Cybersecurity

Functionality

Cybersecurity Functionality

Copyright

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1. Introduction

In the past, substation networks were traditionally isolated, and the protocols and data formats used to transmit information between devices were often proprietary.

Because of this, the substation environment was very safe from cyberattacks. The terms used for this type of inherent security are:

- Security by isolation (if the substation network is not connected to the outside world, it cannot be accessed from the outside world).
- Security by obscurity (if data formats and protocols are proprietary, they are difficult for a third party to interpret).

The increasing sophistication of protection devices, together with the advancement of technology and the requirement for interoperability between manufacturers, have resulted in a standardization of networks and data exchange in substations. Today, devices within substations use standardized protocols for communications. In addition, substations can connect to open networks, such as the Internet and vast corporate networks, which employ standardized protocols for communications. This introduces a greater security risk, making the network vulnerable to cyberattacks, which could result in a higher electrical failure rate. Clearly there is now a need to secure communications and devices in substations.

Cybersecurity provides protection against unauthorized disclosure, transfer, modification or destruction of information or information systems, whether accidental or intentional. To achieve this, there are several security requirements:

- Confidentiality (avoid disclosing information to unauthorized individuals).
- Integrity (avoid modification of unauthorized information).
- Availability (avoid denial of service).
- Non-repudiation (avoid denying that an action was taken or asserting that it was done when it was not done).
- Traceability / Detection (monitoring and logging of activity to detect intrusions and analyze incidents).
- Identification and authentication (avoid access to the device from unauthorized users and applications).
- Authorization (limit the actions a user or application can execute according to their assigned permissions).

The cybersecurity measures implemented in the device to comply with these security requirements include the following:

- Disabling of communication ports and services.
- Role-based access control (RBAC).
- Remote user authentication (LDAP and RADIUS).
- Use of secure sockets.
- Credential management.
- Digital firmware securization.
- Cybersecurity logging.

ZIV cybersecurity solution has been implemented considering the leading cybersecurity standards and guidelines, such as IEC 62443, IEC 62351, IEEE 1686 and NERC CIP.

2. Communication Ports and Services

It is important to be able to guarantee that a device has active only those physical ports and logical ports (services) strictly necessary to comply with the required functionality. Therefore, it is essential to be able to enable or disable physical ports and services. These physical ports and services are managed through the configuration tool and the HMI of the device.

Depending on the model, the device may have the following physical ports:

- Local port.
- Remote port 1.
- Remote port 2.
- LAN 1.
- LAN 2.
- LAN 3.
- LAN 4.
- USB (pendrive).

By default, all ports on the device are disabled except the local port and the LAN ports. This means that remote ports 1 and 2 and USB (pendrive) will not be operational until they are enabled.

The user must initially enable the ports and services to be used, either by connecting to the configuration tool or by accessing the corresponding HMI menus.

When a port is disabled, it means that both transmission and reception are blocked.

The device has the following settings for enabling and disabling physical ports:

Port Enabling				
Configuration Tool	**HMI**	**Range**	**Step**	**Default**
Local Port	`Local Port`	Inactive / Active		Active
Remote Port 1	`Remote Port 1`	Inactive / Active		Inactive
Remote Port 2	`Remote Port 2`	Inactive / Active		Inactive
LAN 1 Port	`LAN 1 Port`	Inactive / Active		Active
LAN 2 Port	`LAN 2 Port`	Inactive / Active		Active
LAN 3 Port	`LAN 3 Port`	Inactive / Active		Active
LAN 4 Port	`LAN 4 Port`	Inactive / Active		Active
USB Port	`USB Port`	Inactive / Active		Inactive

If network redundancy (PRP, HSR, RSTP) is configured on a pair of ports (LAN1-2 or LAN3-4), enabling/disabling the second port (LAN2 or LAN4) has no effect on the device. Enabling/disabling the first port (LAN1 or LAN3) enables/disables both ports (LAN1-2 or LAN3-4).

The communication services supported by the device are the following:

- HTTP.
- HTTPS.
- Telnet.
- SSH.
- FTP.
- SFTP.
- MMS (IEC 61850).
- PROCOME Protocol.
- TCP/IP 1 Protocol (PROCOME, DNP3 or MODBUS).
- TCP/IP 2 Protocol (PROCOME, DNP3 or MODBUS).
- TCP/IP 3 Protocol (PROCOME, DNP3 or MODBUS).
- TCP/IP 4 Protocol (PROCOME, DNP3 or MODBUS).
- Syslog Client.

By default, all services are disabled except those which use secure sockets (HTTPS, SSH, SFTP and PROCOME). When a service is disabled, the socket associated with the service may appear as open (in listen state), but does not support connections, filtering is done by software.

The device has the following settings for enabling and disabling logical ports (services):

Service Enabling				
Configuration Tool	**HMI**	**Range**	**Step**	**Default**
HTTP	`HTTP`	Inactive / Active		Inactive
HTTPS	`HTTPS`	Inactive / Active		Active
SSH	`SSH`	Inactive / Active		Active
SFTP	`SFTP`	Inactive / Active		Active
Telnet	`Telnet`	Inactive / Active		Inactive
FTP	`FTP`	Inactive / Active		Inactive
MMS (IEC 61850)	`MMS (IEC 61850)`	Inactive / Active		Inactive
PROCOME	`PROCOME`	Inactive / Active		Active
TCP/IP Protocol 1	`TCP/IP Protocol 1`	Inactive / Active		Inactive
TCP/IP Protocol 2	`TCP/IP Protocol 2`	Inactive / Active		Inactive
TCP/IP Protocol 3	`TCP/IP Protocol 3`	Inactive / Active		Inactive
TCP/IP Protocol 4	`TCP/IP Protocol 4`	Inactive / Active		Inactive
Syslog Client	`Syslog Client`	Inactive / Active		Inactive

To disable other services such as synchronization by SNTP or by PTP (IEEE 1588), see **Chapter 1**, **Description and Start-Up**, **Synchronization Settings** chapter in the instruction manual of the device.

As an additional security measure, the device offers the possibility of changing the logical ports assigned to the services. The device has the following settings for changing service ports:

Service Ports				
Configuration Tool	**HMI**	**Range**	**Step**	**Default**
HTTP	HTTP	1 - 65535	1	80
HTTPS	HTTPS	1 - 65535	1	443
Telnet	Telnet	1 - 65535	1	23
FTP	FTP	1 - 65535	1	21
SSH/SFTP	SSH/SFTP	1 - 65535	1	22

The logical ports corresponding to the PROCOME, DNP3 and MODBUS protocols can be modified as described in **Chapter 1**, **Description and Start-Up**, **Communications Settings** chapter in the instruction manual of the device.

The logical port of the MMS protocol corresponding to IEC 61850 communications is 102 and cannot be modified.

All the services must have different port numbers assigned to them.

3. Role Based Access Control (RBAC)

The concept of **R**ole **B**ased **A**ccess **C**ontrol (RBAC) is used.

RBAC is not a new concept, many operating systems use it to control access to system resources.

RBAC is an alternative to the total access (superuser) or read-only model.

RBAC complies with the principle of minimum privilege security, which states that no user should be granted more rights than are necessary to perform that user's work.

RBAC allows an organization to separate the different capabilities and group them into special user accounts called roles for assignment to specific individuals according to the needs of their work.

RBAC carries implicitly the concepts of *users*, *roles* and *permissions* that will be developed next.

3.1 Users, Roles and Permissions

RBAC intrinsically carries the concepts of *users*, *roles* and *permissions*. A *user* has an associated *role*. Each *role* has one or more associated *permissions*.

Some examples that can help to understand the concept of roles and permissions would be the following:

Role	**Permissions**
VIEWER	Visualization
OPERATOR	Visualization, Command execution
ENGINEER	Visualization, Setting change, Configuration change
INSTALLER	Visualization, Setting change, Configuration change, Firmware change
RBACMNT	Visualization, User management
SECAUD	Visualization, Audit log
ADMINISTRATOR	All

The device allows the creation of up to **20 users**. Each user has associated a username, a password and a role.

The device allows the creation of up to **10 roles**. At the same time, each role can be associated with a series of permissions, in addition to a name and an identifier (used to univocally relate the user with a role).

Each access to the device by a user or application by remote access to console (Telnet / SSH), web (HTTP / HTTPS), file transfer (FTP / SFTP) and configuration tool (PROCOME), must be authenticated (login) through the introduction of user and password.

Access through other protocols, such as DNP3, MODBUS and MMS (IEC 61850), is still free of username and password.

The following permissions have been defined in the device:

- *Visualization* permission. With this permission it is possible to visualize states, measurements, settings, etc., as well as to collect oscillographs, events and fault reports, configuration files and log files (except for the cybersecurity event file).
- *Command execution* permission. With this permission it is possible to execute commands (e.g., date/time change).
- *Setting change* permission. With this permission it is possible to change the value of the settings, as well as to change the active table of settings and load the settings file.
- *Configuration change* permission. With this permission it is possible to load configuration files.
- *Firmware change* permission. With this permission it is possible to load the firmware to the device.
- *User management* permission. With this permission it is possible to manage users (add, delete or modify a user or a role), configure authentication methods and manage credentials.
- *Audit log* permission. With this permission it is possible to view/collect the cybersecurity event file.

By default, there is a single *admin* user with password *Passwd@02* associated with a role with identifier *-1*. This role is configured by default with all the permissions.

As a minimum, the visualization permission must be active. A user who does not have visualization permission will not have permission to do anything and therefore the device will reject his/her authentication attempts.

There must always be a user with user management permission.

To configure users and roles a user with *user management* permission must be used and next procedures can be used:

- From a website, via HTTP (not recommended because the data is not encrypted) or HTTPS, by accessing the ***Configuration→Access*** menu.
- From command line interface (CLI), via Telnet (not recommended because the data is not encrypted) or SSH.
- From the ***ZIV e-NET Tool***® configuration tool, using *LocalRBAC* and *DevicePreferences* files.

The following parameters are configured for each user:

Users					
Website	**CLI**	**LocalRBAC**	**Range**	**Step**	**Default**
Login	*main/access/user[x]/login*	name	char[32]		admin
Password	*main/access/user[x]/pwd*	pwd	char[32]		Passwd@02
Role Id	*main/access/user[x]/roleid*	role	-32768… 32767	1	-1

where *x* goes from 1 to 20, depending on the number of users that have been created.

Name. It supports up to 32 characters and its default value is *admin*. If it is empty, it means that the user does not exist. The username is unique (no two users with the same name can exist) and is case sensitive (i.e., "Admin" and "admin" are two different users).

Password. Users' passwords are strong. They allow a minimum length of 8 characters and a maximum of 32, are case sensitive and contain:

- At least one uppercase letter.
- At least one lowercase letter.
- At least one number.
- At least one non-alphanumeric character from ASCII 33 to ASCII 126 (both included). Blank space (ASCII character 32) is not a valid character to be used as part of passwords.

Passwords are never shown in clear and are stored encrypted on the device. Its default value is *Passwd@02*.

Role. An existing role identifier is configured.

The following parameters are configured for each role:

Roles					
Website	**CLI**	**DevicePreferences**	**Range**	**Step**	**Default**
Name	*main/access/role[y]/name*	name	char[32]		admin
Id	*main/access/role[y]/id*	id	-32768… 32767	1	-1
View	*main/access/role[y]/rview*	rightMsk (bit 0x01)	NO / YES		YES
Control	*main/access/role[y]/rctrl*	rightMsk (bit 0x02)	NO / YES		YES
Settings	*main/access/role[y]/rchset*	rightMsk (bit 0x04)	NO / YES		YES
Config	*main/access/role[y]/rchcfg*	rightMsk (bit 0x08)	NO / YES		YES
Firmware	*main/access/role[y]/rchfw*	rightMsk (bit 0x10)	NO / YES		YES
Users	*main/access/role[y]/rusrmgmt*	rightMsk (bit 0x20)	NO / YES		YES
Audit	*main/access/role[y]/raudit*	rightMsk (bit 0x40)	NO / YES		YES
Global role concurrency	*main/access/global_conc*	global_conc	NO / YES		YES
Priority	*main/access/role[y]/priority*	priority	1…10		5
Concurrency	*main/access/role[y]/concurrency*	concurrency	NO / YES		NO

where *y* goes from 1 to 10, depending on the number of roles that have been created.

Name. It supports up to 32 characters and its default value is *admin*. It cannot be empty. The role name is unique (there cannot be two roles with the same name) and is case sensitive (i.e., "Admin" and "admin" are two different roles).

This is the text used to check the role returned by remote authentication services (AAA) when they return a text.

Identifier. It uniquely identifies the role and is used on the user to indicate its associated role.

This is the number used to check the role returned by remote authentication services (AAA) when they return a number.

According to the IEC 62351-8 standard, negative numbers will be used for private roles, since this standard reserves positive numbers. The user can assign the desired value within the range -32768...32767, but by default the identifier -1 is assigned to comply with IEC 62351-8 standard.

Permissions. These are the different permissions that a role can have.

A role without associated permissions is not allowed to do anything and the device will reject authentication attempts from users who have that role associated with them.

The characters supported for usernames and roles are those indicated in the POSIX standard (Portable Operating System Interface for Unix) (IEEE Standard 1003.1 2008), with the exception that the hyphen character '-' cannot be used as the first character of the username or role:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 . _ -
```

The last three parameters are related to **Role Concurrency**:

Access Priority. This is a numeric setting indicating the access priority of that role. There will be as many priorities as roles (10). The lowest number (1) indicates the lowest priority and the highest number (10) indicates the highest priority. The default value is 5.

When the session limit has been reached (see **Session Concurrency** chapter), the access of a user with a higher priority role will cause the session of a user with a lower priority role to be automatically closed without any warning. The criterion will be to close the session of the user with a lower priority role and who has been connected for the longest time.

This priority will become clearer later when the different configuration possibilities are explained.

Concurrent Access. This is a Boolean setting that indicates whether concurrency is allowed for that role. Concurrency is when a role is authenticated more than once. The value **YES** indicates concurrency, **NO** indicates non-concurrency (a role can only be authenticated once). The default value will be **NO**.

Once there is an authenticated user with a certain non-concurrent role, authentication of another user with that role is prevented. The exceptions to this general rule are explained in detail below.

Apply non-concurrent roles together. If selected (value **YES**) it means that non-concurrency is additionally checked among all roles marked as non-concurrent, i.e., only one non-concurrent role can be authenticated at any given time. For example, if the roles ENGINEER, INSTALLER and SECADM are configured as non-concurrent and this setting is set to YES, only a user with role ENGINEER, INSTALLER or SECADM can be active (if a user with role ENGINEER is authenticated, a second user with role INSTALLER or SECADM will not be able to access). The default value is YES.

Always taking the criterion of facilitating access to users with priority roles (non-concurrent):

- In case there are authenticated users with non-priority (concurrent) roles, the user with the lowest priority role and who has been logged in the longest is logged out automatically and without warning.
- In case there is another authenticated user with priority (non-concurrent) role, the new user is asked if he/she wants to log out the user with priority role who is currently authenticated. This option is only available via website and HMI.

The following shows how the device behaves in the different cases of role concurrency configuration:

Case in which the *Apply Non-Concurrent Roles Together* parameter is set to YES

- If the accessing user has a concurrent role:
  - If there are available sessions in that service, the access is granted.
  - If there are no available sessions in that service, a check is made to see if there is an authenticated user with less priority role:
    - If there is, the session of the longest authenticated user is automatically closed, access is granted to the new user and the **Logout - session closed by other user** event is generated with the user that has just been expelled.
    - If not, access is denied and the event **Login failed - too many user sessions** is generated.
- If the accessing user has a non-concurrent role:
  - A check is made to see if there is any other authenticated user with non-concurrent role:
    - If there is and it is higher priority than the user trying to access, access is denied and the event **Login failed - user rejected due to role concurrency** is generated.
    - If there is and it is equal or lower priority than the user trying to access and the service is not a website or HMI, access is denied and the event **Login failed - user rejected due to role concurrency** is generated.
    - If there is and it is equal or lower priority than the user trying to access and the service is a website or HMI, a warning is displayed that there is another authenticated privileged user and is offered to the user the possibility of expelling it:
      - If the user accepts, the other user with non-concurrent role is logged out, access is granted to the new user and the **Logout - session closed by other user** event is generated with the user that was just expelled.
      - If the user cancels, access is denied and the event **Login failed - user rejected due to role concurrency** is generated.

- If there is not, a check is made to see if there are available sessions in that service:
  - If there are available sessions, access is granted.
  - If there are no available sessions, a check is made to see if any user with a lower priority role is authenticated:
    - If there is, the session of the user who has been logged in the longest is automatically closed, access is granted to the new user and the **Logout - session closed by other user** event is generated with the user who has just been expelled.
    - If not, access is denied and the event **Login failed - too many user sessions** is generated.

Case where the *Apply Non-Concurrent Roles Together* parameter is set to NO

- If the accessing user has a concurrent role:
  - If there are available sessions in that service, the access is granted.
  - If there are no available sessions in that service, a check is made to see if there is an authenticated user with less priority role:
    - If there is, the session of the user with the lowest priority role who has been logged in the longest is automatically closed, access is granted the new user and the **Logout - session closed by other user** event is generated with the user who has just been expelled.
    - If not, access is denied and the event **Login failed - too many user sessions** is generated.
- If the accessing user has a non-concurrent role:
  - A check is made to see if there is any other authenticated user with the same role:
    - If there is, and the service is not a website or HMI, access is denied and the event **Login failed - user rejected due to role concurrency** is generated.
    - If there is and the service is a website or HMI, a warning is displayed that there is another authenticated privileged user and is offered to the user the possibility of expelling it:
      - If the user accepts, the other user with the same role is logged out, access is granted to the new user and the **Logout - session closed by other user** event is generated with the user that was just expelled.
      - If the user cancels, access is denied and the event **Login failed - user rejected due to role concurrency** is generated.
    - If there is not, a check is made to see if there are available sessions in that service:
      - If there are available sessions, access is granted.
      - If there are no available sessions, a check is made to see if any user with a lower priority role is authenticated:
        - If there is, the user with the lowest priority role who has been logged in the longest is automatically logged out, access is granted to the new user and the **Logout - session closed by other user** event is generated with the user who has just been logged out.
        - If not, access is denied and the event **Login failed - too many user sessions** is generated.

3.2 User and Role Management

User and role management must be performed by a user with *user management* permission. It is managed from several interfaces:

- Web interface (HTTP / HTTPS).
- CLI Command Line Interface (Telnet / SSH).
- ***ZIV e-NET Tool***® Configuration Tool, using *LocalRBAC* and *DevicePreferences* files.

3.2.1 User and Role Management from the Web

In order to manage users from the web interface, some LAN ports and HTTP or HTTPS services need to be enabled. It is recommended to use the HTTPS connection instead of HTTP because the data is encrypted.

- The website of the device is accessed through a browser, entering the IP address configured in the LAN adapter of the device. Initially, a login web page appears where the user's name and password must be entered (*admin* is the default user).

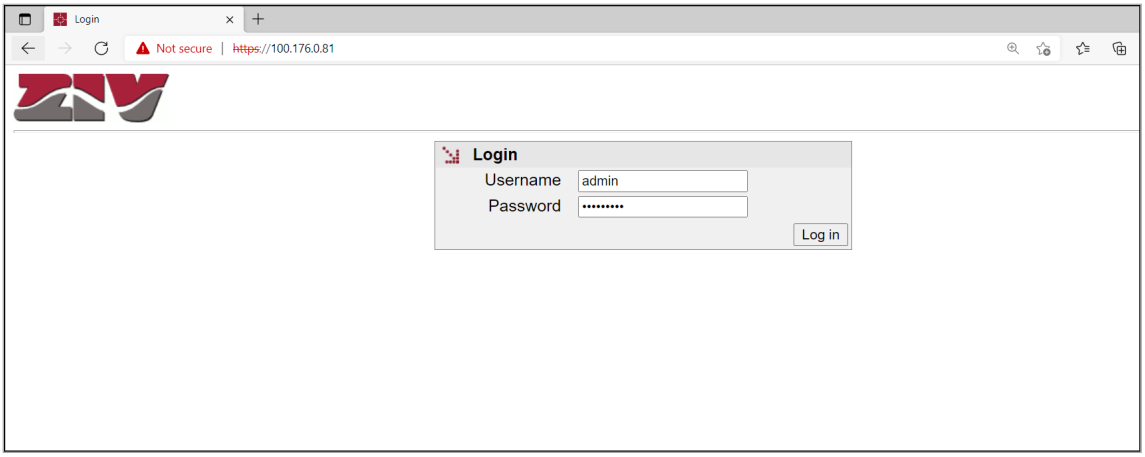


Figure 1 User Authentication on Website.

- The ***Configuration→Access*** menu is accessed to manage user configuration.

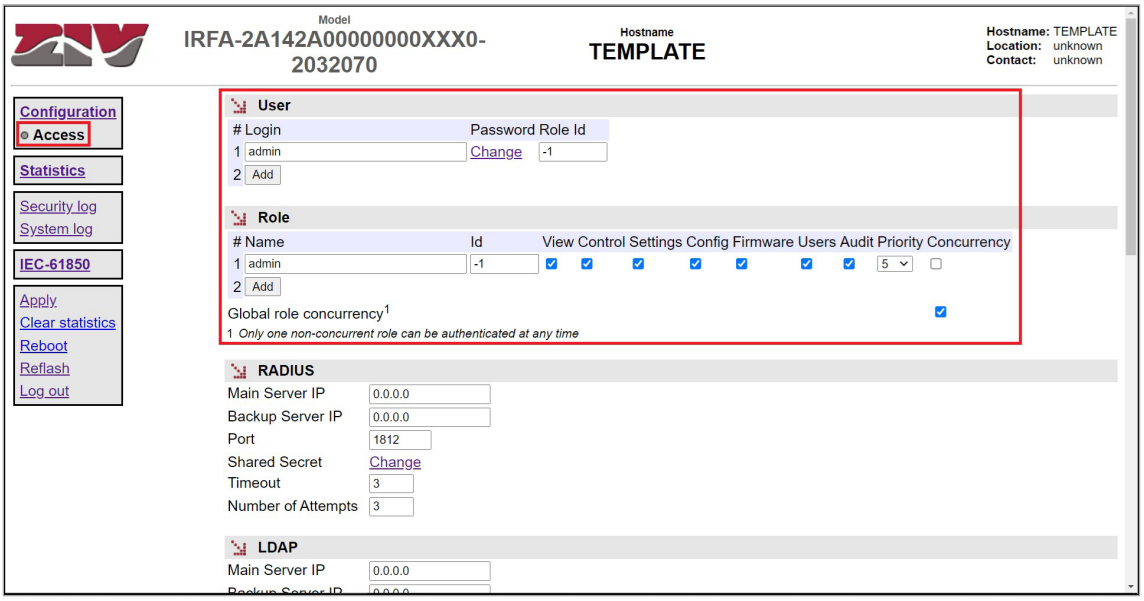


Figure 2 Users and Roles Configuration in Website.

To add a new role, click on the ***Add*** button in the ***Role*** section and enter a name (cannot be repeated), a role identifier (cannot be repeated), select the permissions assigned to the role, adjust the priority if desired and configure its concurrency. In the following example, the *guest* role is created with identifier '-2' and only *visualization* permission, priority 5 and concurrent.

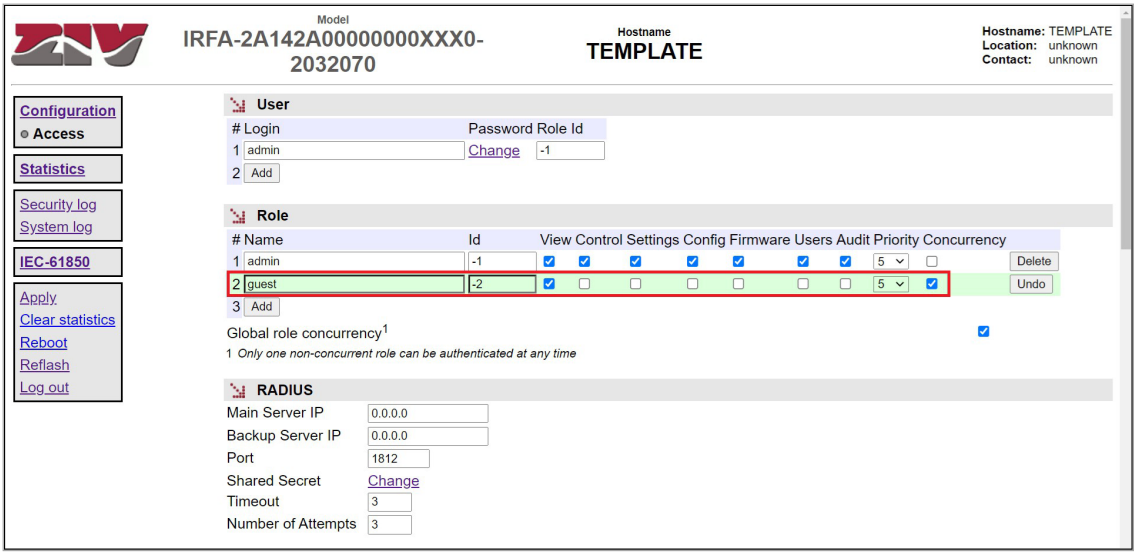


Figure 3 Adding a Role on a Website.

To add a new user, press the ***Add*** button in the ***User*** section and enter the username (cannot be repeated) and the associated role identifier. In the example, a *guest* user is created and assigned the role with identifier -2.

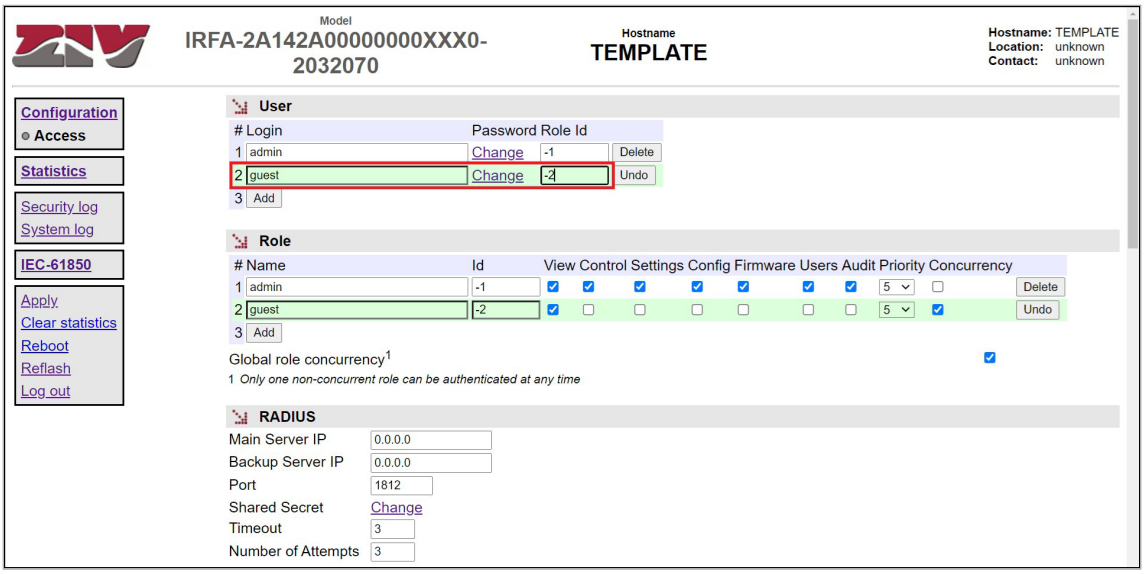


Figure 4 Adding a User on a Website.

You can then change the password of the new user by clicking on the ***Change*** link.

*Before changing your password, validate the above data by clicking on the **Send** button located at the bottom of the web page (scroll down to the bottom of the web page). Otherwise, the data entered will be lost.*

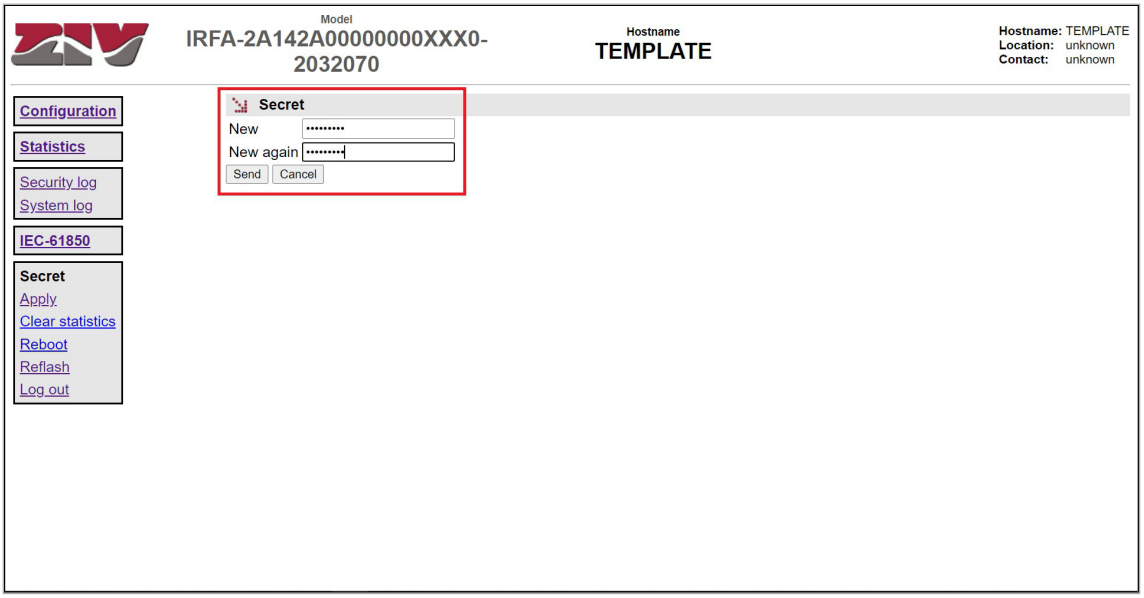


Figure 5 Changing User Password on Website.

On the password change web page, enter the new password twice and click ***Send***.

For new users to take effect, click on the ***Apply*** menu, click ***OK*** and wait for the message *'Configuration applied successfully'* to appear.

If you wish to temporarily disable a role, and therefore the users that include it, you can deactivate the *visualization* permission of a role.

To delete a user or role, press the corresponding ***Delete*** button and then press ***Send*** and ***Apply***. In the following example the user 'Dummy' and the role '-1234' will be deleted.

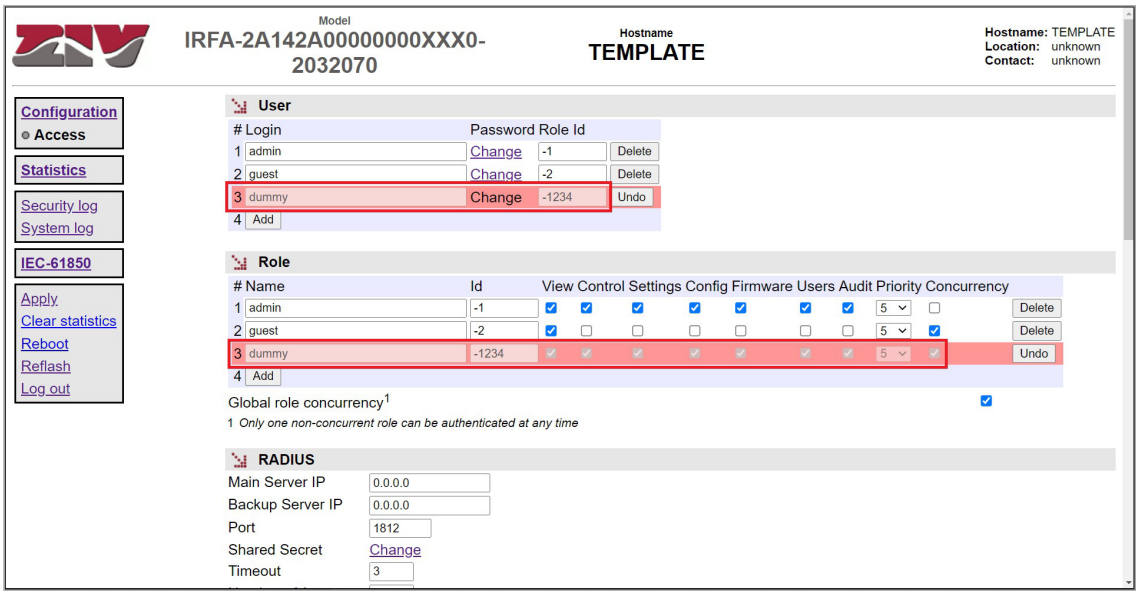


Figure 6 Delete User and Role on Website.

*Before changing your password, validate the above data by clicking on the **Send** button located at the bottom of the web page (scroll down to the bottom of the web page). Otherwise, the data entered will be lost.*

3.2.2 User and Role Management from CLI

To manage users and roles from the CLI command line interface, some LAN ports and Telnet or SSH services need to be enabled. It is recommended to use the SSH connection instead of Telnet because the data is encrypted.

The CLI of the device is accessed by launching the Telnet or SSH application from a host and entering the login of the user who has *user management* permission.

The CLI of the device is automatically accessed. You can see the parameters tree with the ***get*** command. The corresponding user settings are in the ***main/access*** node.

