RTU for the MV/LV substation – Cyber security requirements

This standard specifies the Cyber security requirements for the RTU used in the MV/LV distribution substations.

Edition	Date	List of modifications
0	20.05.2019	First draft
1	12.06.2019	Second draft
2	27.06.2019	First approved edition
3	10.03.2026	Updating the company logo

INDEX

TABLES

ANNEX

1 ACRONYMS

a. **API** Application Programming Interface
b. **ARM** Advanced RISC Machine
c. **B, kB, MB, GB** Memory size expressed with a capitol letter (e.g. kB, MB), etc. means xBYTE
d. **BASH** Bourne Again SHell
e. **CS** MV/LV Distribution Substation (also “Secondary DS”)
f. **DNF** Dandified YUM (software package manager)
g. **DS** Distribution Substation
h. **FW** Firmware
i. **GS Client** Standard
j. **HW** Hardware
k. **IED** Intelligent Electronic Device
l. **IPsec** IP Security
m. **JTAG** Joint Test Action Group
n. **LV** Low Voltage
o. **MCU** RTU main processor
p. **MV** Medium Voltage
q. **NTP** Network Time Protocol
r. **OS** Operating System
s. **PCB** Printed Circuit Board
t. **PSBC** Power Supply Battery Charger of the RTU
u. **RADIUS** Remote Authentication Dial-In User Service
v. **REST** REpresentational State Transfer
w. **RTU** Remote Terminal Unit
x. **SFTPD** Secure FTP Daemon
y. **Syslog-ng** System Log next generation
z. **SR_XX** Security Requirement (XX type, e.g. SW=Software, HW=Hardware, DC=Documentary)
aa. **SSH** Secure SHell
bb. **SSL** Secure Sockets Layer
cc. **SSSD** System Security Services Daemon
dd. **SW** Software
ee. **TCP** Transmission Control Protocol
ff. **TLS** Transport Layer Security
gg. **UE2020** Elaboration Unit of the UP2020
hh. **UEX** I/O interface eXpansion module of the UP2020
ii. **UP** RTU of a MV/LV DS with chance to be used in IP networks
jj. **UP 2020** RTU of a MV/LV DS with enhanced communication capability (e.g. IEC 61850, MQTT)

2 SCOPE OF THE WORK

The Cyber Security requirements for the MV/LV Distribution Substation RTU have been defined especially for the Client product “UP”. However, these requirements may be adopted in other similar devices and so it could be used as reference in the device technical specification.

3 NORMATIVE REFERENCES AND BIBLIOGRAPHY

All the references in this GS are intended in the last revision or amendment.

IEC 61850 series	Communication networks and systems for power utility automation
IEC 60870-5-104	Telecontrol equipment and systems – Part 5-104: Transmission protocols – Network access for IEC 60870-5-101 using standard transport profiles
Client CSG 12	Cyber Security Guideline no. 12 – Version no.1 dated 11/09/2017 Client Operational Technologies (OT) security guideline on industrial control systems
Client CSG 7	Cyber Security Guideline no.7 – Version no.2 dated 30/09/2017 Client “IT Security Guidelines - APPLICATIONS”

4 REPLACED STANDARDS

0	20.05.2019	First draft
1	12.06.2019	Second draft
2	27.06.2019	First approved edition

5 RTU CYBER SECURITY REQUIREMENTS[1] DESCRIPTION

The security requirements belong to three categories:

- **Hardware security requirements**: this section includes the Hardware requirements that "strengthen" the RTU Device against physical attacks aimed at accessing to the internal/logical components. Physical security requirements against acts of vandalism or enabling an overall physical protection of the device are out of scope;
- **Firmware security requirements**: this section prescribes the typical security requirements of the on-board Software. All the SW components in the RTU will be affected. This set of SW is called "firmware";
- **Documentary Requirements**: this section deals with requirements for the documentation attached to the supply.

The requirements specified in the following sections can be:

- **Mandatory**, that means necessary for the award of the contract;
- **Optional**, that could additionally increase the score of the proposal during the technical evaluation.

The Supplier is required to give details concerning the technical procedures used to fulfill both types of requirements.

In this document any requirement referred to communication services or functions is designed to be implemented and used in an effective and reliable way, if the RTU works in IP networks.

6 RTU HARDWARE CYBER SECURITY REQUIREMENTS

6.1 Hardware Architecture

This section contains the Hardware characteristics required

- to meet the security requirements;
- to update and standardize the hardware performance of the RTU and the capability to manage the devices over time.

6.1.1 MCU Platform

6.1.1.1 SR_HR_01

Requirement type: **Mandatory**

In order to support the security services, the hardware platform of the main processor must fulfill the following **minimum** requirements:

- ARM Architecture;
- Cortex-A7 Processor Class;
- One Core;
- Core frequency @ 700 MHz.

These are the minimum performance requirements intended to guarantee (now and in the future) **only the security features of the RTU**. Therefore, the Supplier shall size the MCU of the device taking into account the adequate performance requirements both to deliver the RTU application services and to support the security services.

[1] Hereafter the terms "**will**" and "**shall**" mean "**have the duty to**".

6.1.1.2 SR_HR_01bis

Requirement type: **Optional**

The Supplier can offer more performing solutions for the MCU in terms of number of cores, processor class or frequency (see SR_HR_01). Also 64 bit architecture could be proposed if it provides, during the execution of 32-bit applications, comparable or better performance.

6.1.1.3 SR_HR_02

Requirement type: **Mandatory**

Hardware components, in particular the MCU, must not be classified as "Discontinued" or "End of Life" at the time of supply and, at least, for the **next 10 years**. In addition, at the time of the supply, the MCU must not be classified as NRND (Not Recommended for New Design) or similar.

6.1.1.4 SR_HR_03

Requirement type: **Mandatory**

The “Product Longevity”, that is the minimum supply and support period of the MCU by the Manufacturer, must be **equal or more than 15 years.**

6.1.1.5 SR_HR_04

Requirement type: **Mandatory**

The MCU and the memory devices must be classified as Industrial Grade.

