued automatically.

1. The client generates an RSA key pair.
   The public part of this key pair is sent to the server together with the request later. The private part of the key pair stays in the client.
2. The client sends the public part of the generated key pair (public key) to the server together with information on its identity (name, email address etc.) as a certificate request. This request is signed with the private part of the key pair.
3. The server checks the certificate request and issues a device certificate without further interaction if the data suffices for authentication.
4. The device certificate is forwarded to the client and provided for VPN operation, for example.

### 7.3.2 Manual Processing

The server switches to “manual mode” if it needs more information to authenticate the requester, i.e. the server puts the certificate request in a waiting state until the approval or rejection from the certificate authority is available. In “manual mode,” the certificate is not delivered directly. Instead, the client can continually query whether the authentication has occurred. As soon as it has, the client receives the certificate in response to its request.

1. The client generates an RSA key pair.
2. The client sends the public part of the generated key pair (public key) to the server together with information on its identity (name, email address etc.) as a certificate request. This request is signed with the private part of the key pair.
3. The server checks the certificate request and puts it in a waiting state if the authentication cannot be performed.
4. The client is informed that the authentication cannot occur.
5. Manual authentication occurs, e.g. over the telephone.
6. Through cyclic querying, the client determines whether it can retrieve the certificate.
7. The server checks the certificate request and issues a device certificate without further interaction if the data suffices for authentication.
8. The device certificate is retrieved by the client and provided for VPN operation, for example.

### 7.3.2.1 Set up SCEP Process

**NOTICE**

**Synchronization of the time on the controller:**
For certificate verification via the SCEP protocol, the date and time of all controllers must be synchronized. The simplest way to do with is via the “Network Time Protocol” (NTP).

You can first use the following command to display a list of the available parameters of the SCEP client:

```
ipsec scepclient --help
```

1. Generate a 2048 RSA key pair according to the PKCS1 standard; see the section “Hardening” > … > “Generate Private Keys.”

**Note**

**Export the key pair and load it onto your device.**
Create and export your RSA key pair with the XCA key management software. Select the export format “.der.” The key pair can then be loaded onto the controller via the WBM: **OpenVPN/IPsec** > **Certificate Upload** > **New Private Key**. The private key is stored in the /etc/certificates/keys/ folder.

2. Load the CA certificates for the PKCS7 encryption (request) and the PKCS7 signature verification (response)

```
ipsec scepclient --out cacert --url
http://10.1.101.53/certsrv/mscep/
```

CA certificates are then loaded according to the configuration of the infrastructure and used to protect the SCEP communication. In the example, these are the certificates “caCert-ra-1.der” for the encryption of the SCEP request and “caCert-ra-2.der” for the signature verification of the SCEP response of the server.

3. Load a certificate from the certificate authority on the basis of the generated RSA key pair (initial) and loaded CA certificates from step 2:

```
ipsec scepclient --out cert=pfc200Cert.der --in
pkcs1=pfc200private.der --in cacert-enc=caCert-ra-1.der --in cacert-
sig=caCert-ra-2.der --url http://10.1.101.53/certsrv/mscep/ -p
90261AC82C586743
```

**Note**

**Note that the names are merely placeholders.**
The names of the key pair given above (pfc200Cert.der/pfc200private.der) and of the server and passphrase parameters (--url, -p) are placeholders and can be assigned differently as needed.

After the command is entered, a self-signed certificate is created for the PKCS7 signature. A certificate request (PKCS10) is then created with the PKCS1 key

(signature). A PKCS7 request is then processed with the help of the CA certificates. Since the PKCS7 signature is based on self-signed certificates, a password (-p) must also be provided, which is provided by the certificate authority calling the URL. It might also be possible to omit the password, depending on the configuration of the certificate authority. The trusted certificate “pfc200Cert.der” is then stored in the “/etc/certificates” directory.