```
TEMPLATE /> get
/
   main/
      hostname       = TEMPLATE
      location       = unknown
      contact        = unknown
      product        = any
      model          = IRFA-2A142A00000000XXX0-2032070
      iedInfo        = ZIV   IRFA-2A142A00000000XXX0-2032070   0.10.0-0C03-00
      fw_version     = 0.10.0-0C03-00
      fw_reference   = unknown
      serialnumber   = 160538
      config_version = 00.00
      time           = 2021/06/29,10:25:11
      localtime      = 2021/06/29,10:25:11
      access/
         global_conc = on
         user[]/
            [user] login pwd      roleid
            ---------------------------
            1      admin ******** -1
         role[]/
            [role] name  id rview rctrl rchset rchcfg rchfw rusrmgmt raudit priority concurrency
            ------------------------------------------------------------------------------------
            1      admin -1 on    on    on     on     on    on       on     5        off
         radius/
            server1_ip = 0.0.0.0
            server2_ip = 0.0.0.0
            port       = 1812
            secret     = ********
            timeout    = 3
            attempts   = 3
```

Figure 7 User and Role Settings in CLI.

To add a new role, first add a new element in the ***role[]*** table with the command ***add/main/access/role***. A new role is created with index 2 in the table and with the default values.

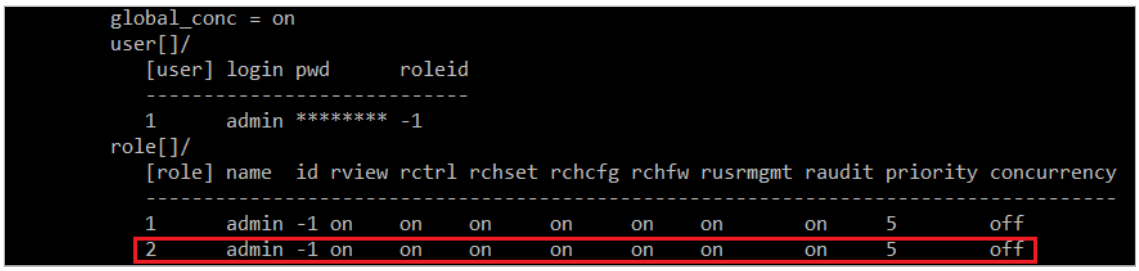


Figure 8 Adding a Role in CLI.

The new role is now modified with the desired name, identifier, permissions, priority and concurrency. It can be done interactively executing the command ***set main/access/role[2]*** where each one of the fields will be asked to complete the role, or directly executing the set of the field to modify, for example:

- ***set /main/access/role[2]/name guest***
- ***set /main/access/role[2]/id -2***
- ***set /main/access/role[2]/rview on***
- ***set /main/access/role[2]/rctrl off***
- ***set /main/access/role[2]/rchset off***
- ***set /main/access/role[2]/rchcfg off***
- ***set /main/access/role[2]/rchfw off***
- ***set /main/access/role[2]/rusrmgmt off***
- ***set /main/access/role[2]/raudit off***
- ***set /main/access/role[2]/priority 5***
- ***set /main/access/role[2]/concurrency on***

To add a new user, you must perform a similar process to the previous one but in the ***user[]*** table.

- ***add main/access/user***

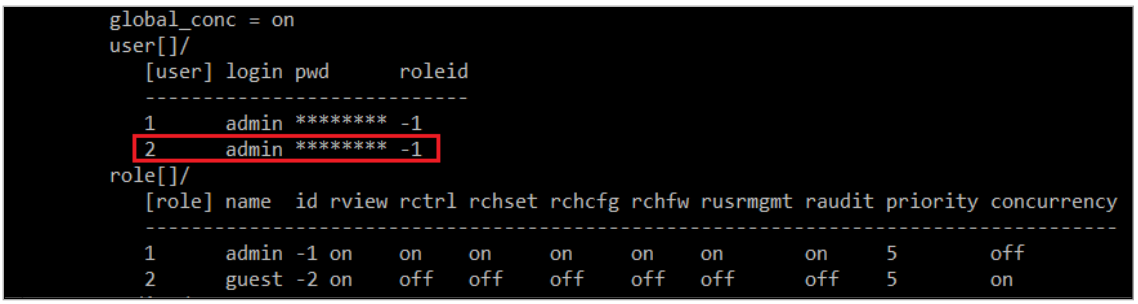


Figure 9 Adding a User in CLI.

- ***set main/access/user[2]/login guest***
- ***set main/access/user[2]/roleid -2***
- ***set main/access/user[2]/pwd*** (by entering the key value twice).

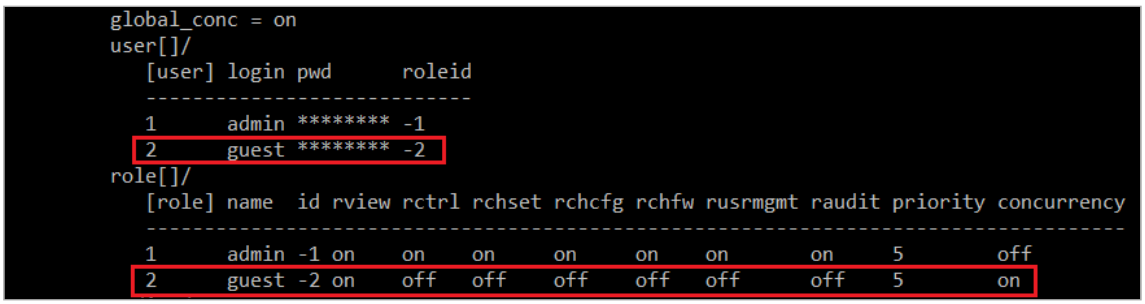


Figure 10 Modified User and Role in CLI.

Once the new user and/or role has been created, the data is saved with the ***save*** command and the changes in the device are made effective with the ***apply*** command.

3.2.3 User and Role Management from *ZIV e-NET Tool*®

To manage users and roles from ***ZIV e-NET Tool***® it is necessary that some of the LAN ports and the SSH and SFTP services are enabled.

A distinction is made between users and roles, as they are configured in different configuration files. The operation is as follows: first the roles are created (they are stored as part of the *DevicePreferences* file), then the local users are created (they are stored in the *LocalRBAC* file) and finally both files are transferred to the device. When collecting information from the device, only the complete *DevicePreferences* file can be collected (this includes the roles). Local users cannot be retrieved from the device, as the *LocalRBAC* file contains the device passwords, it is considered a file only for input to the device.

To configure local users, in the ***Navigation View*** go to the ***Preferences→Security→Local Users*** option.

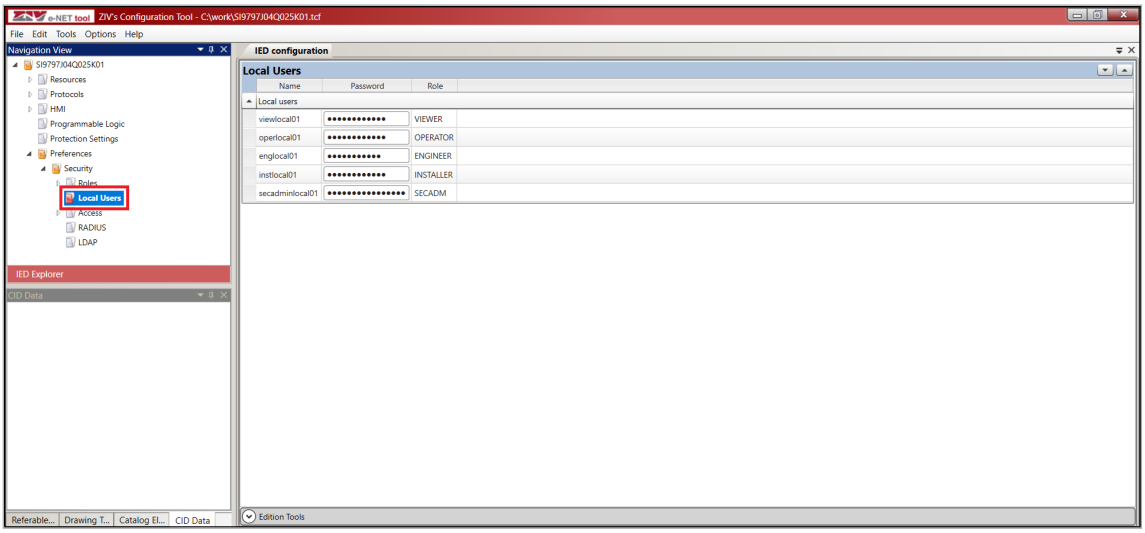


Figure 11 Configuration of Local Users with Configuration Tool.

Following the usual philosophy of the tool, local users can be added or deleted (using the ***Edition Tools*** tab), as well as modifying the current local users. For each user, its name, password and associated role must be configured (therefore the roles must be created previously).

By right-clicking on this option ***Preferences→Security→Local Users***, you can export the users to a *LocalRBAC* file or import them from a previously configured *LocalRBAC* file.

The *LocalRBAC* file is an XML file with the following format:

Node	Description	Child nodes	Attributes
RBACtemplate	Root node.	**Device**	-
device	Main node.	**LOCALUSRDB**	**name**: name of the device. **IP**: IP address of the device.
LOCALUSRDB	Local users.	**LocalUsr**	-
LocalUsr	Definition of a local user: name, password and assigned role.	-	**name**: name of the user. **pwd**: user password. **role**: numeric identifier of the role assigned to the user.

Example of a *LocalRBAC* file:

```
<?xml version="1.0" encoding="utf-8"?>
<RBACtemplate>
 <device name="TEMPLATE" IP="192.168.1.81">
  <LOCALUSRDB>
   <LocalUsr name="viewlocal" pwd="PwdView@01" role="7"/>
   <LocalUsr name="operlocal" pwd="PwdOper@01" role="8"/>
   <LocalUsr name="englocal" pwd="PwdEng@01" role="9"/>
   <LocalUsr name="instlocal" pwd="PwdInst@01" role="10"/>
   <LocalUsr name="secadmlocal" pwd="PwdSecadm@01" role="15"/>
  </LOCALUSRDB>
 </device>
</RBACtemplate>
```

To transfer local users to the device, go to the menu ***Tools→Device access***. In this window, check that SFTP + SSH is selected in ***Protocol***, configure the IP address of the device and port (it must match the SSH/SFTP port of the device) and enter the user's credentials (user and password). In the ***Transfer Files*** tab, check the ***LRBAC*** option and click ***Transfer*** to upload the *LocalRBAC* file to the device. The upload progress is indicated in the ***Information*** window. If the *LocalRBAC* file is successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if any error occurs in the validation of the *LocalRBAC* file by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

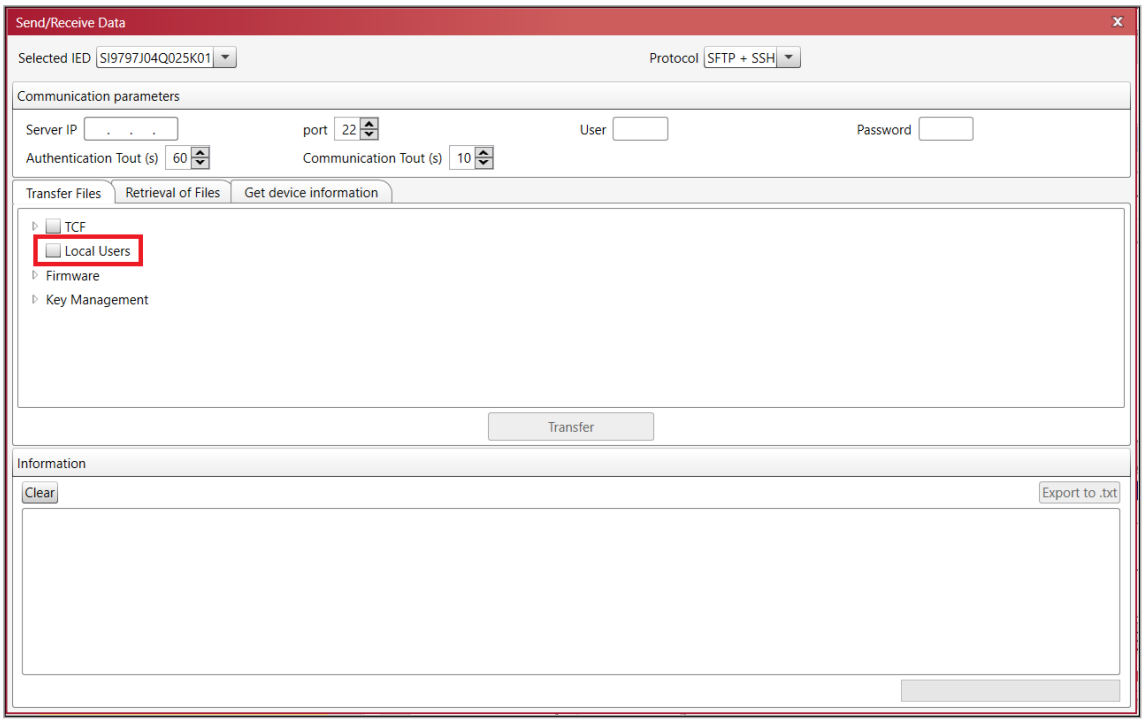


Figure 12 Sending Local Users with Configuration Tool.

To configure the roles, in the Navigation View go to the ***Preferences→Security→Roles*** option.

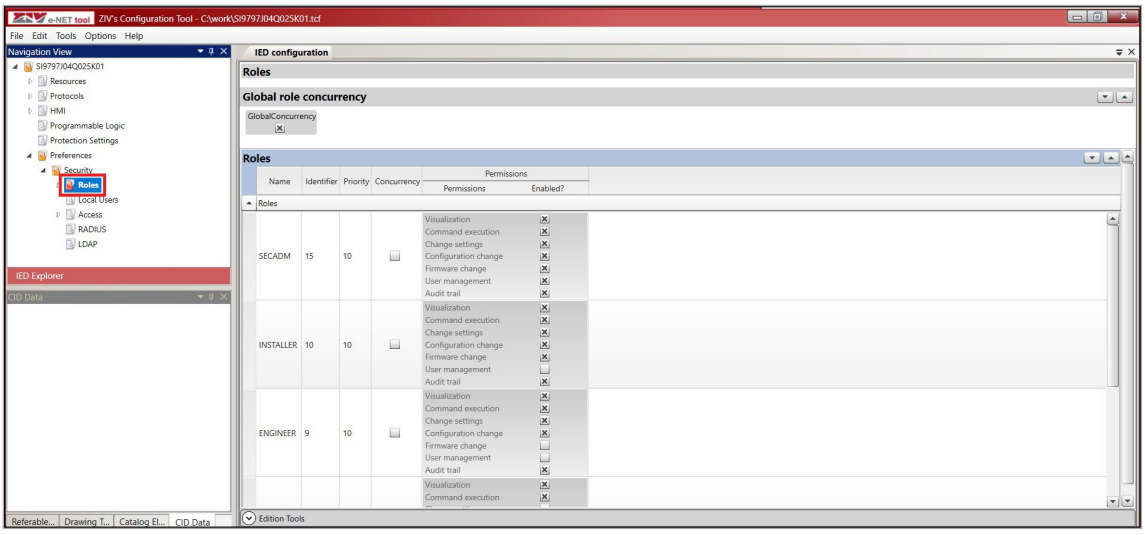


Figure 13 Role Configuration with Configuration Tool.

Following the usual philosophy of the tool, roles can be added or deleted (using the ***Edition Tools*** tab), as well as modifying the current roles. For each role, its name, identifier, priority, concurrency and associated permissions must be configured.

By right-clicking on this option ***Preferences→Security→Roles***, you can export the roles to a *DevicePreferences* file or import them from a previously configured *DevicePreferences* file.

The roles are part of the *DevicePreferences* file. This is an XML file whose roles section has the following format:

Node	Description	Child nodes	Attributes
device	Root node.	**ROLES**, RADIUS, LDAP, ACCESS	**tversion**: file version. **filetype**: file type ("DevicePreferences").
ROLES	Roles.	**Role**	**global_conc**: indicates whether only one or more non-concurrent roles can be authenticated at the same time.
Role	Role definition: name, identifier, permission mask, priority and concurrency.	-	**name**: name of the role. **id**: numerical identifier of the role. **rightMsk**: permissions mask, where: - 0x01: *Visualization* permission. - 0x02: *Command execution* permission. - 0x04: *Setting change* permission. - 0x08: *Configuration change* permission. - 0x10: *Firmware change* permission. - 0x20: *User management* permission. - 0x40: *Audit log* permission. **priority**: role priority. **concurrency**: indicates whether the role is concurrent or not.

Example of *DevicePreferences* file (only with roles section):