6.1.1.6 SR_HR_05

Requirement type: **Mandatory**

The memory/storage devices of the RTU must have, at least, the following characteristics:

- Minimum 512 MB RAM memory;
- Minimum 2GB Flash memory;

they must be welded directly on the PCB (without socket) and hardly removable.

These are the minimum performance requirements necessary to guarantee (now and in the future) **only the security features of the RTU**. Therefore, the Supplier shall size the memory/storage devices of the RTU taking into account the performance requirements adequate both to deliver the application services and to support the security services.

6.1.1.7 SR_HR_05bis

Requirement type: **Optional**

The Supplier can offer more performing solutions in terms of RAM and/or Flash memory. (see SR_HR_05)

6.1.2 Security-specific functions

6.1.2.1 SR_HR_06

Requirement type: **Mandatory**

The MCU must fully support and, where possible, accelerate via ad-hoc instructions or HW components the security protocols that are currently classified as secure in the reference document published by the ENISA[2] "*Algorithms, Key Sizes and Parameters Report*" (latest available version).

6.1.2.2 SR_HR_07

Requirement type: **Optional**

The MCU or any additional component in the RTU may provide the following (optional) security feature: the integrity validation of the RTU Bootloader or Firmware, by verifying their digital signature during the device start-up.

As an example, please, refer to the Arm Trusted Firmware[3] functionality.

6.1.2.3 SR_HR_08

Requirement type: **Optional**

The RTU should support the complete encryption of its Flash memory through algorithms that are currently classified as secure in the reference document published by the ENISA[4] "*Algorithms, Key Sizes and Parameters Report*" (latest available version). A secure storage system (according to the Anti-tampering mechanisms) for the archiving of the encryption keys or of the certificates can be provided. It's up to the Supplier to propose the solution deemed most effective.

6.1.3 Interfaces and physical ports/connectors security

6.1.3.1 SR_HR_09

Requirement type: **Mandatory**

During the operation of the RTU the programming and/or debugging interfaces must be blocked at HW level. Therefore, the Supplier is requested to remove all the programming ports available on the electronic boards of the RTU intended for the operation in the substation; particular care must be taken of the JTAG interfaces.

This requirement can be waived only for RTUs used for testing or debugging purposes.

Alternatively to HW disabling, permanent blocking via SW is allowed (e.g. refer to Secure JTAG[5].)

6.1.3.2 SR_HR_10

Requirement type: **Mandatory**

All of the physical interfaces and ports not expressly required by the Client functional specifications, according to this security annex, must be blocked at Hardware level (e.g. USB ports, additional serial interfaces, etc.). However wireless interfaces (e.g. Bluetooth, Wi-Fi, Infrared, etc.) are not allowed.

6.2 Hardware Solutions

This section of the document contains the requirements for hardware solutions used to strengthen the security of the HW system. Some of these requirements are classic anti-tampering mechanisms, others are measures introduced to hinder the Reverse Engineering.

[2] https://www.enisa.europa.eu/topics/data-protection/security-of-personal-data/cryptographic-protocols-and-tools
[3] https://www.arm.com/products/security-on-arm/trustzone
[4] https://www.enisa.europa.eu/topics/data-protection/security-of-personal-data/cryptographic-protocols-and-tools
[5] https://www.digi.com/resources/documentation/digidocs/90001546/concept/trustfence/c_secure_jtag_android.htm

6.2.1 Anti-intrusion mechanisms

6.2.1.1 SR_HR_11

Requirement type: **Mandatory**

The device must be equipped with protection solutions that are Tamper Resistant, in particular:

- all the screws outside the chassis must be of a non-standard type, so use screws with a Security Torx or TRI-WING drive.

6.2.1.2 SR_HR_11bis

Requirement type: **Optional**

The device must be provided with paint seal or label on screws that denounces an attempted break-in.

6.2.1.3 SR_HR_12

Requirement type: **Mandatory**

The device must be equipped with Tamper Detection solutions, in particular:

- one or more switches must be provided to detect the opening of the enclosure of all the RTU modules (PSBC, UE2020 and UEX)
- moreover, the switch status change must trigger the execution of an arbitrary command to the system (e.g. a system shut down, a script execution, etc.). The trigger will be activated also by the unauthorized configuration modification of tamper switch logic. The type of action must be customizable by Client after the production phase of the RTU.

The adopted solution, as a part of the device itself, shall be resilient to any mechanical or electromagnetic test, which is described and/or required in the device technical specification.

6.2.1.4 SR_HR_13

Requirement type: **Mandatory**

The device must be equipped with Tamper-Evidence solutions, in particular:

- In case of status change of the Tamper Detection switch(es) (ref.
- SR_HR_12), an event must be generated and reported with a Syslog message including the time stamp and the trigger event type.

6.2.1.5 SR_HR_14

Requirement type: **Optional**

The device can be equipped with Tamper Detection solutions capable of detecting the status change of the switch(es) even in case of powered-off RTU (e.g. using a memory register and a buffer battery): an event will be memorized and used at the next power-up of the RTU, according to the requirement

SR_HR_12 and SR_HR_13)

6.2.1.6 SR_HR_15

Requirement type: **Optional**

The device can be equipped with Tamper Response solutions, such as:

- When the status change of the Tamper Detection switch(es) is detected, execute a customizable kill switch that can perform, for example, the following operations:

- Lock the device by disabling the bootloader,
- Erase the Flash memory the next time the RTU is restarted,
- Delete the cryptographic keys,
- Physically burn (destroy) the Flash memory;

- In addition, for maintenance purposes, a normally open and difficult to identify switch must be provided; once activated, it allows to temporarily disable the kill-switch.

It's up to the Supplier to propose possible solutions that he considers most effective, motivating them.