If a new certificate request is made after the current certificate expires, it is possible to use the trusted certificate that is used currently. It is not necessary to create a new self-signed certificate. Since the certificate is a trusted certificate, password entry (-p) is omitted. If the parameter “–in cert-self=pfc200Cert.der” is provided, the current certificate “pfc200Cert.der” is used, not a self-signed certificate:

```
ipsec scepclient --out cert=pfc200Cert.der --in cert-self=pfc200Cert.der
--in pkcs1=pfc200private.der --in cacert-enc=caCert-ra-1.der --in cacert-
sig=caCert-ra-2.der --url http://10.1.101.53/certsrv/mscep/
```

**Note**

**Further information on SCEP:**
You can find further information on the “scepclient” application at http://manpages.ubuntu.com/manpages/artful/man8/scepclient.8.html

# 8 Appendix

## 8.1 FAQ zu IPsec

There are various possible causes for a faulty setup of an IPsec connection. This section provides information about possible problems that can arise with an IPsec configuration. Error analysis measures on the PFC200/PFC100 controllers are described at the same time.

Table 12: Notes and Actions

| Notes | Action Plan |
|---|---|
| Since the encryption and authentication methods are configurable, problems may arise when setting up a “Security Association” (SA) with different VPN products. If the configurations of the IPsec remote stations are different, an error may occur. The connection then cannot be established. | Ensure that the IPsec remote stations use the same authentication and encryption methods. For the controller, this configuration is made in the “ipsec.conf” configuration file; see the section “IPsec” > “Create Configuration Files.” |
| | |
| For routing between the networks in a site-to-site scenario, the address ranges of the subnets that are to be connected must differ. | • Specify a separate address range (different subnets) for the subnets that are to be connected.<br>• Once the networks are interconnected, do not perform multiple IP address assignment, since this will lead to a faulty IPsec connection.<br>• If the setup is successful, note that the IPsec application “strongSwan” automatically adds a route to routing table 220.<br>Linux® command:<br>`root@PFC200-405679:~ ip route list table 220` |
| | |
| The IKEv2 key exchange protocol is more stable and user-friendly than the IKEv1 key exchange protocol. | Use the IKEv2 key exchange protocol instead of IKEv1 to prevent more significant problems with the NAT technology, dynamic IP addresses and mobile devices. |
| | |
| Strong encryption/cryptographic protection leads to high resource usage. This can cause delays and anomalies in the program operation, so the controller can no longer handle its own tasks. | Use encryption that is not too strong, and/or pay attention to the choice of the key lengths for the cryptographic method corresponding to the BSI TR-02102-4 technical guidelines (version 2017-01). |
| | |
| If you use certificates, the time must be identical on all IPsec remote stations. Otherwise, problems may arise in verifying certificates, preventing an IPsec connection from being established. | • Make sure that the time is the same on all systems.<br>• If you are using a time server, you can change the time/date manually via the console (date --set "YYYY-MM-DD HH:MM") or the WBM. |

**Note**

**You can get more information directly from stronSwan:**
A strongSwan FAQ is available at
https://wiki.strongswan.org/projects/strongswan/wiki/FAQ!

### 8.1.1 Additional IPsec Errors/Status Analysis

- For IPsec error analysis, analyze the IPsec log entries listed at the path "`/var/log/messages`."
- It is possible to raise the log level of the logging output in the "ipsec.conf" configuration file in order to get more detailed information in the event of an error.
  https://wiki.strongswan.org/projects/strongswan/wiki/LoggerConfiguration
- View the incoming and outgoing network packets (e.g., via the SSH service) with the following command:
  `tcpdump port not 22 -n -i eth0.`
- You can find more detailed information on configuring and testing an IPsec connection at
  http://wiki.strongswan.org/projects/strongswan/wiki/UserDocumentation.

# List of Figures

# List of Tables

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D - 32385 Minden
Hansastraße 27 • D - 32423 Minden
Phone: +49 571 887 – 0
Fax: +49 571 887 – 844169
E-Mail: info@wago.com
Internet: www.wago.com