```
<?xml version="1.0" encoding="utf-8"?>
<device tversion="0.1" fileType="DevicePreferences">
  <ROLES global_conc="Yes">
    <Role name="SECADM" id="15" rightMsk="127" priority="10" concurrency="No" />
    <Role name="INSTALLER" id="10" rightMsk="95" priority="10" concurrency="No" />
    <Role name="ENGINEER" id="9" rightMsk="79" priority="10" concurrency="No" />
    <Role name="OPERATOR" id="8" rightMsk="3" priority="1" concurrency="Yes" />
    <Role name="VIEWER" id="7" rightMsk="1" priority="1" concurrency="Yes" />
  </ROLES>
</device>
```

To transfer the roles to the device, go to the menu ***Tools→Device access***. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the IP address and port data of the device (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Transfer Files*** tab, check the option ***TCF→Preferences*** (if you want to transfer the complete *DevicePreferences* file) or the option ***TCF→Preferences→Roles*** (if you want to transfer only the roles) and click ***Transfer*** to upload the complete *DevicePreferences* file or only the roles to the device. The upload progress is indicated in the ***Information*** window. If it has been successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if there is any error in the validation by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

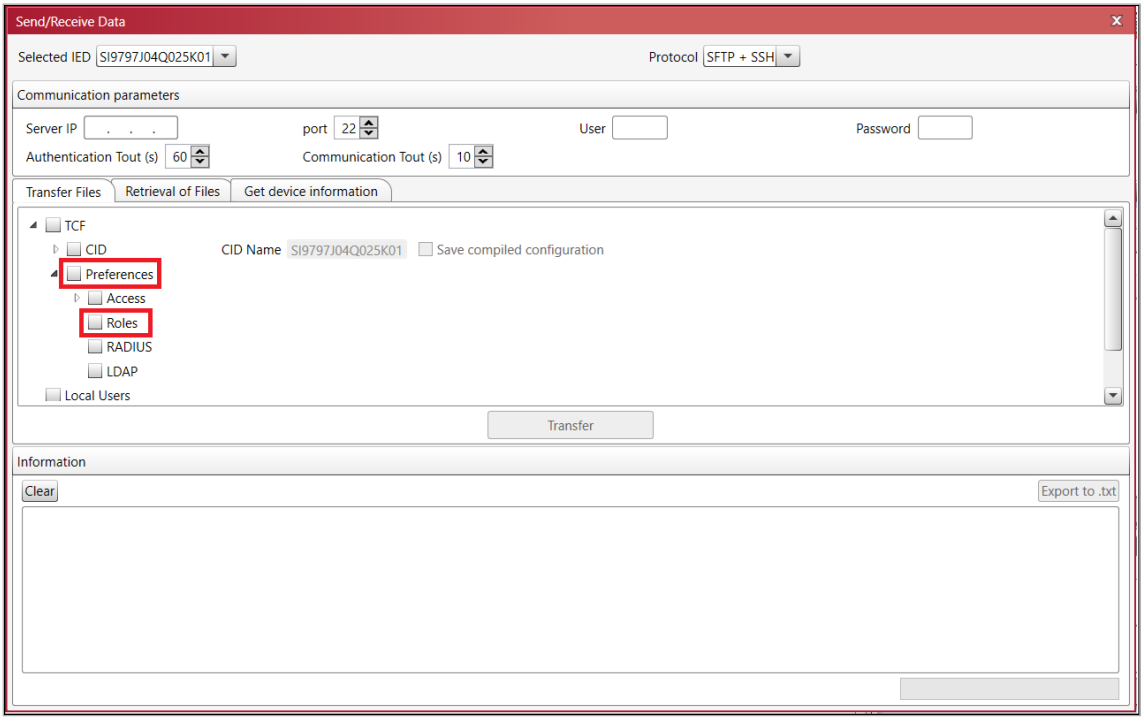


Figure 14 Sending Roles with Configuration Tool.

To collect the roles from the device, access the menu ***Tools→Device access***. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the IP address of the device and port (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Retrieval of Files*** tab, check the ***Preferences*** option (only the complete *DevicePreferences* file can be downloaded) and click ***Retrieval*** to download the *DevicePreferences* file. The download progress is displayed in the ***Information*** window. Once the file has been downloaded, the **Configuration downloaded** cybersecurity event is generated.

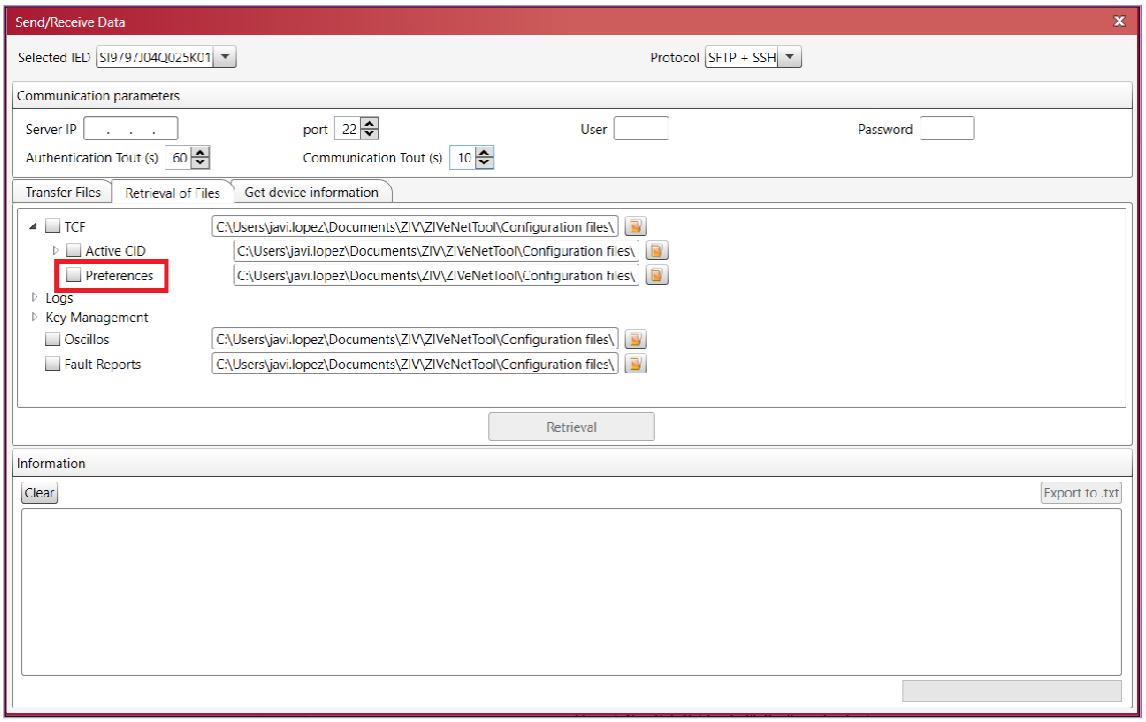


Figure 15 Role Retrieval with Configuration Tool.

3.3 Session Concurrency

In order to control the resources used by the device, it was decided to limit the number of authenticated concurrent sessions per type of service available in the device.

The device has the following settings, which can be modified from the configuration tool and from the HMI:

Concurrent sessions				
Configuration Tool	**HMI**	**Range**	**Step**	**Default**
The same user can occupy all available sessions	`Same usr all sess`	YES / NO		YES
Telnet	`Telnet`	1 – 5	1	2
FTP	`FTP`	1 – 5	1	2
HTTP	`HTTP`	1 – 5	1	2
SSH/SFTP	`SSH/SFTP`	1 – 10	1	4
HTTPS	`HTTPS`	1 – 5	1	2

Some clients wish to prevent the same user from being able to occupy all available sessions of a service, preventing others from authenticating. For this purpose, the setting ***The same user can occupy all available sessions*** is used as follows:

- If the setting takes the value **YES**, all sessions of a service can be occupied by the same user and an attempt to open a new session when all of them are occupied will mean that the session is not accepted, generating the **Login failed - too many user sessions** cybersecurity event.
- If the setting is set to **NO**, all sessions of a service may be occupied by the same user and an attempt to open a new session when all of them are occupied will mean that:
  - If the session is of another user, this is accepted and the oldest (longest running) session of the other user who had all sessions occupied is closed without any warning. The **Logout - session closed by other user** cybersecurity event is generated.
  - If the session is of the same user who has all other sessions occupied, this new connection is rejected and the **Login failed - too many user sessions** cybersecurity event is generated.

3.4 Local Access

Local accesses correspond to accesses via HMI and USB (pendrive). As these are face-to-face accesses, which take place within the physical security perimeter of the electrical installations (considered secure), it is generally enough for them to be protected by a numerical password. Even under these conditions, there are customers who require all accesses to be protected with a username and password, and the HMI is no exception.

The device is able to operate in both ways depending on the following setting, which can be modified from the website, command line (CLI) or ***ZIV e-NET Tool***® configuration tool (using the *DevicePreferences* file).

User required for HMI authentication					
Website	**CLI**	**DevicePreferences**	**Range**	**Step**	**Default**
User required for HMI authentication	*main/access/hmi/ usrauthhmi*	usrauthhmi	NO / YES		NO

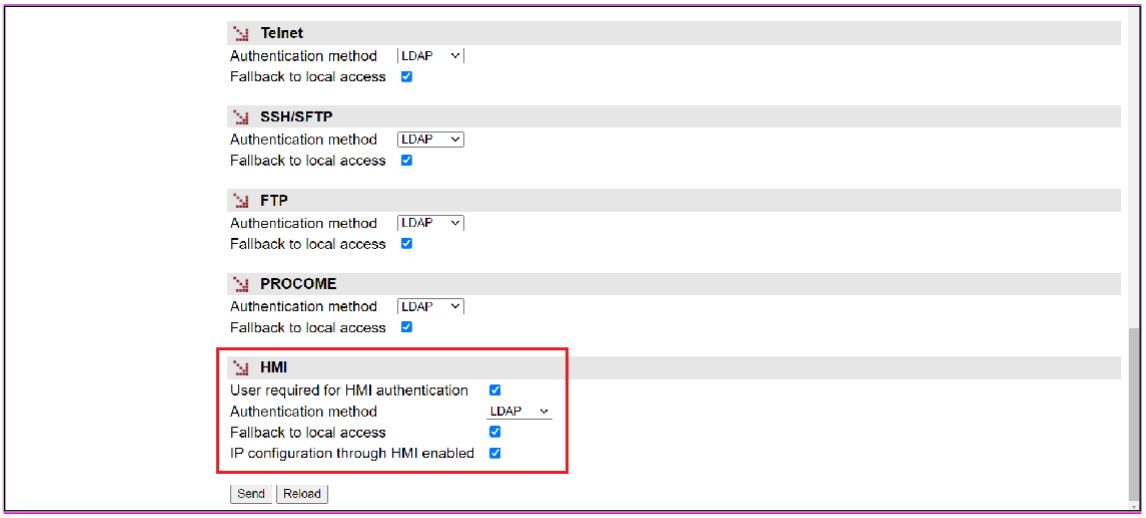


Figure 16 Setting to Require User for HMI Authentication on Website.

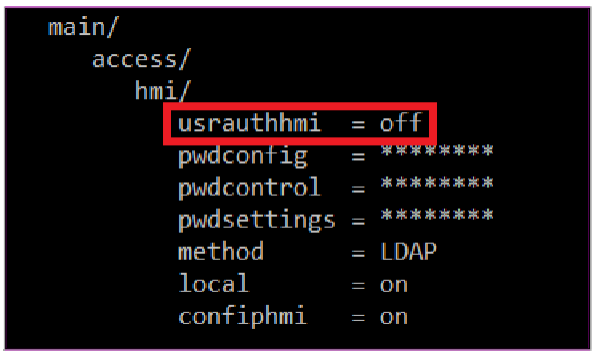


Figure 17 Setting to Require User for HMI Authentication on CLI.

To change this setting via website or CLI, it is necessary that, after modifying the setting, the ***Send*** and ***Apply*** buttons/commands are pressed/executed in website/CLI.

To change this setting by ***ZIV e-NET Tool***® it is necessary that some of the LAN ports and the SSH and SFTP services are enabled. In the ***Navigation View***, go to the ***Preferences→Security→Access→HMI*** option.

By right-clicking on this option ***Preferences→Security→Access→HMI***, you can export the HMI settings (the present one plus others explained in other sections) to a *DevicePreferences* file or import them from a previously configured *DevicePreferences* file.

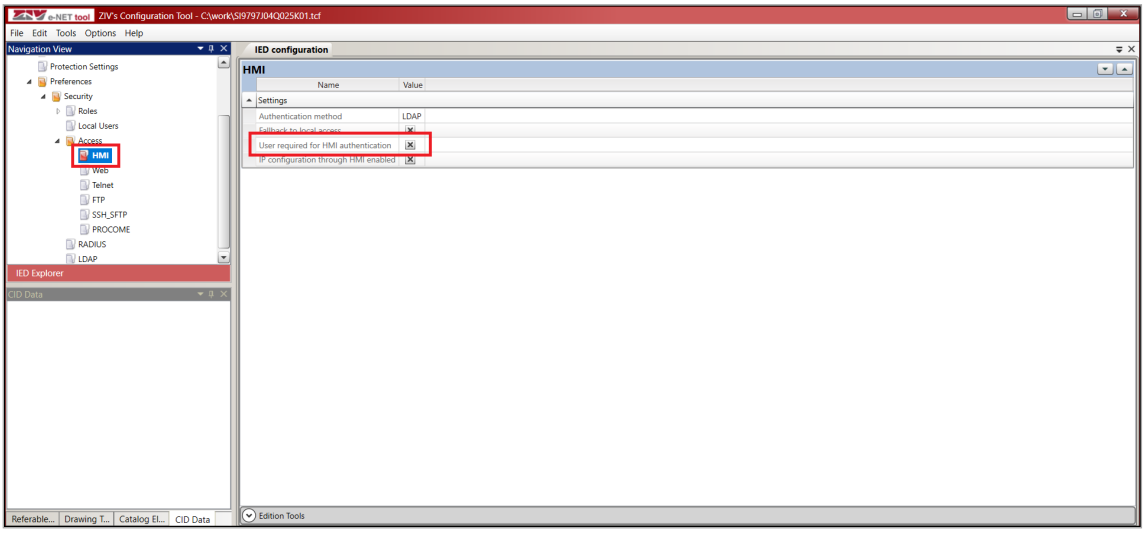


Figure 18 Setting to Require User for HMI Authentication on Configuration Tool.

This setting is part of the *DevicePreferences* file. It is an XML file and this setting is located inside the HMI section (highlighted):

Node	Description	Child nodes	Attributes
Device	Root node.	ROLES, RADIUS, LDAP, **ACCESS**	**tversion**: file version. **filetype**: type of file ("DevicePreferences").
ACCESS	Parameters to control access	**HMI**, WEB, TELNET, FTP, SSH_SFTP, PROCOME	-
HMI	HMI access parameters.	**Setting**	-
Setting	One **Setting** node for each HMI access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for HMI access (local, LDAP or RADIUS). - **local**: alternative use of local users for HMI access. - **userauthhmi**: name of the role. - **confiphmi**: it is allowed to change the IP via HMI. **value**: setting value.

Example of *DevicePreferences* file (with complete HMI section, the setting of require user for HMI authentication is highlighted):

```
<?xml version="1.0" encoding="utf-8"?>
<device tversion="0.1" fileType="DevicePreferences">
  <ACCESS>
    <HMI>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
      <Setting name="usrauthhmi" value="Yes" />
      <Setting name="confiphmi" value="Yes" />
    <HMI>
  </ACCESS>
</device>
```

To transfer the setting to the device, access the ***Tools→Device access*** menu. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the IP address data of the device and port (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Transfer Files*** tab, check the option ***TCF→Preferences*** (if you want to transfer the complete *DevicePreferences* file), the option ***TCF→Preferences→Access*** (if you want to transfer the complete access section) or the option ***TCF→Preferences→Access→HMI*** (if you want to transfer the complete HMI access section, this is the minimum section that can be sent) and press ***Transfer*** to upload the information to the device. The upload progress is indicated in the ***Information*** window. If it has been successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if there is any error in the validation by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

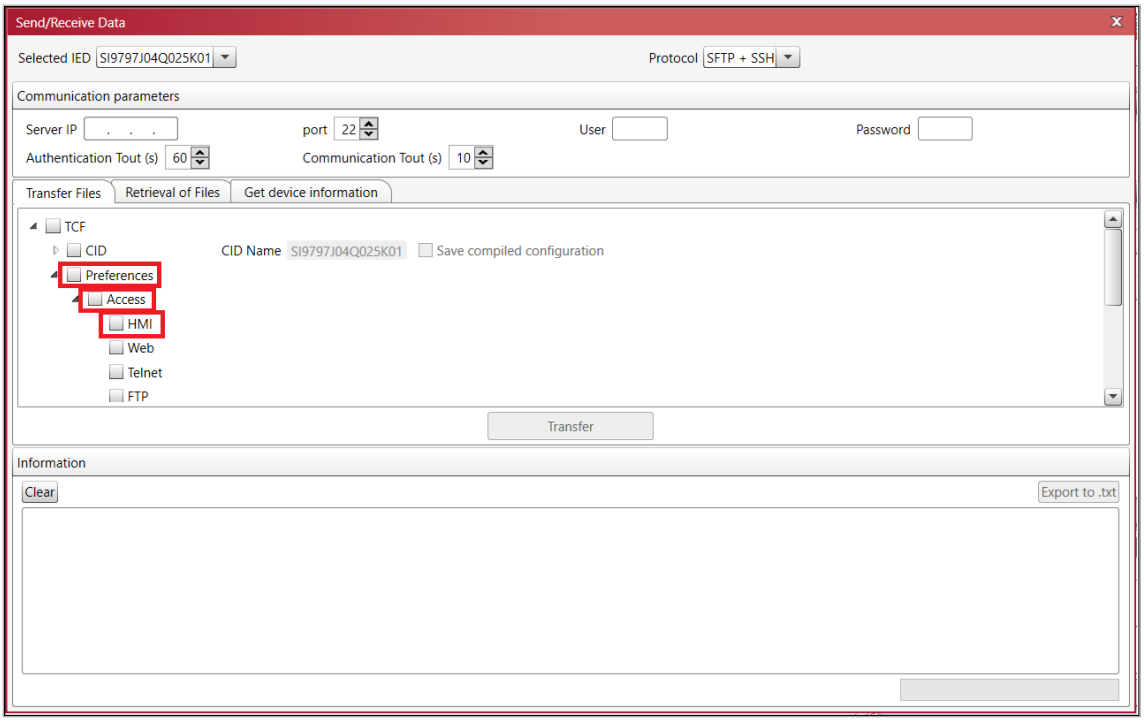


Figure 19 Sending Setting to Require User for HMI Authentication with Configuration Tool.

To collect this setting from the device, proceed as indicated in the **User and Role Management from ZIV e-NET Tool®** chapter for the roles, as it is only possible to collect the complete *DevicePreferences* file.

Depending on this setting, the device behaves as follows:

- The value **NO** will indicate that the HMI authentication is performed using only a 4-number password.
- The value **YES** indicates that the HMI authentication is performed using user and password.

Both authentication modes are explained below.

3.4.1 Password Authentication

The device has three passwords to access different levels of HMI management:

- *Configuration*: used to change parameters in the HMI configuration menu, as well as to download information to the USB pendrive.
- *Operations*: used to perform operations (commands) from HMI.
- *Settings*: used to change settings, including changing the active settings table.

The default value of the three passwords is 2140.

Two procedures can be used to change the value of these passwords:

- From HMI, by accessing the ***Configuration→Passwords*** menu.
- From the command line interface (CLI), through Telnet (not recommended because the data is not encrypted) or SSH.

The settings corresponding to the three local passwords are as follows:

Local Passwords				
CLI	**HMI**	**Range**	**Step**	**Default**
main/access/hmi/pwdconfig	Configuration	0 - 9999	1	2140
main/access/hmi/pwdcontrol	Operations	0 - 9999	1	2140
main/access/hmi/pwdsettings	Settings	0 - 9999	1	2140

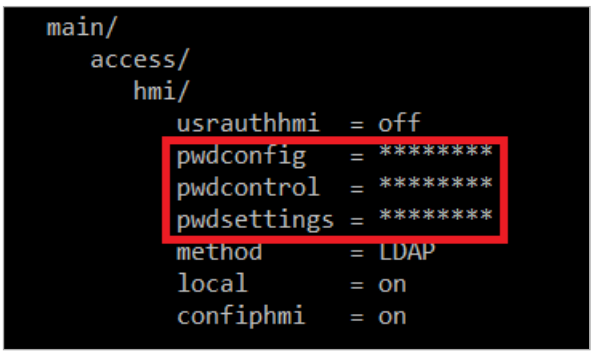


Figure 20 HMI Passwords in CLI.

The remote change of local passwords via CLI facilitates the integration of device into centralized cybersecurity systems, allowing them to meet one of their main requirements: to periodically update device passwords.

When entering the password value, leading zeros are not entered. If, for example, the password is '0012', the digits to be entered will be '12'.

The menu options that are available with each of the passwords are listed below:

Password	HMI Menu Option
No password	*Information* menu. *USB* menu (if USB port activated and pendrive inserted): *Eject USB*. Access to *CLR* key to reset LEDs, trips, etc. Access to the *commandable objects* of the graphic display (accessible with SEL and executable with the O and | buttons). Access to *push buttons P1 to P6*.
Configuration	*Configuration* menu. *USB* menu (if USB port activated and pendrive inserted): *Storage dump*.
Operations	*Operations* menu.
Settings	*Activate Group* menu. *Change Settings* menu.

3.4.2 User and Password Authentication

When username/password use is required, the entered users are validated against the users available on the device, i.e., remote users (when using remote authentication methods such as LDAPS or RADIUS) and/or local users (when not using remote authentication methods or, if using remote authentication methods, they are not available and you have configured to use local users alternatively).

This user and password authentication option is only available on devices with enhanced cybersecurity.

Only access to the following menus will be possible without authentication:

- *Information*:
  - *Relay information*.
  - *Status*.
  - *Measurements*.
  - *Records*.

To access the rest of the menus it will be necessary to authenticate with username and password.

To log in or log out a user by HMI, one of the following two ways must be used:

- When in the default screen, pressing ENTER will take you to the first menu option:
  - When there is no user authenticated by HMI, the first menu option is **Login** and when you enter that option the login screen explained below is displayed.
  - When there is a user authenticated by HMI, the first menu option is **Logout** and when that option is entered, the logout screen explained below is displayed.
- If there is no HMI authenticated user and an attempt is made to access an area that requires a user, the login screen is automatically displayed.

The **login** screen initially has the focus on the user line.

Initially the character 'a' is displayed, with ↑ the next character option ('b') is displayed, with ↓ the previous character option is displayed, with → you advance to the next character, with ← you go back to the previous character (the current character is deleted). Pressing and holding down the ↑ or ↓ arrow speeds up the search for the desired character.

The possible character options are as follows (shown in order of appearance):

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 . _ -
```

Press ENTER to enter the password; the focus moves to the password field.