6.2.2 Concealment of the components

6.2.2.1 SR_HR_16

Requirement type: **Mandatory**

Silk screen omission: the PCBs constituting the RTU must have no silkscreen except for the PCB code and the Manufacturer logo/data.

6.2.2.2 SR_HR_17

Requirement type: **Optional**

The silkscreens on top of the integrated circuits can be removed or hidden to limit the attacker's ability to understand the used components.

6.2.2.3 SR_HR_18

Requirement type: **Mandatory**

The multilayer PCB must not expose copper tracks on the outer layers that could easily be identified as linked to the pins of the Flash memories or of the MCU, e.g. the copper tracks of Serial ports, modems, JTAG or other interfaces.

6.2.2.4 SR_HR_19

Requirement type: **Optional**

The Supplier may propose the use of epoxy encapsulation or resin coating of the components as a supplementary (e.g. encryption of solid state memories) or alternative solution in case some of the mandatory security requirements can't be met, including: removal of the marking, removal of programming interfaces, etc.

6.2.3 Power supply control

6.2.3.1 SR_HR_20

Requirement type: **Optional**

The Supplier may propose the implementation of circuits monitoring the power supply of the electronic boards, with particular attention to the supplies on the external interfaces (e.g. USB ports). The detection of anomalies must be traced via a Syslog message and must trigger an operation on the system (e.g. the kill-switch activation).

7 RTU FIRMWARE CYBER SECURITY REQUIREMENTS

This section contains the requirements related to the Firmware features and configurations in order to meet the security requirements, to allow the adaptation and standardization of the RTU configurations and the device management and maintenance over time.

In this document the term Firmware means the entire set of the RTU software components, including;

- Root File System;
- Kernel;
- Bootloader;
- Middleware (basic and functional applications, including Application Software).

The FW is stored in the Flash memory.

7.1 Features of the Operating System

This section defines the requirements concerning the type of OS to which the UP must comply.

7.1.1 Bootloader

7.1.1.1 SR_SW_01

Requirement type: **Mandatory**

The Supplier shall disable the interactive Boot features offered by the Bootloader and completely preclude the possibility to modify the Bootloader configurations. Furthermore, the Bootloader must be configured to allow the OS to boot only from the on-board Flash memory (it is forbidden, for example, to boot from a USB Flash drive or any other external peripheral device).

7.1.2 Operating System

7.1.2.1 SR_SW_02

Requirement type: **Mandatory**

The Supplier shall equip the RTU with a Linux OS; "Bare Metal" solutions are not accepted. The Operating System must have, at least, the following characteristics:

- updatable and extensible in terms of functionality during the service life of the UP;
- maintained by a Supplier or a community providing updates and Security Patches;
- replaceable if discontinued and, therefore, independent of the MCU's hardware Manufacturer;
- generated through "Yocto Project[6]" or a similar framework that guarantees the same functionality in terms of upgradeability and flexibility. The framework must be of the latest version available and supported by the MCU. Client suggests to use the Poky distribution or a distribution derived from it. The latest final OS version or distribution branch must be used (release candidate or beta versions are not allowed).

7.1.2.2 SR_SW_03

Requirement type: **Mandatory**

The Operating System must be equipped with proper resources (Kernel modules, binary and libraries) in order to manage and fully support:

[6] https://www.yoctoproject.org/

- secure communications established through TLS[7], SSH[8], IPSEC[9];
- application protocols and standards like IEC 60870, IEC 61850, IEC 62351, etc.

7.1.2.3 SR_SW_04

Requirement type: **Mandatory**

Every RTU must be equipped with an ad-hoc set of Cryptographic Keys and Digital Certificates (Security Tokens) used to establish secure communications.

The Supplier will install these Tokens during the manufacturing phase of the RTU, and will also provide methods (as described in the following sections) to update the aforementioned Tokens during the operation of the device.

Indicatively, the following Tokens must be present in the UP:

- private/public x509 Certificates pair for the TLS communication service to the SCADA infrastructure; these Certificates must be different on each RTU;
- X509 Certificates pair for the https service; these Certificates must be different on each RTU;
- Public Key pool for the remote access via SSH; these Certificates must be different on each RTU;
- Root-CA public x509 Certificate, common to all devices.

Client , in general, will provide all necessary Digital Certificates; if not, the Supplier shall generate Self-Signed Certificates.

7.2 Middleware Components

This section contains the Middleware requirements the RTU has to comply with. Middleware refers to all the SW required for the execution of the functions the RTU has been designed to, including the application software when not provided by Client , the management system and the basic security software.

7.2.1 Remote Management functionalities of the device

7.2.1.1 SR_SW_05

Requirement type: **Mandatory**

In addition to the management functionalities specified in the Technical Specification, the FW must be provided with an easy (to implement) access to the essential security features (ref. SR_SW_06) for a correct centralized management of the RTUs, such as:

- SSH service configuration:
  - activation/deactivation of the service (activated by default),
  - Public Keys addition and removal in "authorized_keys";
- SCP/SFTP Service Configuration:
  - identification of the users enabled for the service,
  - definition of the users' access rights (reading or reading/writing);
- Network services configuration:
  - NTP server configuration for clock synchronization,
  - system hostname configuration,
  - system DNS configuration,
  - IP addresses configuration;
- Firewall service:
  - activation/deactivation,
  - bulk download or bulk upload of iptables rules configuration file;

[7] https://en.wikipedia.org/wiki/Transport_Layer_Security
[8] https://en.wikipedia.org/wiki/Secure_Shell
[9] https://en.wikipedia.org/wiki/IPsec

- Credentials/Keys Service:
  - creation/deletion of system user and http service,
  - change of user's password and role assignment,
  - upload/update of Cryptographic Keys and Digital Certificates (as defined in SR_SW_04);
- Syslog service:
  - configuration of destination server IPs, ports and protocols for logs transmission,
  - log download;
- System functions:
  - reset of the factory configurations by removing all data and restoring initial configurations,
  - device restart;
- Updates:
  - upload of security update/new firmware,
  - configuration of the repository for the download of the update in accordance with the methods defined in the following sections,
  - execution of the update command, also with related scheduling;
- Information and characteristics of the device:
  - Hardware informations (at least, Manufacturer, Product Name, Version and univocal, for any device, Serial Number), Firmware version, Operating System version, Patching Level, Kernel version, https server version, Application Software version.

