Cybersecurity Policy Manual

MU214604 Rev. E

March 30, 2026

www.altus.com.br

According to current legislation in Brazil, the Consumer Defense Code, we are giving the following information to clients who use our products, regarding personal safety and premises.

The industrial automation equipment, manufactured by Altus, is strong and reliable due to the stringent quality control it is subjected to. However, any electronic industrial control equipment (programmable controllers, numerical commands, etc.) can damage machines or processes controlled by them when there are defective components and/or when a programming or installation error occurs. This can even put human lives at risk. The user should consider the possible consequences of the defects and should provide additional external installations for safety reasons. This concern is higher when in initial commissioning and testing.

The equipment manufactured by Altus does not directly expose the environment to hazards, since they do not issue any kind of pollutant during their use. However, concerning the disposal of equipment, it is important to point out that built-in electronics may contain materials which are harmful to nature when improperly discarded. Therefore, it is recommended that whenever discarding this type of product, it should be forwarded to recycling plants, which guarantee proper waste management.

It is essential to read and understand the product documentation, such as manuals and technical characteristics before its installation or use. The examples and figures presented in this document are solely for illustrative purposes. Due to possible upgrades and improvements that the products may present, Altus assumes no responsibility for the use of these examples and figures in real applications. They should only be used to assist user training and improve experience with the products and their features.

Altus warrants its equipment as described in General Conditions of Supply, attached to the commercial proposals.

Altus guarantees that their equipment works in accordance with the clear instructions contained in their manuals and/or technical characteristics, not guaranteeing the success of any particular type of application of the equipment.

Altus does not acknowledge any other guarantee, express or implied, mainly when end customers are dealing with third-party suppliers. The requests for additional information about the supply, equipment features and/or any other Altus services must be made in writing. Altus is not responsible for supplying information about its equipment without formal request. These products can use EtherCAT® technology (www.ethercat.org).

COPYRIGHTS

OPEN SOURCE SOFTWARE NOTICE

To obtain the source code under GPL, LGPL, MPL and other open source licenses, that is contained in this product, please contact opensource@altus.com.br. In addition to the source code, all referred license terms, warranty disclaimers and copyright notices may be disclosed under request.

Contents

1. Introduction

Cybersecurity plays a crucial role in the industrial automation environment. With the alarming increase in security incidents in factories, plants and other automated applications, effective measures to protect these systems becomes imperative. This document's objective is to present and justify the cybersecurity measures implemented in Altus products, notably MasterTool, the development environment for programmable logical controllers (PLCs), and the Nexto, Nexto Xpress and Hadron Xtorm series.

Government institutions such as ICS-Cert and the German Federal Department for Information Security (BSI) have been closely monitoring these increasing incidents. Given this scenario, the development of methodologies to ensure the integrity and protection of systems has become an urgent necessity. A significant milestone in this regard is the international standard guideline IEC 62443, initially published by the Industrial Automation and Control Systems Security Committee (ISA99) of the International Society of Automation (ISA), often referred to as the ISA/IEC 62443 standard.

The scope of cybersecurity measures encompasses the protection of various aspects, including the availability of controller functionalities, application functionality, the confidentiality of source code and the application, the integrity of application functions, of the development system, and of the components employed, as well as the authenticity of the controller and its data.

In this context, this document highlights the cybersecurity strategies adopted by Altus and its products, aiming to safeguard customers and their industrial operations from increasingly sophisticated and persistent threats. The use of the ISA/IEC 62443 standard as a solid reference reflects a commitment to excellence in safeguarding the industrial automation environment against potential cyber risks.

In addition to aligning the products with the ISA/IEC 62443 standard, a study was conducted involving another important cybersecurity-related document: Module 5 of the ONS (Operador Nacional do Sistema) Operation Procedures Manual. Chapter 8 provides a description of how each proposed requirement relates to Altus products.

The content in this document covers the functionalities of the following products: Mastertool IEC XE, NX3003, NX3004, NX3005, NX3008, NX3010, NX3020, NX3030, XP300, XP315, XP325, XP340, XP350, XP351, HX3040, and NL717.

2. Terms and Definitions

2.1. Vulnerabilities

Automation systems might be attacked through different points in its structure:

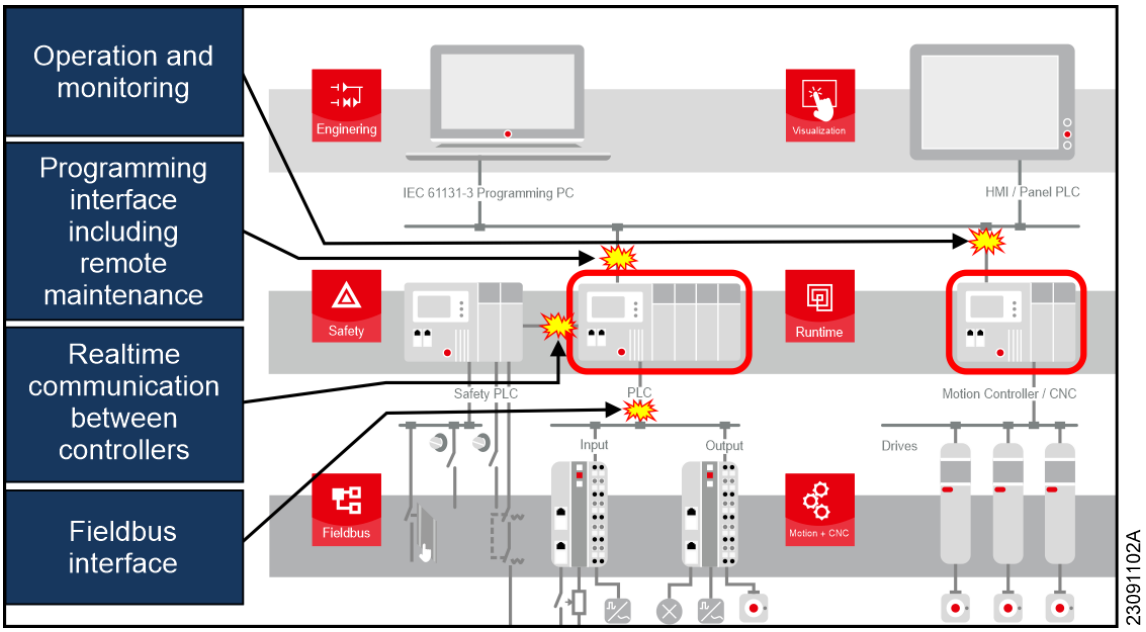

Figure 1: Possible vulnerabilities of a typical automation system.

2.2. Threat

It refers to a set of circumstances and associated sequence of events with the potential to adversely affect operations (including mission, functions, image, or reputation), assets, control systems, or individuals through unauthorized access, destruction, disclosure, modification of data, and/or denial of service. In essence, it is the possibility of malicious or unwanted events occurring that compromise the integrity, confidentiality, or availability of resources and information in an industrial automation environment.

2.3. Security Levels

To address this comprehensive approach, the ISA/IEC 62443 standard establishes four main levels of protection on an ascending scale, each tailored to address different threats:

- Level 1: Occasional and accidental threats;
  Examples: Hard drive failure, operational errors.
- Level 2: Intentional threats through simple means;
  Example: Successfully guessing a password.
- Level 3: Intentional threats through sophisticated means;
  Example: Utilizing hacking tools.
- Level 4: Intentional threats through sophisticated means and extensive resources.
  Examples: Specialized development, knowledge of the application, or insider corruption.

2.4. Programmable Controller

A computer used in industrial automation systems, which can also be referred to as a PLC (Programmable Logical Controller) or simply a Controller, is a crucial component in these systems. These devices can be targeted in cyberattacks due to their unique characteristics, and they rely on programming designed for specific applications, which can potentially introduce vulnerabilities. The Altus controllers, discussed in this document, belong to the Nexto, Nexto Xpress, or Hadron Xtorm product lines.

2.5. MasterTool

MasterTool IEC XE is a comprehensive tool for programming, debugging, configuring, and simulating user applications. The software is based on the integrated tool concept, providing flexibility and ease of use, allowing users to program in six languages defined by the IEC 61131-3 standard: Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), and Continuous Function Chart (CFC).

2.6. Protected Environment

Every system and piece of equipment needs to be accessed during its installation, operation, and maintenance. However, unrestricted access should be avoided to prevent operational failures and unintentional or intentional damage to the product. To achieve this, the system should be divided into subsystems, with controlled access to each subsystem, ensuring that only authorized individuals can access them, thereby safeguarding the environment.

3. Responsibilities of Different Agents in Industrial Systems Security

In the configuration of industrial control applications, several active parties and suppliers are involved: software and hardware component providers, system integrators or builders of industrial control applications, and operators. As information technology security is a comprehensive task, all the mentioned parties must make a significant effort to protect the application against attacks.

- Software Provider:
  - Analyze assets and threats;
  - Provide approved security measures;
  - Supply technical documentation;
- Automation Component Provider:
  - Analyze assets and threats;
  - Implement software and hardware security measures;
  - Supply technical documentation;
- System Integrator and Machinery Manufacturer:
  - Analyze assets and threats;
  - Implement software and hardware security measures;
  - Implement system security measures;
- Plant Operator/Manager:
  - Analyze assets and threats;
  - Implement software, hardware, and system security measures;
  - Test, audit, and certify the system;
  - Train employees;

4. General Protections for Industrial Automation Systems

First and foremost, all commonly known security measures for computers should be applied in networks with industrial automation equipment, such as:

- Virus protection;
- Strong passwords that are regularly changed;
- Firewall protection;
- Use of VPN tunnels for inter-network connections;
- Caution when dealing with removable storage devices like USB removable media drives.

Additionally, it is mandatory to have well-defined user and permission management for access to controllers and their interconnected networks.

4.1. Use in a Protected Environment

Locating the controller in a protected environment is absolutely necessary to prevent accidental or unauthorized intentional access to the controller or its application, which is crucial for the operation of the machinery or installation.

This protected environment can be, for example, within:

- Locked electrical control cabinets with no external communication access;
- An intranet network with well-defined user rights and no external access;
- A network with internet access only through a well-configured firewall via a VPN tunnel.

Clearly, the level of protection decreases as you move down this list.

To establish such a protected environment, several rules must be followed:

- Keep the trusted network as small as possible and independent of other networks;
- Safeguard cross-communication between controllers and communication between controllers and field devices using standard communication protocols (fieldbus systems) through appropriate measures;
- Isolate and strictly separate these networks from common access;
- Use fieldbus systems exclusively in protected environments, as they lack additional security measures like encryption. Open physical or data access to fieldbus systems and their components poses a significant security risk.

4.2. Security-Aware Users

Security-aware users play a crucial role in cybersecurity as most reported security incidents occur unintentionally due to handling errors or device misuse. Therefore, both machine and facility manufacturers, as well as operators, need to be aware of potential threats and the necessary infrastructure measures to prevent them. To achieve this goal, it is advisable for users to participate in specialized training provided by security experts, whether within the company or by external professionals. These training sessions are designed to empower users to adopt proper security practices and understand how to apply the appropriate protective measures in the development and operation of industrial controllers.

5. Security Measures Present in MasterTool

This chapter provides information on the cybersecurity features within the MasterTool program, emphasizing their significance, and how to locate them in the product manuals. Below the subchapter titles, the relevant component requirement (RC) from the IEC 62443-4-2:2019-02 standard to which each feature corresponds is specified.

5.1. User Management

The configuration of users and groups (CR 1.3 of the standard) is done in the *Project* dialog in the *Project Settings* window. In the *Users and Groups* tab, a user can be registered for each person working on the project, and these users can be organized into groups. During user creation, passwords are not evaluated based on strength, but a strong password policy, if followed, meets the CR 1.7 requirement of the standard. In this screen, it is also possible to configure the maximum number of authentication attempts, meeting the CR 1.11 requirement of the standard.

More detailed information on the use of each tool presented in this chapter can be found in the chapter "User Management and Access Rights" of the Mastertool Manual.

5.1.1. User Management at Project Levels
RC 1.1, RE (1), 1.3, 1.4, 1.5, 1.7, 1.10, 1.11, 2.1, RE(1), RE(2), 2.5, 2.6 e 3.3 from IEC 62443-4-2 standard

MasterTool offers the capability of read/write protection for individual objects within the project with user administration. This protection can be defined for menu commands as well as for specific object types (e.g., task creation, POUs, methods, GVLs, etc.) or existing objects in the project (such as project settings or POUs or specific tasks).

Through user administration, it's possible to limit the range of functionalities in a more granular way, allowing tailored access rights to specific security needs, thereby safeguarding the confidentiality of intellectual property as well as the integrity of the application code.

User management in a project is only useful when combined with access rights management. In a new project, access rights are not automatically defined but are instead configured to a default value, meaning that rights are usually "granted". During project execution, each right can be explicitly granted or denied and reconfigured to the default setting. Access rights management is done through the *Permissions* dialog or — for object access rights — through the *Access Control* dialog (which is part of the objects *Properties* dialog).

Once access to a function is restricted to the *Everyone* group, logging in as a user with access permissions is required to use it. During login, password characters are hidden by asterisks.

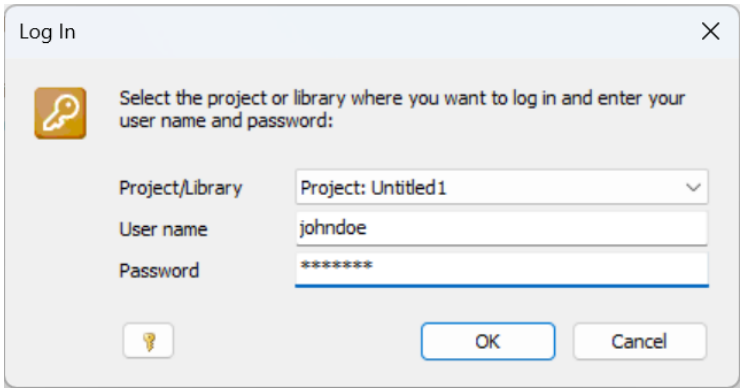

Figure 2: Login screen.

Creating Users and Adding Them to Groups

A new user can be added to the project via *Project* > *Project Settings* > *Users and Groups* > *Users* > *Add...*. In the same menu, users can be assigned to a group, ensuring that all group settings apply to the new user. Permissions for users within the project are defined at the group level. Each user is uniquely identified by their *Login Name*.

The software does not evaluate password strength, but implementing and enforcing an internal policy can help ensure user security in compliance with item 1.7 of the standard.

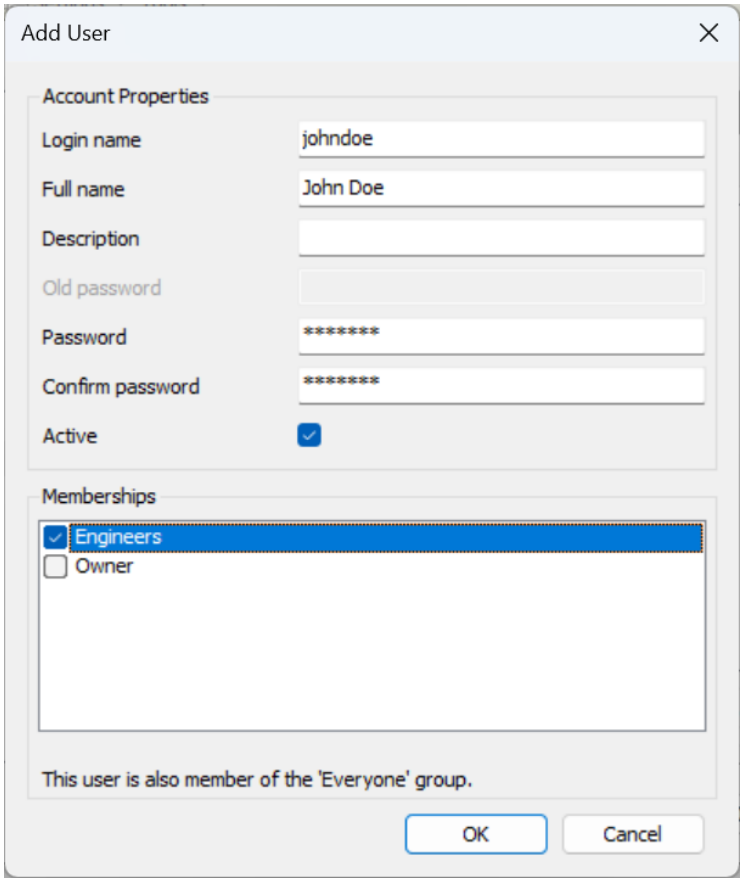

Figure 3: Creating a new user.

User creation must be performed by a user in the *Owner* group. In Mastertool, the project starts with an *Owner* user, which, by default, has an empty password.

User Management

The *User Manager* allows administrators to define user access levels, make modifications, and assign users to predefined groups. To grant a specific user access, they must be added to a group, which in turn defines different levels of access. If a user attempts an operation that requires authorization, Mastertool will prompt for valid credentials to confirm access or modifications.

In Mastertool, the administrator user is the *Owner*, which, by default, has an empty password. To log in as this user, navigate to *Project* > *User Management* > *User Login...*. Only users in the *Owner* group can add or edit user and group configurations, as well as change user passwords.

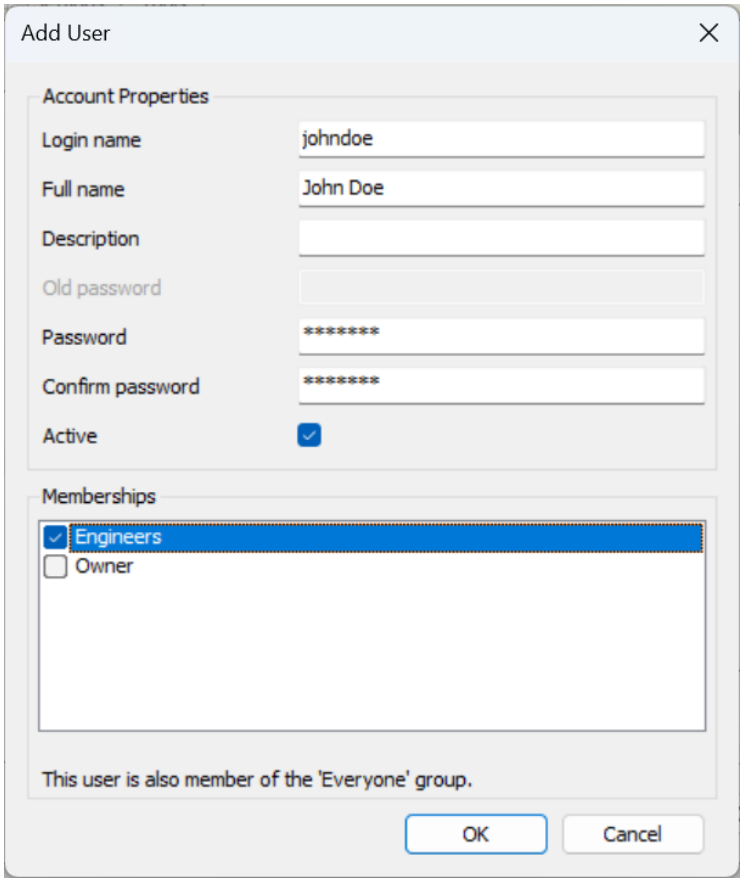

Figure 4: Creating a new user.