The way of entering the characters of the password is equivalent to that of the user. The order of displaying characters is: lowercase, uppercase, numbers and special characters in the order of the ASCII table:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 ! " # $ % & ' ( ) * + , - . / @
: ; < = > ? [ \ ] ^ _ ` { | } ~
```

Each character of the confirmed password (when → is pressed) is graphically replaced by the '*' character to prevent the password from being seen.

Pressing ENTER completes the user and password entry and the device proceeds to authenticate the user. While the user is being authenticated, the message LOGIN IN PROGRESS is displayed.

If the user authentication is accepted by the device, an "L" character (login) is displayed on the HMI in the upper right corner. This allows to clearly identify when there is a user authenticated by HMI ("L" character present) or not ("L" character absent).

If the user authentication is rejected by the device, the message LOGIN FAILED / PRESS ANY KEY appears. When any key is pressed, the login screen is displayed again.

The **logout** screen requires confirmation: YES (←) confirms the logout and NO (→) or ESC cancels it and the user remains authenticated by HMI. While the user is being logged out, the message LOGOUT IN PROGRESS is displayed.

If ESC is pressed at any time within the login or logout screen, the login/logout window closes (in the case of logout the user would still be authenticated).

To log out (close user session) there are two options:

- A timeout after inactivity (no key presses, see **Automatic Logout** chapter): in this case the user returns to the main screen automatically.
- The user selects the option from the menu and confirms the logout.

A user authenticated by HMI counts as any other access for role concurrency (see **Users, Roles and Permissions** chapter). Therefore, a user by HMI can:

- Be expelled by another user (e.g., by web). In this case, the user is returned to the main screen, the HMI authenticated user indication is removed ("L" character in the display) and the **Logout - session closed by other user** cybersecurity event is generated with the user that has just been expelled.
- Expel another user. If the HMI user can expel a user and confirmation is required to do so (because it is about expelling another privileged user), after the login screen and confirmation that the user exists and is valid, a screen is presented to be able to expel the other user. This screen requires confirmation. If YES (🡐) is chosen, the other user will be expelled and this HMI user will be authenticated. If NO (🡒) is chosen or ESC is pressed, the HMI user authentication process is aborted and the **Login failed - user rejected due to role concurrency** cybersecurity event is generated.

The menu options that are available with each of the permissions are listed below:

Permission	HMI menu option
Visualization	*Information* menu. *USB* menu (if USB port activated and pendrive inserted): *Eject USB*.
Command execution	*USB* menu (if USB port activated and pendrive inserted): *Storage Dump*. *Operations* menu. Access to *CLR* key to reset LEDs, trips, etc. Access to the *commandable objects* of the graphic display (accessible with SEL and executable with the O and | buttons). Access to *push buttons P1 to P6*.
Setting change	*Configuration* menu. *Activate Group* menu. *Change Settings* menu.
Configuration change	-
Firmware change	-
User management	-
Audit log	-

When a NOT allowed operation/option is selected, a PERMISSION DENIED message is displayed for a few seconds.

Examples where the PERMISSION DENIED message is displayed:

- A user without *command execution* permission presses P1.
- A user without *command execution* permission presses CLR.
- A user without *configuration change* permission tries to access *the Activate Group* menu.

3.5 Automatic Logout

A user's session is automatically logged out by the device after a period of inactivity. The concept of inactivity varies depending on the access:

- Communications: absence of communications messages (keep-alive in case of sockets are not taken into account). It affects HTTP, HTTPS, Telnet, SSH, FTP, SFTP and PROCOME services.
- HMI: no user interaction (keystrokes).
- USB (pendrive): no user interaction (key presses causing file download operations).

The device has the following setting, which can be modified from the configuration tool and from the HMI:

Configuration Tool	HMI	Range	Step	Default
Inactivity Timeout	`Inactivity Timeout`	1 – 60 min	1	5

4. Remote User Authentication

As described above, there is the possibility of local user authentication in the device. A series of local users and a series of roles with permissions are defined and a role is associated to each user. When a user authenticates, the device validates it against these local users.

Clients do not usually use local authentication on the device. They usually use centralized repositories where they configure their users with their associated roles. The device must authenticate users against these remote repositories using different standards. These repositories will return to the device, in case of positive user/password authentication, the user's role. In this way, the device will check internally if such role is defined in the device, the permissions assigned to such role in the device and will allow such user to perform the actions associated to such permissions.

The definition of users, roles and permissions using remote authentication is deployed as follows:

- Outside the device, in the remote repository, users are defined and their roles are associated with them.
- On the device, the roles and their associated permissions are defined (and additionally the local users with their roles).

This remote authentication option is only available on devices with enhanced cybersecurity.

As a general rule, the device can be configured to use remote or local authentication (local users of the device) and, in case of remote authentication, which remote authentication method to use (RADIUS or LDAP).

In addition, in case of using remote authentication, you can configure whether you want to use local users (local authentication) as an alternative source of authentication in case of impossibility to access the configured remote authentication repositories. If this option is chosen, the user/password authentication will be done locally (taking into account the local users defined in the device) when the requests to the remote repositories fail.

The following settings are defined for these two functionalities:

Authentication					
Website	**CLI**	**DevicePreferences**	**Range**	**Step**	**Default**
Authentication method	*main/access/xxxx/method*	method	local RADIUS LDAP		Local
Fallback to local access	*main/access/xxxx/local*	local	on (Yes) / off (No)		on (Yes)

where *xxxx* is the service name: web, telnet, ssh_sftp, ftp, procome and hmi.

These settings can be adjusted individually for each of the services requiring authentication: web access, telnet access, SSH/SFTP access, PROCOME access and HMI access.

*If all **Fallback to local access** settings are set to **No** and the authentication methods for all services are set to a value other than **local**, the device will not be able to authenticate if the device is taken to the factory for repair or diagnostics, as it will not be possible to connect to the remote authentication servers and there will be no possibility of alternative use of local users.*

*It is therefore recommended that at least the HMI access be configured with **Fallback to local access** set to **Yes**.*

The device uses two remote authentication methods which are explained in detail below: RADIUS and LDAP.

4.1 RADIUS (Remote Authentication Dial In User Service)

The device acts as a RADIUS client to authenticate the user/password against up to two RADIUS servers (primary and secondary).

The device uses the RADIUS UDP client version specified by RFC 2865. The password is encrypted, but not the user. If more security is desired for RADIUS, alternative bump-in-the-wire methods, such as the use of VPN, should be employed.

For role notification by the server, the device processes the *Management-Privilege-Level* attribute (Type=136), explained in RFC 5607. This attribute contains the following fields:

- Type (1 byte): takes the value 136.
- Length (1 byte): takes the value 6.
- Value (4 bytes): this is an integer indicating the user's role. As private roles not predefined in IEC 62351-8 must take a negative value, this value is interpreted by the device as a signed integer (INT32).

When the device receives an Access-Accept message, it takes the role of this *Management-Privilege-Level* attribute and checks it against the list of roles it has defined:

- If it does not match any role or matches any role, but this one does not have *Visualization* permission, it shall refuse to authenticate the user.
- Otherwise, it must accept the user's authentication, take note of the permissions associated with that role and allow the user to execute only the operations associated with the permissions available to that role.

The RADIUS client settings are as follows:

RADIUS					
Website	**CLI**	**DevicePreferences**	**Range**	**Step**	**Default**
Main Server IP	*main/access/radius/server1_ip*	server1_ip	XXX.XXX.XXX.XXX		0.0.0.0
Backup Server IP	*main/access/radius/server2_ip*	server2_ip	XXX.XXX.XXX.XXX		0.0.0.0
Port	*main/access/radius/port*	port	1 – 65535	1	1812
Shared Secret	*main/access/radius/secret*	secret	8 – 128 chars		ziv12345
Timeout	*main/access/radius/timeout*	timeout	1 – 30 s	1	3
Number of Attempts	*main/access/radius/attempts*	attempts	1 – 10	1	3

The meaning of the settings is as follows:

- *Main Server IP*: IP address of the primary RADIUS server. If **0.0.0.0** (default value) is set, it means that this server will not be used.
- *Backup Server IP*: IP address of the secondary RADIUS server. If **0.0.0.0** (default value) is configured, it means that this server will not be used.
- *Port*: Port on which the RADIUS servers operate. The default value is **1812** (UDP), which is the one reserved by RFC 2865.
- *Shared Secret*: Shared secret key between client and RADIUS server for password encryption. The default value is **ziv12345**. The length will range from 8 to 128 characters.
- *Timeout*: Maximum waiting time (in seconds) for obtaining the response from the RADIUS server. The default value is **3 seconds**.
- *Number of Attempts*: Number of attempts (not retries) to authenticate against a server before deciding that the server could not be accessed. The default value is **3 attempts**.

If both IPs are set to 0.0.0.0, it means that the RADIUS is disabled and therefore tries to validate the incoming user against local users, even if the *Authentication Method* setting is set to **RADIUS** and the *Fallback to Local Access* setting is set to **Off**.

For all other cases, upon receiving a login from a user, the device executes the following process if the *Authentication Method* setting is set to **RADIUS**:

1) Send an *Access-Request* to the primary server (or go to step 2 if the primary's IP is 0.0.0.0):

   1.1) If it replies, it no longer tries the secondary:

   - If the response is *Access-Reject*, the user is rejected and the process is terminated.
   - If the answer is *Access-Accept*:
     - And the role does not exist on the device, the user is rejected and the process is terminated.
     - And the role does not have *visualization* permission assigned, the user is rejected and the process is terminated.
     - Otherwise, the user is accepted and the permissions associated to the role received from the server are granted and the process ends.

   1.2) If the user does not answer in the *Timeout* setting:

   - If it has tried less than *Number of Attempts* times, go to step 1.
   - If it has tried *Number of Attempts* times, go to step 2 and the **RADIUS server not accessible** cybersecurity event is generated.

2) Repeat step 1, but now with the secondary server.

   If the IP of the secondary is 0.0.0.0, continue to step 3.

3) If the *Fallback to Local Access* setting is set to:

   - **On**, it attempts to validate the user and password against the local users of the device and terminates the process.
   - **Off**, the user is rejected and the process is terminated.

4.2 LDAP (Lightweight Directory Access Protocol)

The device acts as an LDAP client to authenticate the user/password against up to two LDAP servers (primary and secondary).

Authentication is performed by validating the user's credentials on the LDAP server. The LDAP client used is OpenLDAP 2.4.50. Simple synchronous authentication is supported according to the following summary procedure:

- Anonymous bind to LDAP server using anonymous authentication ("*Anonymous Authentication Mechanism of Simple Bind*" according to RFC 4513).
- Verification by the LDAP server of the username and password entered when opening a session via web, file transfer (FTP/SFTP), console access (Telnet/SSH), configuration tool (ICT) or other means, verifying that the username/password pair is correct.
- Reading the attribute defining the user's role (defined below), to apply the appropriate permissions to the user before granting access to the device by limiting, according to the role, the actions that the user can perform on the device.

The LDAP client settings are as follows:

LDAP					
Website	**CLI**	**DevicePreferences**	**Range**	**Step**	**Default**
Main Server IP	*main/access/ldap/ server1_ip*	IpAuth	XXX.XXX.XXX.XXX		0.0.0.0
Backup Server IP	*main/access/ldap/ server2_ip*	IpAuth2	XXX.XXX.XXX.XXX		0.0.0.0
Port	*main/access/ldap/ port*	Port	1 – 65535	1	389
Timeout	*main/access/ldap/ timeout*	tRetryAuth	1 – 600 s	1 s	3 s
Number of Attempts	*main/access/ldap/ attempts*	NRetryAuth	1 – 10	1	3
Base DN	*main/access/ldap/ base_dn*	base_dn		0-100 chars	ou=users,dc= ziv,dc=es
Search Filter	*main/access/ldap/ search_filter*	search_filter		0-100 chars	objectClass= equiposSAS
Role Attribute	*main/access/ldap/ role_attribute*	role_attribute		0-60 chars	perfil-usu
Version	*main/access/ldap/ version*	version	2 – 3	1	3
Secure Connection	*main/access/ldap/ secure_cx*	secure_cx	Yes / No		Yes
Secure Protocol	*main/access/ldap/ secure_protocol*	secure_protocol	LDAPS / StartTLS		StartTLS

The meaning of the settings is as follows:

- *Main Server IP*: IP address of the primary LDAP server. If **0.0.0.0** (default value) is set, it means that this server will not be used.
- *Backup Server IP*: IP address of the secondary LDAP server. If **0.0.0.0** (default value) is configured, it means that this server will not be used.
- *Port*: Port on which the LDAP servers operate. The default value is **389** (TCP), which is the one reserved according to RFC 4511.
- *Timeout*: Maximum waiting time (in seconds) for obtaining the response from the LDAP server. The default value is **3 seconds**.
- *Number of Attempts*: Number of attempts (not retries) to authenticate against a server before deciding that the server could not be accessed. The default value is **3 attempts**.
- *Base DN*: organizational unit (OU) in which users are stored in the LDAP server. The default value is **ou=users,dc=ziv,dc=en**. The length will range from 0 to 100 characters.
- *Search Filter*: additional search filter for the user, obviously together with the username (this will be done by the firmware by default). The default value is **objectClass=equiposSAS**. If no such filter is found, the user will not be validated. If an empty value is set, it means that the device will not perform this filter check to validate the user. The length will be between 0 and 100 characters.

- *Role Attribute*: name of the attribute that will contain the role of the user belonging to *Base DN* and that complies with the *Search Filter*. The default value is **perfil-usu**. If this attribute is not found on the server, the user will not be validated. If an empty value is set, the user will never be validated. The length will range from 0 to 60 characters.
- *Version*: version of LDAP to be used. The default value is **3** (LDAPv3).
- *Secure Connection*: indicates whether the connection to the LDAP server is secure or not. By default, a secure connection is used (**Yes**).
- *Secure Protocol*: secure protocol used if *Secure Connection* = **Yes**. The options are **LDAPS** or **StartTLS**. As the use of StartTLS is reserved for LDAPv3, if *Secure Connection* = **Yes** and *Version* = **2** are chosen, the LDAPS secure protocol (LDAP over SSL/TLS) will always be used regardless of the *Secure Protocol* parameter. The default value is **StartTLS**.

If both IPs are set to 0.0.0.0, it means that LDAP is disabled and therefore tries to validate the incoming user against local users, even if the *Authentication Method* setting is set to **LDAP** and the *Fallback to Local Access* setting is set to **Off**.

For all other cases, upon receiving a login from a user, the device, if the *Authentication Method* setting is set to **LDAP**, executes the following process:

1) If the device is able to connect to the primary server, continue to step 1.1.

   If it is not able to connect to the primary server, continue to step 1.2.

   If the IP of the primary is 0.0.0.0, continue to step 2.

   1.1) Send a *bindRequest* to the primary server indicating simple authentication, but without username and password (both zero length). This is known as *Anonymous authentication*.

   - If the response is *bindResponse* success, continue to step 1.1.1.
   - Otherwise, the user's authentication is rejected and the process is terminated.

   1.1.1) Send a *searchRequest* to search for the user's role, consisting of:

   - *baseObject* field equal to the *Base DN* setting.
   - *Filter* field that is an AND of:
     - *Search Filter* setting (e.g., **objectClass=equiposSAS**).
     - *uid* equal to the name of the user to be authenticated (e.g., *uid=admin*).
   - *attributes* field containing a single attribute whose *AttributeDescription* is the value of the *Role Attribute* setting (e.g., *perfil-usu*).

   It must receive from the LDAP server as many *searchResEntry* as entries in the *Base DN* directory and match the indicated search criteria (theoretically one, if it hypothetically responds with more, the device will keep the first one), ending in a *searchResDone*. This means that if only *searchResDone* is returned, there are no entries in the directory.

- If the response is at least one *searchResEntry*, continue to step 1.1.1.1.
- Otherwise, the user's authentication is rejected and the process is terminated.

1.1.1.1) It sends a *bindRequest* to the primary server indicating simple authentication, with username and password (in clear). This is known as the *Name/Password Authentication Mechanism of Simple Bind*.

- If the response is *bindResponse* success, continue to step 1.1.1.1.1.
- Otherwise, the user authentication is rejected and the process is terminated.

1.1.1.1.1) The device takes the first *SearchResEntry* received in step 1.1.1.

The device checks the value of the role received from the server against the list of roles it has defined. The role attribute can contain a text string or an integer. Therefore, to be versatile, the device compares the value of that variable with (in this order):

- The role *name* field. It must match exactly (case sensitive).
- The role *identifier* field.

First the names are checked and then the role identifiers. The first role of the device that matches this search will be the chosen role. For better understanding, the following example of device roles is included:

# (role)	*Name*	*identifier*
1	VIEWER	-7
2	OPERATOR	-5
3	-7	3

If the role attribute received by LDAP is "-7", the chosen role will be #3, because the *name* field is checked first.

- If it does not match any role or it matches one, but it does not have *visualization* permission, the user authentication is rejected and the process is terminated.
- If the check has been successful, the user authentication is accepted, takes note of the permissions associated with said role and allows the user to execute only the operations associated with the permissions available to said role and the process is finished.

1.2) If it does not answer or refuses the socket connection in the *Timeout* setting:

- If it has tried less than *Number of Attempts* times, skip to step 1.
- If it has tried *Number of Attempts* times, go to step 2 and generate the **LDAP repository not accessible** cybersecurity event.

2) Repeat step 1, but now with the secondary server.

   If the secondary's IP is 0.0.0.0, continue to step 3.

3) If the *Fallback to Local Access* setting is a:
   - **On**, it attempts to validate the user and password against the local users of the device and terminates the process.
   - **Off**, the user authentication is rejected and the process is terminated.

At the end of the process, if it has been able to connect to the server, the device sends an *unbindRequest* before closing the connection to the LDAP server.

In the case of using a secure method (LDAPS or StartTLS), when connecting to the LDAP server, the device performs a TLS mutual authentication based on certificates. On the one hand, the device sends its device certificate so that it can be validated by the LDAP server. On the other hand, the device asks the LDAP server for its certificate and checks that the certificate sent by the LDAP server is valid based on the following checks:

- The remote certificate must be signed by one of the certificate authorities (CA) present in the list of CAs of the device. If it is not, the connection to the LDAP server is rejected and the **Certificate validation failed - certificate signature check failed** cybersecurity event is generated.
- The remote certificate must not be revoked, which is checked using the certificate revocation lists (CRL) of the device. If the certificate is revoked, the connection to the LDAP server is rejected and the **Certificate validation failed - certificate revoked** cybersecurity event is generated.
- The remote certificate must not be expired (checking that the current date of the device is between the *Valid from* and *Valid to* dates of the server certificate). If the certificate is expired, the connection to the LDAP server is rejected and the **Certificate validation failed - certificate expired** cybersecurity event is generated.
- The remote certificate must contain the IP of the LDAP server in one of the following certificate fields: CN (Common Name) or SAN (Subject Alternative Name). If the IP does not appear in either of these two fields or appears but does not match the IP of the LDAP server, the connection to the LDAP server is rejected and the **Certificate validation failed - certificate does not comply with authorization policies** cybersecurity event is generated.

For details on how the certificate and lists of CAs and CRLs of the device are managed, see **Credential Management** chapter.

4.3 Remote User Authentication Configuration

The configuration of remote user authentication must be performed by a user with *user management* permission. Remote user authentication is managed from several interfaces:

- Web interface (HTTP / HTTPS).
- CLI command line interface (Telnet / SSH).
- ***ZIV e-NET Tool***® configuration tool, using the *DevicePreferences* file.

4.3.1 Remote User Authentication Configuration from the Web

To manage remote user authentication from the web interface, some of the LAN ports and the HTTP or HTTPS services must be enabled. It is recommended to use HTTPS connection instead of HTTP because the data is encrypted.

The website of the device is accessed through a browser by entering the IP address configured in the LAN adapter of the device. Initially, a login web page appears where the user's name and password must be entered (*admin* is the default user).

The ***Configuration→Access*** menu is accessed to manage the following information related to remote authentication:

- Configuration of the authentication method for each access:

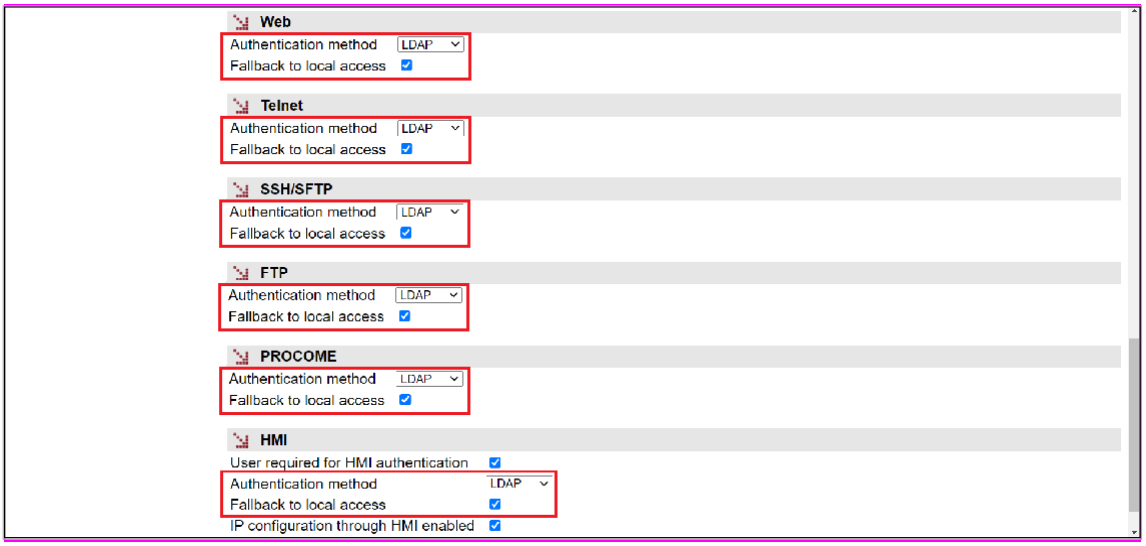


Figure 21 Authentication Method Settings in Website.

- RADIUS Configuration:

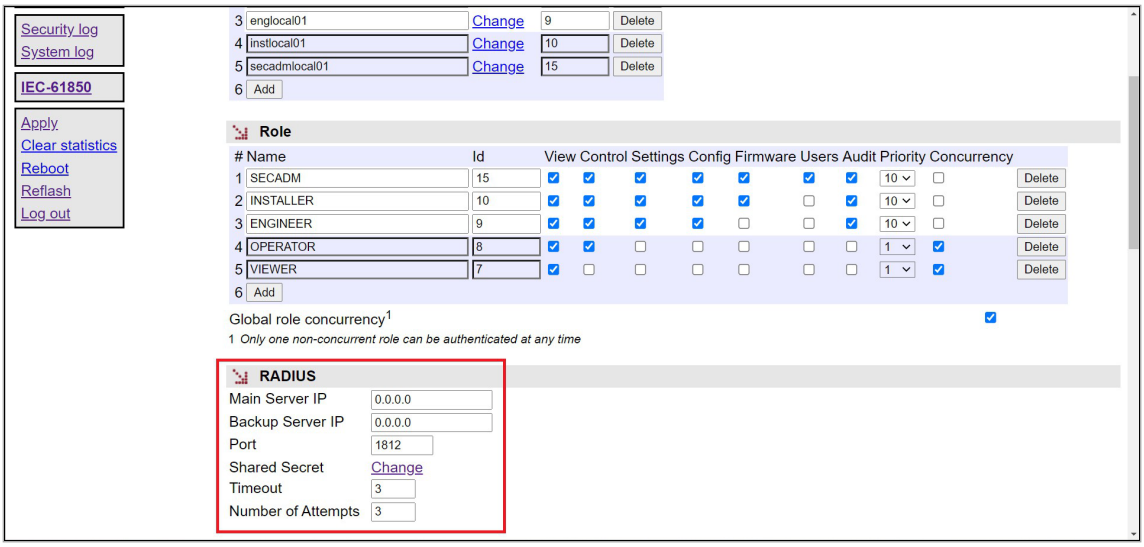


Figure 22 RADIUS Settings in Website.

To change the secret key, proceed in the same way as for changing a user's password (Figure 5).

- LDAP Configuration:

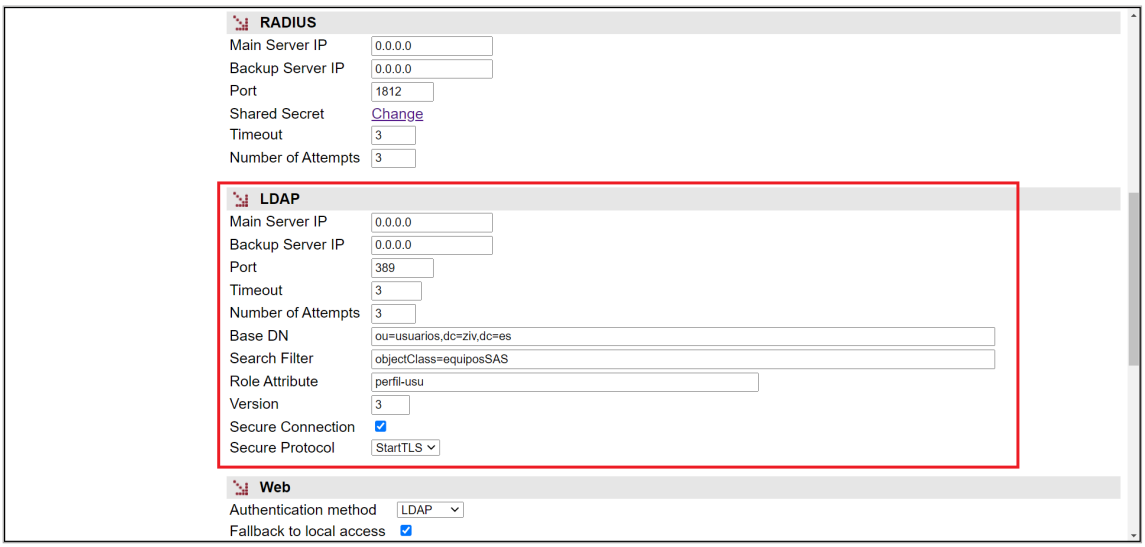


Figure 23 LDAP Settings in Website.

4.3.2 Remote User Authentication Configuration from CLI

To manage remote user authentication from the CLI command line interface, one of the LAN ports and the Telnet or SSH services must be enabled. It is recommended to use the SSH connection instead of Telnet because the data is encrypted.

The CLI of the device is accessed by launching the Telnet or SSH application from a host and entering the login of the user who has user management permissions.

The CLI of the device is automatically accessed. The parameter tree can be viewed with the ***get*** command. The settings for remote user authentication are in the ***main/access*** node:

- Configuration of the authentication method for each of the accesses:

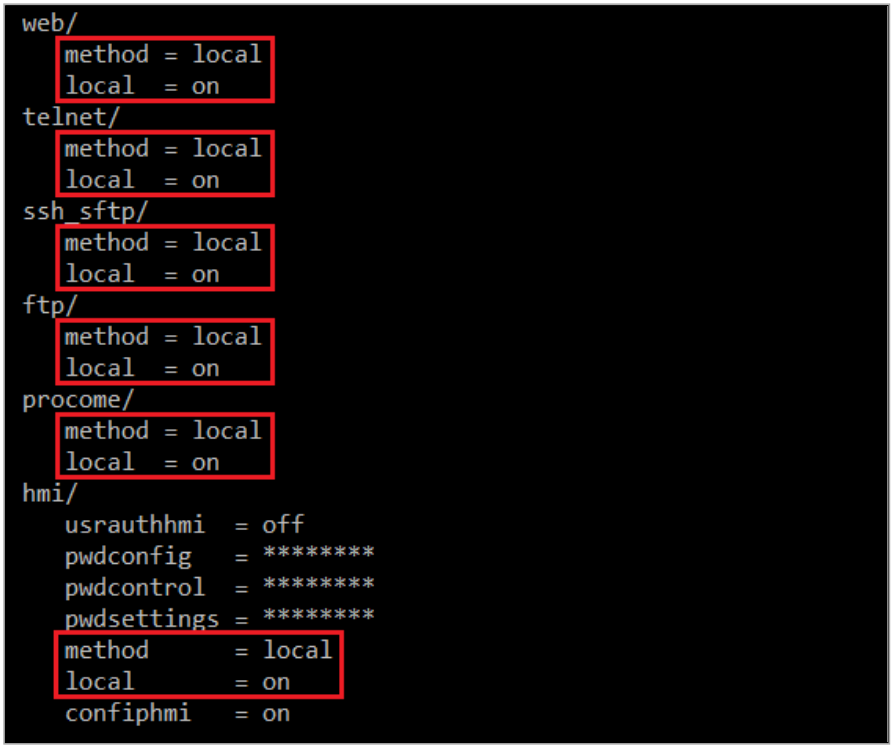


Figure 24 Authentication Method Settings in CLI.

- RADIUS Configuration:

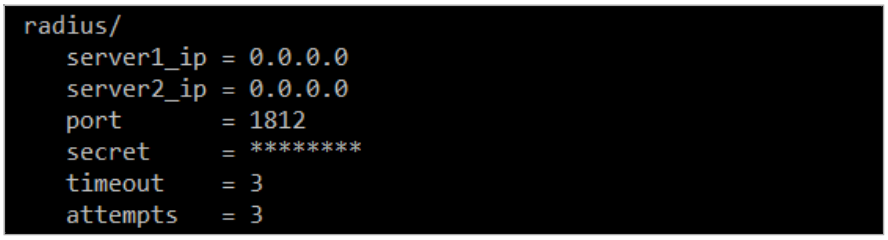


Figure 25 RADIUS Settings in CLI.

- LDAP Configuration:

```
ldap/
   server1_ip       = 0.0.0.0
   server2_ip       = 0.0.0.0
   port             = 389
   timeout          = 3
   attempts         = 3
   base_dn          = ou=usuarios,dc=ziv,dc=es
   search_filter    = objectClass=equiposSAS
   role_attribute   = perfil-usu
   version          = 3
   secure_cx        = on
   secure_protocol  = StartTLS
```

Figure 26 LDAP Settings in CLI.

To change a setting, proceed by executing the ***set*** command with the name of the setting and the device will ask for the value, or by directly executing the ***set*** command of the field to be modified, for example:

- ***set main/access/web/method local***
- ***set main/access/radius/timeout 5***
- ***set main/access/radius/secret*** (entering the key value twice).
- ***set main/access/ldap/port 636***

Once the desired settings have been configured, the data are saved with the ***save*** command and the changes are made effective on the device with the ***apply*** command.

4.3.3 Remote User Authentication Configuration from *ZIV e-NET Tool®*

To manage remote user authentication from the ***ZIV e-NET Tool®*** it is necessary that some of the LAN ports and the SSH and SFTP services are enabled.

Remote user authentication is configured with the *DevicePreferences* file.

- Configuration of the authentication method for each of the accesses:

  To configure the authentication methods for the different accesses, in the ***Navigation View*** go to the ***Preferences→Security→Access*** option.

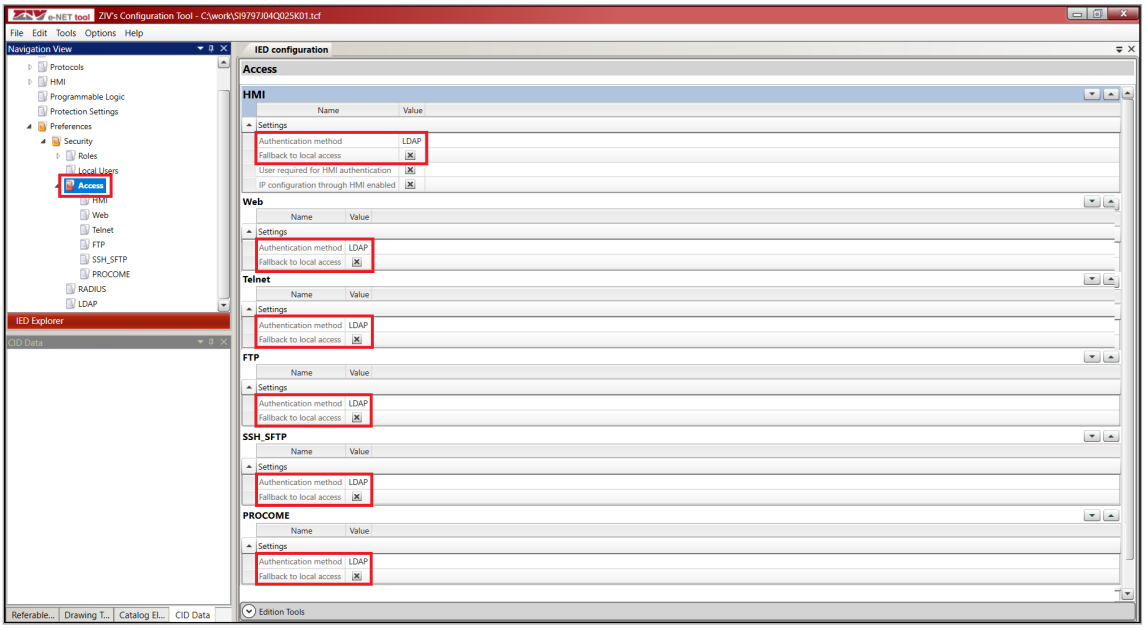


Figure 27 Configuring the Authentication Method for each Access with the Configuration Tool.

By right-clicking on this option ***Preferences→Security→Access***, you can export the authentication methods to a *DevicePreferences* file or import them from a previously configured *DevicePreferences* file.

The authentication methods for each access are part of the *DevicePreferences* file. This is an XML file whose authentication methods section has the following format (settings are highlighted):

Node	Description	Child nodes	Attributes
device	Root node.	ROLES, RADIUS, LDAP, **ACCESS**	**tversion**: file version. **filetype**: type of file ("DevicePreferences").
ACCESS	Access parameters.	**HMI**, **WEB**, **TELNET**, **FTP**, **SSH_SFTP**, **PROCOME**	-
HMI	Definition of authentication method for HMI access: method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each HMI access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for HMI access (local, LDAP or RADIUS). - **local**: alternative use of local users for HMI access. - **userauthhmi**: role name. - **confiphmi**: it is allowed to change the IP by HMI. **value**: setting value.
WEB	Definition of authentication method for web access (HTTP/HTTPS): method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each web access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for web access (local, LDAP or RADIUS). - **local**: alternative use of local users for web access. **value**: setting value.
TELNET	Definition of authentication method for Telnet access: method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each Telnet access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for Telnet access (local, LDAP or RADIUS). - **local**: alternative use of local users for Telnet access. **value**: setting value.

Node	Description	Child nodes	Attributes
FTP	Definition of authentication method for FTP access: method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each FTP access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for FTP access (local, LDAP or RADIUS). - **local**: alternative use of local users for FTP access. **value**: setting value.
SSH_SFTP	Definition of authentication method for SSH/SFTP access: method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each SSH/SFTP access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for SSH/SFTP access (local, LDAP or RADIUS). - **local**: alternative use of local users for SSH/SFTP access. **value**: setting value.
PROCOME	Definition of authentication method for PROCOME access: method and alternative use of local users.	**Setting**	-
Setting	One **Setting** node for each PROCOME access setting.	-	**name**: setting name. The following settings are available: - **method**: authentication method for PROCOME access (local, LDAP or RADIUS). - **local**: alternative use of local users for PROCOME access. **value**: setting value.

Example of *DevicePreferences* file (only with access section, authentication method settings are highlighted):

```
<?xml version="1.0" encoding="utf-8"?>
<device tversion="0.1" fileType="DevicePreferences">
  <ACCESS>
    <HMI>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
      <Setting name="usrauthhmi" value="Yes" />
      <Setting name="confiphmi" value="Yes" />
    </HMI>
    <WEB>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
    </WEB>
    <TELNET>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
    </TELNET>
    <FTP>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
    </FTP>
    <SSH_SFTP>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
    </SSH_SFTP>
    <PROCOME>
      <Setting name="method" value="LDAP" />
      <Setting name="local" value="Yes" />
    </PROCOME>
  </ACCESS>
</device>
```

To transfer the authentication methods for the different accesses to the device, access the ***Tools→Device access*** menu. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Transfer Files*** tab, check the ***TCF→Preferences*** option (if you want to transfer the complete *DevicePreferences* file), the ***TCF→Preferences→Access*** option (if you want to transfer the complete access section) or the ***TCF→Preferences→Access→XXX*** option (where XXX=HMI, Web, Telnet, FTP, SSH_SFTP or PROCOME if you want to transfer a particular access section, this is the minimum section that can be sent) and click ***Transfer*** to upload the information to the device. The upload progress is indicated in the ***Information*** window. If it has been successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if there is any error in the validation by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

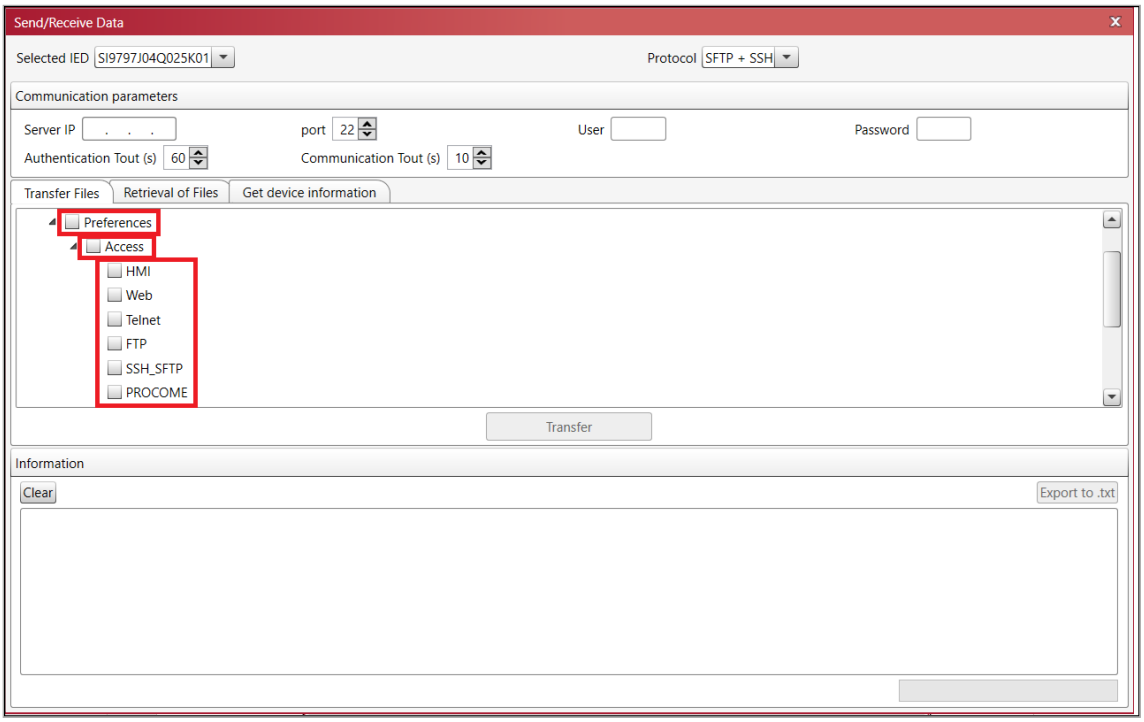


Figure 28 Sending Authentication Methods with Configuration Tool.

To retrieve the authentication methods for the different accesses of the device, proceed as indicated in the **User and Role Management from ZIV e-NET Tool®** chapter for the roles, as it is only possible to retrieve the complete *DevicePreferences* file.

- RADIUS Configuration:

  To configure RADIUS, in the ***Navigation View*** go to the ***Preferences→Security→RADIUS*** option.

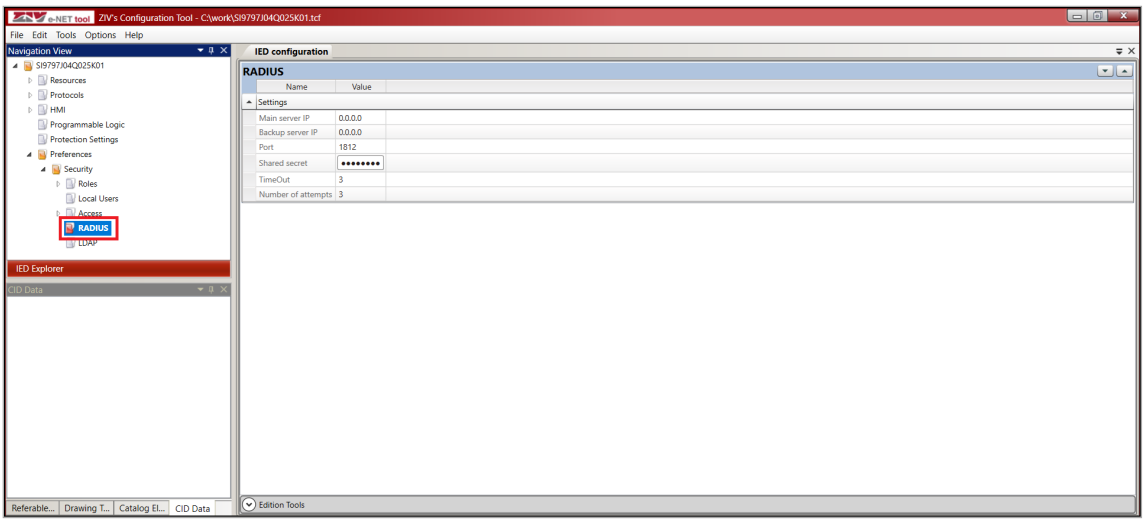


Figure 29 RADIUS Configuration with Configuration Tool.

By right-clicking on this option ***Preferences→Security→RADIUS***, you can export the RADIUS settings to a *DevicePreferences* file or import them from a previously configured *DevicePreferences* file.

The RADIUS settings are part of the *DevicePreferences* file. This is an XML file whose RADIUS section has the following format:

Node	Description	Child nodes	Attributes
device	Root node.	ROLES, **RADIUS**, LDAP, ACCESS	**tversion**: file version. **filetype**: type of file ("DevicePreferences").
RADIUS	Parameters for RADIUS remote authentication	**Setting**	-
Setting	One **Setting** node for each RADIUS setting.	-	**name**: setting name. The following settings are available: - **server1_ip**: IP address of the primary RADIUS server. - **server2_ip**: IP address of the secondary RADIUS server. - **port**: RADIUS server port. - **secret**: password encryption key. - **timeout**: RADIUS server response waiting time. - **attempts**: authentication attempts against the RADIUS server. **value**: setting value.

Example of *DevicePreferences* file (only with RADIUS section):

```
<?xml version="1.0" encoding="utf-8"?>
<device tversion="0.1" fileType="DevicePreferences">
   <RADIUS>
   <Setting name="server1_ip" value="0.0.0.0" />
   <Setting name="server2_ip" value="0.0.0.0" />
   <Setting name="port" value="1812" />
   <Setting name="secret" value="ziv12345" />
   <Setting name="timeout" value="3" />
   <Setting name="attempts" value="3" />
  </RADIUS>
</device>
```

To transfer RADIUS settings to the device, access the ***Tools→Device access*** menu. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Transfer Files*** tab, check the ***TCF→Preferences*** option (if you want to transfer the entire *DevicePreferences* file) or the ***TCF→Preferences→RADIUS*** option (if you want to transfer only the RADIUS section) and click ***Transfer*** to upload the information to the device. The upload progress is indicated in the ***Information*** window. If it has been successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if there is any error in the validation by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

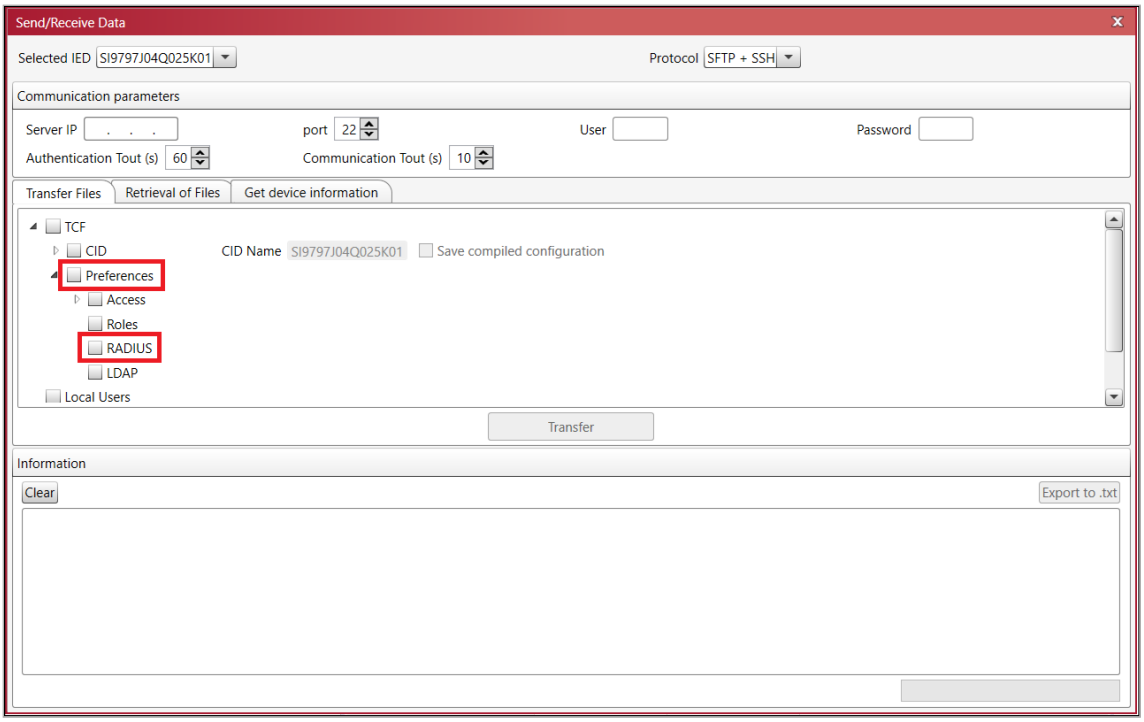


Figure 30 Sending RADIUS Settings with Configuration Tool.

To retrieve the RADIUS settings of the device, proceed as indicated in the **User and Role Management from ZIV e-NET Tool®** chapter for the roles, as it is only possible to retrieve the complete *DevicePreferences* file.

- LDAP Configuration:

  To configure LDAP, in the ***Navigation View*** go to the ***Preferences→Security→LDAP*** option.

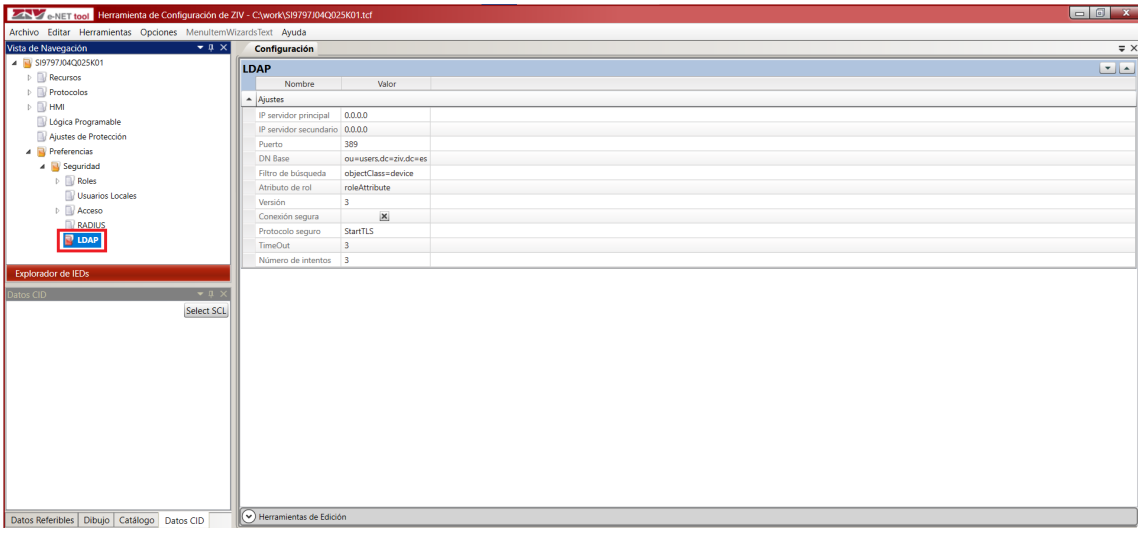


Figure 31 LDAP Configuration with Configuration Tool.

By right-clicking on this option ***Preferences→Security→LDAP***, you can export the LDAP settings to a *DevicePreferences* file or import them from a previously configured *DevicePreferences* file.

The LDAP settings are part of the *DevicePreferences* file. This is an XML file whose LDAP section has the following format:

Node	Description	Child nodes	Attributes
device	Root node.	ROLES, RADIUS, **LDAP**, ACCESS	**tversion**: file version. **filetype**: type of file ("DevicePreferences").
LDAP	Parameters for LDAP remote authentication.	-	**IpAuth**: IP address of the primary LDAP serve. **IpAuth2**: IP address of the secondary LDAP serve. **port**: LDAP server port. **tRetryAuth**: LDAP server response time. **NRetryAuth**: authentication attempts against the LDAP server. **base_dn**: organizational unit (OU) in which users are stored on the LDAP server. **search_filter**: additional search filter for the user. **role_attribute**: name of the attribute containing the user role. **version**: LDAP version. **secure_cx**: indicates whether the LDAP connection is secure or not. **secure_protocol**: indicates the secure protocol to be used.

Example of *DevicePreferences* file (only with LDAP section):