7.2.1.2 SR_SW_06

Requirement type: **Mandatory**

The security configurations, as defined in the SR_SW_05 requirement, must be accessible both via API and via Web interface.

The Supplier must use the REST API type as defined in ANNEX 1. Hereafter, Annex 1 has to be referred everytime API method is mentioned. Furthermore, these APIs should share same channel of the Web application (coexistence).

7.2.2 Security of the Software code developed by the Supplier

7.2.2.1 SR_SW_07

Requirement type: **Mandatory**

The Supplier undertakes to develop the SW code according to the security guidelines defined by Client in the document *Guideline n.7 – Client "IT Security Guidelines - APPLICATIONS".*

Client reserves the right to perform Security Static/Dynamic Code Analysis of the software components developed by the Supplier. The Supplier will undertake all necessary corrective actions at its own expense if, during the testing phase of the product, discrepancies with the requirements in the Client guidelines are identified.

7.2.2.2 SR_SW_08

Requirement type: **Mandatory**

All applications developed by the Supplier running on the RTU must perform the tracking of the security logs by generating a Syslog messages (according for example to RFC 5424) in the device.

Client considers the security events concerning

- the (both successful and failed) authentication to the system;
- all the administrative operations performed on the device of high importance.

7.2.2.3 SR_SW_09

Requirement type: **Optional**

If the Supplier executes Security Static Code Analysis with own tools, aimed at identifying potential security vulnerabilities in the code he developed, when issuing the RTU, he can provide Client with the findings identified by such instruments. In case of unresolved reports, the Supplier shall explain the reason why they have not been solved.

7.2.3 Required Security Software

7.2.3.1 SR_SW_10

Requirement type: **Mandatory**

The RTU must be equipped with specific security software, in particular the Supplier is required to equip the Firmware with the following software updated to the latest version:

- Iptables with related dependencies (libraries and Kernel modules);
- OpenSSH configured to support the SFTPD protocol;
- OpenSSL;
- SELinux;
- RADIUS centralized authentication modules;
- SSSD[10] centralized authentication modules;
- Syslog-ng daemon;
- NTP daemon;
- Bash or sh scripting environment;
- Python interpreter.

7.2.4 Remote Management Software

7.2.4.1 SR_SW_11

Requirement type: **Mandatory**

The supplier will have to equip the device with a web interface to manage the functionality of the application software and security configurations (ref. next requirement).

If explicitly required by Client, the software for managing and configuring the parameters of the RTU (i.e. "UP Management SW") must rely on a communication protocol based on TLS or SSH.

7.3 Hardening

This section specifies the security configurations (Hardening) the RTU must be equipped with. These configurations allow to reduce the perimeter of attack exploitable by an attacker that attempts to take the control of the device.

7.3.1 Hardening Guideline

7.3.1.1 SR_SW_12

Requirement type: **Mandatory**

[10] https://en.wikipedia.org/wiki/System_Security_Services_Daemon

The Supplier shall configure the RTU so that the only demons or (IP) network services are the ones authorized by Client, according to the following list:

- Web Server relying on the https service, using TCP port 443, for exposing Web interfaces/device management APIs;
- Secure Shell relying on the SSH service, using TCP port 22, for administrative access to the device;
- Services and protocols strictly required by the communications of the device applications (e.g. IEC 60870-5-104, MQTT, etc).

The presence or the use of any network service exposed by the device and different from those mentioned above must be previously authorized by Client upon presentation of written documentation explaining the Supplier's requirements and their reasons.

7.3.1.2 SR_SW_13

Requirement type: **Mandatory**

All the physical interfaces to access the system not explicitly required in this technical specification and that have not been removed during the hardware design phase must be disabled via software (e.g. additional network interfaces, serial interfaces, etc.) in order to block the interface in any device status or condition. However wireless interfaces (e.g. Bluetooth, Wi-Fi, Infrared, etc.) are not allowed.

7.3.1.3 SR_SW_14

Requirement type: **Mandatory**

The Firmware configuration must follow secure configuration guidelines (defined according to the selected software version and type) at least for the following components:

- SSH service[11];
- Web server selected by the Supplier[12];
- Linux OS[13].

Virtualization of no more supported O.S. is not allowed. However the Supplier, unless specifically indicated by Client when technical proposal has been provided (during the tender and/or TCA), shall choose these guidelines among those publicly available and make them known to Client. As an example, some guidelines are reported in the footnotes.

7.3.1.4 SR_SW_15

Requirement type: **Mandatory**

For the following functions only:

- Clock update via NTP and other authorized synchronization protocols;
- Sending log messages via Syslog;
- Application communication through standard remote control protocols (IEC 60870-5-104) and automation/IoT protocols (MQTT).

the use of unencrypted (IP) network communications, although the RTU is required to natively support the secure versions too, could be approved by Client just in particular circumstances; in this cases, the communications must be secured via Network Security technologies/solutions (e.g. enabling/configuring security features in the network devices).

Unless explicit Client approval, all the other communications must rely on underlying security protocols, that will only support encryption algorithms considered, to date and for the entire life cycle of the RTU, secure, as

[11] https://wiki.centos.org/HowTos/Network/SecuringSSH
[12]https://www.owasp.org/index.php/SCG_WS_nginx , https://www.owasp.org/index.php/SCG_WS_Apache
[13] https://www.sans.org/score/checklists/linux

defined in the document published by the ENISA[14] "*Algorithms, Key Sizes and Parameters Report*" (latest available version).