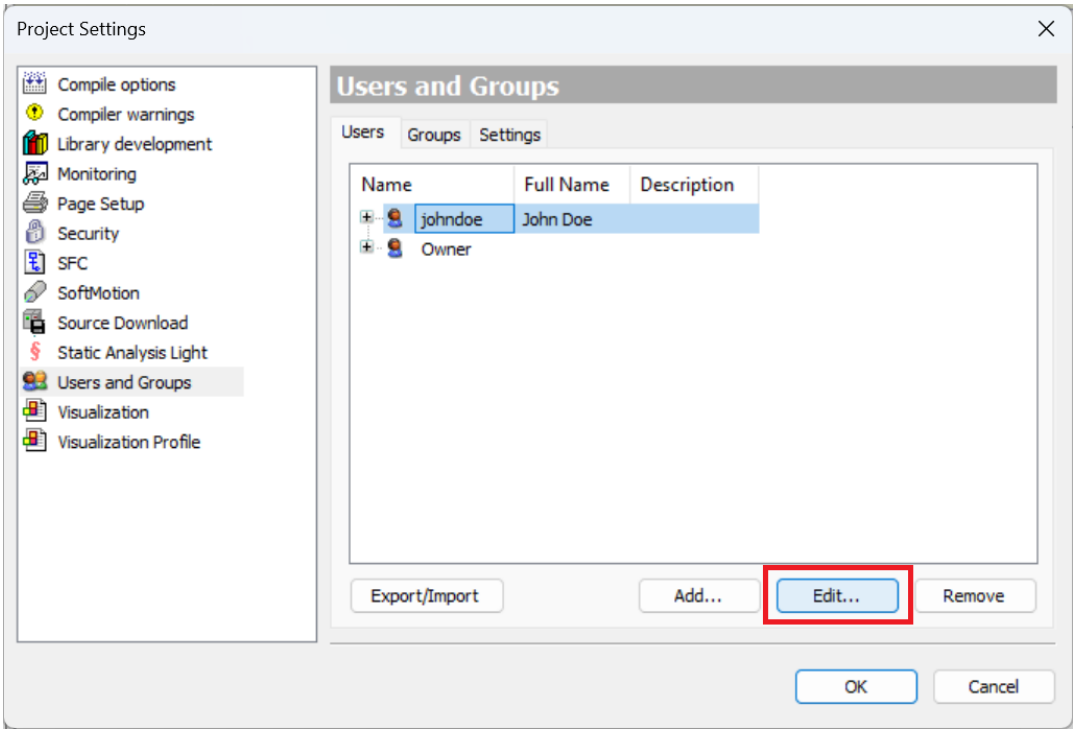

Figure 5: User edit button.

User activation is also performed in the edit screen by selecting the *Active* checkbox. A user may be deactivated either manually or due to exceeding the login attempt limit. This setting must be configured by a system administrator.

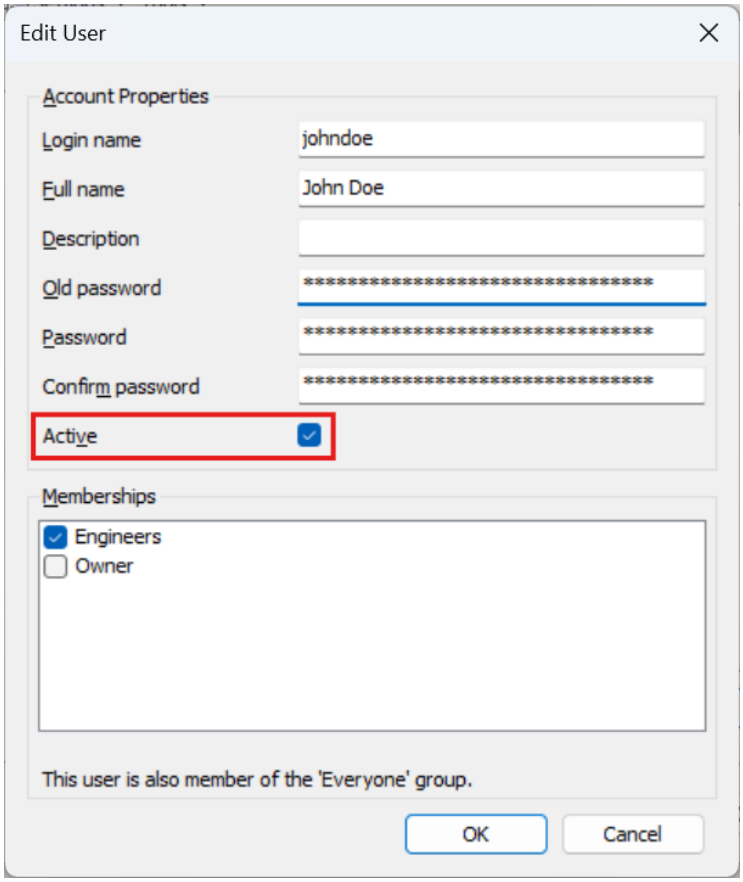

Figure 6: User editing.

Once logged in as *Owner*, a group can be created via *Project* > *Project Settings* > *Users and Groups* > *Groups* > *Add...*.

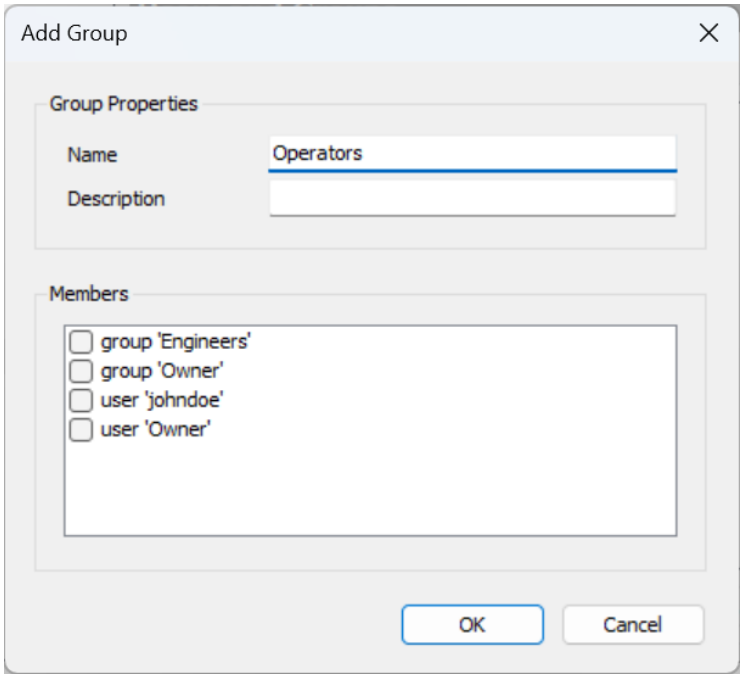

Figure 7: Adding groups.

To check a user group's project permissions, go to *Project* > *User Management* > *Permissions*. This menu allows permissions to be granted or removed. Within the folders, all project permission-related commands are listed. Clicking on a command opens a panel on the right, where permissions can be assigned or denied for registered groups.

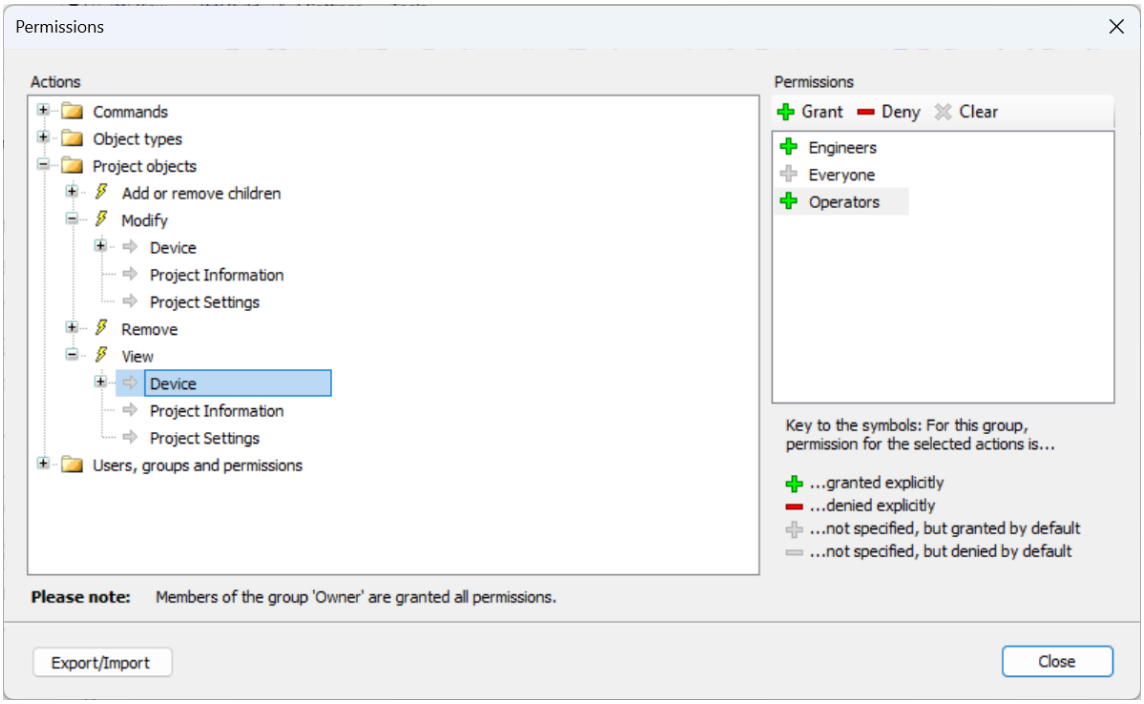

Figure 8: Group-based permissions screen.

Configuring Login Options

To prevent brute force login attacks, the number of authentication attempts can be configured via *Project* > *Project Settings* > *Users and Groups* > *Settings*. If the limit is exceeded, the user is deactivated and will remain so until an administrator

reactivates them. Section *User Management* in Chapter 5.1.1 explains how to reactivate a user.

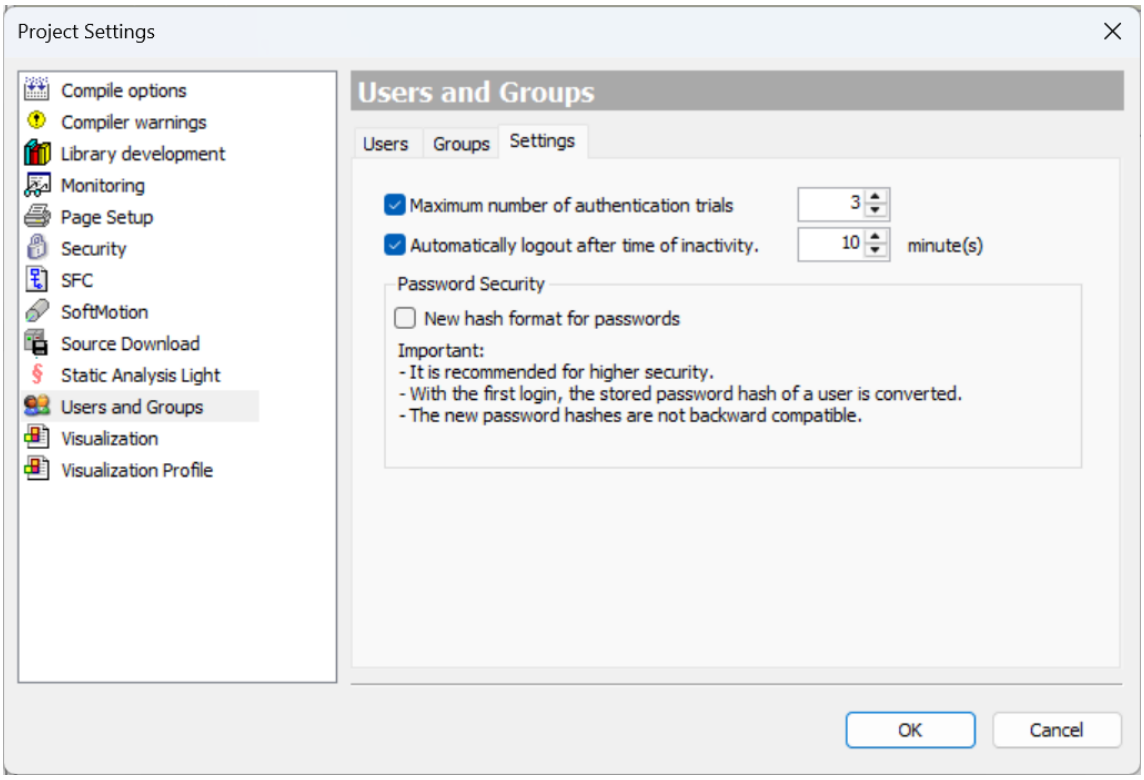

Figure 9: User and Group Configuration screen.

On the same screen, a maximum inactive session time can be set, after which the user must log in again. This setting also applies to remote access to devices.

Verifying Identification and Access Control for Unauthorized Accounts

To verify that an unauthorized user cannot access the project, go to *Project* > *User Management* > *User Login* and enter the credentials of a non-existent user. The following message should appear on the screen:

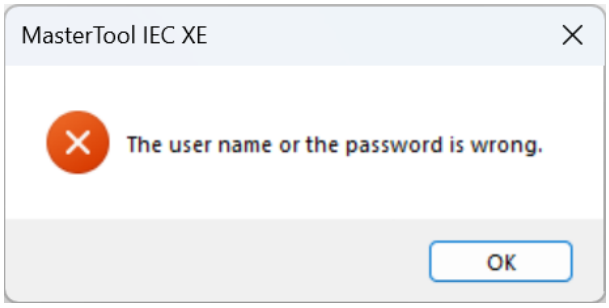

Figure 10: Error message when entering invalid credentials.

5.1.2. Managing Users of the Integrated Visualization
RC 2.1 from IEC 62443-4-2 standard

The integrated visualization of MasterTool allows for direct operation of the controller and the application. It is strongly recommended to separate operation into different sections or screens according to their level of functional and security influence. MasterTool provides the capability to protect individual visualization elements as well as entire visualization screens in the project through a special user visualization management.

This user management allows for limiting the scope of functionality for specific operators. Critical safety operation modes, such as exporting production data, the plant startup and shutdown process, and access to dedicated service functions, can be restricted to operators with explicitly assigned permissions, ensuring the confidentiality of intellectual property as well as the availability and reliability of the machine or plant process.

5.2. PLC IP Settings

RC 1.2 from IEC 62443-4-2 standard

The *Easy Connection* feature allows for scanning all PLCs in the same network.

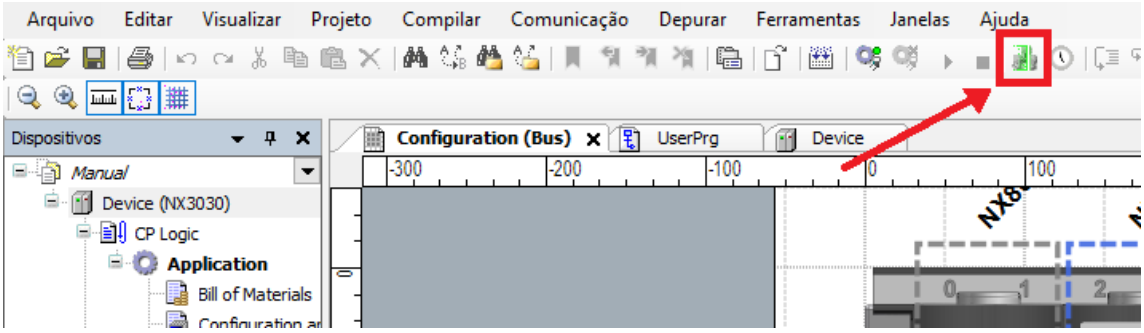

Figure 11: Easy Connection button.

After the scan, a list of connected PLCs will appear. From this list, it is possible to identify the device by its IP or by clicking on *Identify device*. When using *Identify device*, the PLC's DG LED will start blinking rapidly, allowing you to physically verify which PLC it is before logging in.

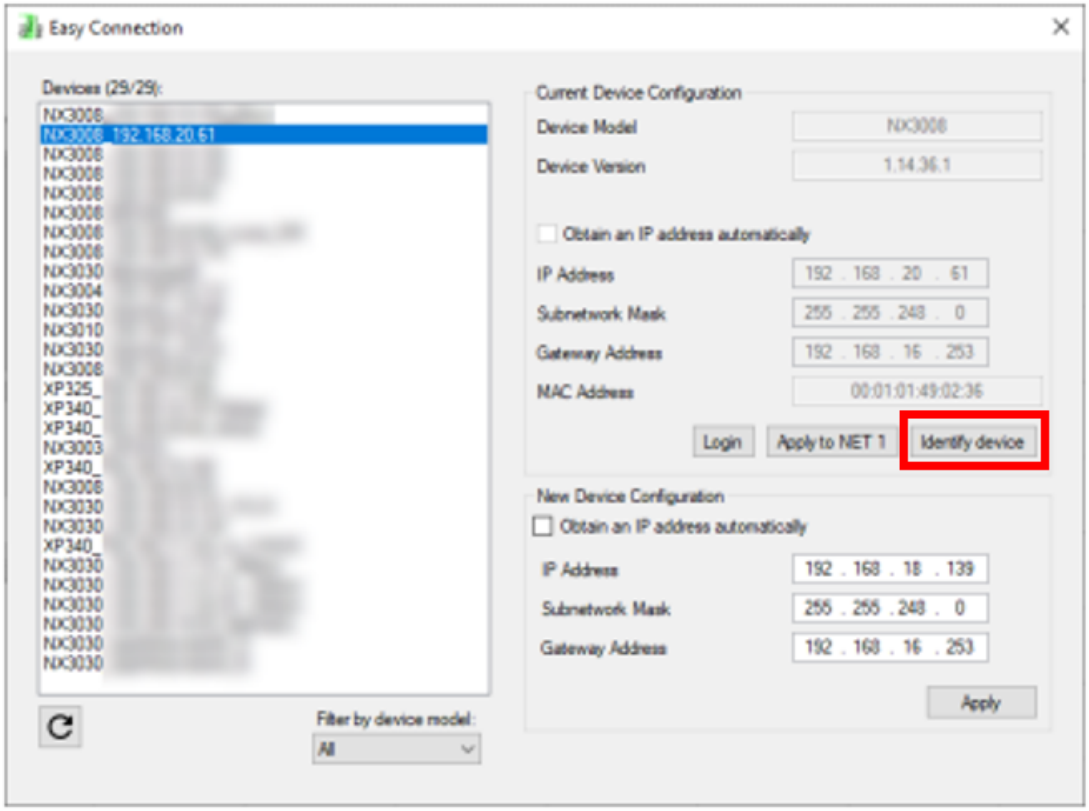

Figure 12: Easy Connection.