```
<?xml version="1.0" encoding="utf-8"?>
<device tversion="0.1" fileType="DevicePreferences">
  <LDAP tRetryAuth="3" NRetryAuth="3" port="389" IpAuth="192.168.1.69"
  IpAuth2="0.0.0.0" role_attribute="perfil-usu" version="3" secure_cx="Yes"
  secure_protocol="StartTLS" base_dn="ou=usuarios,dc=ziv,dc=es"
  search_filter="objectClass=equiposSAS" />
</device>
```

To transfer LDAP settings to the device, access the ***Tools→Device access*** menu. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Transfer Files*** tab, check the ***TCF→Preferences*** option (if you want to transfer the entire *DevicePreferences* file) or the ***TCF→Preferences→LDAP*** option (if you want to transfer only the LDAP section) and click ***Transfer*** to upload the information to the device. The upload progress is indicated in the ***Information*** window. If it has been successfully uploaded to the device, the **Configuration uploaded successfully** cybersecurity event is generated. However, if there is any error in the validation by the device, the **Configuration upload failed - invalid configuration** cybersecurity event is generated.

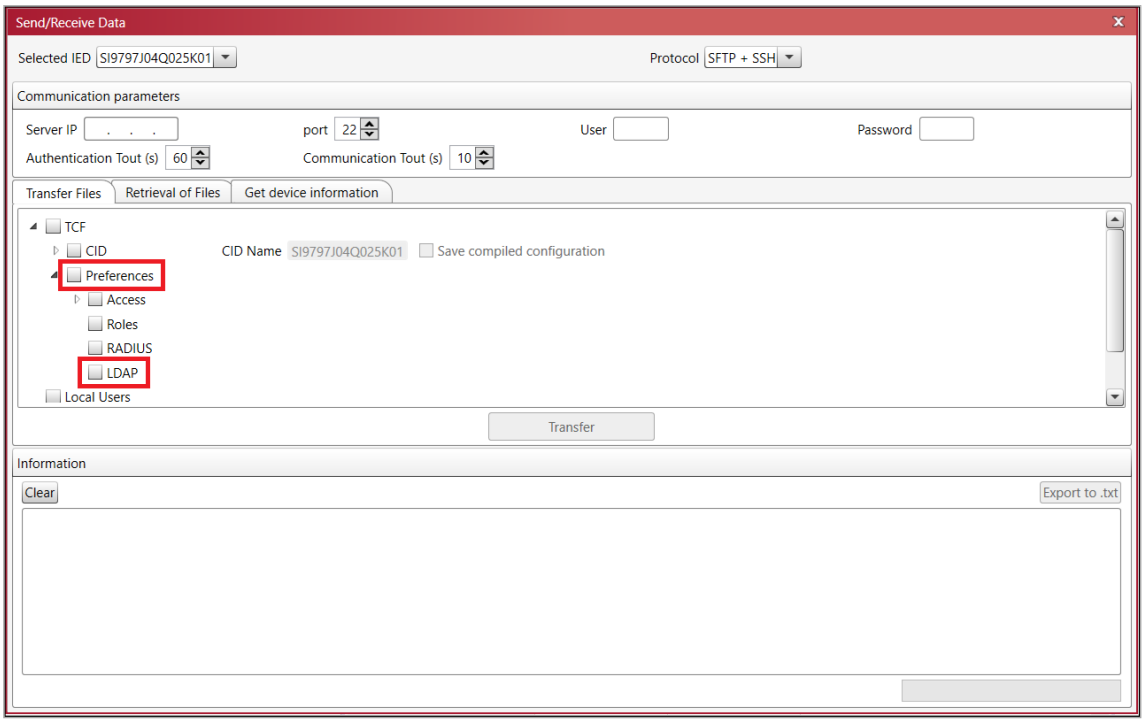


Figure 32 Sending LDAP Settings with Configuration Tool.

To retrieve the LDAP settings of the device, proceed as indicated in the **User and Role Management from ZIV e-NET Tool®** chapter for the roles, as it is only possible to retrieve the complete *DevicePreferences* file.

5. Communication with Configuration Tool using PROCOME

Zivercomplus® and ***ZIV e-NET Tool***®[1] communications with the device use PROCOME protocol. This PROCOME protocol has been strengthened so that communications are authenticated and encrypted (the latter only in the case of LAN port communication).

Authentication is based on a new private ASDU that includes user and password, replacing ASDU 116, which only uses a password.

On the other hand, for LAN port communications, there is the possibility for the protocol to use encryption, using TLS v1.2.

The different instances of communication with PROCOME protocol that can be present in the device are:

- Local Port. - Remote Port 1 and 2.	- LAN ports: one fixed PROCOME instance and four more configurable instances (TCP/IP Protocol 1, 2, 3 and 4).

The settings for configuring whether each of the PROCOME instances is authenticated and/or encrypted (the latter only in case of LAN port communication) are detailed in Chapter 1, **Description and Start-up**, **Communication Settings** chapter in the instruction manual of the device.

By default, all instances are set with authentication except for the local port. Therefore, any session opened through an authenticated connection will require the user/password to be entered on the device at the start of the connection.

The authenticated local port setting can only be adjusted from HMI.

Only instances that communicate over LAN ports (PROCOME Protocol and TCP/IP Protocols), can be configured to work with encryption, and by default they are adjusted to encrypted mode. For more details on the type of encryption used, see **Secure Sockets** chapter. Each of these instances also has an adjustable logical port number.

In TCP/IP protocols, encryption and authentication only affect the PROCOME protocol. The other configurable protocols, DNP3 and MODBUS, are not affected by these settings.

*When communicating the configuration tools with the device via LAN, the user must configure the **secure** / **non-secure** communications option in the tool, according to the corresponding PROCOME instance that has been configured as **encrypted** / **not encrypted** on the device. If they do not match, communication will not be possible.*

[1] ***ZIV e-NET Tool***® also uses other protocols such as SFTP and SSH to communicate with the device.

6. Secure Sockets

In order to prevent communications from being spied on, a suitable countermeasure is to encrypt communications using secure sockets.

The device uses the secure (encrypted) versions of the following protocols:

Protocol	Secure Version
Telnet	SSH
FTP	SFTP
HTTP	HTTPS
PROCOME	PROCOME over TLSv1.2
LDAP	LDAPS/StartTLS

6.1 SSH (Secure Shell)

Secure Shell is a cryptographic network protocol for securing communications. It establishes a secure channel over an insecure network in a client/server architecture.

In the case of the device, it provides an SSH server that allows SSH clients to connect in an authenticated manner (login) for maintenance functions by using the command line interface (CLI).

This protocol is a direct and secure replacement for the Telnet interface.

The encryption used by SSH provides confidentiality and data integrity over an insecure network, such as the Internet.

SSH uses public key cryptography to authenticate the remote machine and allow the user to authenticate.

The protocol version used on the device is SSH Version 2 (SSHv2).

The device supports both Telnet and SSH (see **Communication Ports and Services** chapter to see how to enable / disable one, the other or both).

6.2 SFTP (SSH File Transfer Protocol)

SSH File Transfer Protocol (also known as Secure File Transfer Protocol) is a network protocol that provides file access, file transfer and file management functionality over SSH. It works together with SSH to provide secure file transfer capabilities.

This protocol assumes that it is running on a secure channel, such as SSH, that the server has already authenticated the client, and that the client's user identity is available to the protocol.

The encryption and authentication aspects for this protocol are handled by the SSH server also running on the device. Therefore, the encryption and authentication methods are identical to those for SSH described above.

The device supports both FTP and SFTP (see **Communication Ports and Services** chapter for how to enable/disable one, the other or both).

6.3 TLS / SSL (Transport Layer Security / Secure Socket Layer)

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communication security over a TCP/IP network. They use the X.509 certificate and, therefore, asymmetric cryptography to authenticate the machine they are communicating with and to exchange a symmetric session key. This session key is used to encrypt the data flowing between the two parties. This allows data confidentiality and message authentication codes for message integrity and, in addition, message authentication.

In Internet Protocol Suite, TLS and SSL encrypt the network connection data at the application layer. In OSI model equivalencies, TLS/SSL is initialized at layer 5 (session layer) and functions as layer 6 (presentation layer). The session layer has a link protocol that uses asymmetric encryption to set up encryption configurations and a shared key for that session. The presentation layer then encrypts the rest of the communication using symmetric encryption and that session key. In both models, TLS and SSL work on behalf of the underlying transport layer, whose segments carry encrypted data.

In the device, TLS is only used for the Secure Web Server (HTTPS) application and for PROCOME, which is explained in the **Communication with Configuration Tool using PROCOME** chapter.

The TLS version supported is TLSv1.2. During TLS negotiation, the most secure option common to both client and server is always chosen.

The SSL1, SSL2, SSL3, TLSv1.0 and TLS v1.1 protocols are not available on the device as they are considered insecure.

6.4 HTTPS (Secure Web Server)

The Secure Web Server application uses the Hypertext Transfer Protocol (HTTPS) which is a communications protocol for secure communication over a network.

Technically it is not a protocol per se but is the result of simply placing the Hypertext Transfer Protocol (HTTP) over the SSL/TLS protocol (explained in the previous section), thus adding the security capabilities of SSL/TLS for standard HTTP communications.

The security of HTTPS is that of the underlying SSL / TLS, which uses long-term public and secret keys to exchange a short-term session key to encrypt the data flow between a client and a server.

The X.509 certificate is used to ensure that you are communicating to the peer you want to communicate with.

In its popular implementation on the Internet, HTTPS provides authentication of the website and the associated web server. In addition, it provides bi-directional encryption of communications between a client and a server, which protects against eavesdropping and manipulation and / or falsification of the communication content.

A website must be fully supported over HTTPS, without having some of its content uploaded over HTTP, or the user will be vulnerable to some attacks and surveillance.

The device supports both HTTP and HTTPS (see **Communication Ports and Services** chapter for how to enable/disable one, the other or both).

6.5 Mutual Authentication

Mutual authentication or two-way authentication (not to be confused with two-factor authentication) refers to two ends of a communication authenticating each other at the same time in an authentication protocol. Mutual authentication is a desired feature in verification schemes that transmit sensitive data, in order to ensure the security of such data. Mutual authentication can be achieved with two types of credentials: user/password and public key certificates.

Two scenarios are distinguished in the device:

- Communications using TLS protocol:

  Mutual authentication is performed based on X.509 certificates. The device sends its certificate so that it can be validated by the other end and additionally requires the remote end to send its certificate to perform the pertinent checks and validate it. In HTTPS and PROCOME communications over TLS (in this second case if it is configured to be authenticated) a second authentication factor consisting of user and password is additionally required.

- Communications using SSH protocol:

  Mutual authentication is performed as follows:

  - Server authentication is performed based on the server's public key. That is, all clients communicating with the device using SSH will need to include the public key of the device among their known hosts if they want to check the validity of the server.

- Client authentication is based on username/password authentication. In other words, clients connecting to the device must enter username and password in order to be validated by the device (they will be checked against remote servers or against local users, as explained in the **Remote User Authentication** chapter).

The device manages a single common private key for TLS and SSH communications. There is, therefore, a single place where the private key and certificate of the device are configured. When new credentials (private key/certificate) are uploaded to the device, the public key used by SSH is derived internally from those credentials by software within the device.

Root certificates, used to validate remote certificates received by the device during TLS communications, are common to all services using TLS (HTTPS, secure LDAP or encrypted PROCOME LAN). They correspond to the list of Certificate Authorities (hereinafter CA) of the device. When a device has a list of CAs configured (by default the device is configured with the ZIV CA), the device will require TLS mutual authentication, although this mutual authentication can be disabled per service as explained in the **Credential Management** chapter.

Devices with basic cybersecurity do not perform TLS mutual authentication.

The process for validating a remote certificate is as follows:

- The remote certificate is validated against the list of CAs of the device. If the certificate is not trusted (not signed by any of the CAs present in the list), the communication is rejected and the **Certificate validation failed - certificate signature check failed** cybersecurity event is generated.
- The revocation of the remote certificate is checked against the CRL (Certificate Revocation List) of the device. If the certificate is revoked, the communication is rejected and the **Certificate validation failed - certificate revoked** cybersecurity event is generated.
- The remote certificate is checked for expiration (the certificate is expired or not yet valid). If the certificate is expired, the communication is rejected and the **Certificate validation failed - certificate expired** cybersecurity event is generated.
- Additional validations can be performed on the remote certificate expressed in an authorization policy file based on certificate details (fingerprint and CN and OU fields) which are detailed in the **Credential Management** chapter. If the certificate does not comply with these authorization policies, the communication is rejected and the **Certificate validation failed - certificate does not comply with authorization policies** cybersecurity event is generated.

7. Credential Management

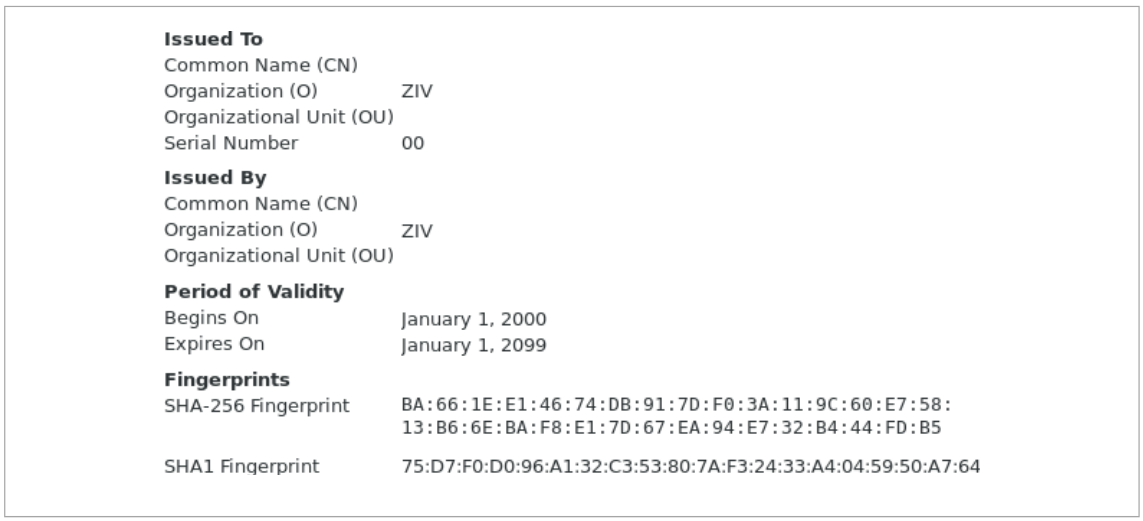
Credential management refers to the way in which the entire public key infrastructure (hereinafter PKI) is managed within the device. The following shows how the certificate and private key of the device are managed (the only thing available in devices with basic cybersecurity), as well as the list of certification authorities (CAs), the certificate revocation list (CRLs) and the authorization policy file for devices with enhanced cybersecurity.

7.1 Devices with Basic Cybersecurity

In devices with basic cybersecurity, the PKI consists only of a certificate and private key that allow the device to perform secure communications without TLS mutual authentication capability, i.e., without performing any type of remote certificate verification.

By default, the devices at the first boot up generate a self-signed X.509 certificate with a 2048-bit RSA public key and a 2048-bit private key, both in PEM format.

The self-signed certificate generated has the following identifiers and validity period:



Issued To

Common Name (CN)	
Organization (O)	ZIV
Organizational Unit (OU)	
Serial Number	00

Issued By

Common Name (CN)	
Organization (O)	ZIV
Organizational Unit (OU)	

Period of Validity

Begins On	January 1, 2000
Expires On	January 1, 2099

Fingerprints

SHA-256 Fingerprint	BA:66:1E:E1:46:74:DB:91:7D:F0:3A:11:9C:60:E7:58: 13:B6:6E:BA:F8:E1:7D:67:EA:94:E7:32:B4:44:FD:B5
SHA1 Fingerprint	75:D7:F0:D0:96:A1:32:C3:53:80:7A:F3:24:33:A4:04:59:50:A7:64

Figure 33 Characteristics of the Self-Signed Certificate for Devices with Basic Cybersecurity.

This certificate and its private key are provisional and should be replaced by the client with a trusted certificate and its corresponding private key. For this purpose, the device has a menu on the website that allows the upload of a new certificate and private key, as well as the download of the current certificate.

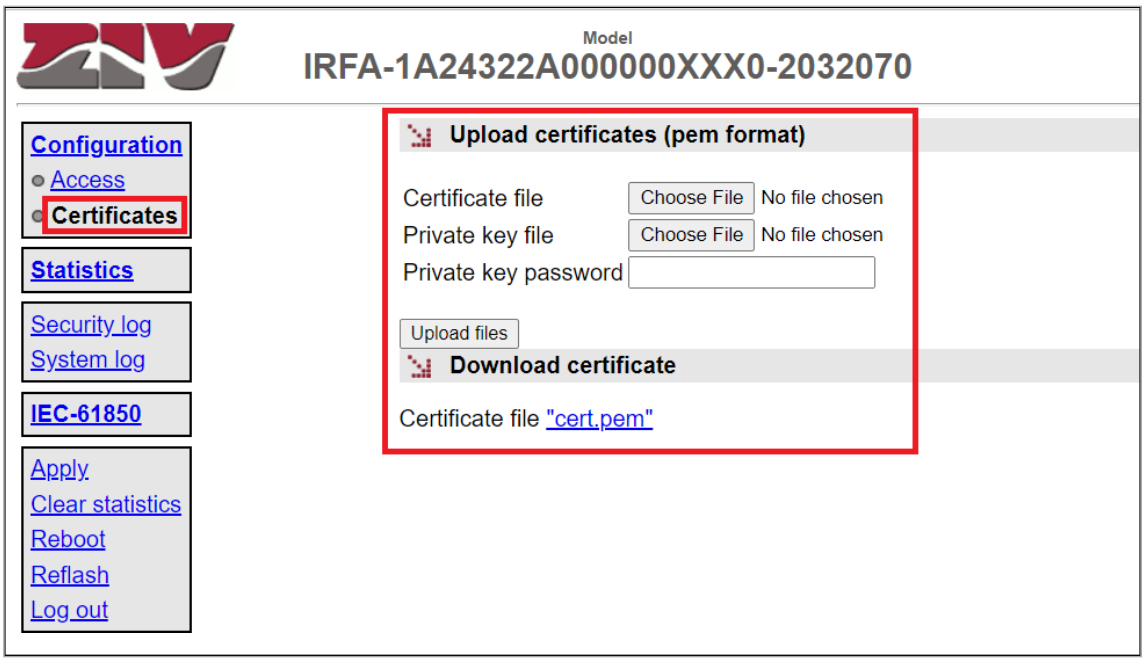


Figure 34 Certificate and Private Key Configuration on Website for Devices with Basic Cybersecurity.

The certificate management operation is as follows:

- Access to the certificate menu of the device is only possible with a user with *user management* permission.
- The certificate and private key upload option are only available with HTTPS connections. In case of using an unsecured connection (HTTP), only the certificate download is allowed.
- Only certificates and private keys in PEM format can be uploaded to the device. The private key can be either clear or encrypted with AES encryption.
- From the website you can select the certificate (***Certificate File***), the private key (***Private Key File***) and the encryption key of the private key (***Private Key Password***).
- You can upload the certificate and the private key individually or simultaneously with the ***Upload files*** button, selecting the ***Choose File*** button to choose the file corresponding to the certificate and/or the private key.
- When only the certificate is uploaded, it must be paired with the private key already stored in the device, or else a verification error will occur. In this case, the uploaded file is discarded. The same thing happens with the private key.
- If no value is entered in the encryption key of the private key, then the private key is considered not to be encrypted. If the password is entered, the private key is decrypted with said password and re-encrypted with a password specific to the device using the AES 256 CBC algorithm. The password entered by the user is not stored in the device in any case.
- After pressing the ***Upload files*** button, a certificate and/or private key check is performed. If there is an error, it will be indicated to the user with a message and the uploaded files will be discarded.

- From the website it is possible to download the current certificate (***Download Certificate***), but in no case the private key.
- Once the files have been uploaded, it is necessary to apply the changes to the device (press the ***Apply*** button) so that the new certificate and/or private key files take effect.

7.2 Devices with Enhanced Cybersecurity

In devices with enhanced cybersecurity, the PKI consists not only of a certificate and a public key, but also of a series of additional files (list of CAs, list of CRLs, authorization policy file) that allow the device both to perform secure communications with mutual authentication capability and to check the correct securization of the firmware.

A device with enhanced cybersecurity capability can be manufactured without such enhanced cybersecurity in terms of PKI at the request of a customer who does not wish to include the features described below. In this case, in terms of credential management, it will behave as described in the previous **Devices with Basic Cybersecurity** chapter.

The files that make up the PKI are the following:

- Master password:

  Used to encrypt certain sensitive information (private key/certificate) when it is sent to the device for updating. The sensitive information is encrypted inside the device using an own password of the device with AES 256 CBC algorithm.

  When you want to update this password in the device, the file sent with the new master password must be encrypted AES 256 CBC with the current master password.

- Certificate / Private key:

  This is the private key and certificate similar to those explained in the previous chapter (**Devices with Basic Cybersecurity**). In this case they are not self-generated and the certificate is not self-signed, different ones are installed for each device during the manufacturing process and the certificate is signed by the ZIV CA and is valid for 20 years. The key is 4096-bit RSA.

  When you want to update the device, either only the certificate or both, certificate and private key, you must use a PKCS#12 file whose key is the current master password of the device.

- List of Certification Authorities (CAs):

  This is a file containing all the root certificates that will be used to verify both the remote certificates in the case of TLS mutual authentication (verifying that they are signed by one of the CAs on the list), and the signatures of the firmware as explained in the **Digital Firmware Securization** chapter. The file format is PEM and contains a concatenation of all the possible CAs managed by the device, both for communications and for firmware signature verification.

- Certificate Revocation List (CRLs):

  This is a file containing all the certificate revocation lists (CRLs) associated with all the CAs of the device, which are going to be used to verify the revocation of remote certificates in the case of TLS mutual authentication, as well as of the firmware signatures. The file format is PEM and contains a concatenation of all possible CRLs managed by the device.

- Authorization policy file:

  This is a file that includes a set of policies that allow checking additional parameters of the remote certificates in the case of TLS mutual authentication, as well as the signatures of the firmware.

  Of course, the validity of the certificates must first be checked: signing by CAs, revocation and expiration. Then these additional rules (policies) are checked. Each policy affects one service (firmware signature verification, PROCOME communications, HTTPS communications) and you can select whether any or all certificates must comply with these rules. The rules are specified per certificate and can include the certificate fingerprint (of the subject and/or issuer), the CN attribute (of the subject and/or issuer) and/or the OU attribute (of the subject and/or issuer).

  The file format is XML and is as follows:

Node	Description	Child nodes	Attributes
AuthPolicy	Root node.	**Policy** (only one)	-
Policy	Authorization policy.	**Certificate** (several)	**effect**: indicates the service to which the policy applies. Possible values are: - **allow-signature-verify**: firmware signature verification service. After verification that the firmware is signed by the certificate(s) and that the certificate(s) is (are) signed by a CA from the list of CAs of the device and is (are) neither expired nor revoked, the certificate(s) must comply with the rules of this policy for the firmware signing process to be valid for the device. - **allow-procome-connection**: PROCOME TLS communications service. After verifying that the remote certificate is signed by a CA from the list of CAs of the device and that it is neither expired nor revoked, the remote certificate must comply with the rules of this policy for the PROCOME TLS connection to be accepted by the device. - **allow-https-connection**: HTTPS communications service. Identical operation to **allow-procome-connection**, but for HTTPS communication.

Node	Description	Child nodes	Attributes
Policy	Authorization policy.	**Certificate** (several)	**condition**: indicates whether the policy affects all certificates or only some of them. The possible values are: - **allow-if-any-certificate-matches**: the policy will be successful if any of the certificates used complies with the certificate rules included in the policy. - **allow-if-all-certificates-match**: the policy will be successful if all certificates used comply with the certificate rules included in the policy. - **allow-always**: only for TLS communication, it is used when you do not want to use mutual authentication despite having configured the list of CAs and CRLs in the device. This option allows you to disable mutual authentication by service (PROCOME or HTTPS). This value is ignored for firmware signature verification.
Certificate	Rules to check for a certificate.	**Thumbprint**, **CN**, **OU** (up to 2 of each)	-
Thumbprint	Fingerprint of the certificate. The fingerprint (SHA256) of the certificate to be checked (**from**="subject") or of the certificate signing the certificate to be checked (**from**="issuer") must match this attribute for the rule to be successful.	-	**from**: indicates whether the rule (Thumbprint) is to be checked for the subject or issuer of the certificate.
CN	CN attribute of the certificate. The CN attribute of the subject (**from**="subject") or issuer (**from**="issuer") of the certificate to be checked must contain this attribute (case sensitive) for the rule to be successful.	-	**from**: indicates whether the rule (CN) is to be checked for the subject or the issuer of the certificate.
OU	OU attribute of the certificate. The OU attribute of the subject (**from**="subject") or issuer (**from**="issuer") of the certificate to be checked must contain this attribute (case sensitive) for the rule to be successful.	-	**from**: indicates whether the rule (OU) is to be checked for the subject or the issuer of the certificate.

An example of the file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
 <AuthPolicy>
  <Policy effect="allow-signature-verify" condition="allow-if-all-
  certificates-match">
   <Certificate>
   <Thumbprint
   from="issuer">3500c3951b03ef20e4954575705fe50b9b0b69ee32e46b72ab0f5a1b170
   42010</Thumbprint>
   <CN from="subject">ziv-firmware-signing</CN>
   <OU from="subject">ZIV</OU>
   </Certificate>
  </Policy>
  <Policy effect="allow-https-connection" condition="allow-if-any-
  certificate-matches">
   <Certificate>
   <CN from="issuer">ziv-ca</CN>
   </Certificate>
  </Policy>
  <Policy effect="allow-procome-connection" condition="allow-always">
  </Policy>
 </AuthPolicy>
```

Credentials management must be performed by a user with *user management* permission and is done using the ***ZIV e-NET Tool***® configuration tool.