If, for technical reasons, the Supplier intends to use security protocols with encryption algorithms that currently are considered safe, but may not be considered as such for the entire life cycle of the RTUs as described in the previous paragraph, the Supplier shall implement methods to update the RTU in order to replace the algorithms that will be considered insecure with secure ones.

7.3.2 Security Logging

The traceability of the actions performed on the device during the operation is a key element for its security. The Supplier shall configure the RTU in a way that the it will be able to trace the operations performed on/by the device.

7.3.2.1 SR_SW_16

Requirement type: **Mandatory**

The RTU must be configured to trace the main administrative operations performed on it, like:

- Successful user login to the system through any implemented interface in remote/local access (e.g. SSH access, Web access, API access, "UP Management SW");
- Failed login attempts to the system through any implemented interface;
- Execution of administrative operations using SSH access;
- Execution of security commands using the Web interface, API or the "UP Management SW";
- system restart.

These operations must be tracked inside the system via the Syslog service likewise the other security logs defined in the requirements of this document.

7.3.2.2 SR_SW_17

Requirement type: **Mandatory**

The Supplier shall provide a storage space dedicated to the logs with a minimum capacity of 200 Megabytes. These logs must be non-volatile and, therefore, saved in the RTU's Flash memory.

Therefore the "Log Rotation" function must be provided to guarantee the storage of the most recent logs and the elimination of the old ones.

7.4 Security Patching

The Firmware of the RTU is equipped with Software components that, regardless of the quality of the validation process adopted during the selection, development, integration and configuration phases, may be affected by not-yet-known vulnerabilities. Client requires that the RTU software be updatable with security patches that guarantee the requested level of device security over time.

7.4.1 Updates during the RTU supply

7.4.1.1 SR_SW_18

Requirement type: **Mandatory**

Unless explicitly authorized by Client, at the time of the supply the Firmware on the supplied devices must consist of Software free of known security vulnerabilities; vulnerabilities classified as CVE >= 4[1] are not tolerated/admitted.

[14] https://www.enisa.europa.eu/topics/data-protection/security-of-personal-data/cryptographic-protocols-and-tools
[1] https://cve.mitre.org/

7.4.2 Security updates during the UP operation

7.4.2.1 SR_SW_19

Requirement type: **Mandatory**

During the entire life cycle of the supply, the Supplier shall support Client to update the Firmware security level, in order to fix new vulnerabilities that could affect the supply and the devices in the field. Basically, as long as the contract is in place, the Supplier is required to proactively release Software updates (Security Patches), at least every 6 months, aimed to resolve the security vulnerabilities made public[15] during the same time period.

Furthermore, Client can explicitly require the release of Security Patches, for example in the following circumstances:

- As a result of a Security Assessment carried out by Client or a third-party company;
- Faced with the publication of a new vulnerability affecting the systems and that Client considers necessary to mitigate with high priority. If the reference software-house has not yet released the relevant Security Patch, it shall implement, at least, "workaround" configurations;
- React to a targeted Cyber Attack.

The Supplier is also required to release the Security Patch within 1 month from the security update request by Client. The Supplier shall previously test the new Security Patches on all the supplied versions of RTU.

The Supplier is not required to distribute the Security Patches on individual equipment already deployed in the field: Client is in charge of this activity.

7.4.2.2 SR_SW_20

Requirement type: **Mandatory**

There must be three ways of distributing the security updates:

- Manual installation on the device via Web interface or “UP Management SW”;
- Update via API;
- Update via centralized repository (through DNF[16] or similar).

Due to potential limitations in the network connectivity, the updates must be performed via Patches only (Differential Update), not via a new entire version of the FW.

7.4.2.3 SR_SW_21

Requirement type: **Mandatory**

In the event that a security update is required when a supply is in progress (ref. SR_SW_19), the devices to be supplied must already include it.

7.4.3 Update Security

7.4.3.1 SR_SW_22

Requirement type: **Mandatory**

At least the following technical requirements regarding the security of updates must be fulfilled:

[15] “made public means published on the (Web) sites of the reference software-house
[16] https://en.wikipedia.org/wiki/DNF_(software)

- The Supplier shall digitally sign the released updates;
- The device must be able to perform the hash check (with a known and, currently, safe algorithm) of the update package (transferred to it) before its installation.

7.5 Users, credentials and certificates management

7.5.1 Credentials Security

7.5.1.1 SR_SW_23

Requirement type: **Mandatory**

The default credentials must be removed and each credential configured on the system must comply with the minimum complexity requirements (length and character pattern) according to the Client policy (ref. Cyber Security Guideline no.7). The connection to the system with “root” user is prohibited, both remotely (e.g. via SSH) and locally (e.g., via the serial interface).

7.5.1.2 SR_SW_24

Requirement type: **Mandatory**

"Hardcoded" credentials, that means included directly in the application code, are prohibited. If the credentials are saved in configuration files, they must be properly obfuscated through the use of non-proprietary and non-deprecated Hashing[17] algorithms at least for a time equal to the life cycle of the RTU.

7.5.2 Centralized authentication

7.5.2.1 SR_SW_25

Requirement type: **Mandatory**

The RTUs must support centralized authentication modes that can be optionally activated by Client during the start-up phase for administrative access to the device (via SSH or Web); in particular:

- Radius centralized authentication;
- SSSD centralized authentication[18].

7.5.3 Update of certificates and cryptographic keys

7.5.3.1 SR_SW_26

Requirement type: **Mandatory**

It must be possible to update all the credentials and cryptographic keys of the device during its operation and without affecting/downgrading its functionalities.

7.5.4 Techniques for the protection of the administrative access to the device

7.5.4.1 SR_SW_27

Requirement type: **Mandatory**

[17] https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_1/Fundamentals_of_data_structures/Hash_tables_and_hashing
[18] https://en.wikipedia.org/wiki/System_Security_Services_Daemon

It must be possible to access the device with administrative privileges in the following **three modes**:

1. Remote access via a web interface;
2. Remote access via SSH service;
3. Remote access via API.

The allowed authentication procedures are the following:

- Login via username/password: modes 1 and 2;
- Access through Digital Certificate or mutual authentication: modes 2 and 3.