For a controller from the Altus NX series, another way to check the IP address is through the button on its top, which displays various diagnostic information on the screen, including the IP address.

To change the device's IP, configure the new address, subnet, and gateway in the *NET (1, 2, or 3)* interface, and then log in to the device using the current IP that is still configured in the *Device* properties for communication with the computer.

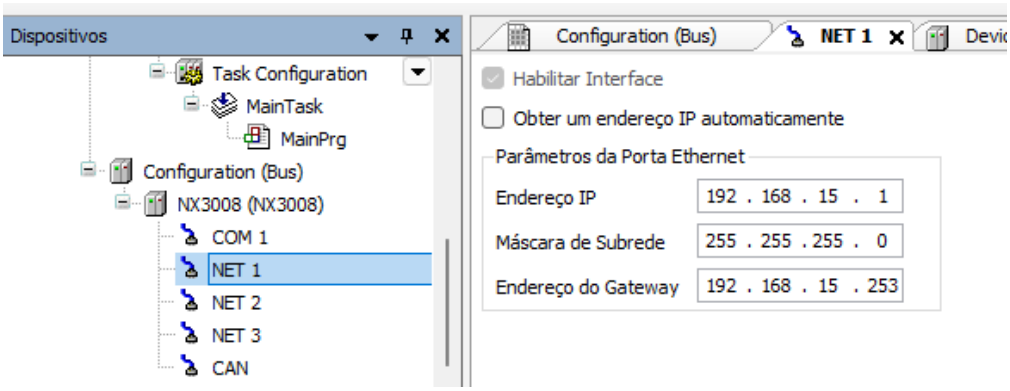

Figure 13: Network configuration changes via Mastertool.

Another way to edit the network settings is through the device's web interface, described in Section 6.5.2.

5.3. Webvisu Communication Ecryption

RC 3.1 from IEC 62443-4-2 standard

To prevent the interception of communication between the controller and the web browser on the computer, you can use an encrypted HTTPS connection. This can be configured with a self-signed certificate or with one generated by a Certificate Authority (CA). In the *Device Security Settings* screen, the use of HTTPS can be configured as mandatory or also allow HTTP connections.

This functionality is implemented in the *Device* screen > *Communication Settings* > *Device* > *Security Settings*....

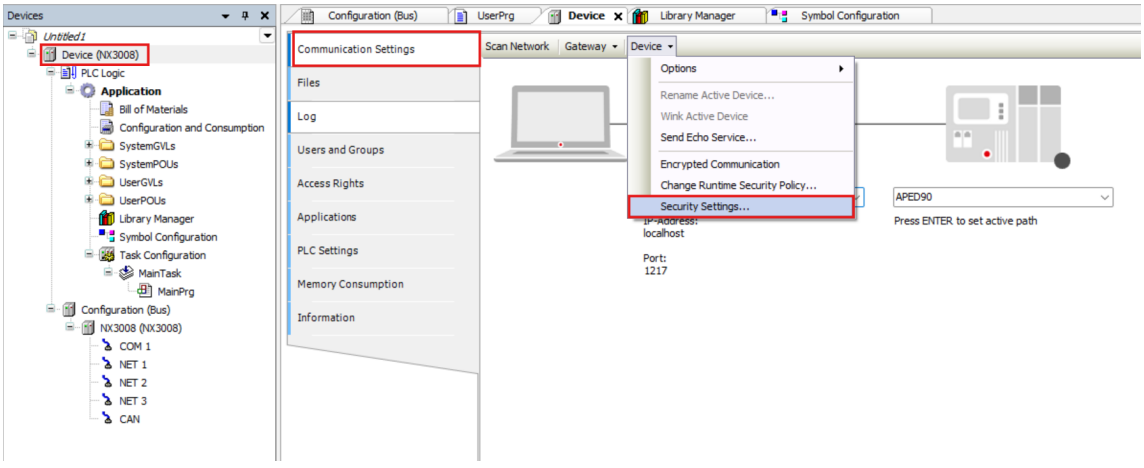

Figure 14: Security Settings

The *CommunicationMode* option allows selecting the Webvisu communication protocol.

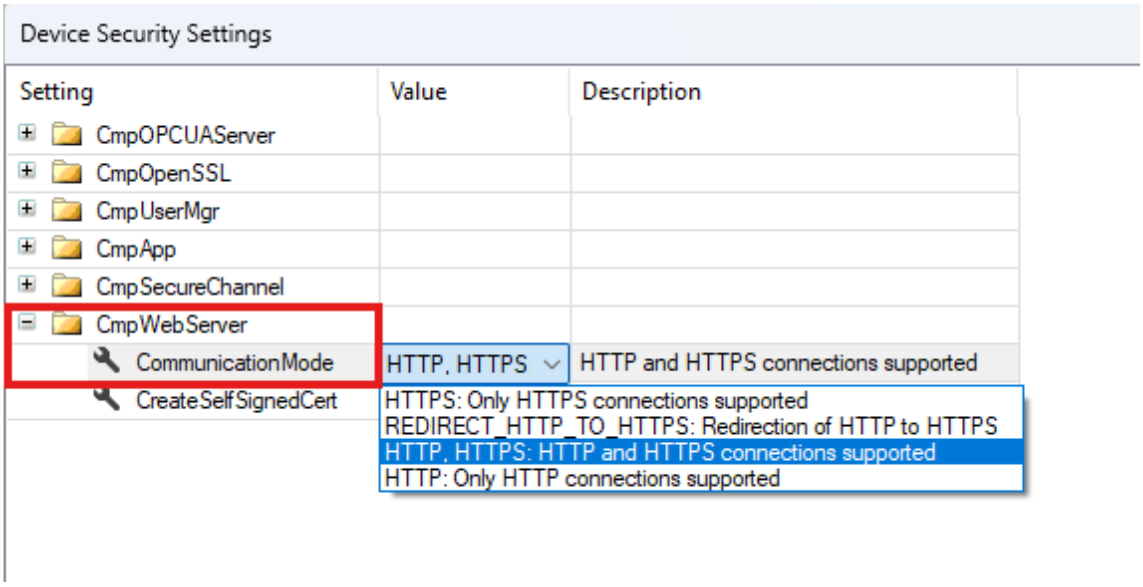

Figure 15: Webvisu Communication Protocol

5.4. Security Screen

RC 1.8, 4.1, 4.3 from IEC 62443-4-2 standard

The security screen manages the X.509 certificates, project and device wise, and the security level of the project. It can be accessed in the *View* > *Security Screen*.

In the "User" tab, you are able to manage the profile and it's respective certificates for file signing and decryption. It is also possible to manage the security level of the project, enforcing features like encrypted communication, encryption of files, signing of files, signing of compiled libraries, and many other security configurations.

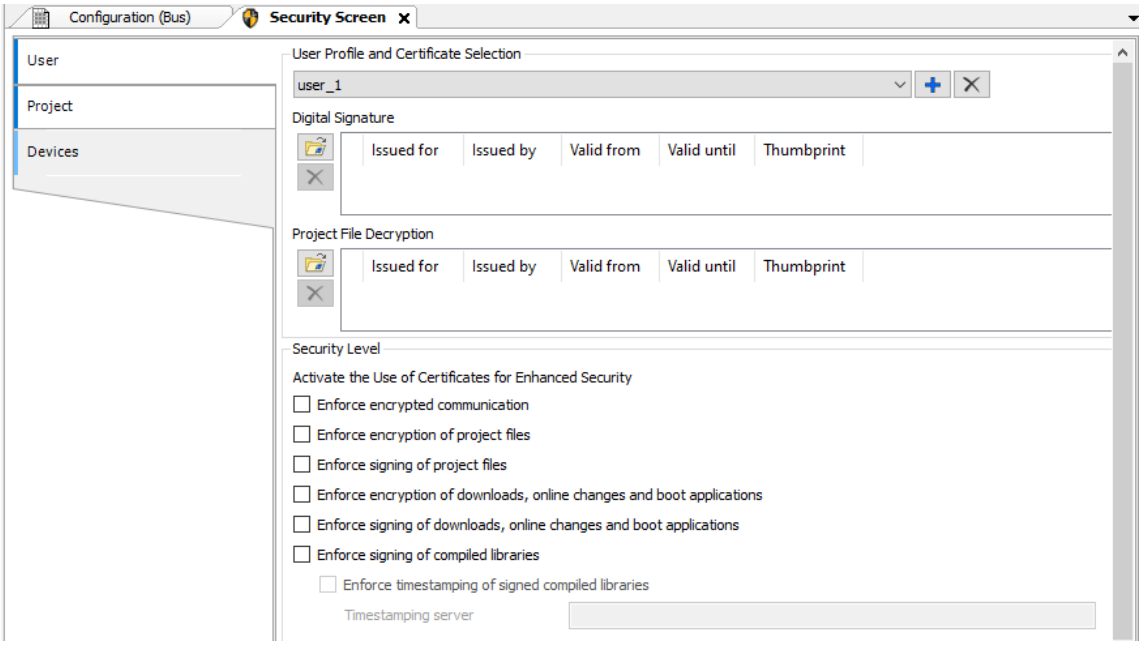

Figure 16: User tab of the security screen.

In the "Project" tab, it is possible to select the type of encryption of the project file, as well as the certificates of the user with access to it. It is also possible to encrypt the boot application, the download and the online change. In the "Devices" tab, you can configure encryption for OPC UA communication using the Basic256SHA256 profile, for a secure connection.

Functionalities like Signature of Compiled IEC Libraries, Encryption of the Application Source Code, and Project Protection Methods can be seen in their respective chapters.

To generate a new certificate check the appendix TLS Certificates and Keys Management.

5.5. Signature of Compiled IEC Libraries

RC 4.1 from IEC 62443-4-2 standard

An IEC library can be signed with an X.509 certificate if it is saved as a compiled library. While compiled libraries ensure the protection of the source code, the signature allows for verification of its authenticity.

The status of library signatures can be observed through icons in the *Library Manager* or through the Details in the *Add Library* menu.

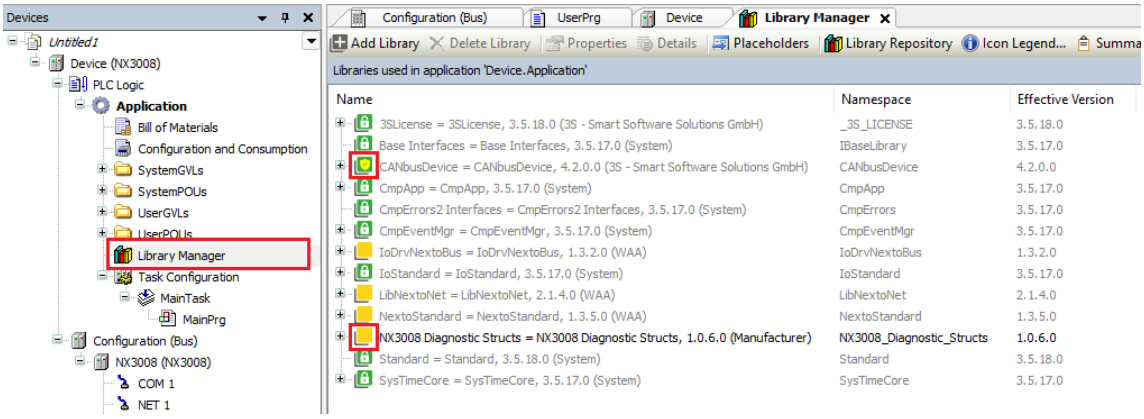

Figure 17: Signed and compiled libraries.

The green icon indicates that the library is signed by a reliable certificate. If the icon is yellow, that means that the library in not signed.

5.6. Encryption of the Application Source Code

RC 4.1, 4.3 from IEC 62443-4-2 standard

The application source code contains detailed information about the system in question and, therefore, the intellectual property of its manufacturer. Thus, protecting the application source code is a priority in the presence of confidential information.

MasterTool allows for project-wide encryption using passwords or physical security keys like USB Dongles. As described in IEC 62443-4-2 under “Component Requirement 4.3”, password encryption is based on the AES (Advanced Encryption Standard) methods, while solutions based on security keys are provided by the company WIBU Systems. Using passwords has the advantage of not requiring additional hardware, but using security keys provides a much higher level of protection since a password can be hacked or leaked.

Multiple different keys can also be linked to a project simultaneously, limiting access to the source code based on the number of keys and minimizing the risk of losing access to the code if a key is destroyed or lost. For this purpose, it is recommended to associate more keys than what would be strictly necessary.

The source code can also be protected using X.509 certificates. In this scenario, the source code will be symmetrically encrypted (AES algorithm). The symmetric key will then be asymmetrically encrypted (RSA algorithm) using the public key of each user who shares the source code. Optionally, the source code can also be digitally signed using the private key associated with the current user’s X.509 certificate. The signature will be saved alongside the source code in a file with the “.p7s” extension, following the PKCS #7 format for digital signatures.

If encryption is not possible, it is established that the project file is saved in a proprietary format, and its integrity is verified each time the project is loaded, thus protecting the confidentiality of intellectual property.

Section 5.13 provides a more detailed description of the encryption configuration in the project.

5.7. Logs

RC 2.8, 2.9, RE(1), 2.11, RE(1), RE(2), 2.12, 3.3 from IEC 62443-4-2 standard

Logs are provided and can identify errors, failures, warnings, project information, and actions taken by users.

To check the logs of the connected device, you should open the *Device* from the project treeview. Then, click on *Log* in the side menu of the *Device* window.

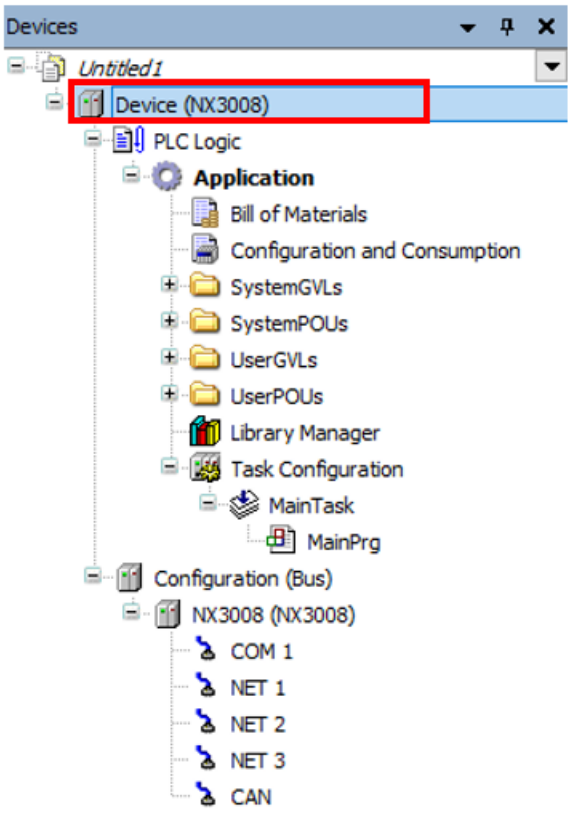

Figure 18: Project treeview

Through SD cards, it is possible to save the system logs, and if the card becomes full, a notification is issued. The device logs are stored in internal memory. It is possible to export a file containing all the logs, allowing it to be saved anywhere the user desires.

To perform the export, access the device logs and click on the icon marked in the image below.

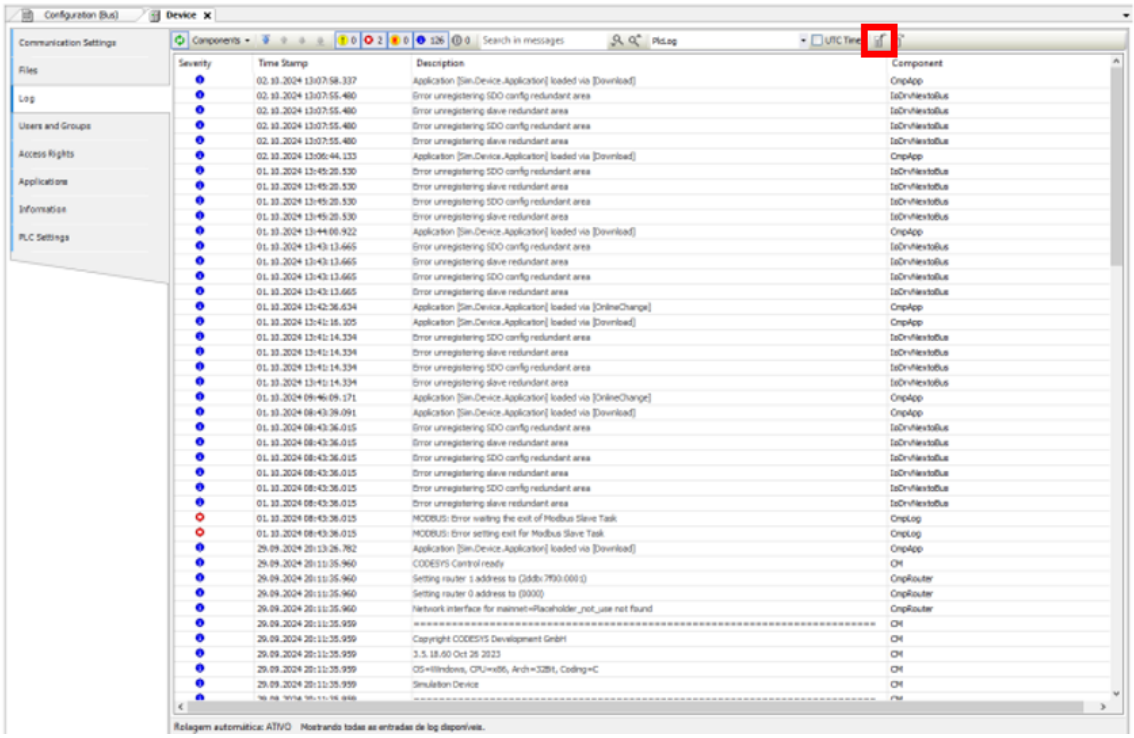

Figure 19: Logs screen

The timestamp of the device's RTC (Real-Time Clock) is provided for each log, and the device's RTC can be configured using the *Clock Settings* button, located next to *Easy Connection*.

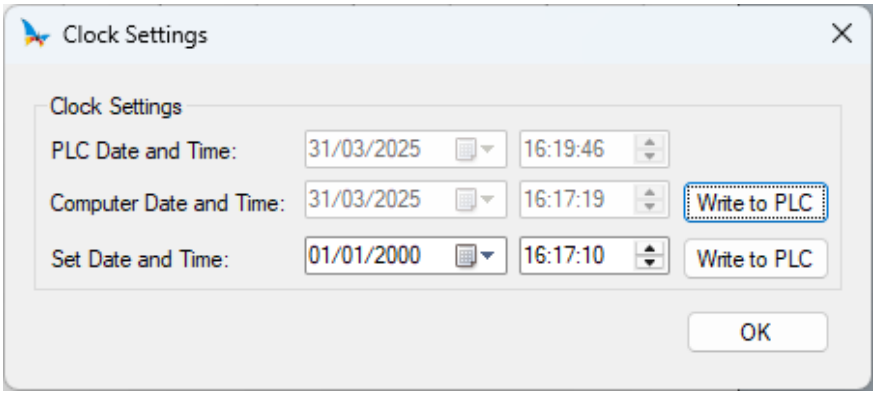

Figure 20: Clock Settings.