To manage the credentials of the device from the ***ZIV e-NET Tool***® it is necessary that some of the LAN ports and the SSH and SFTP services are enabled.

This is accessed from the ***Tools***→ ***Device access*** menu.

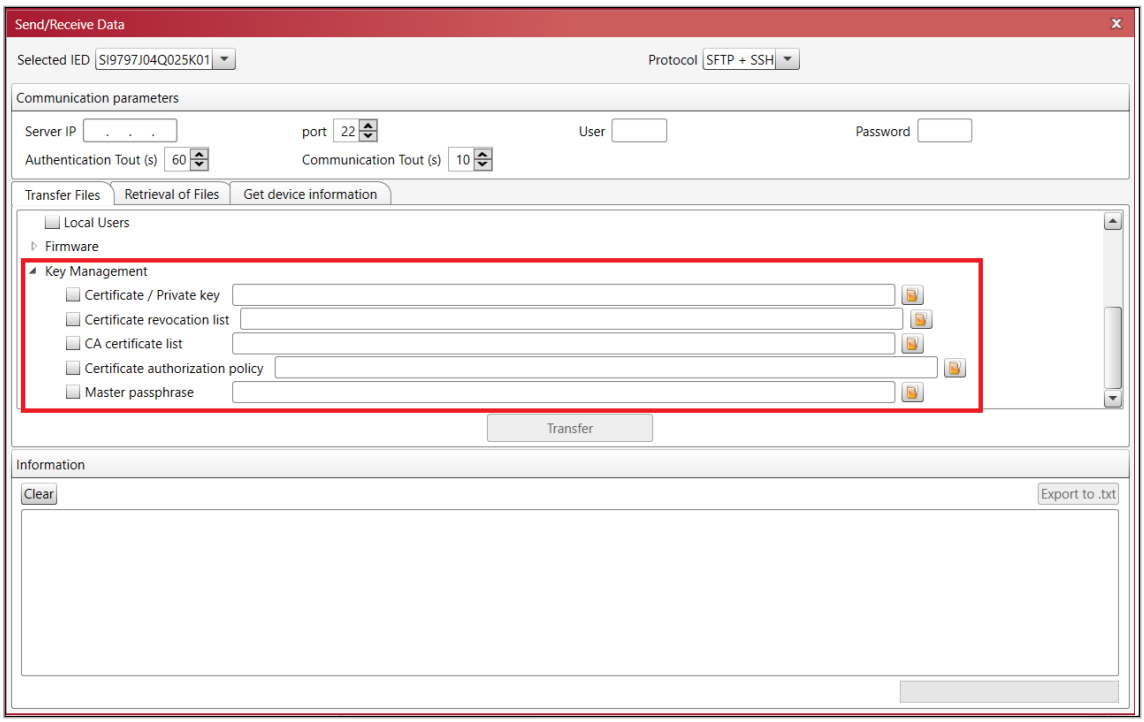


Figure 35 Sending of PKI Files with Configuration Tool.

In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (username and password). In the ***Transfer Files*** tab, under the ***Key Management*** option, there are several options for sending the different PKI files available on the device:

- ***Certificate / Private key***: Option to upload a certificate or a certificate + private key to the device. A file in PKCS#12 format must be used whose key is the current master password of the device. If it contains only the certificate, it must be paired with the current private key of the device.
- ***Certificate revocation list***: Option to upload a certificate revocation list (CRLs). A file in PEM format must be used and must contain a concatenation of all possible CRLs managed by the device. It cannot be empty; it must contain at least one CRL.
- ***CA certificate list***: Option to upload a list of CA certificates to the device. A file in PEM format must be used and it must contain a concatenation of all the possible CAs managed by the device, both for communications and for firmware signature verification. It cannot be empty; it must contain at least one CA and the CAs cannot be revoked or expired.
- ***Certificate authorization policy***: Option to upload an authorization policy file to the device. The XML format file explained above must be used.
- ***Master passphrase***: Option to upload a new master password to the device. A file must be used in which the new master password is encrypted AES 256 CBC with the current master password of the device.

To select the PKI file to be sent, press the button  on the right. After selecting the desired option (only one option must be selected at a time), the ***Transfer*** button is used to upload the PKI file to the device. The upload progress is displayed in the ***Information*** window. If the PKI file is successfully uploaded to the device, the **Security credentials changed successfully** cybersecurity event is generated. However, if any error occurs in the validation of the PKI file by the device, the **Security credentials change failed** cybersecurity event is generated.

When a list of CAs is sent to the device, it checks that, for each CA present in the list, its associated CRL is present in the device. If not, the device rejects the list of CAs.

When a list of CRLs is sent to the device, the device checks that, for each CA present in the device, its associated CRL is present in the list. If not, the device rejects the list of CRLs.

That is, in the device, for each CA, there must be its associated CRL (only one CRL for each CA) and other additional CRLs may also be present.

Therefore, when you want to add one or more CA+CRLs to the device, you must first send the list of CRLs and then the list of CAs; and when you want to remove one or more CA+CRLs from the device, you must first send the list of CAs and then the list of CRLs.

To retrieve the PKI files from the device, access the menu ***Tools→Device access***. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (user and password). In the ***Retrieval of Files*** tab, under the ***Key Management*** option, there are several options to collect the different PKI files available from the device:

- ***Certificate***: Option to download the certificate of the device in PEM format.
- ***Certificate revocation list***: Option to download the Certificate Revocation List (CRLs) of the device in PEM format.
- ***CA certificate list***: Option to download the CA certificate list of the device in PEM format.
- ***Certificate authorization policy***: Option to download the authorization policy file of the device in XML format.

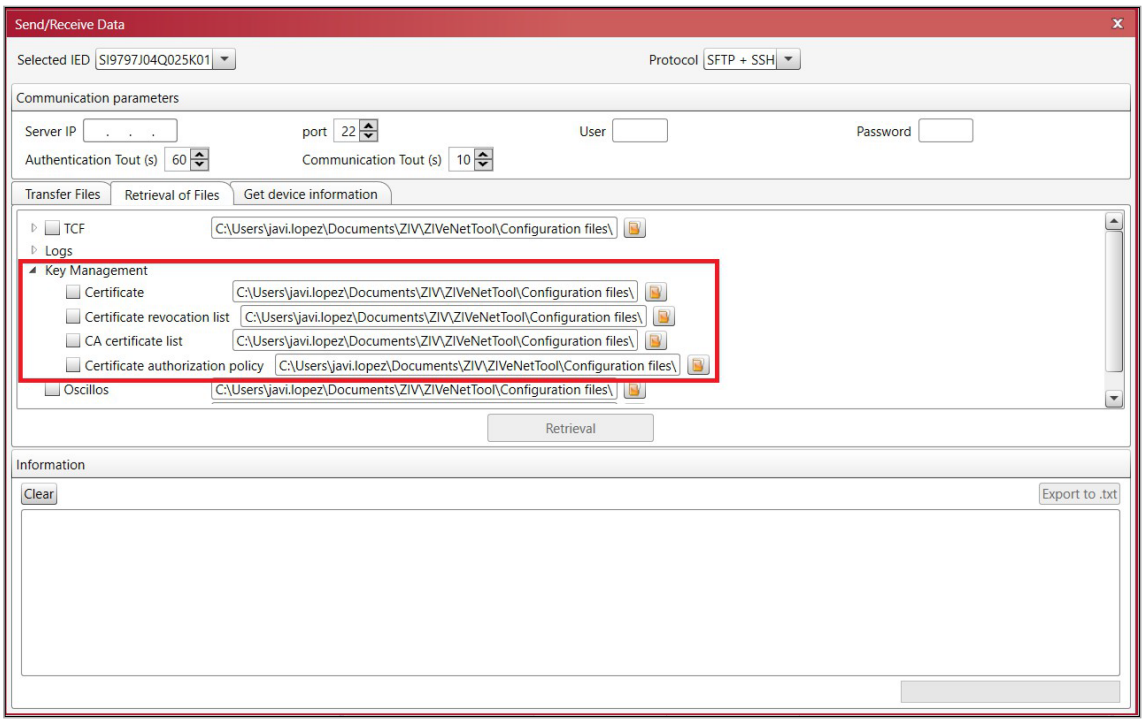


Figure 36 Retrieval of PKI files with configuration tool.

To select the destination directory for each PKI file, click the button on the right. After selecting the desired options, click on the ***Retrieval*** button to download the PKI files from the selected device. The download progress is displayed in the ***Information*** window.

8. Digital Firmware Securization

In the world of cybersecurity and due to its continuous evolution, it is very common that certain security measures become insecure over time, security gaps are discovered in software components and it is necessary to update the firmware of the devices to eliminate such security gaps.

Firmware upload is one of the most critical processes for devices. It is vital to ensure that the firmware uploaded to the device is valid and has not been altered or modified by third parties.

8.1 Devices with Basic Cybersecurity

For devices with basic cybersecurity, firmware securization consists of including a hash of the firmware and performing encryption using fixed criteria known by manufacturer and devices, which provides confidentiality and integrity.

When new firmware is uploaded to the device (via website), the device decrypts the firmware, checks the hash, performs additional internal checks to ensure that the firmware structure is valid, and only then restarts to take the new firmware, generating the **Firmware uploaded successfully** cybersecurity event. If something fails in the validation process of the new firmware, the device remains with the firmware it had uploaded before the failed update, generating the **Firmware upload failed - Invalid firmware** cybersecurity event.

8.2 Devices with Enhanced Cybersecurity

In devices with enhanced cybersecurity, firmware securization is more sophisticated. The firmware of the devices is digitally encrypted, which provides confidentiality. In addition, the encrypted firmware is signed based on X.509 certificates using CMS/PKCS#7 DER format, allowing multiple signatures (in addition to the original signature by ZIV), which provides integrity.

Signatures may be associated with a set of metadata (manifest), encoded as signed attributes of the CMS/PKCS#7 signature (*signedAttributes*). In general, this manifest will have a common structure that includes generic information about the manifest and a particular structure for each type of manifest. Specifically, for the ZIV signature, the following signed attributes have been defined to include firmware related information. For each one, the internal name, its ***OID*** (identifier) and its meaning are indicated:

- *manifestInformation* (***1.3.6.1.4.1.15732.8191.1***): general information about the manifest. It includes the following sequential information (SEQUENCE):
    - *class*: is an enumerated (ENUMERATED) indicating the type of manifest. At the moment it only takes the value 0 (firmware). In the future it may take other values for possible client signatures.
    - *version*: is an integer (INTEGER) indicating the version of the manifest. At the moment it only takes the value 1.
    - *owner*: a text string (VISIBLE STRING) from 0 to 256 characters indicating the owner of the manifest. The value is ZIV.

- *copyright*: in a text string (VISIBLE STRING) from 0 to 256 characters indicating the manifest copyright. The value is ZIV.

- *basicFirmwareInformation* (***1.3.6.1.4.1.15732.8191.2.1***): basic firmware information. It includes the following sequential information (SEQUENCE):
  - *version*: in a text string (VISIBLE STRING) from 1 to 64 characters indicating the full firmware version, e.g., 0.10.0-0C03-02.
  - *description*: is a text string (VISIBLE STRING) from 0 to 512 characters indicating the following information:
    - A first text string **Firmware** indicating that it corresponds to the complete firmware of the device (at the moment the device firmware is atomic, it is uploaded in full). In the future it may contain other words to indicate that it is a security patch or a specific part of the firmware.
    - The character | to separate both text strings.
    - A second text string indicating the complete firmware model, including hardware and software parts, e.g., **IRFA-mA14mA00000000x0xX-2032070**.

    Example: **Firmware|IRFA-mA14mA00000000x0xX-2032070**.
  - *supportedModels*: this is a sequence (SEQUENCE) of text strings (VISIBLE STRING) from 1 to 256 characters indicating the list of hardware models for which the signed firmware is valid. Examples: **IRFA-1A140A00000000XXXX**, **IRFA-1A142A00000000XXXX**, **IRFA-1A143A00000000XXXX**, **IRFA-1A146A00000000XXXX**, **IRFA-1A147A00000000XXXX**, **IRFA-2A140A00000000XXXX**, **IRFA-2A142A00000000XXXX**, **IRFA-2A143A00000000XXXX**, **IRFA-2A146A00000000XXXX**, **IRFA-2A147A00000000XXXX**.

These signed attributes can be viewed by applying a CMS/PKCS#7 viewer or analyzer to the signed firmware.

When new firmware is uploaded to the device (either via website or using the ***ZIV e-NET Tool***®), it performs the following checks on the received firmware file:

- Verify that the firmware is correctly signed by certificates that comply with the PKI of the device explained in the **Credential Management** chapter. The certificates of the signatures, therefore, must be signed by CAs contained in the list of CAs of the device, must not be revoked (checked using the list of CRLs of the device) or expired and must comply with the firmware signature verification authorization policy. If any of these checks are not met, an invalid firmware message is displayed due to a signing error and the firmware upload process is aborted.
- Verify that the signed attribute information (if available, only mandatory for ZIV signing) *manifestInformation* and the particular attribute (*basicFirmwareInformation* in the case of ZIV signing) is correctly structured and valid. If it is not, an invalid firmware message is displayed due to an error in the associated information and the firmware upload process is aborted.
- Extracts the firmware version from the signed *basicFirmwareInformation* attribute. From the website you can simply upload firmware and upgrade and downgrade is always allowed. But from the ***ZIV e-NET Tool***® it is possible to upgrade and downgrade separately. In this case, the device compares the current firmware version with that of the new firmware you are trying to upload.

- If the new version is later than the current version and the operation is a downgrade, an invalid firmware message is displayed because it is not possible to downgrade to a later version and the firmware upload process is aborted.
- If the new version is lower than the current version and the operation is an upgrade, an invalid firmware message is displayed because it is not possible to upgrade to a previous version and the firmware upload process is aborted.
- If the new version is the same as the previous one, both upgrade and downgrade operations are allowed.

- Checks that the new firmware is valid for the hardware model of the device, comparing that the current hardware model of the device is included in the hardware models for which the firmware is valid present in the signed *basicFirmwareInformation* attribute. If it is not valid, an invalid firmware message is displayed because the model is not supported and the firmware upload process is aborted.
- It proceeds to decrypt the firmware with a key uploaded during the device manufacturing process. If it fails to decrypt it, an invalid firmware message is displayed because of a decryption error and the firmware upload process is aborted.

Finally, the device performs additional internal checks to ensure that the firmware structure is valid and, only then, reboots to take the new firmware, generating the **Firmware uploaded successfully** cybersecurity event. If something fails in the validation process of the new firmware (cases detailed above), the device remains with the firmware it had uploaded before the failed update, generating the **Firmware upload failed - Invalid firmware** cybersecurity event.

A device with enhanced cybersecurity capability may be manufactured without such enhanced PKI cybersecurity at the request of a customer who does not wish to include this enhanced firmware securization feature. In that case, in terms of firmware securization, it will behave as a device with basic cybersecurity and both upgrade and downgrade options of ***ZIV e-NET Tool***® will work the same, without version checking.

8.3 Firmware Upload Methods

In order to upload firmware, the user must have *firmware change* permission.

There are two methods to upload firmware to the device:

- Website. To upload firmware from the web interface, one of the LAN ports and the HTTP or HTTPS services must be enabled. It is recommended to use HTTPS connection instead of HTTP because the data is encrypted. Access the ***Reflash*** menu, authenticating first if necessary.

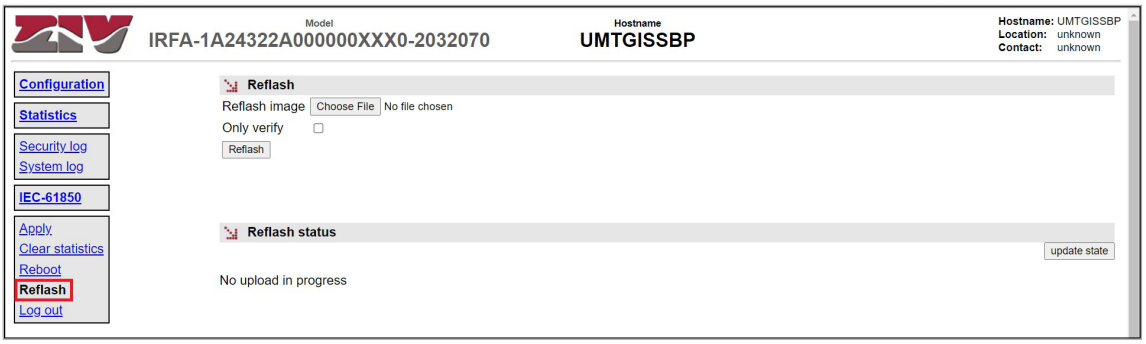


Figure 37 Uploading Firmware via Website.

The firmware file to be uploaded to the device is selected by clicking on the ***Choose File*** button.

The ***Only verify*** option is used to validate if the firmware is valid, but without updating the device with the new firmware.

Pressing the ***Reflash*** button starts the firmware upload process to the device (or validation if ***Only verify*** is checked). The firmware upload/validation progress is displayed in the ***Reflash status*** window. This progress is refreshed periodically or can be refreshed on demand by pressing the ***update state*** button.

As indicated above, from the website it is always possible to upgrade and downgrade the firmware version, no version checking is performed.

- Configuration tool. To upload firmware from the ***ZIV e-NET Tool®*** it is necessary that some of the LAN ports and the SSH and SFTP services are enabled. This is accessed from the ***Tools→Device access*** menu.

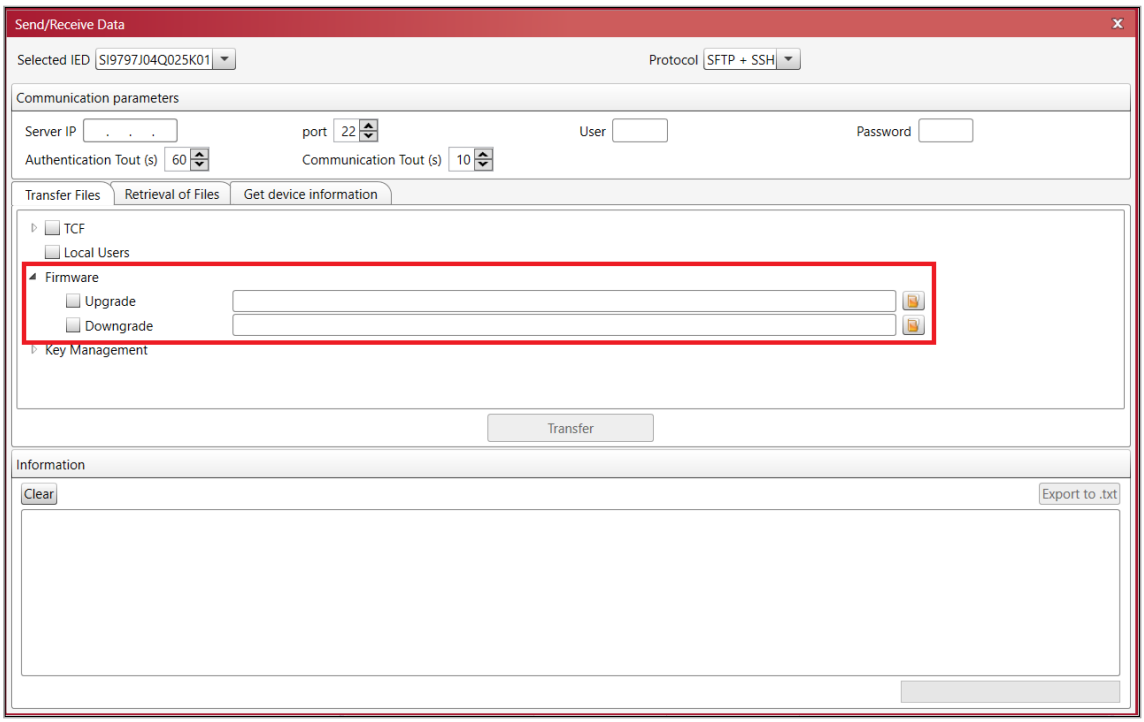


Figure 38 Firmware Upload with Configuration Tool.

In this window it is verified that **SFTP + SSH** is selected in ***Protocol***, the device IP address and port data are configured (it must coincide with the SSH/SFTP port of the device) and the user credentials (user and password) are entered. In the ***Transfer Files*** tab, under the ***Firmware*** option, there are two options:

- ***Upgrade***. Option for uploading a firmware version equal to or higher than the current one.
- ***Downgrade***. Option to upload a firmware version equal to or lower than the current version.

In both options, the firmware file to be sent to the device must first be selected by pressing the button on the right. After selecting the desired option, the firmware is uploaded to the device by clicking on the ***Transfer*** button. The firmware upload progress is indicated in the ***Information*** window.

9. Cybersecurity Logging

One of the classic cybersecurity requirements is *non-repudiation*, which is to avoid the possibility of denying that the device performed an action or claiming that the device performed an action when it really did not. The main countermeasure to ensure *non-repudiation* is the recording of cybersecurity activities or events.

The device has a cybersecurity event log. These events, in addition to being displayed online on the website of the device and stored in a file for later collection, can be sent to a centralized cybersecurity system using the **Syslog** protocol, complying with RFC5424, using a format strongly based on IEC 62351-14 and complying with IEEE 1686 and IEC 62443 standards. This will allow the centralized system to analyze cybersecurity events instantly and detect and mitigate possible attacks, including coordinated attacks on several devices simultaneously.

9.1 Syslog Format

For the Syslog format of cybersecurity events and their transmission to possible Syslog servers, see Annex A, **Syslog Protocol**.

9.2 Events

The cybersecurity events that can be generated in the device are listed in the following table. For each event, its universal identifier (see ***ID*** field in Annex A, **Syslog Protocol**), its severity (event or alarm), its associated text (see ***Text*** field in Annex A, **Syslog Protocol**) and a brief description are indicated.

Id	Severity	Text	Description
***Access Control* Events**			
0000001	Event	Login successful	Successful login (authentication).
0000038	Event	Logout	Logout (manual or automatic).
0000039	Event	Login failed	Authentication failure.
0000069	Alarm	Login failed – too many user sessions	Authentication failed due to no sessions available.
0000070	Alarm	Login failed – user rejected due to role concurrency	Authentication failed because it cannot authenticate due to role concurrency.
0000071	Alarm	Logout – session closed by other user	Logout due to being expelled by a user.
0000072	Event	CLI access initiated	CLI access initiated (may be by SSH or Telnet).
0000073	Event	FTP access initiated	FTP access initiated (via SSH).