The following features are required in order to protect the login with username/password:

- The password complexity must comply with the guidelines provided by Client (Cyber Security Guideline no.7);
- Implement password complexity validation mechanisms limited to the mode 1;
- Provide timed lock-out techniques for the credentials.

7.5.4.2 SR_SW_28

Requirement type: **Mandatory**

Besides the user profiles of the device operators defined in the main specification, a profile for the administrative access via the Web interface for the Security Management of the device must be provided:

- Security Administrator user that can only modify the security parameters and configurations as defined in SR_SW_05.

8 DOCUMENTARY REQUIREMENTS

8.1 Detailed Technical Documentation to be provided

8.1.1 Required technical details

8.1.1.1 SR_DC_01

Requirement type: **Mandatory**

The Supplier, in addition to the supply of the RTU, shall provide detailed documentation and software regarding the adopted security configurations, highlighting all the aspects of compliance with the requirements in this GS. Furthermore, in case of changes impacting the Cyber security of the device, the abovementioned documentation shall be updated accordingly.

In particular, the required documentation must include the following information and SW:

- Detailed specification of the security configurations adopted at HW level;
- Detailed description of the selected OS, versions of the SW packages, libraries and Kernel;
- List of the incremental patches with respect to the adopted version of the OS;
- Detailed specification of the Hardening configurations performed on the system compared to the basic configurations of the OS;
- Detailed list of applications, utilities, scripts, databases included in the system that aren't part of the basic OS;
- Evidence of the tests or security checks carried out;
- Changes to the system compared to the basic configurations of the OS;
- Development Environment used to implement the FW;
- All of the Credentials configured and set in the device.

9 CYBER SECURITY REQUIREMENTS BIDDING FORM

The Supplier/Bidder shall fill the following table, related to the Cybersecurity requirements described in Chapters 0, 6 and 7.

Table 1 – RTU Cyber Security - Level of compliance with the requirements					
N.	**Technical Specification**	**Mandatory**	**Yes**	**No**	**Remarks for any deviation**
6.1 HW Cyber security requirements - Architecture					
SR_HR_01		X			
SR_HR_01bis					
SR_HR_02		X			
SR_HR_03		X			
SR_HR_04		X			
SR_HR_05		X			
SR_HR_05bis					
SR_HR_06		X			
SR_HR_07					
SR_HR_08					
SR_HR_09		X			
SR_HR_10		X			
6.2 HW Cyber security requirements – HW solutions					
SR_HR_11		X			
SR_HR_11bis					
SR_HR_12		X			
SR_HR_13		X			
SR_HR_14					
SR_HR_15					
SR_HR_16		X			
SR_HR_17					
SR_HR_18		X			
SR_HR_19					
SR_HR_20					
7.1 RTU FW Cyber security requirements – Features of the OS					
SR_SW_01		X			
SR_SW_02		X			
SR_SW_03		X			
SR_SW_04		X			
7.2 RTU FW Cyber security requirements – MW components					
SR_SW_05		X			
SR_SW_06		X			
SR_SW_07		X			
SR_SW_08		X			
SR_SW_09					
SR_SW_10		X			
SR_SW_11		X			
7.3 RTU FW Cyber security requirements – Hardening					
SR_SW_12		X			
SR_SW_13		X			
SR_SW_14		X			
SR_SW_15		X			
SR_SW_16		X			
SR_SW_17		X			
7.5 RTU FW Cyber security requirements – Security Patching					
SR_SW_18		X			
SR_SW_19		X			
SR_SW_20		X			

Table 1 – RTU Cyber Security - Level of compliance with the requirements					
N.	**Technical Specification**	**Mandatory**	**Yes**	**No**	**Remarks for any deviation**
SR_SW_21		X			
SR_SW_22		X			
7.6 RTU FW Cyber security requirements – Users, Credentials and Certificates Management					
SR_SW_23		X			
SR_SW_24		X			
SR_SW_25		X			
SR_SW_26		X			
SR_SW_27		X			
SR_SW_28		X			
8.1 Detailed Technical Documentation to be provided					
SR_DC_01		X			

ANNEX

Annex 1 - SECURITY CONFIGURATIONS API

General information

API must adhere to the REST architectural constraints (RESTful APIs).
API must be available at the following URI:

```
https://<hostname>:<port>/securityConfigurations/v1
```

Security

Only HTTP/1.1 or higher protocol can be used.

Authentication

The API must support a custom HMAC authentication named XWS and defined here:

XWS authentication
Clients will provide HTTP Authentication and Date headers in the following format:
(XWS stands for ICS Web Services)

```
Authentication: XWS <username>:<digest>
Date: <timestamp>
```

<digest> = base64(hmac-sha256("<password>", "<verb> <pathname> <timestamp>"))

- <verb> is the http verb in uppercase (for example "GET")
- <pathname> is the pathname of the http request without the hostname, with a leading slash and with the eventual parameters (for example "/securityConfigurations/v1/ssh/service")
- <timestamp> is the number of seconds since Jan 01 1970. (UTC) of the request (for example "1557131233")
- <username> and <password> are system users credentials.

base64() means base64 encoding

hmac-sha256(<secret key>, <text to be hashed>) means hashing with the Hash-based message authentication code (HMAC) with digest algorithm SHA-256 and with secretKey <secret key> and text <text to be hashed>.