The logs are saved in a circular manner, meaning that once the available memory limit is reached, the oldest records begin to be overwritten.

RTC settings protection

The protection of the RTC configuration functions can be done through access management to its configuration library. It is possible to block changes to the RTC settings from the *Device* menu. First, a group must be created, and a user should be added to this group.

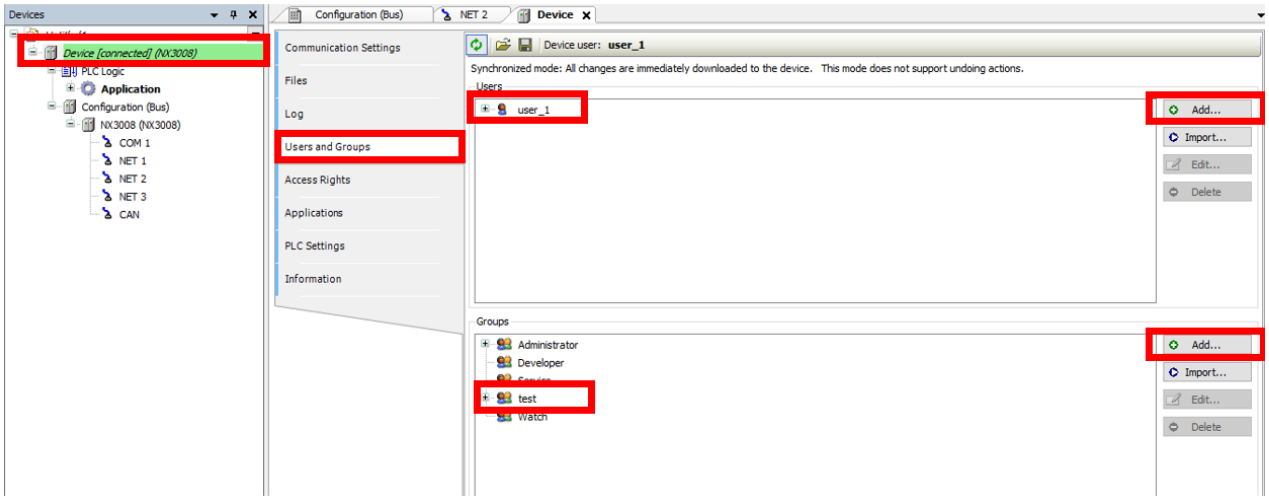

Figure 21: Adding a User to the Device

On the *Access Rights* page, you can grant or deny the execution of certain actions to registered users. To block changes to the RTC settings, you must deny modification of the *Application* parameter for the desired group, as highlighted in the image.

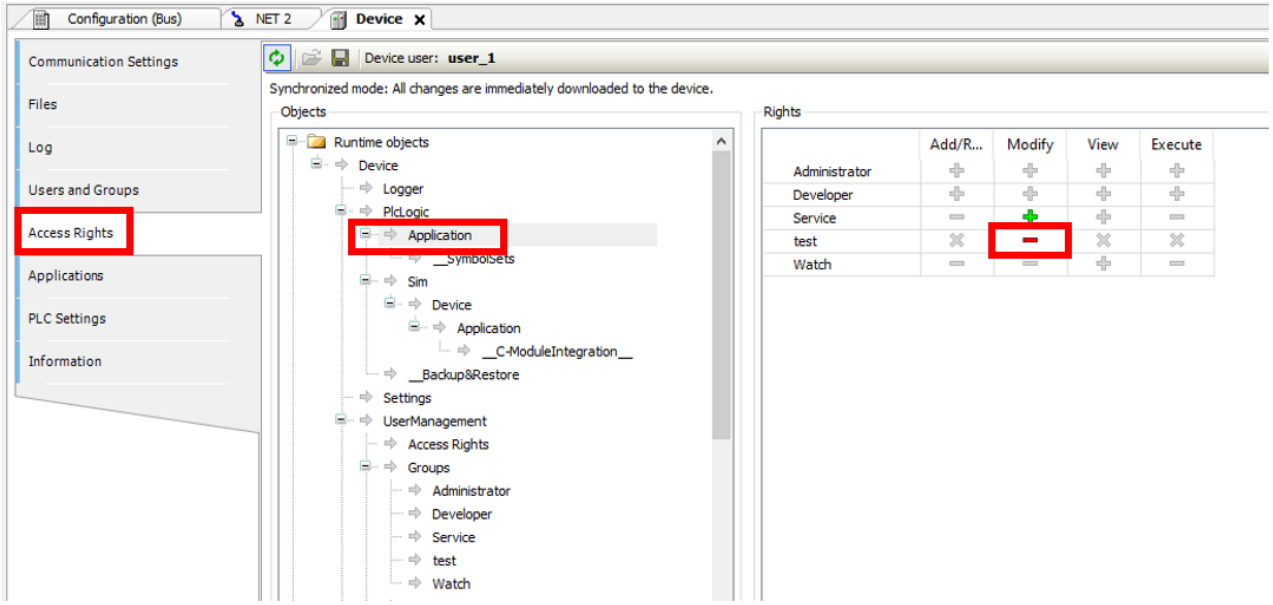

Figure 22: Configuring Access Rights on the Device

Confirmation that the device logs continue to be saved.

It can be confirmed that the device logs continue by accessing *Device* > *Log* and checking for updated data entries in the list. One way to force entries is by logging into the PLC, which will display the message "*User logged in*" at the top of the list.

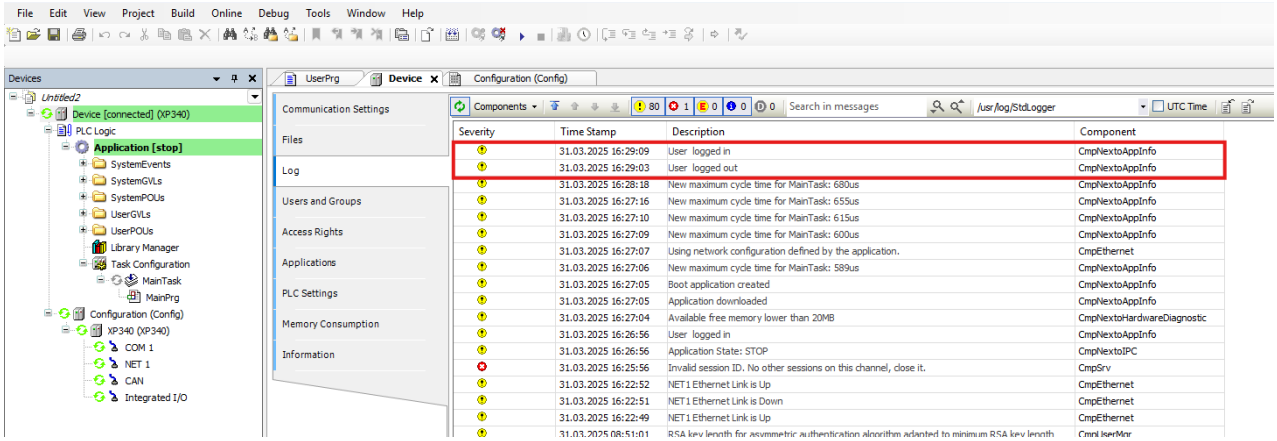

Figure 23: Demonstration of Logs Being Saved

5.8. Preset Outputs

RC 3.6 from IEC 62443-4-2 standard

In Mastertool, it is possible to configure the behavior of the component's outputs after a malfunction or failure. This is done in the *Device* > *PLC Settings* menu under the *Behavior for outputs in stop* option.

- Keep current values: The current values are maintained.
- Set all outputs to default: The default values derived from the I/O mapping are assigned.
- Execute program: The management of output values is governed by a program included in the project, which is executed in STOP mode. Enter the program name in the field on the right.

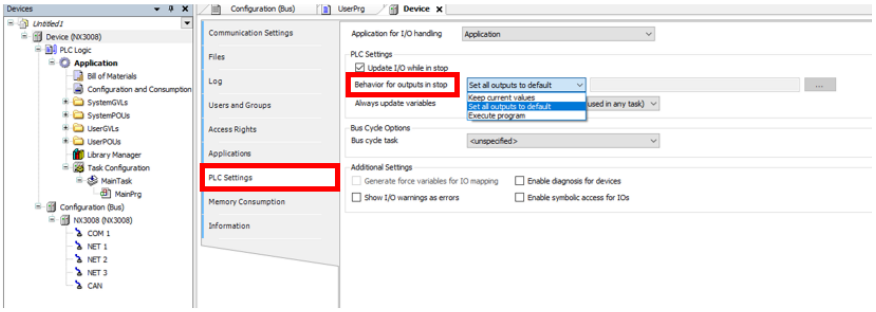

Figure 24: Output behavior

5.9. Error Visualization

RC 3.7 from IEC 62443-4-2 standard

The errors occurring in the system are presented clearly, objectively, and quickly, providing the user with the necessary information for correction or further diagnosis. Additionally, it does not provide data that is too detailed to aid in attacks.

There are two places where the system presents errors to the user: in the Logs (as shown in section 5.7) and in the message window of Mastertool.

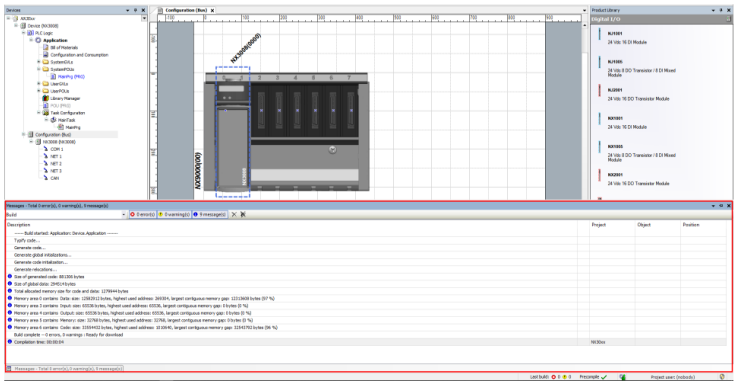

Figure 25: Messages window

In the message window, errors related to the application development are displayed, such as configuration or syntax errors and improper use of libraries. Additionally, this tab also shows messages and warnings related to the project.

5.10. Control System Backup

RC 7.3 from IEC 62443-4-2 standard

It is possible to download the source code from the internal memory of the CPU for backup purposes.

When attempting to log in to the PLC after some modification in the project, a selection box will appear for choosing between online change or downloading the modifications. After that, the user is asked whether they want to download the source code. The user can also download the source code in the "File" menu. This ensures that the application's source code is stored as a backup within the PLC.

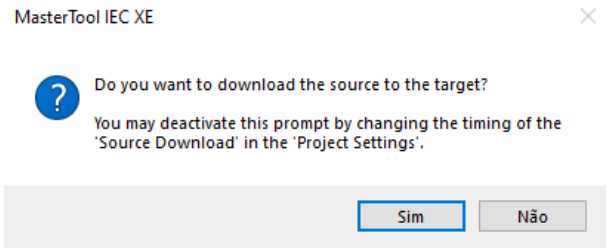

Figure 26: Download Option for the PLC.

Another way to keep a complete copy of the code is by extracting the Project Archive. This will generate a file that contains not only the code but also all the libraries used to run the application.

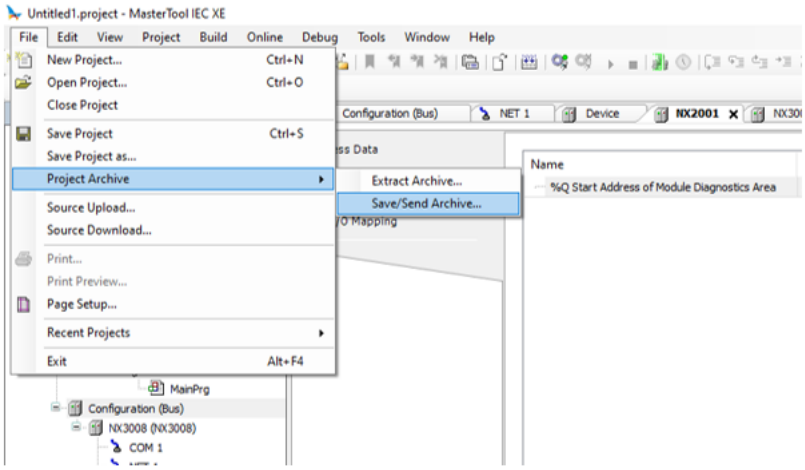

Figure 27: Project Archiving.

To store the PLC firmware as a backup, you need to access https://altusautomation.com/suport/downloads/ and download the file. It is not possible to keep the firmware backup in the PLC's internal memory.

The firmware update can be performed through the device's web page. To access it, enter the IP address in the browser's search bar. Then, go to *CPU Management*. You will need to log in to perform this action (admin/admin).

The previously downloaded file must be uploaded. The update may take a few minutes.

5.11. Inventory of Installed Components

RC 7.8 from IEC 62443-4-2 standard

It is possible to check the inventory of installed components in Mastertool by accessing the *Help* menu.

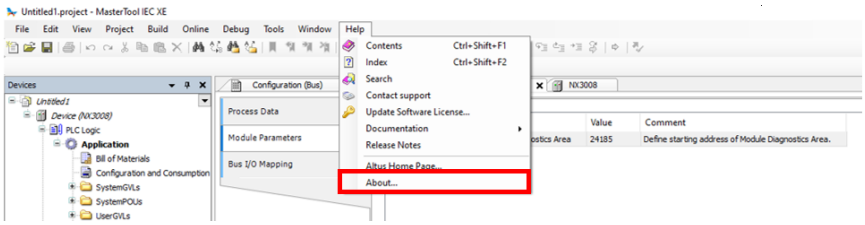

Figure 28: Installed components inventory

On this page, you can check all the components installed in Mastertool.

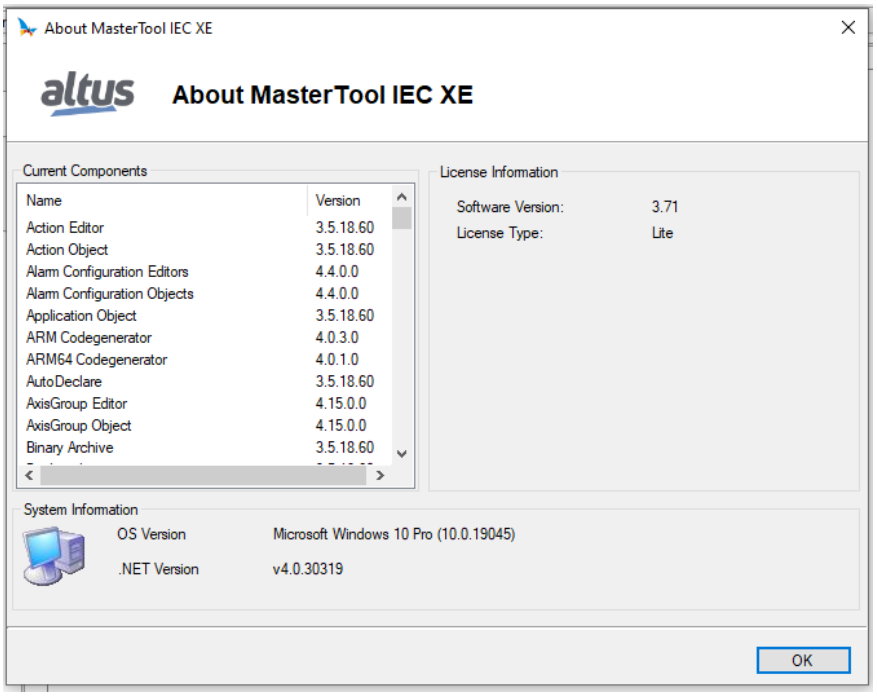

Figure 29: Installed components

5.12. Protection Against Malicious Code

RC 3.2 from IEC 62443-4-2 standard

As it is a closed system, the ways to load malicious code are through application loading or firmware updates. The application loading has the user protections mentioned earlier, and for firmware updates, the file is encrypted and has content consistency checks.

5.13. Project Protection Methods

RC 4.3 from IEC 62443-4-2 standard

It is possible to access the project security settings from the menu *Project* > *Project Settings* > *Security*. In this screen, the user can select the project's security method. The default method is *Integrity Check*, where the file is saved in a proprietary format, and its integrity is verified each time the project is loaded.

The most secure method is using encryption, where the user can configure a password to encode the project's file content. The password will need to be entered every time the project is opened. There is also the option to configure encryption using certificates, requiring that the certificates of all users sharing the project be saved in local storage.

To import and configure the project protection certificates, check Security Screen.

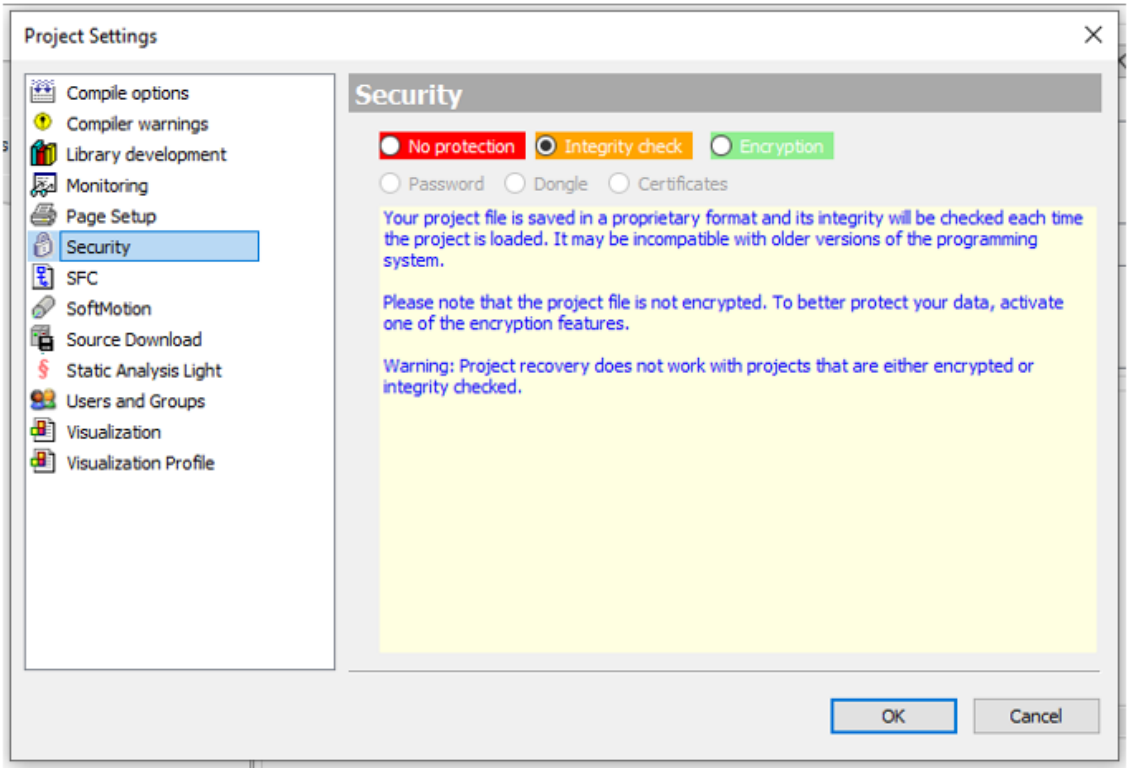

Figure 30: Security Menu.