Id	Severity	Text	Description
***User and Credential Management* Events**			
0000030	Event	User account created successfully	User created.
0000031	Event	User account modified successfully	User modified.
0000032	Event	User account deleted successfully	User deleted.
0000043	Event	Role created successfully	Role created
0000044	Event	Role modified successfully	Role modified
0000045	Event	Role deleted successfully	Role deleted
0000065	Alarm	Security credentials changed successfully	Successful change of operation PKI credentials.
0000066	Alarm	Security credentials change failed	Failed change of operation PKI credentials.
0000067	Event	LDAP repository not accessible	The LDAP remote authentication server is not available.
0000068	Event	RADIUS server not accessible	The RADIUS remote authentication server is not available.
0000111	Alarm	Default security credentials changed successfully	Successful change of default PKI credentials.
0000112	Alarm	Default security credentials change failed	Failed change of default PKI credentials.
0000113	Alarm	Security credentials restored to default due to integrity problems	An integrity error has been detected in the PKI information and certain values have been restored to their default values.
***Settings and Configuration Change* Events**			
0000012	Event	Parameters changed successfully	Change of settings. The number of settings that have been changed is indicated in parentheses.
0000015	Event	Configuration downloaded	Successful configuration download.
0000016	Event	Configuration uploaded successfully	Successful configuration upload.
0000018	Alarm	Configuration upload failed – invalid configuration	Wrong configuration upload (invalid file).
***Firmware Change* Events**			
0000020	Event	Firmware uploaded successfully	Successful firmware upload.
0000022	Alarm	Firmware upload failed – invalid firmware	Wrong firmware upload (invalid firmware).
***Device Restart* events**			
0000023	Alarm	Device reset to factory default	Device reset to factory settings.
0000024	Alarm	Manual reset	Device manually restarted.
0000025	Alarm	IED startup	Device started.

Id	Severity	Text	Description
		***Hardware Change* Events**	
0000026	Alarm	Hardware change detected	Hardware change detected (valid model, HW compatible).
0000027	Alarm	Hardware change detected – invalid hardware	Hardware change detected (invalid model, HW incompatible).
		***Hardware/Software Error* Events**	
0000075	Alarm	Critical error - Read/write settings error	Existence of incorrect settings or loss of the value of a setting. Corresponds to alarm 0x00000001 of the device.
0000076	Alarm	Non-critical error - Local port HW error	Error when operating on the front USB communications port. Corresponds to alarm 0x00000002 of the device.
0000077	Alarm	Critical error - Protection error	Protection error, protection not operative. Corresponds to alarm 0x00000004 of the device.
0000078	Alarm	Critical error - ADC error	Error in the ADC or in the auxiliary microcontroller that receives and controls the samples. Corresponds to alarm 0x00000008 of the device.
0000079	Alarm	Critical error - Digital I/O error	Problems in the digital I/O module. Corresponds to alarm 0x00000010 of the device.
0000080	Alarm	Critical error - Flash error	Flash memory problems. Corresponds to alarm 0x00000020 of the device.
0000081	Alarm	Critical error - Internal power failure	Internal power failure. Corresponds to alarm 0x00000040 of the device.
0000082	Alarm	Non-critical error - IEC 61850 error	Problems with data files, CID attached to the SW, unknown device model, inability to choose a data model or error interpreting a previously validated CID. Corresponds to alarm 0x00000080 of the device.
0000083	Alarm	Non-critical error - Error in configuration	The error occurs if the system cannot be configured with the requested configuration. Corresponds to alarm 0x00000100 of the device.

Id	Severity	Text	Description
***Hardware/Software Error* Events**			
0000084	Alarm	Critical error - Program error	Unexpected problems in the SW that prevent its operation. Corresponds to alarm 0x00000200 of the device.
0000085	Alarm	Non-critical error - Remote port 1 error	Remote port 1 hardware error. Corresponds to alarm 0x00000400 of the device.
0000086	Alarm	Critical error - Hardware error	Incorrect hardware configuration. The HW detected by the device does not correspond to the internal models provided by the SW. Corresponds to alarm 0x00000800 of the device.
0000087	Alarm	Critical error - Microcontroller error	Error in the communication with the auxiliary microcontroller that controls the different boards. Corresponds to alarm 0x00001000 of the device.
0000088	Alarm	Non-critical error - RTC error	Problems with RTC. Corresponds to alarm 0x00002000 of the device.
0000089	Alarm	Critical error - Transducer input error	Error in the communication with the input transducer. Corresponds to alarm 0x00004000 of the device.
0000090	Alarm	Non-critical error - Remote port 2 error	Remote port 2 hardware error. Corresponds to alarm 0x00008000 of the device.
0000091	Alarm	Critical error - External flash error	Problem in flash memory external to the CPU board. Corresponds to alarm 0x00010000 of the device.
0000092	Alarm	Critical error – 87L microcontroller error	Internal communications error between microcontrollers of the 87L communications board. Corresponds to alarm 0x00020000 of the device.
0000093	Alarm	Non-critical error – 87L communication error	Error in the communication with the 87L communications board. Corresponds to alarm 0x00040000 of the device.
0000114	Alarm	Certificate validation failed – certificate expired	The certificate received by communications is expired (or not yet valid).
0000115	Alarm	Certificate validation failed – certificate revoked	The certificate received by communications is revoked.

Id	Severity	Text	Description
***Communication Certificate Validation* Events**			
0000116	Alarm	Certificate validation failed – certificate signature check failed	The certificate received by communications is not trusted (it has not been successfully validated against the list of CAs of the device).
0000117	Alarm	Certificate validation failed – certificate does not comply with authorization policies	The certificate received by communications does not comply with the authorization policies (it has not been successfully validated against the authorization policy file of the device).
***Ports Enabling / Disabling* Events**			
0000049	Event	Local port disabled	Local port disabled.
0000050	Event	Local port enabled	Local port enabled.
0000051	Event	Remote port 1 disabled	Remote port 1 disabled.
0000052	Event	Remote port 1 enabled	Remote port 1 enabled.
0000053	Event	Remote port 2 disabled	Remote port 2 disabled.
0000054	Event	Remote port 2 enabled	Remote port 2 enabled.
0000055	Event	LAN port 1 disabled	LAN port 1 disabled
0000056	Event	LAN port 1 enabled	LAN port 1 enabled
0000057	Event	LAN port 2 disabled	LAN port 2 disabled
0000058	Event	LAN port 2 enabled	LAN port 2 enabled
0000059	Event	LAN port 3 disabled	LAN port 3 disabled
0000060	Event	LAN port 3 enabled	LAN port 3 enabled
0000061	Event	LAN port 4 disabled	LAN port 4 disabled
0000062	Event	LAN port 4 enabled	LAN port 4 enabled
0000063	Event	USB port disabled	USB port disabled
0000064	Event	USB port enabled	USB port enabled
Other events			
0000028	Event	Date and time set successfully	Change in device date/time > 100ms.
0000029	Event	Security events log downloaded	Cybersecurity events log downloaded.
0000041	Event	USB connected	It occurs when a USB pendrive is connected to the USB-A port.
0000042	Event	USB disconnected	It occurs when a USB pendrive is disconnected from the USB-A port.
0000074	Alarm	Unauthorized physical access detected	Detection of unauthorized physical access, for the moment, removal/insertion of hardware boards (except CPU and power supply) with the device powered up.
0000106	Event	CLI command performed “XXXX”	Indicates that the XXXX command has been successfully executed by CLI. XXXX corresponds to the complete command executed on the CLI.

9.3 Storage File

The device has the capacity to hold at least 2048 cybersecurity events in a circular buffer (FIFO).

The events are stored in a FLASH file, so that they are not lost in the event of a shutdown.

The size of the file is 256 kB, and the number of events it can hold will depend on the text occupied by each event with its additional information, but always greater than 2048 events.

The file is called */audit/security.log* and cannot be modified or deleted in any way.

9.4 Viewing and Downloading the File

Only a user with *audit log* permission can view the cybersecurity events and/or download the file.

There are several ways to access the cybersecurity event file (*security.log*):

- By FTP or SFTP file transfer. In this case the file location is */audit/security.log*.
- From the website, ***Security log*** menu. The events are displayed directly on the web page in order of occurrence. In addition, the file can be downloaded by clicking the ***Download File*** button.

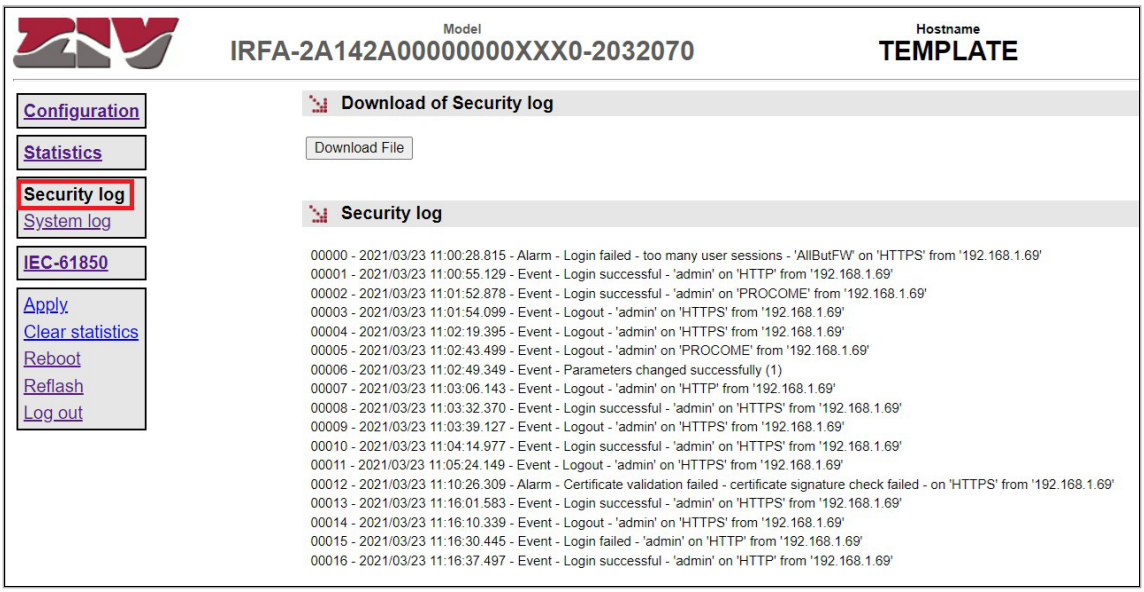


Figure 39 Display of Cybersecurity Events on Website.

- Using the ***ZIV e-NET Tool®*** configuration tool. To collect cybersecurity events using the ***ZIV e-NET Tool®***, access the ***Tools→Device access*** menu. In this window, check that **SFTP + SSH** is selected in ***Protocol***, configure the device IP address and port data (it must match the SSH/SFTP port of the device) and enter the user credentials (username and password). In the ***Retrieval of Files*** tab, check the ***Logs→Security Events*** option and click ***Retrieval*** to download the cybersecurity events file. The download progress is displayed in the ***Information*** window.

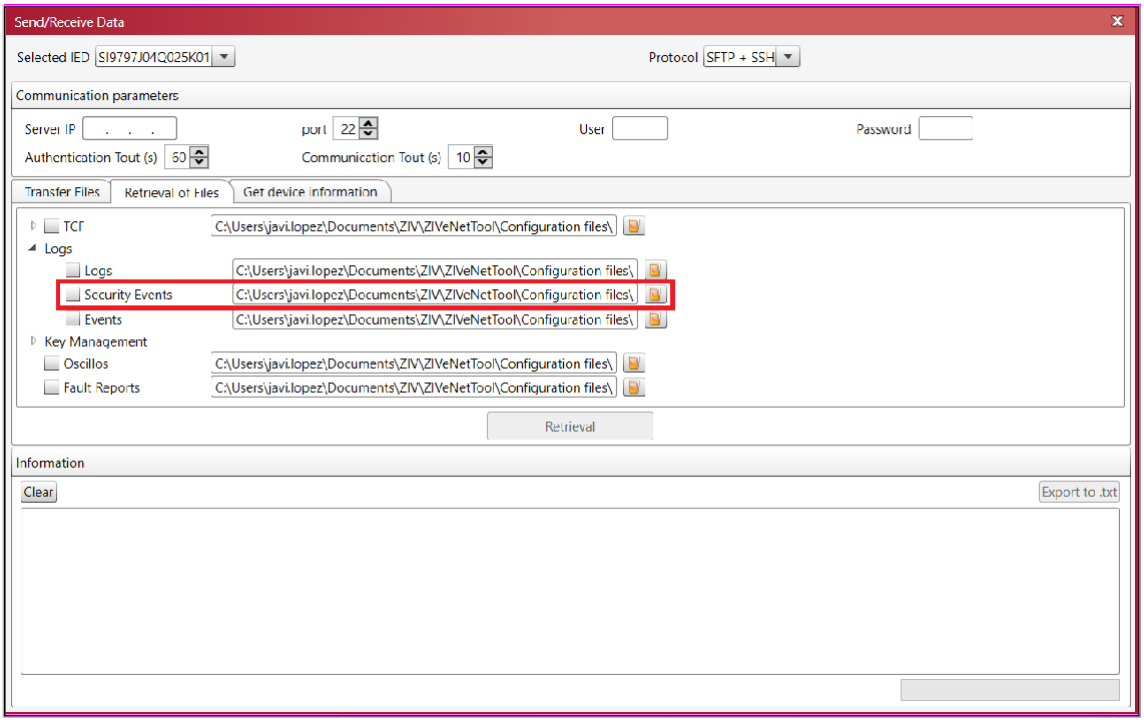


Figure 40 Collection of Cybersecurity Events with Configuration Tool.

Each time the cybersecurity event file is downloaded from the device, the **Security events log downloaded** cybersecurity event is generated.

Cybersecurity events are displayed on the website and stored in the file *security.log* with the following format:

YYYY/MM/DD hh:mm:ss.mmm - eveala - evtext - 'user' on 'service' from 'ipaddr' (interface)

where:

- ***YYYY/MM/DD hh:mm:ss.mmm***: it is the date and time of the event (up to milliseconds). Corresponds to the TIMESTAMP field of the Syslog HEADER.
- ***eveala***: reflection of Syslog HEADER field PRI. It will be able to take the values:
  - Event (PRI=<108>)
  - Alarm (PRI=<105>)

- ***evtext***: text of the event. It is the faithful reflection of the *Event* field of Syslog STRUCTURED-DATA. If after ***evtext*** there is no more information to show or it is only shown (*interface*), then the hyphen (-) following ***evtext*** will not be shown. Otherwise, it will be shown.
- ***user***: username. It is the faithful reflection of the *UsrID* field of the Syslog STRUCTURED-DATA. If the *UsrID* field is not present or has a NILVALUE ("-") value, then the text "*user*" will not be presented in the event.
- ***service***: name of the service. It is the faithful reflection of the *Param(0)* field of the Syslog STRUCTURED-DATA. If the *Param(0)* field is not present or has a NILVALUE ("-") value, then the text "*on service*" will not be presented in the event.
- ***ipaddr***: IP address of the device causing the event. It is the faithful reflection of the *PeerInfo* field of the Syslog STRUCTURED-DATA. If the *PeerInfo* field is not present or has a NILVALUE ("-") value, then the text "*from ipaddr*" will not be presented in the event.
- ***interface***: physical interface through which the event occurred. It is the faithful reflection of the *Param(1)* field of the Syslog STRUCTURED-DATA. If the *Param(1)* field is not present or has a NILVALUE ("-") value, then the text "(*interface*)" will not be presented in the event.

The following are examples of event format in file:

- Successful authentication of user "admin" by SSH from IP=192.168.1.69 with device SI3197E1Q11F01:

  ```
  2016-04-17 22:36:41.358 - Event - Login successful - 'admin' on
  'SSH' from '192.168.1.69'
  ```

- Starting up of device TEMPLATE:

  ```
  2016-04-17 21:24:32.498 - Alarm - IED startup
  ```

- Logout of user "Ramon.Perez" by HMI in the device SWT650:

  ```
  2016-04-17 23:35:49.423 - Event - Logout - 'Ramon.Perez' on 'HMI'
  ```

- Failed authentication of user “secadm” by PROCOME through local port with device IED_TEST: `2016-04-17 14:29:36.187 - Event - Login failed - 'secadm' on 'PROCOME' (LocalPort).`

Annex A. Syslog Protocol

A.1 General

Due to its simplicity and wide acceptance and use, the format chosen for sending events is **Syslog**. Throughout the evolution of Syslog, several RFCs have been published, all with the aim of achieving greater compatibility and security between the different implementations. Of these, RFC 5424 is used, as it is widely accepted and provides all the required features.

The device has a cybersecurity event log. These events, in addition to being viewed online on the website of the device and stored in a file for later collection, can be sent to a centralized cybersecurity system using the Syslog protocol, in compliance with RFC5424, using a format strongly based on IEC 62351-14 and complying with IEEE 1686 and IEC 62443 standards. This will allow such a centralized system to analyze cybersecurity events instantaneously and detect and mitigate possible attacks, including coordinated attacks on several devices simultaneously.

The following acronyms and structures will be used throughout this Annex:

UTF-8-STRING	= *OCTET (UTF-8 string as specified in RFC 3629).
OCTET	= Characters from %d00 to %d255.
SP	= Blank space (%d32), used as field separator.
PRINTUSASCII	= Characters from %d33 to %d126.
NONZERO-DIGIT	= Characters from %d49 to %d57.
DIGIT	= Character %d48 + NONZERO-DIGIT.
NILVALUE	= “-“

The structure of a Syslog event is as follows:

HEADER SP **STRUCTURED-DATA** [SP **MSG**]

The **HEADER** field has the following structure:

HEADER = PRI VERSION SP TIMESTAMP SP HOSTNAME SP APP-NAME SP PROCID SP MSGID

The **STRUCTURED-DATA** field has the following structure

STRUCTURED-DATA	= 1 * SD-ELEMENT
SD-ELEMENT	= "[" SD-ID *(SP SD-PARAM) "]"
SD-ID	= SD-NAME
SD-PARAM	= PARAM-NAME "=" %d34 PARAM-VALUE %d34
PARAM-NAME	= SD-NAME
PARAM-VALUE	= UTF-8-STRING; the characters '"' (double quotation mark: %d34)', '\' (%d92) and ']' (close bracket: %d93) must be escaped using the character '\' (%d92).
SD-NAME	= 1*32 PRINTUSASCII, except for the characters '=' (equals: %d61), ' ' (blank space: %d32), ']' (close bracket: %d93) and '"' (double quotation mark: %d34).

A.2 Syslog Format

For cybersecurity events, the MSG field is not used and, therefore, will never be present in the generated Syslog events.

Therefore, the structure of a cybersecurity Syslog event is as follows:

HEADER SP **STRUCTURED-DATA**

A.3 HEADER Format

RFC 5424 Information Fields		

HEADER	***PRI***	= "<PRIVAL>"
		Priority of the event:
		The ***PRIVAL*** field (1~3 DIGIT) is calculated as follows
		PRIVAL = (8 x ***Facility***) + (***Severity***)
		- ***Facility***: The value log audit is set =13.
		- ***Severity***:
		  - Events have severity level of "Warning" = 4
		  Event → PRIVAL = 8 x 13 + 4 = *108*.
		  - Alarms have severity level of "Alert" = 1
		  Alarm → PRIVAL = 8 x 13 + 1 = *105*.

	VERSION	1
	TIMESTAMP	Date and time of the event. The format is: ***FULL-DATE "T" FULL-TIME*** ***FULL-DATE*** = ***DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY*** ***DATE-FULLYEAR*** = 4 DIGIT ***DATE-MONTH*** = 2 DIGIT (01-12) ***DATE-MDAY*** = 2 DIGIT (01-28, 01-29, 01-30, 01-31, depending on the month). ***FULL-TIME*** = ***PARTIAL-TIME TIME-OFFSET*** ***PARTIAL-TIME*** = ***TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND "." TIME-SECFRAC*** ***TIME-HOUR*** = 2 DIGIT (00-23) ***TIME-MINUTE*** = 2 DIGIT (00-59) ***TIME-SECOND*** = 2 DIGIT (00-59) ***TIME-SECFRAC*** = 3 DIGIT representing milliseconds 000~999 ***TIME-OFFSET***="Z" (represents the date/time in UTC). UTC format is always used. Example: 2016-04-17T22:36:41.358Z
	HOSTNAME	Own IP (global or in service origin). Since the device can have several IP addresses, the value of the HOSTNAME takes the value of the IP address of the first enabled LAN port, following the order LAN1, LAN2, LAN3, LAN4.
	APP-NAME	Name of the device where the event occurred (device name). When the device has IEC 61850, the name of the device obtained from the CID configuration file will be assigned. Otherwise, it can be modified from the website and from CLI.
	PROCID	Takes the NILVALUE value = "-", as IEC 62351-14 states.
	MSGID	Takes the fixed value "IEC62351-14:1" as IEC 62351-14 states.

A.4 STRUCTURED-DATA Format

RFC 5424 Information Fields		
STRUCTURED-DATA (SD)	***SD-ID***	Takes the fixed value “62351-14@41912” as IEC 62351-14 states.
	SOE	Event Sequence Counter. It is a sequential number, which is incremented with each cybersecurity event that occurs. It is a 32-bit unsigned integer. Therefore, possible values range from 0 to 4294967295. It is encoded with a text string of up to 10 characters. It is used to order events according to their occurrence, as the date/time could be altered.
	ID	Universal identifier that identifies the event in a unique way. It is a text string of 7 characters. See column ***Id*** of the table in the ***Cybersecurity Logging→Events*** chapter.
	Text	Descriptive text of the event. It is a text string of up to 127 characters. See column ***Text*** of the table in the ***Cybersecurity Logging→Events*** chapter.
	UsrID	Identifier of the user who caused the event. Text string of up to 256 characters with the username that generated the event.
	PeerInfo	A text string of up to 39 characters that identifies the IP address (IP4 or IP6) of the machine that accessed the device and caused the event.
	Param(0)	Identifier of the service through which the event occurred. HMI and USB are considered as services. Text string with the following possible values: “LDAP”, "HTTP", "HTTPS", "FTP", "Telnet", "SSH", "MMS", "PROCOME", “HMI” and “USB”.
	Param(1)	A text string of up to 32 characters that identifies the physical interface through which the event occurred. This field is available for *Access Control* type events generated on interfaces other than LAN. Possible values are: "LocalPort", "RemotePort1" and "RemotePort2".

A.5 Examples

The following are examples of cybersecurity events in Syslog format:

- Successful authentication of user "admin" by SSH from IP=192.168.1.69 with device SI3197E1Q11F01 with IP=192.168.1.81:

```
<108>1 2016-04-17T22:36:41.358Z 192.168.1.81 SI3197E1Q11F01 - IEC62351-14:1 [62351-14@41912 ID="0000001" Text="Login successful" SOE="131" UsrID="admin" PeerInfo="192.168.1.69" Param(0)="SSH"]
```

- Starting the device TEMPLATE with IP="192.168.2.81":

```
<105>1 2016-04-17T21:24:32.498Z 192.168.2.81 TEMPLATE - IEC62351-14:1 [62351-14@41912 ID="0000025" Text="IED startup" SOE="128"]
```

- Logout of user "Ramon.Perez" by HMI on the device SWT650 with IP="192.168.3.81":

```
<108>1 2016-04-17T23:35:49.423Z 192.168.3.81 SWT650 - IEC62351-14:1 [62351-14@41912 ID="0000038" Text="Logout" SOE="132" UsrID="Ramon.Perez" Param(0)="HMI"]
```

- Failed authentication of user "secadm" by PROCOME through local port with device IED_TEST with IP=192.168.4.81:

```
<108>1 2016-04-17T14:29:36.187Z 192.168.4.81 IED_TEST - IEC62351-14:1 [62351-14@41912 ID="0000039" Text="Login failed" SOE="189" UsrID="secadm" Param(0)="PROCOME" Param(1)="LocalPort"]
```

A.6 Syslog Transmission

The device can send events via Syslog protocol to up to 3 Syslog servers.

To communicate these events in Syslog format to said servers, the UDP protocol is used (standard port UDP/514) as specified in RFC 5426.

To connect to each of the 3 possible Syslog servers, the following settings can be modified from the configuration tool and from HMI, corresponding to the IP addresses of each server and the logical port to communicate with it:

Syslog				
Configuration Tool	**HMI**	**Range**	**Step**	**Default**
IP Address (Server 1)	IP Address (Server 1)	XXX.XXX.XXX.XXX		0.0.0.0
Port Number (Server 1)	Port Number (Server 1)	1-65535	1	514
IP Address (Server 2)	IP Address (Server 2)	XXX.XXX.XXX.XXX		0.0.0.0
Port Number (Server 2)	Port Number (Server 2)	1-65535	1	514
IP Address (Server 3)	IP Address (Server 3)	XXX.XXX.XXX.XXX		0.0.0.0
Port Number (Server 3)	Port Number (Server 3)	1-65535	1	514

Sending events to a server can be disabled by setting the IP address of the Syslog server to 0.0.0.0. The Syslog client can also be deactivated by disabling the Syslog service, as described in the **Communication Ports and Services** chapter.