Strings must be UTF-8 encoded and the newline separator is LF (unix style)
HMAC and SHA-256 are defined in RFC4634 (https://tools.ietf.org/html/rfc4634)

Example
Assuming a request with the credential myUser1/myPassword1:

```
GET /securityConfigurations/v1/ssh/service
Authentication: XWS
myUser1:MTA1NGM2YmRiZDdjY2U0ZDg2ZWUxMmM2MjBmYzAwZjI4ZWYzMGIwZDQ4ZTMyNDgwZWY4ODcxMW
I5YWY2YTRlMQ==
Date: 1557131233
```

The digest is calculated according to this string:

```
hmac-sha256("myPassword1","GET /securityConfigurations/v1/ssh/service
1557131233")= "1054c6bdbd7cce4d86ee12c620fc00f28ef30b0d48e32480ef88711b9af6a4e1"
```

```
base64("1054c6bdbd7cce4d86ee12c620fc00f28ef30b0d48e32480ef88711b9af6a4e1") =
"MTA1NGM2YmRiZDdjY2U0ZDg2ZWUxMmM2MjBmYzAwZjI4ZWYzMGIwZDQ4ZTMyNDgwZWY4ODcxMWI5YWY2Y
TRlMQ=="
```

Authorization

The server must calculate the <digest> value of the request and, only if the calculated digest is equal to the provided digest, the API is authorized: otherwise a "401 Unauthorized" must be returned.

Even in presence of a correct digest, <timestamp> must be within a configurable timeframe, with a default value of 24 hours, compared to the actual time of the ICS.
No specific API is available for such a configuration, which must be handled via a firmware update, e.g. providing a specific file in a determined location, that must be declared at design time for the product, or including a specific field in an already existing configuration file.

The API server onboard on the RTU must check the associated privilege before initiating the API execution. All the APIs described here must be allowed ONLY to users of the "administrator" type.

Response
If the responses contains data must be declared the content type "application/json".

In the definition of the API, response is defined only if contain a JSON content.

Successful response must be 200 OK.

Unsuccessful response must use common http rules.

4XX Response must provide an error code and description in the JSON content.

Example:

Response OK
```
HTTP/1.1 200 OK
```

Response KO
```
HTTP/1.1 4XX
Content-Type: application/json;
{
  "errorCode": "<errorCode>",
  "errorDescripton": "<errorDescription>"
}
```

SSH service configuration

Activation and deactivation of the service ssh

Activation of the service ssh
Request
```
POST  https://<hostname>:<port>/securityConfigurations/v1/ssh/service/start
```

Deactivation of the service ssh
Request
```
POST  https://<hostname>:<port>/securityConfigurations/v1/ssh/service/stop
```

Get of the state of the service ssh
Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/ssh/service
{
  "state": "<active|inactive>"
}
```

Public Keys addiction and removal in "authorized keys"

Public key addiction

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/ssh/publicKeys
{
  "name": "<name>",
  "publicKey": "<publicKey>"
}
```

Public key deletion

Request

```
POST
https://<hostname>:<port>/securityConfigurations/v1/ssh/publicKeys/delete/<name>
```

Retrieve Public keys list

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/ssh/publicKeys
```

Response

```
[
  {
    "name": "<name 1>",
    "publicKey": "<public Key 1>"
  },
  {
    "name": "<name N>",
    "publicKey": "<public Key N>"
  }
]
```

SCP/SFTP Service Configuration

Identification of the users enabled for the service

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/scpSftp/users
```

Response

```
[
  {
    "name": "<name 1>",
    "access": "<readOnly|readWrite>"
  },
  {
    "name": "<name N>",
    "access": "<readOnly|readWrite>"
  }
]
```

Network services configuration

Retrieve network services configuration info

Request

```
GET   http://<hostname>:<port>/securityConfigurations/v1/networkServicesConfig
```

Response

```
{
  "hostname": "<hostname>",
  "ipAddresses": [
      {"ipAddress":"<ip address 1>","netMask":"<net mask 1>",
"defaultGateway":"<default gateway 1>", ”nic”:”<network interface card>” },
      {"ipAddress":"<ip address N>","netMask":"<net mask N>",
"defaultGateway":"<default gateway N>", ”nic”:”<network interface card>” }
  ],
  "dns": [
    "<ip dns server 1>",
    "<ip dns server N>”
  ],
  "ntpServer": "<ntp server>"
}
```

Update Network service configuration info

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/networkServicesConfig
{
  "hostname": "<hostname>",
  "ipAddresses": [
      {"ipAddress":"<ip address 1>","netMask":"<net mask 1>",
"defaultGateway":"<default gateway 1>", ”nic”:”<network interface card>” },
      {"ipAddress":"<ip address N>","netMask":"<net mask N>",
"defaultGateway":"<default gateway N>", ”nic”:”<network interface card>” }
  ],
  "dns": [
    "<ip dns server 1>",
    "<ip dns server N>”
  ],
  "ntpServer": "<ntp server>"
}
```

Firewall service

Activitation and deactivitation of the firewall service

Activation of the firewall service

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/firewall/service/start
```

Deactivation of the firewall service

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/firewall/service/stop
```

Retrieve the state of the firewall service

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/firewall/service
```

Response

```
{
  "state": "<active|inactive>"
}
```

bulk download or bulk upload of iptables rules configuration file

Retrieve the current used iptable rules

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/firewall/iptable
```

Response

```
{
  "iptable": "<iptable file content>"
}
```

Update the iptable rules

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/firewall/iptable
{
  "iptable": "<iptable file content>"
}
```

NOTE: iptable rules must be actualized in real time
NOTE2: in the iptable rules file, newline is the ASCII LINE-FEED character (“\n”) (unix/linux default)

Credentials/Keys Service

Bulk Download

Download bulk user settings

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/users/bulkSettings
```

Response

```
[
  {
    "etcPasswd": "</etc/passwd content file>",
    "etcShadow": "</etc/shadow content file>",
    "etcGroup": "</etc/group content file>"
  }
]
```

System users

Retrieve the list of the system users
Note: “group” is optional

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/users/system
```

Response

```
[
  {
    "name": "<name 1>",
    "role": "<users|administrator>",
    "group": "<group>"
  },
```

```
  {
    "name": "<name N>",
    "role": "<users|administrator>",
    "group": "<group>"
  }
]
```