6. Security Measures of Altus PLCs

Altus PLCs are equipped with various security features to prevent vulnerabilities during their operation. Some measures are present in only certain controller models, so always check on the specific documentation of the product for the desired security features. Below the subchapter titles, the component requirement (CR) or Embedded Device Requirement (EDR) of the IEC 62443-4-2:2019-02 standard to which it pertains is provided.

6.1. User Management and Access Rights of the UCP

RC 1.1, RE (1), 1.3, 1.4, 1.5, 1.7, 2.1, RE(1), RE(2) from IEC 62443-4-2 standard

Nexto CPUs have a user rights management system that blocks or allows certain actions for each user group in the CPU. To edit these rights in the CPU, the user must access a project in MasterTool IEC XE without being logged in to the CPU. He must then click in the Device tree on the left of the program, double click on the *Device* item and then select the CPU in the *Communication Settings* tab that opens. Only the *Users and Groups* and *Access Rights* tabs refer to this topic. The figure below illustrates the steps to access this CPU tab.

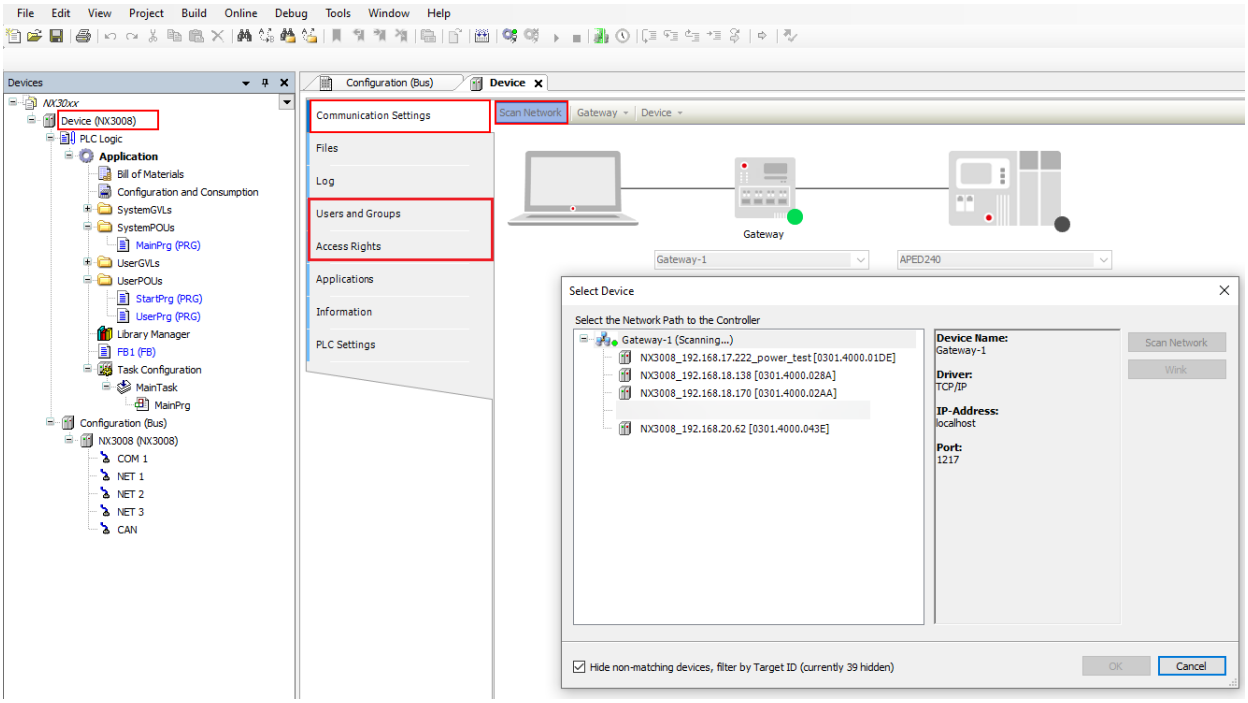

Figure 31: Access to Users and Groups and Access Rights Tabs

ATTENTION

If the user forgets the password(s) of the account(s) with access to the CPU, the only way to recover this access is by updating its firmware.

ATTENTION

After performing a CPU user Logoff command, the *Device* tab of this project must be closed, so that all access rights are effectively closed.

6.1.1. Users and Groups

The *Users and Groups* tab is located in the *Devices* tab. It allows the configuration of user and group accounts that which, together with access rights management, control access to objects in the PLC in online mode.

6.1.1.1. Common

It might be desired that certain functions of a controller can only be executed by authorized users. For this purpose, the *Online User Management* feature provides the possibility to set up user accounts, to assign access rights for user groups and to force an user authentication at login.

The device specific user management might be predefined by the device description and it also depends on this description to what extent the definitions can be edited in the configuration dialogs in the programming system.

Like in the project user management, users have to be members of groups and only user groups can get certain access rights.

6.1.1.2. Using the Configuration Dialog Box

Basically, the handling of the online user management dialogs is very similar to that of the project's user management. There is even the possibility to *import* user account definitions from the project's user management.

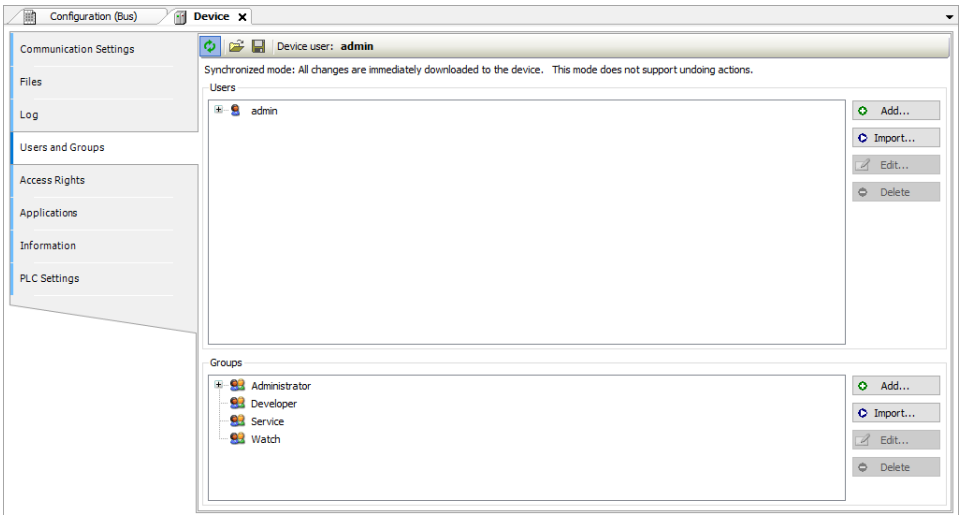

Figure 32: Device Dialog, Users and Groups

6.1.1.2.1. Users

The following buttons are available for setting up user accounts:

Add: The dialog *Add User* opens where you can define a user name and a password. The password must be repeated in the *Confirm password* field.

ATTENTION

By opening this dialog, the *Password* and *Confirm Password* fields are going to be filled with fictional characters, the user must replace these characters with a valid password.

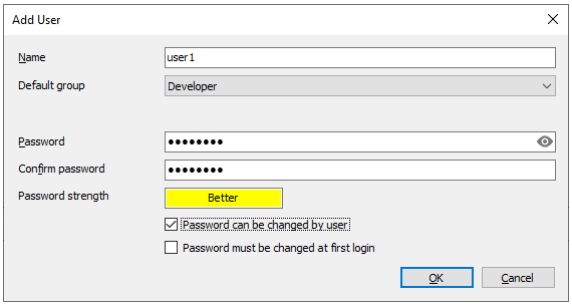

Figure 33: Add User Dialog

Import: The dialog *Import Users* opens showing all user names which are currently defined in the project user management. Select one or several entries and confirm with *OK*. The dialog *Enter* password will open where you have to enter the corresponding password as it is defined in the project user management, in order to get the user account imported to the device-specific user management.

Modify: The currently selected user account can be modified concerning user name and password. This *Edit User* <user name> dialog corresponds to the *Add User* dialog.

Delete: The currently selected user account will be deleted.

6.1.1.2. Groups

Add: The dialog *Add Group* opens where you can define a new group name and select from the currently defined users those who should be members of this group.

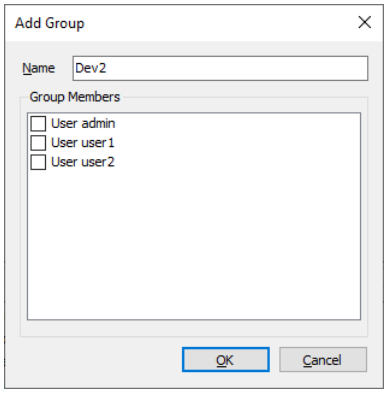

Figure 34: Add Group Dialog

Import: The dialog *Import Groups* opens where the groups currently defined in the project user management are listed. You can select one or several entries and confirm with *OK* to get them integrated in the groups list of the device-specific user management.

Modify: The currently selected group can be modified concerning group name and associated users. For this purpose the *Edit Group* <group name>, dialog opens, which corresponds to the *Add Group* dialog.

Delete: The currently selected group can be deleted.

6.1.1.3. Applying and Storing the Current Configuration

Enables or disables synchronization between the editor and the device's user management system.

When the button is not activated, the editor remains either blank or displays a configuration that you have loaded from the hard disk.

If you enable synchronization while the editor holds a user configuration that hasn't been synced with the device, you'll be prompted to decide how to handle the editor's contents. You have the following options:

- **Upload from the device and overwrite the editor content**: This will load the configuration from the device into the editor, replacing the current contents.
- **Download the editor content to the device and overwrite the user management there**: This will transfer the configuration from the editor to the device, applying it and overwriting the existing user management settings.

 Save to Disk,  Load from Disk: The current configuration can be stored in a *.dum2 file and re-loaded from this file, which is useful to set up the same user configuration on multiple systems. The standard dialog for browsing in the file system will be provided for this purpose. The file filter automatically is set to *.*dum2*, which means *device user management* files.

Note: Before CODESYS V3.5 SP16, the Device user management files (*.dum) file type was used which did not require any encryption.

The actual settings can also be documented as printed version by use of the command *Print...* (*File* menu) or *Document...* (*Project* menu).

6.1.1.4. Considerations on Default Users and Groups

In firmware versions 1.3.x.x or lower, the existing users and groups are: Everyone and Owner, as seen in the table below:

Users	Groups
Everyone	Everyone
Owner	Owner

Table 1: Users and Groups in Versions 1.3.x.x

However, in firmware versions 1.4.x.x or higher there are the users: Administrator and Everyone, and the groups: Administrator, Developer, Everyone, Service and Watch. As seen in the table below:

Users	Groups
Administrator	Administrator
Everyone	Developer
	Everyone
	Service
	Watch

Table 2: Users and Groups in Versions 1.4.x.x

6.1.1.4.1. Group Administrator

This group has all privileges and it is not possible to remove it in the firmware versions 1.4.x.x or higher. The group Developer is part of this group.

6.1.1.4.2. Group Developer

Group created to define access rights to users that are application developers. The group Service is part of this group. If not used, this group can be removed.

6.1.1.4.3. Group Everyone

For firmware versions 1.3.x.x or lower: This is the default group to perform accesses in a CPU while there are no defined users and groups.

For firmware versions 1.4.x.x or higher: This is the default group to perform accesses in a CPU while there are no defined users and groups.

6.1.1.4.4. Group Service

Group created to define access rights to users that provide some kind of service in the PLC, for example, maintenance teams. The group Watch is part of this group. If not used, this group can be removed.

6.1.1.4.5. Group Watch

Group created to define access rights to user that can only visualize, without making any modification in the application. If not used, this group can be removed.

6.1.1.4.6. User Administrator

The user Administrator is defined in the groups Everyone and Administrator. The password of the user Administrator is *Administrator* and can be modified.

6.1.1.4.7. User Everyone

For firmware versions 1.3.x.x or lower: The user Everyone is defined in the group Everyone. This user doesn't have a defined password.

For firmware versions 1.4.x.x or higher: The user Everyone is defined in the groups Everyone and Administrator. This user doesn't have a defined password.

6.1.1.5. User and Groups from Old Projects

To maintain this data from old projects in a new project after updating the CPU firmware or in a new Nexto CPU, it's necessary to execute the command *Synchronization* () in the old project with the original firmware, thus fetching the CPU configuration, and then execute the command *Save to disk*, saving the current configuration in a file.

In the new Nexto CPU or in an updated CPU, run the command *Load from disk*, then select the file generated before, *Synchronization* again, thus sending the configuration to the CPU.

ATTENTION

If the old project is with firmware versions 1.3.x.x or lower, a user and a group with the name *Administrator* must be created before saving the configurations in a file. This procedure guarantees that the configuration will be loaded in projects with firmware versions 1.4.x.x or higher.

6.1.2. Access Rights

This dialog is provided on a tab of the *Device* dialog (Device Editor). It is part of the *Online User Management* feature and is used to grant or deny certain permissions to the currently defined user groups, thus defining the user's access rights to files or objects (such as an application) on the PLC at runtime.

Note that these rights can only be assigned to groups, not to individual users. For this reason, a user must be defined as a member of a group. Users and groups are configured in the *Users and Groups* tab of the Device Editor.

The figure below shows the rights to add and remove children to/from the Device object for the user groups *Administrator*, *Developer*, *Service* and *Watch*.

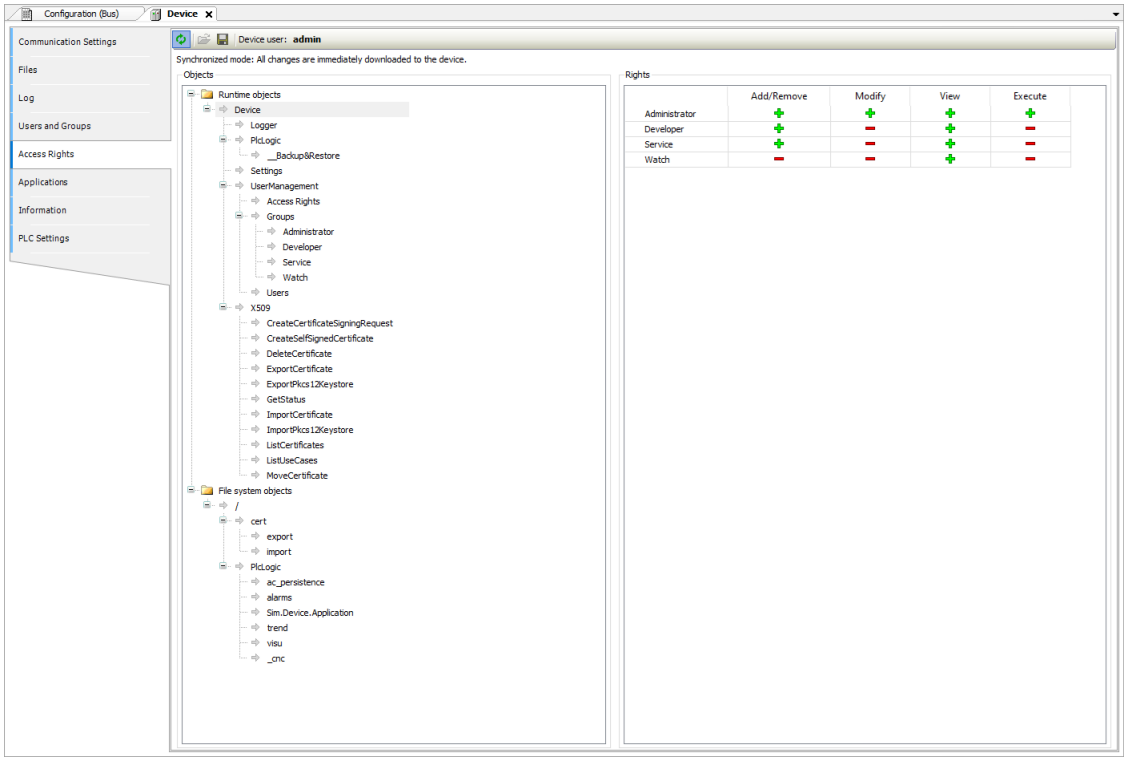

Figure 35: Device Access Rights

See in the following how to define the access permissions and how to get them loaded to the device or stored in a reloadable file.

6.1.2.1. Defining the Access Rights

To define the permission for performing an action on one or multiple object(s), you have to select the object entries below the desired action type in the *Objects* window, then select the desired group in the *Rights* window, and then click on the *Grant* or *Deny* button (also in the *Rights* window).

See in the following a description of the particular dialog windows.

6.1.2.1.1. Objects

This part of the dialog lists the possible actions which might be performed during runtime on files in the PLC file system resp. runtime objects like for example applications. The tree is structured in the following way:

- :
  - Top-level, for structuring purposes, there are two *folders* for File system objects and Runtime objects.
- :
  - Indented below there are nodes for the four types of actions, which might be performed on the particular objects. These nodes also just serve for structural purposes:
  - Add/remove children (adding or removing of *child* objects to an existing object).
  - Execute (for example start/stop application, setting breakpoints etc.)
  - Modify (for example downloading application, etc.)
  - View (monitoring)
- Objects (action *targets*)

ATTENTION

Assigning an access right definition to a *father* in the objects tree usually means that the *children* will inherit this definition, as long as they do not get an explicit own definition. However, depending on the device, this might be handled differently. In any event, inheritances are not visualized here in the dialog.

6.1.2.1.2. Rights

This field shows the defined user groups and their rights in a table. By selecting an Object in the *Objects* tab, you can change its rights using the following buttons:

- : The object(s) currently selected in the *Objects* window are granted for the group.
- : The object(s) currently selected in the *Objects* window are denied for the group.
- : Currently there is no explicit access right definition for the object(s) currently selected in the *Objects* window.

6.1.2.2. Applying the Current Configuration Access Rights

Enables or disables synchronization between the editor and the device's user management system.

When the button is not activated, the editor remains either blank or displays a configuration that you have loaded from the hard disk.

If you enable synchronization while the editor holds a user configuration that hasn't been synced with the device, you'll be prompted to decide how to handle the editor's contents. You have the following options:

- **Upload from the device and overwrite the editor content**: This will load the configuration from the device into the editor, replacing the current contents.
- **Download the editor content to the device and overwrite the user management there**: This will transfer the configuration from the editor to the device, applying it and overwriting the existing user management settings.

Save to Disk, Load from Disk: The current configuration can be stored in a *.drm file and re-loaded from this file, which is useful to set up the same user configuration on multiple systems. The standard dialog for browsing in the file system will be provided for this purpose. The file filter automatically is set to **.drm*, which means *device rights management* files.

The actual settings can also be documented as printed version by use of the command *Print...* (*File* menu) or *Document...* (*Project* menu).

6.1.2.3. User and Access Right Management of Old Projects

To maintain this data from old projects in a new project after updating the CPU firmware or in a new Nexto CPU, it's necessary to execute the command *Synchronization* ( ) in the old project with the original firmware, thus fetching the CPU configuration, and then execute the command *Save to disk*, saving the current configuration in a file.

In the new Nexto CPU or in an updated CPU, run the command *Load from disk*, then select the file generated before, *Synchronization* again, thus sending the configuration to the CPU.

ATTENTION

If the old project is with firmware versions 1.3.x.x or lower, a user and a group with the name *Administrator* must be created before saving the configurations in a file. This procedure guarantees that the configuration will be loaded in projects with firmware versions 1.4.x.x or higher.

6.1.3. Access to the Runtime System with Permission/Authentication Management RC 1.4 e 1.5 from IEC 62443-4-2 standard

There are different phases in an industrial application, from the initial development of the source code to its commissioning, production with the machine or plant, and maintenance. These phases are typically operated by different technicians with appropriate levels of qualification.

Considering these qualification levels and the threats of potential misuse beyond the assigned task or competence, it makes sense to restrict usage to certain user groups.

MasterTool supports authentication and permissions management for an individual user or a group of administrators. Depending on the security policy of the runtime system, user management may or may not be enabled by default. If not, all users are members of the administrator group and have unlimited rights on the controller until user management is activated. When using it, it's necessary for the first activation to occur during the initial login, specifying an administrator user.

Once at least one new user is added, all users must authenticate with their usernames and passwords for each online connection to the controller. Passwords are transmitted encrypted (by default, using asymmetric encryption) and stored encoded as cryptographic hashes in the runtime system.

This measure reduces the threat of accidental or intentional access to the running controller, which could impact the availability or integrity of the compiled application executed on the controller.

Secure login on the programmable controller provides a way to protect the user's application from any unauthorized access. By enabling this feature, the Nexto Series PLC will request a user password before executing any commands between MasterTool IEC XE and the PLC, such as stopping and programming the application or forcing outputs on a module.

6.2. Protection Against Flood-type Attacks

RC 7.1 from IEC 62443-4-2 standard

The NX3035 NX3008, and HX3040 controllers, and the NX5000 (Ethernet) module are equipped with protection against flood-type attacks. This essential security feature is designed to detect and effectively mitigate flood attacks, in which a large amount of data is sent simultaneously to overload the system and cause unavailability or service disruption. However, it is always recommended to use firewall rules to allow traffic only from known addresses, enhancing this protection.

6.3. Log Storage

RC 2.9, RE(1), CR 3.9 RE(1), from IEC 62443-4-2 standard

The device logs are stored in the internal memory. It is possible to export a file containing all records, allowing the user to save it anywhere they prefer. To perform the export, you need to access the device logs and click on the icon marked in the image.

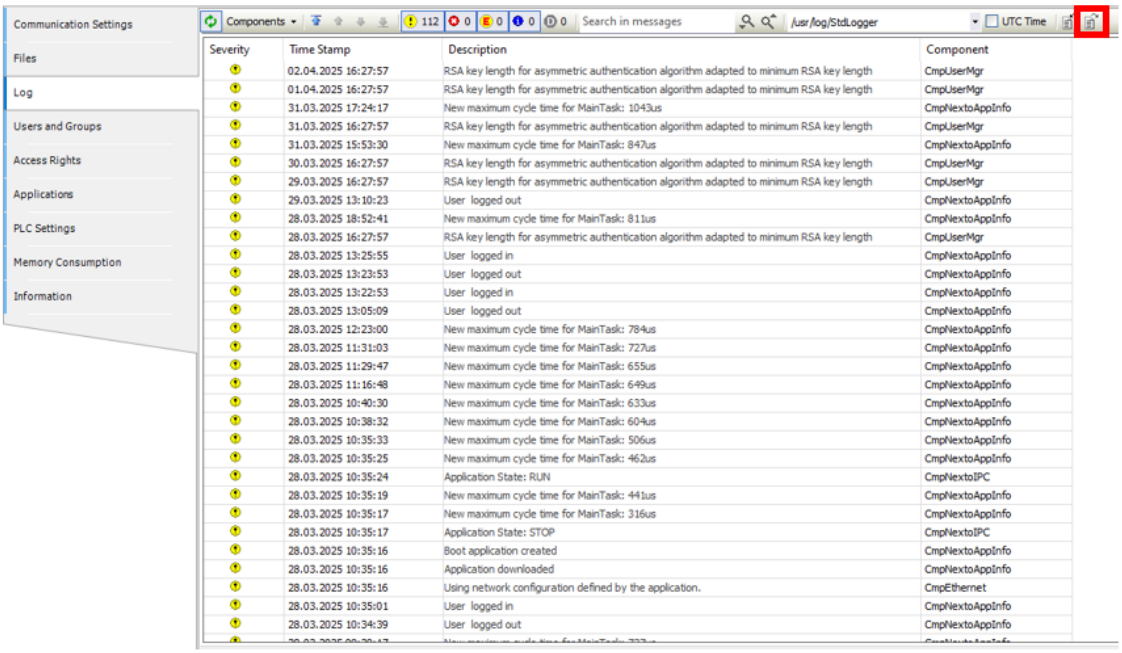

Figure 36: Export Logs

The logs are saved in a circular manner, meaning that once the available memory limit is reached, the oldest records start to be overwritten.

6.4. SysLog

RC 2.9, RE(1) from IEC 62443-4-2 standard

Syslog (System Logging Protocol) is a protocol that allows devices to send log messages to a centralized log collection and storage system. These messages can include information about system events, errors, warnings, and other activities relevant to system administration and security.

The UCP can be configured as a SysLog Client from the System section in the Management tab of the UCP System web page. As shown in the figure below.

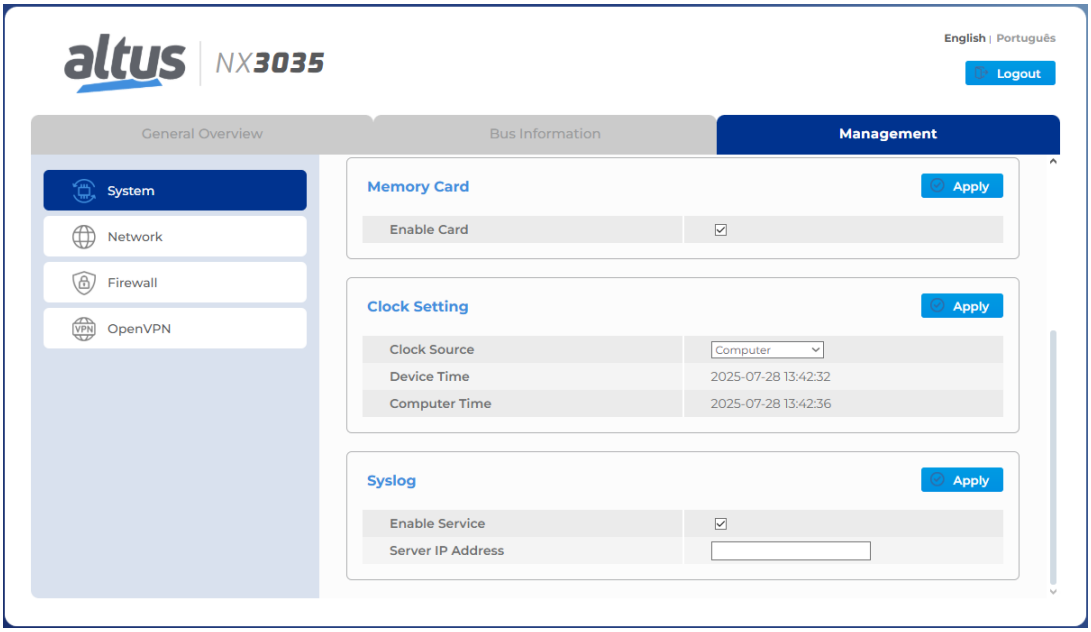

Figure 37: SysLog Configuration

6.4.1. SysLog Configuration

The SysLog configuration consists of only two fields. The *"Enable Service"* checkbox is disabled by default and must be checked to activate the service. The IP address of the SysLog server to be used is configured in the *"Server IP Address"* field.

After configuring the server IP address and enabling the service, the settings are applied by clicking the *"Apply"* button. It is important that the Syslog server is properly configured to communicate througth UDP protocol, using port 514.

When configured, SysLog will send all logs present in Mastertool's System Logs to the server, as detailed in the Logs section. The SysLog service only sends logs with a priority of WARNING or higher, as indicated in the classification below:

Mastertool Category	Syslog Category	Priority
Exception	Emergency	0
Error	Error	3
Warning	Warning	4

Table 3: Syslog Message Priority

6.5. Web Page Features

EDR 3.10 e RE(1) e RC 5.1 from IEC 62443-4-2 standard

The products offer a configuration page accessible through a web browser by simply typing the device's IP address in the search bar. From there, it is possible to configure and monitor various functionalities, depending on the product.

6.5.1. Update PLC

It is possible to update the PLC firmware from the web page. To access it, type the IP address in the browser's search bar. Then, go to *UCP Management*. Login will be required to perform this action (admin/admin by default).

You must upload the firmware file available at https://altusautomation.com/suport/downloads/. The update may take a few minutes.

Before the new firmware is actually written, integrity checks of the file and session are performed. First, the user is asked to log in with a password on the PLC, ensuring that only authorized people have access. After that, the firmware undergoes a series of checks on the provided binary, verifying the model, CRC, and the number of files.

After this, the firmware files in the binary are unpacked and decrypted, and are also subjected to a CRC check. Only then are they written.

6.5.2. PLC's IP Address Change

On the device's web page, access the *Management* menu and find the *Network Settings* screen. In this screen, modify the values and save the changes.

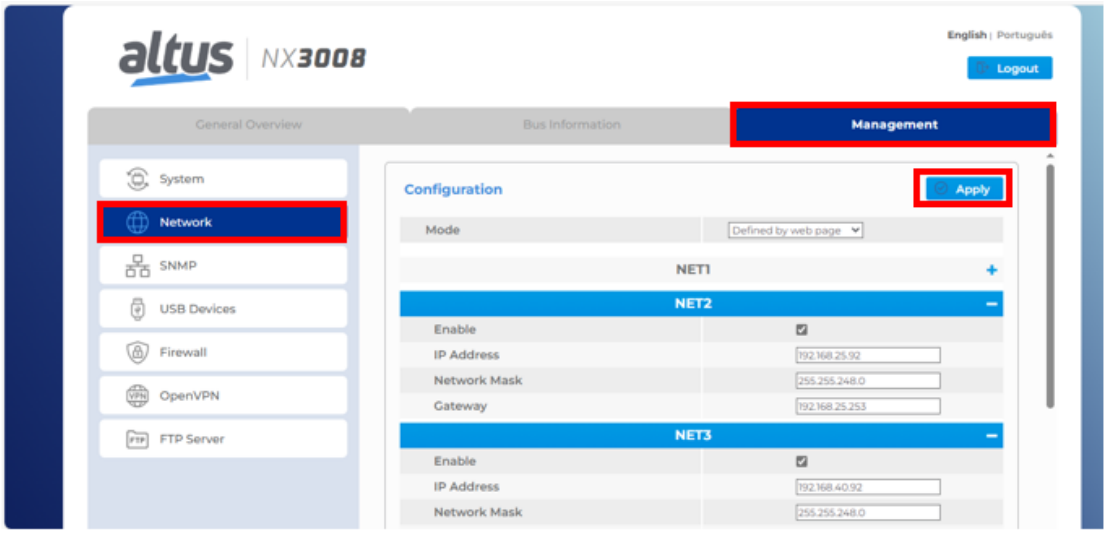

Figure 38: Changing Network Settings via the Device's Web Page

6.6. Memory Card

RC 2.9, RE(1), CR 3.9 RE(1), from IEC 62443-4-2 standard

Among other memories, Nexto Series CPUs allow the user the utilization of a memory card. It is defined according the features described in the PLC manual.

When the card is inserted in the CPU and it presents a file type different from FAT32, it automatically identifies those files and questions the user if he wants to format the files. In negative case the user cannot use the card, as it is not mounted. The card presence is not displayed. If the user decides to format the files, the CPU takes a few minutes to execute the operation, depending on the cycle time (execution) of the application which is running in the CPU. Once the memory card is mounted, the CPU will read its general information, leaving access to the memory card slower in the first few minutes. This procedure is done only when the card is inserted, in case of the CPU reset or through the device's web page.

ATTENTION

It is recommended to format the memory card directly in the Nexto CPU in order to avoid possible use problems, mounting time increase or even the incorrect functioning.
It is not recommended to remove the memory card or de-energize the CPU during the formatting or during the files transfer as it can cause the loss of data as well as irreversible damages.

6.6.1. Memory Card Configuration

A web page was developed for managing the memory card. Among the features offered are formatting, the option to unmount and remove the card, and the ability to enable and disable the card interface. These settings were developed in the "Memory Card" section of the CPU webpage, under the "Management" tab. In addition to the settings, information about the device's current status, total and free storage space, both measured in *kB*, are also displayed. The image below shows the home page, with the interface enabled and no devices connected.

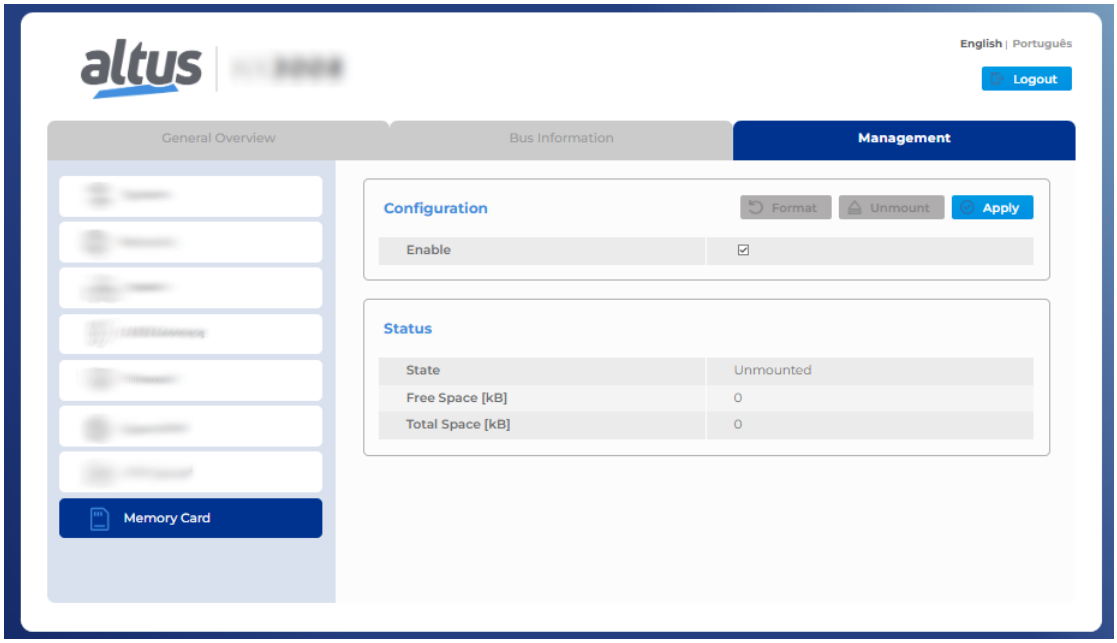

Figure 39: Memory Card Home Page

As long as there is no memory card inserted and mounted in the CPU, the *Format* and *Unmount* buttons remain locked for use. The **Status** table indicates the *Unmounted* state. When a memory card is inserted into the CPU, the *Format* button is enabled for use. After the card is mounted, the *Unmount* button also becomes available for use. When inserting a memory card into the CPU, it may take a few moments for the card to be mounted and the information updated on the web page. The image below shows the web page when a card is connected and mounted in the CPU.

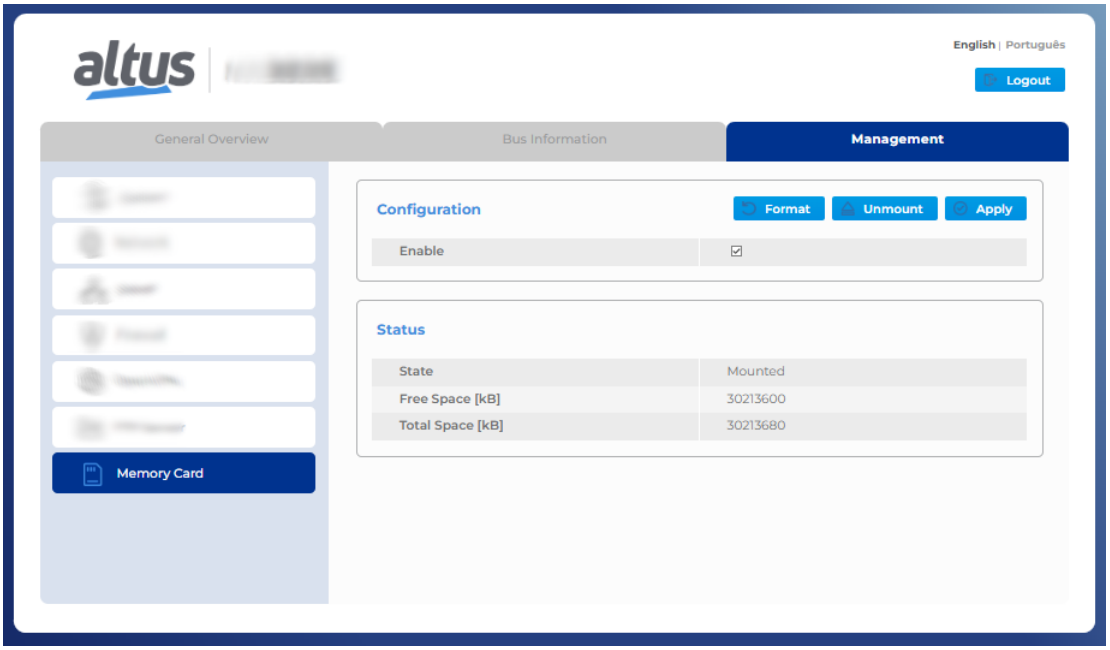

Figure 40: Memory Card Device Mounted

6.6.1.1. Formatting the Memory Card

To format the device, use the **Format** button. When you click, a *pop-up* style message will appear, asking you to confirm the operation. The image below presents this message.