Create a system user

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/users/system
{
  "name": "<name1>",
  "role": "<users|administrator>",
  "group": "<group>"
}
```

Change password of a system user

Request

```
POST
      https://<hostname>:<port>/securityConfigurations/v1/users/system/changePassw
ord
{
  "name": "<name>",
  "password": "<new password>"
}
```

Delete a system user

Request

```
POST
https://<hostname>:<port>/securityConfigurations/v1/users/system/delete/<name>
```

Web server users

Retrieve the list of the web server users

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/users/web
```

Response

```
[
  {
    "name": "<name 1>",
    "role": "<read|write|security>",
    "group": "<group>"
  },
  {
    "name": "<name N>",
    "role": "<read|write|security>",
    "group": "<group>"
  }
]
```

Create a web server user

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/users/web
{
  "name": "<name>",
  "role": "<read|write|security>"
}
```

Change password of a web server user
Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/users/web/changePassword
{
  "name": "<name>",
  "password": "<new password>"
}
```

Delete a web server user
Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/users/web/delete/<name>
```

Upload update and get of Cryptographic Keys and Digital Certificates

Cryptographic Keys
Retrieve the list of the cryptographic keys.

NOTE: tokenType can assume the following value:

- "private public x509" private/public x509 Certificates pair for the TLS communication service to the SCADA infrastructure; these Certificates must be different on each RTU;
- "X509" X509 Certificates pair for the https service; these Certificates must be different on each RTU;
- "Public Key" Public Key pool for the remote access via SSH; these Certificates must be different on each RTU;
- "Root CA public x509" Root-CA public x509 Certificate, common to all devices.

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/cryptographicKeys
```

Response

```
[
  {
    "name": "<name 1>",
    "key": "<cryptographic key 1>",
    "tokenType": "<token type 1>"
  },
  {
    "name": "<name N>",
    "key": "<cryptographic key N>",
    "tokenType": "<token type N>"
  }
]
```

Create a cryptographic key
Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/cryptographicKeys
{
  "name": "<name>",
  "key": "<cryptographic key>",
  "tokenType": "<token type>"
}
```

Delete a cryptographic key
Request

```
POST
https://<hostname>:<port>/securityConfigurations/v1/cryptographicKeys/delete/<name
>
```

Digital Certificates
Retrieve the list of the digital certificates. All certificates must be <cer> type

NOTE: tokenType can assume the following value:

- "private public x509" private/public x509 Certificates pair for the TLS communication service to the SCADA infrastructure; these Certificates must be different on each RTU;
- "X509" X509 Certificates pair for the https service; these Certificates must be different on each RTU;
- "Public Key" Public Key pool for the remote access via SSH; these Certificates must be different on each RTU;
- "Root CA public x509" Root-CA public x509 Certificate, common to all devices.

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/digitalCertificates
```

Response

```
[
  {
    "name": "<name 1>",
    "certificate": "<digital certificate 1>",
    "tokenType": "<token type 1>"
  },
  {
    "name": "<name N>",
    "certificate": "<digital certificate N>",
    "tokenType": "<token type N>"
  }
]
```

Upload a digital certificate

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/digitalCertificates
{
  "name": "<name>",
  "certificate": "<certificate file>"
    "tokenType": "<token type>"
}
```

Delete a digital certificate

Request

```
POST
      https://<hostname>:<port>/securityConfigurations/v1/digitalCertificates/dele
te/<name>
```

Syslog service

Configuration

Retrieve Syslog service configuration info

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/syslogService/config
```

Response

```
{
  "destinationServerIP": [
      {"ip":"<destination server IP 1>","port":"<port 1>", "protocol":"<protocol
1>"},
      {"ip":"<destination server IP N>","port":"<port N>", "protocol":"<protocol
N>"}
  ]
}
```

Log download

Retrieve the system Log.
Log must be in <zip> format and must contain all the log files present in the device.

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/syslogService/log
```

Response

```
{
  "log": "<log file>"
}
```

SysLog configuration download

Retrieve the Syslog configuration files.
“syslog” content must be in <zip> format and must contain all the syslog configuration files present in the device.

Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/syslogService/syslog
```

Response

```
{
  "syslog": "<syslog files>"
}
```

System functions

Device restart
Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/systemFunctions/restart
```

Updates

Upload security update or firmware file

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/update/upload
{
  "update": "<update file>"
}
```

Configuration of the repository for the package management system

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/update/repository
{
  "repositoryURL": "<repository URL>"
}
```

Configuration of the URL of the update filename

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/update/download
{
  "filenameURL": "<filename URL>"
}
```

Start the update
NOTE: filename it's without pathname

Request

```
POST  https://<hostname>:<port>/securityConfigurations/v1/update
{
  "filename": "<file name>",
  "type": "<upload|download|repository>",
  "scheduling": "DDMMYYYY-HH:MM"
}
```

In case of update in realtime, scheduling can be empty.
If type is "upload" the update will use the file previously uploaded
If type is "downwload" the update will be downloaded at the previously defined filename URL
If type is "repository" the update will be downloaded with the package management system at the previously defined repository URL

Information and characteristics of the device

Retrieve the information of the device
Request

```
GET   https://<hostname>:<port>/securityConfigurations/v1/deviceInformation
```

Response

```
{
  "Manufacturer": "<Manufacturer>",
  "ProductName": "<Product Name>",
  "Version": "<Version>",
  "SerialNumber": "<SerialNumber>",
  "FirmwareVersion": "<Firmware version>",
  "OperatingSystemVersion": "<Operating System version>",
  "Patching Level": "<PatchingLevel>",
  "Kernel version": "<KernelVersion>",
  "httpsServerVersion": "<https server version>",
  "ApplicationSoftwareVersion": "<Application Software version>"
}
```

NOTE: Serial number must be univocal for any device