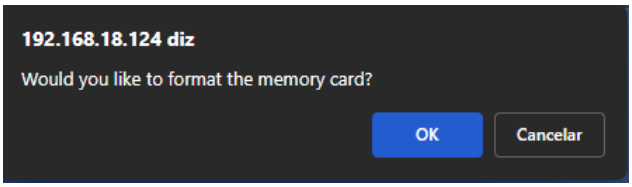

Figure 41: Format Confirmation Message

Upon confirming with the **OK** button, the operation begins, after which all configurations are blocked. The **Format**, **Unmount** and **Apply** buttons, as well as the checkbox, become unavailable during formatting. The formatting process is indicated in the **Status** table, with the **State** value changed to **Formatting...**, as shown in the figure below.

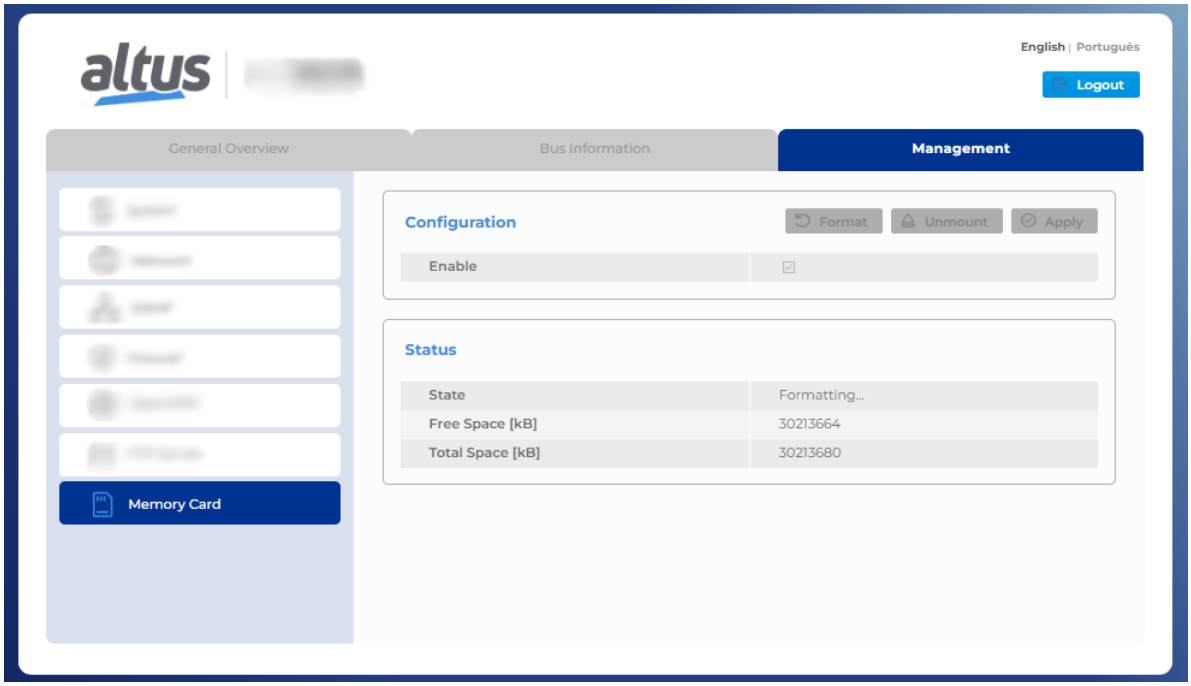

Figure 42: Formatting Memory Card

At the end of the formatting process, a message is displayed indicating that the operation has been completed on the device. The following figure shows this message.

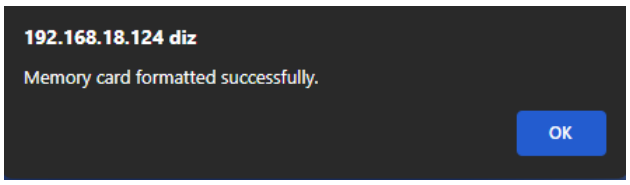

Figure 43: Formatting Complete Message

Upon completion of the operation, the web page returns to its initial state, unlocking all buttons, as well as the checkbox, as shown in figure Memory Card Device Mounted.

6.6.1.2. Unmounting the Memory Card

To unmount and remove the device, click the **Unmount** button. A confirmation *pop-up* will appear asking you to confirm the operation. The image below shows this message.

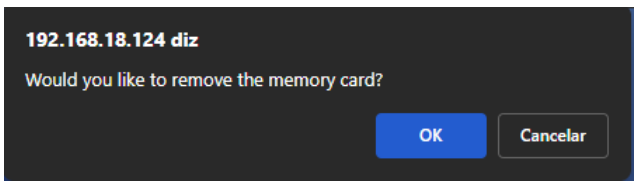

Figure 44: Confirmation Message to Unmount

Upon confirming with the **OK** button, the operation begins, after which all configurations are blocked. The **Format**, **Unmount** and **Apply** buttons, as well as the checkbox, become unavailable during unmounting operation. The unmounting process is indicated in the **Status** table, with the **State** value changed to **Unmounting...**, as shown in the figure below.

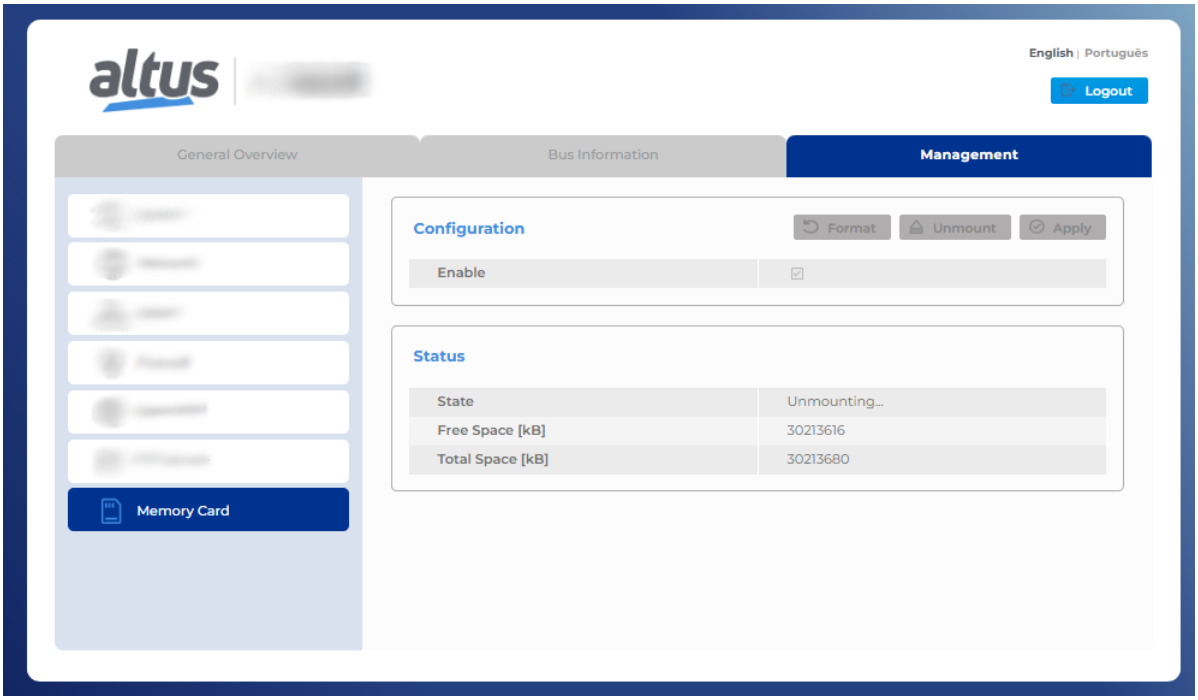

Figure 45: Unmounting Memory Card

At the end of the unmounting process, a message is displayed indicating that the operation on the device has been completed. The following figure shows this message.

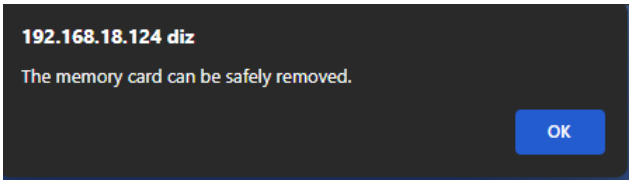

Figure 46: Unmount Complete Message

Upon completion of the operation, the **Format** and **Apply** buttons, as well as the checkbox, become available for use. The **Unmount** button remains locked because the card has already been unmounted. The *Status* table displays the data for an unmounted card, as can be seen in the image below:

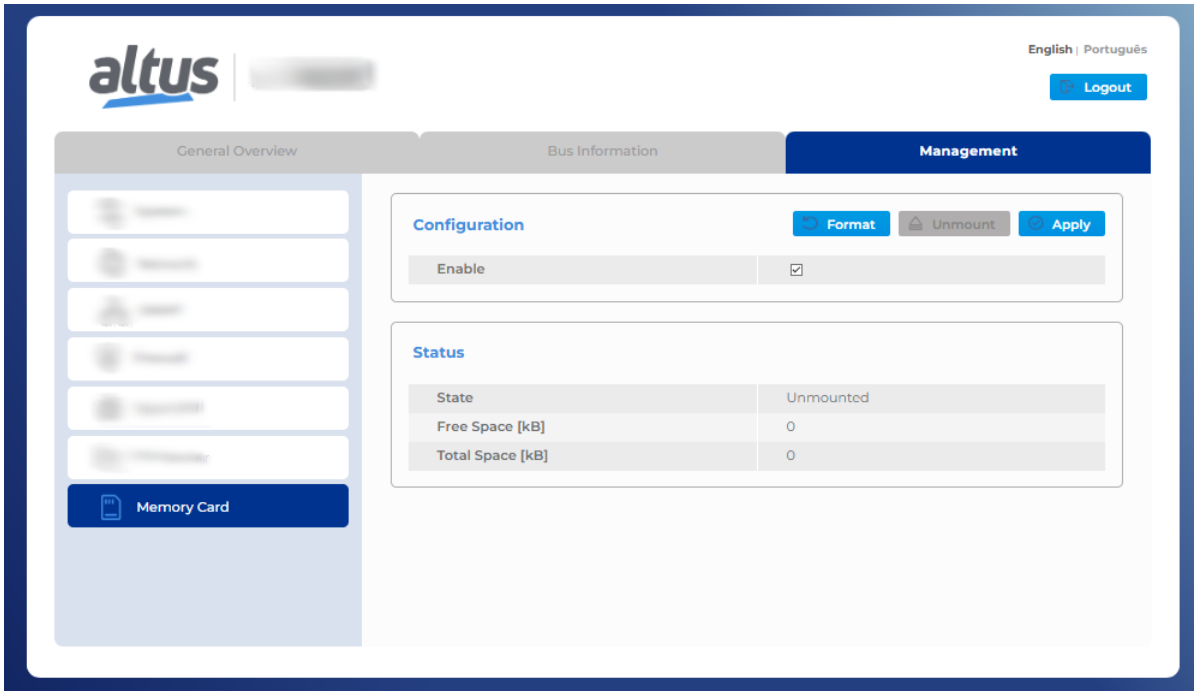

Figure 47: Memory Card Unmounted

6.6.1.3. Memory Card Interface Management

A setting was developed on the card's web page to enable and disable the memory card interface, this functionality is part of the level one cybersecurity requirements according to IEC 62443. To enable, check the **Enable** checkbox in the **Configuration** table. Then use the **Apply** button to submit the new configuration. To disable, uncheck the **Enable** checkbox and use the same button to apply the setting. Clicking the **Apply** button will display a *pop-up* message asking you to confirm the operation. The image below shows the message.

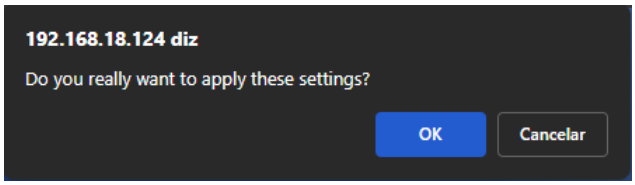

Figure 48: Confirmation Message to Apply Configuration

After confirming by clicking the **OK** button, the new configuration is sent to the CPU. If the interface has been enabled, the information displayed on the web page depends on whether or not a memory card is inserted for mounting. If there is no device, the page will display the information shown in figure Memory Card Home Page. When a device is connected, the information displayed will be as shown in figure Memory Card Device Mounted, after the card has been properly mounted.

If the new configuration disables the interface, the information displayed will be as shown in the image below.

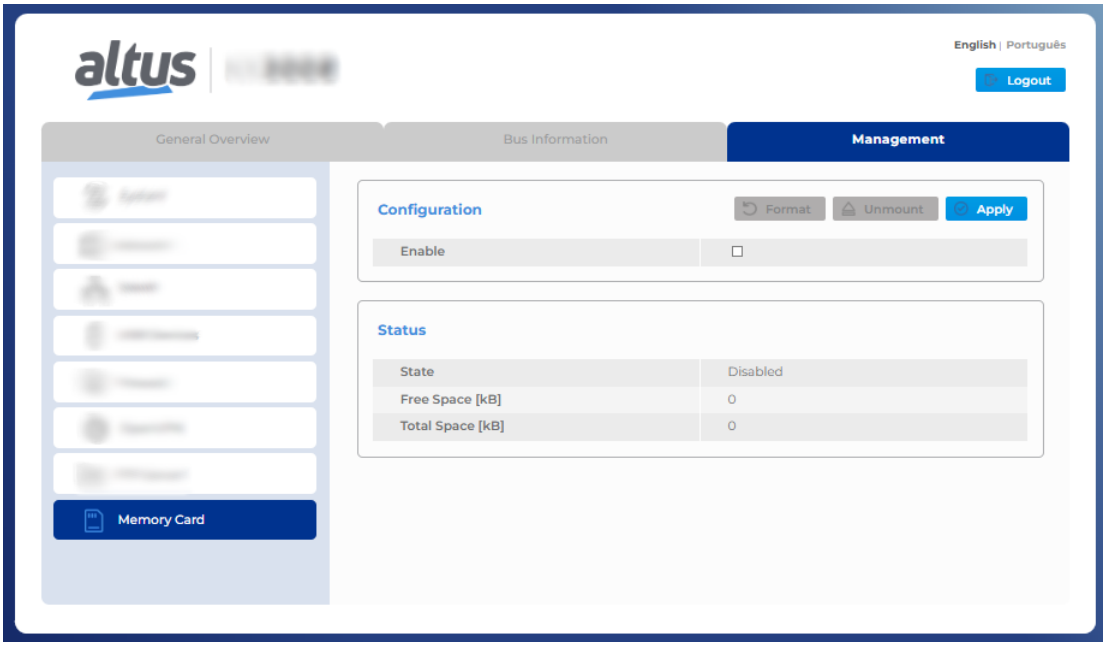

Figure 49: Memory Card Disabled Interface

ATTENTION

If the memory card deactivation is in effect, the MemoryCard folder will not be mounted.

6.6.1.4. Memory Card Interface Management by Application

To facilitate the management of the memory card interface, a function was developed that can be called directly by the user's application code. The **SetMemCardState** function was implemented within the **NextoStandard** library. The image below shows the library information, as presented in the *Library Manager*.

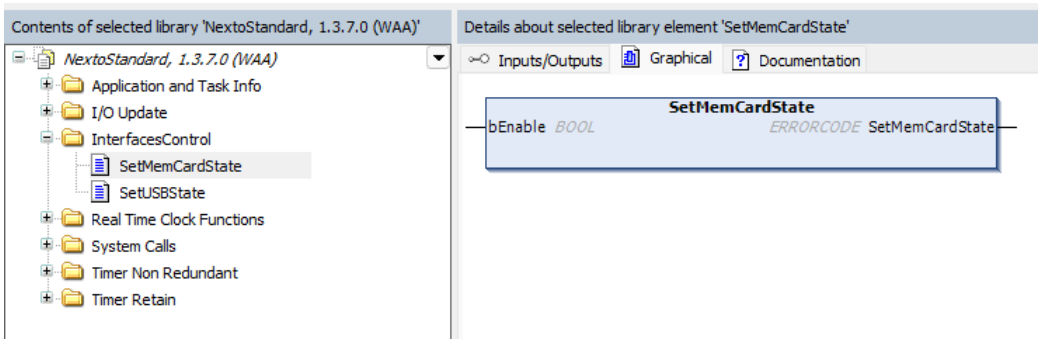

Figure 50: SetMemCardState information in Library Manager

The function has an input variable of type *bool*, **bEnable**, which receives the value to enable or disable the card interface. The function has three return values: *NoError* on success, *SetMemCardStateFail* on failure, or *ImportFunctionNotFound* if the function is not supported. The image below shows a basic example of variable declaration and function call.

```
PROGRAM UserPrg
VAR
  bSetMemoryCardInterfaceState : BOOL;
  stErrorCode : NextoStandard.ERRORCODE;
END_VAR
-------------------------------------------------------------------------
// Function call example to configurate memory card's interface
stErrorCode:= NextoStandard.SetMemCardState(bEnable :=
   bSetMemoryCardInterfaceState);
```

ATTENTION

The function executes the command to set the desired value for the memory card interface. It is neither necessary nor recommended that the function be called cyclically.

6.7. Firewall

RDR 5.2 from IEC 62443-4-2 standard

The Firewall is designed to increase the security of the device while it is in use. The main function of the Firewall is to filter data packets coming into and leaving the device. The implemented filter uses information from each data packet to decide whether that packet is allowed. The main parameters used are the input/output interfaces, the port, the transport layer protocol, and the source and destination addresses.

6.7.1. Settings

Firewall configuration is done through a dedicated section located in the *Management* tab of the controller's System Web Page, as shown below:

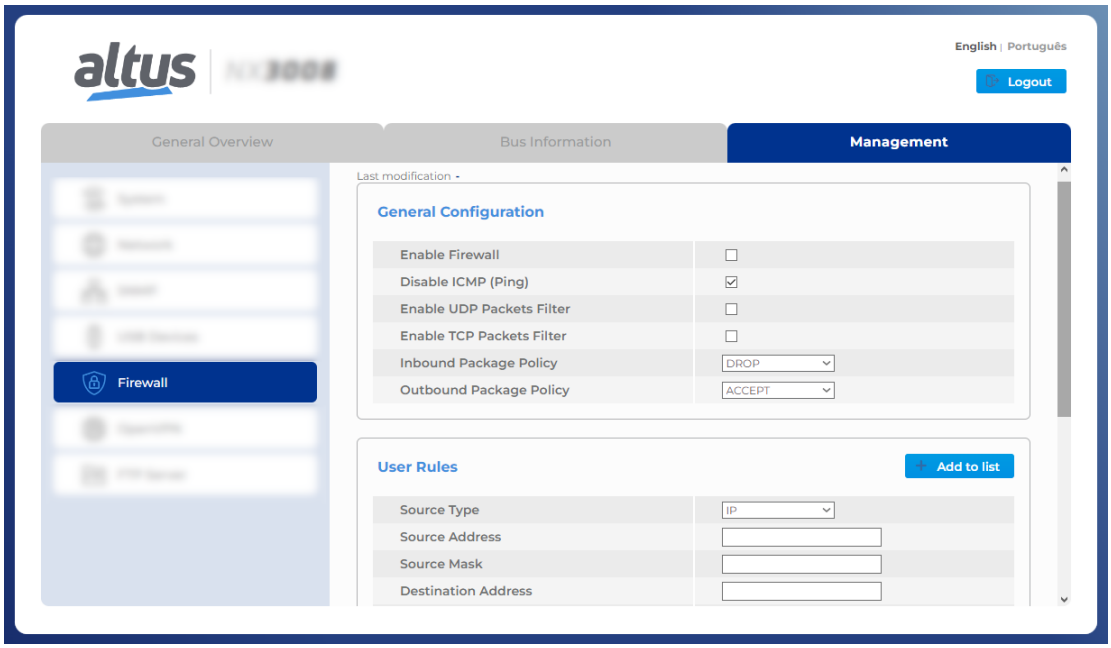

Figure 51: Firewall Configuration Screen

The Firewall is a separate feature from the programming tool, that is it doesn't require any interaction with the programming tool. Settings applied on the *Firewall* section take effect when confirmed with the *Apply* button, and are automatically saved in the controller. If the feature is enabled, it will operate again even after rebooting the device.

The following sections describe the possible settings for the Firewall, divided according to the tables of the *Firewall* section.

6.7.2. General Settings

The image below shows all the settings in the *General Configuration* table:

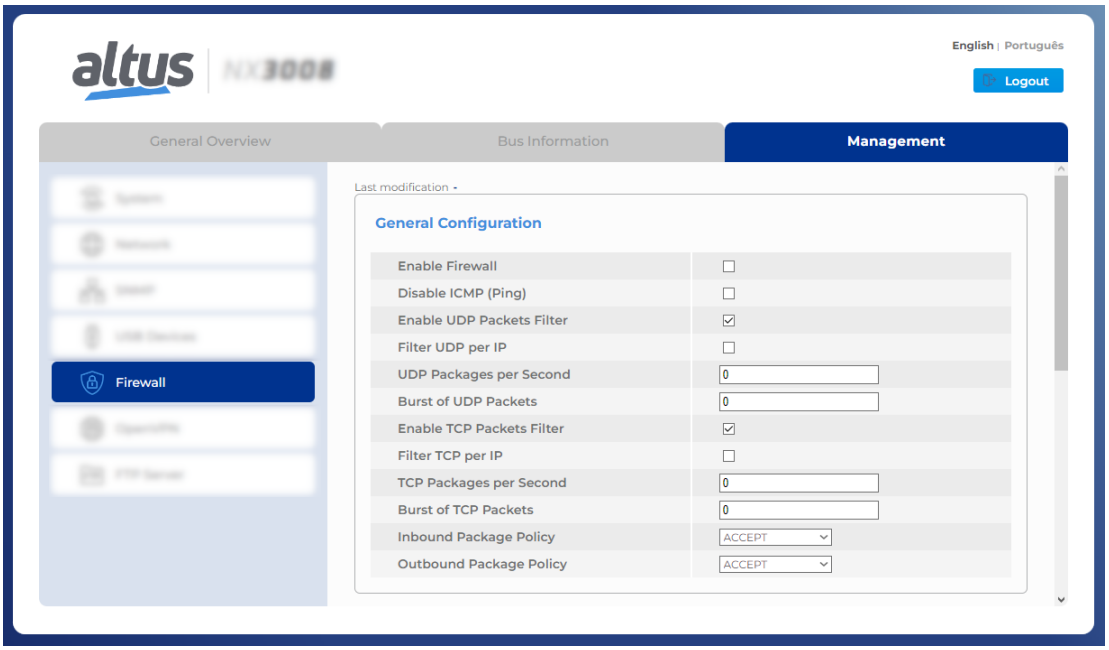

Figure 52: Firewall General Settings Table

This table expands dynamically by selecting the options to enable UDP and TCP packet filters, revealing all the items that can be configured. The first item in this table, *Enable Firewall*, is used to enable and disable this functionality. When the Firewall is enabled, the web page settings, when submitted to the device, will be applied to the configuration files, and then the Firewall will filter what has been configured. If the Firewall is disabled, the configuration that was made is stored, but the rules are not applied in the controller.

The field *Disable ICMP (Ping)* enables or disables protection against the ICMP protocol. When protection is enabled, the controller will not respond to *Ping* requests, since it will drop packets that use the ICMP protocol. When disabled, the operation of the device for *Ping* responses maintains its normal behavior.

When enabled, the fields that enable UDP and TCP packet filtering, filter these protocols according to the limits configured in their respective fields. The packet filtering rule works like this: for a packet to be accepted, there must be "credits" available, and one credit is used to accept a data packet.

The setting of the field *Burst of XXX Packets* sets the initial value of packages (credits), which will be accepted. In this way, it is possible to set an overflow limit for these packets, where if there is a large flow of packets, only the configured amount will be accepted. The *XXX Packages per Second* field sets how many credits that rule will earn per second. For example, if the value is 5, each second, the rule will receive five new credits, so it will be able to accept five more packages. The limitation for this increment in the number of credits is the configuration of *Burst of XXX Packets* itself, and the limit set here is not exceeded, even with the increment of packets every second. These settings are applied as a *stock*, where upon receiving a data packet, it is first checked if there is any credit available in the stock, and then a decision is made whether or not to accept the package. If the packet is accepted in this quantity filter, it is forwarded to the filter of the other firewall rules.

The setting *Filter XXX per IP* causes the rule to differentiate the source addresses of each packet and apply the packet per second and packet overflow filters individually to each IP address. So, going back to the previous example, it can be considered that each source address has its *stock* of credits, and one address cannot use the credits that are in the *stock* reserved for others.

ATTENTION

Negative values are not allowed for the *XXX Packages per Second* and *Burst of XXX Packets* fields. If negative values are set, when applying the settings an error message will appear on the screen indicating the field that had a conflict. If the filter is enabled, but the values in these fields are left at 0, the filter is not applied.

The settings in this table are applied with the *Apply* button that appears in figure 54.

The fields for selecting both incoming and outgoing policies have options to accept and drop. If the Firewall is active, when data packets arrive, all the rules that have been configured are checked, and then the policy configured for these packets is applied, whether *Accept* or *Drop*. So if an accept policy is set, *Accept*, all packets that do not match any configured rule will be accepted by the firewall, and if a reject policy is set, *Drop*, they would all be dropped.

6.7.3. User Rules

The *User Rules* table was created to allow greater control over the firewall's rule settings. With it, you can configure different rules dynamically and with more precise filters.

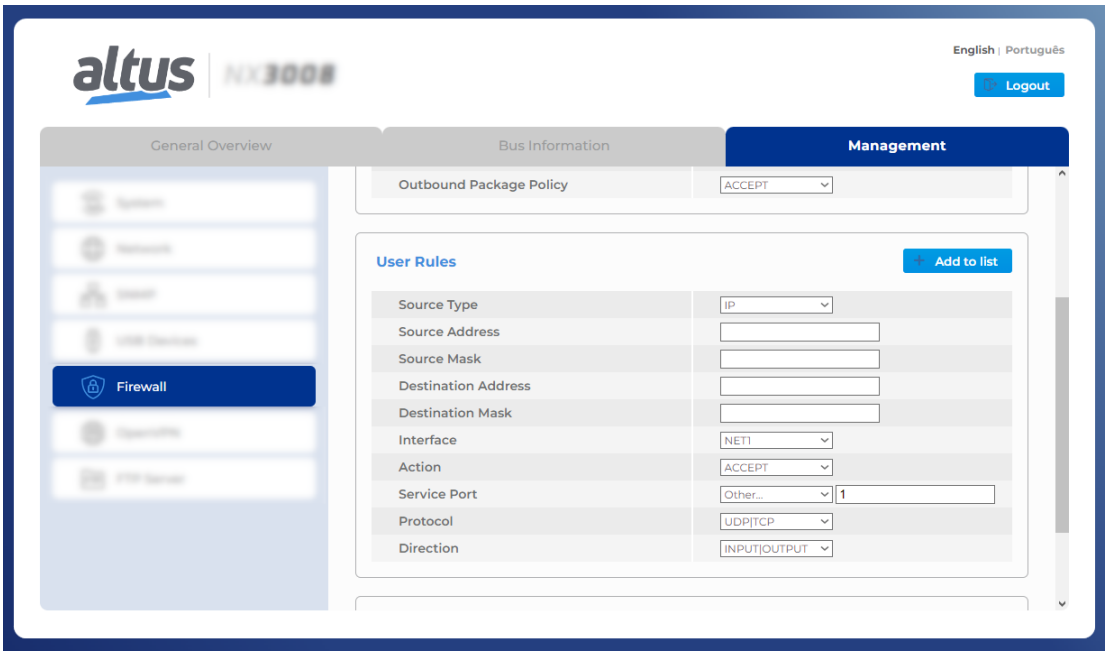

Figure 53: Firewall User Rules Configuration Table

This table changes its format according to the selected *Source Type*, which can be IP or MAC. When the type is *IP*, the table has the items shown in figure above, but when the type is selected as *MAC*, the source and destination mask fields disappear, as well as the *Destination Address* field. The item *Source Address* now accepts a MAC address as input in a format of six groups of two hexadecimal digits separated by colons, e.g. "1A:2B:3C:4D:5E:6F". Also, an address-based *MAC* rule can only be configured as an input rule. In other words, the *Direction* field will be forced with the value *INPUT*.

With the *Source Address* and *Destination Address* fields, you can enter the addresses that will be configured for that specific rule, and using the *Source Mask* and *Destination Mask* fields, you can configure a network range for this rule. If you only configure the address, only the address will be assigned to the rule but with different netmask configurations, you can get IP groups of various sizes to be applied to the rule.

Interface configuration makes it possible to individually select each physical or virtual interface available to the controller. Based on which interface you select for a given rule, only data packets entering or leaving the interface will be filtered by the Firewall. If you use the option *Any*, this rule will have no interface filter. So the filtering rule will be valid for all available interfaces.

The *Action* field has three configuration options: *ACCEPT*, *DROP*, and *REJECT*. The action sets up what should be done with the package whose characteristics match the rule applied. If the chosen action is *ACCEPT*, the data packet having characteristics according to the rule will be accepted. If it is *DROP*, the packet will be dropped, and no reply will be sent to the sender of the package. Finally, if it is set to *REJECT*, the packet will be rejected, and a reply will be sent to the sender, stating that the requested *host* is inaccessible.

The *Service Port* field, is used to indicate which ports will be configured in this rule. All service ports that have a certain protocol or communication *standard* for the controller, such as the MODBUS protocol that has the standard port 502, are available with the service name and port used next to it. Thus, if you configure the rule for the MODBUS protocol, port 502 will be applied if you configure the rule for the WebVisu service, port 8080 will be applied, and so on for the other protocols listed in the checkbox.

This field also has two other settings, which are *Any* and *Other*. When you select the *Any* option, the rule is applied to all service ports except port 80 then two rules are created using the following port ranges: *1:79* and *81:65535*. If you select the *Other* option, a text box appears in which you can configure the port you want, except for port 80. To configure a port, you can type its number in the text box, but if you want to add more than a single port, you must use the *"&"* separator, and if you want to insert a range of ports, simply enter the start and end port using the separator *":"*.

Example of configuring ports 120, 144, and the range 1300 to 1450 in the same field: *120 & 144 & 1300:1450*.

This field doesn't accept values outside the range 1:65535, port 80, or port repeats.

The HTTP port 80 can only be set by selecting it from the list of known protocols and cannot be applied to the NET 1 interface. So if the HTTP protocol is chosen, the Interface field *NET 1* and *Any* won't be selectable.

In the *Protocol* field, you can select between UDP, TCP, and UDP|TCP protocols. If you select the UDP|TCP option, two rules will be created on the firewall, one for each transport protocol.

In the *Direction* field, you can select between *INPUT*, *OUTPUT*, and *INPUT\OUTPUT*. These options cause the rule to be applied to packets arriving at the device, option *INPUT*, or leaving it, option *OUTPUT*. If the joint option is configured, two rules will be created, one with each direction option.

The figure below demonstrates how a rule is applied:

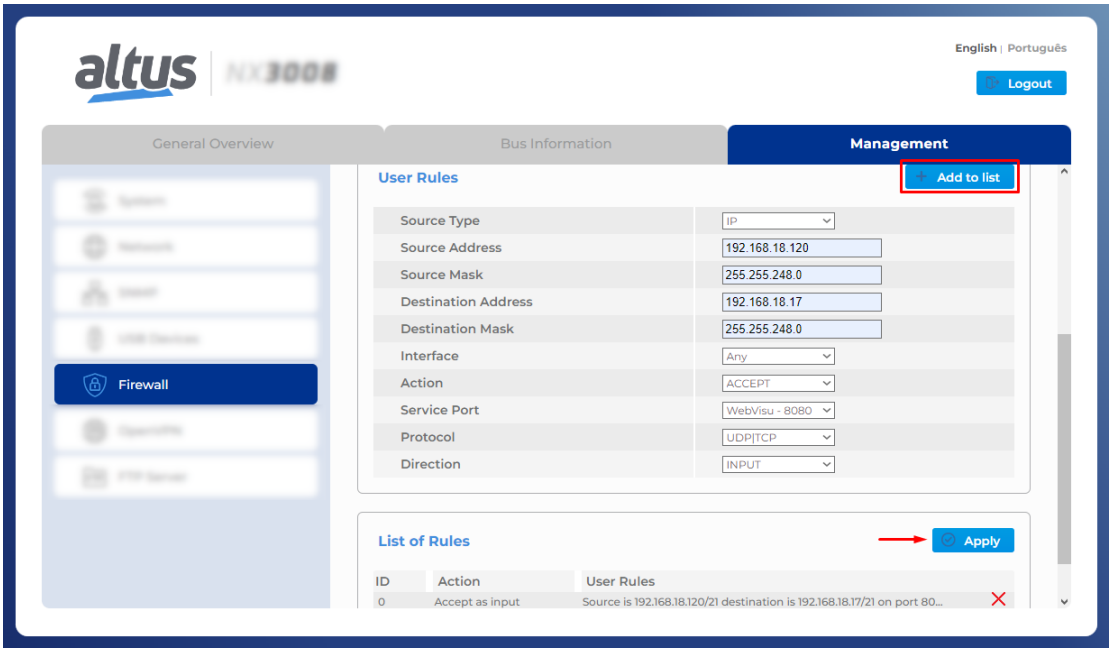

Figure 54: Firewall User Rules Enforcement Table

After filling in the fields as you wish to configure the firewall rule, you must click the *Add to list* button. By doing this, all the settings will be analyzed to check if there are invalid values or if there is any duplicate rule. It's impossible to add two rules with the same address, mask, interface, port, and direction parameters. If a conflict is found, a message will be displayed indicating the field that contains an invalid setting or the ID number of the rule in the table whose settings caused the conflict with the newly configured one.

After all, parameters are checked, the rule will be added to the list below the configuration table. This list expands automatically as rules are added or deleted. If you want to exclude a rule from the list, you can place the mouse over the one you want to exclude. When you do this, a red *X* button will appear on the right, as shown in the previous figure. By clicking it, the rule will be deleted from the table.

When adding new rules, or deleting an existing one, in the rules table, the *Apply* button below must be clicked for the configuration to be applied to the device.

ATTENTION

During the application of firewall rules, there may be a momentary instability in Ethernet communication.

6.8. OpenVPN

RDR 5.3 from IEC 62443-4-2 standard

VPN (Virtual Private Network), used for surfing unsecured networks, transmitting data, or simply accessing the Internet with a high level of security and privacy. The VPN virtual network can be understood as a tunnel in which information travels securely, protected by security certificates and keys. OpenVPN is an *open-source* service, which means that it is free to use and distribute, and its source code is open for modifications if needed.

The main purpose of a VPN is to communicate securely over an unsecured network. To make this possible, data encryption is used based on certificates and keys generated using TLS, Transport Layer Security, a protocol that performs 256-bit encryption, one of the most secure.

To perform the configuration of the OpenVPN client or server, the OpenVPN page was created in the *Management* tab of the CPU's System Web Page. As shown in the figure below.

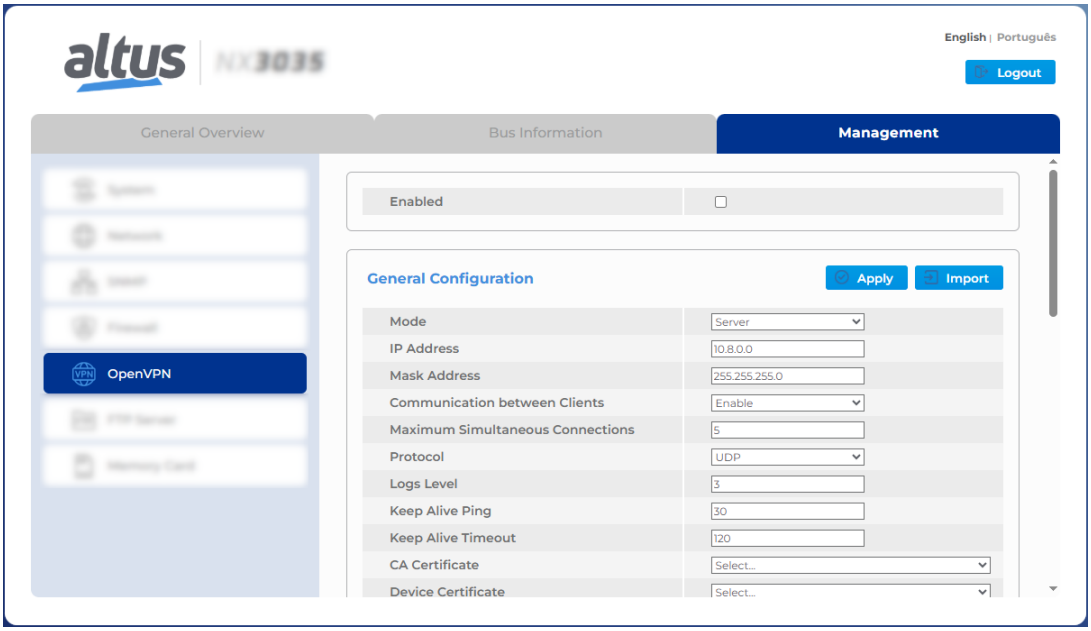

Figure 55: OpenVPN Configuration Screen

Because it is located within the *Management* tab, access to this page is password protected. The following sections describe the settings and functionality of this page.

6.8.1. Importing Configurations

To quickly and easily configure the VPN on your device, you can use the *Import* button that appears in the picture 55 in the upper right corner of the page. Clicking on this button opens a file explorer window where you can select a configuration file. Files with extension *.conf* or *.ovpn* should be selected. When you select a file, its contents will be read and the configuration parameters present will fill their respective configuration fields on the web page.

For the file's parameters to be interpreted correctly, they must follow standard OpenVPN configuration file syntax.

If there are security files, certificates, or keys, written in the configuration file, along with the other parameters, they will be read and separated into separate files within the controller for use.

ATTENTION

Do not use spaces to separate the words in the name of the *".conf"* files. Instead, use *"_"* to separate them.

6.8.2. OpenVPN Configuration

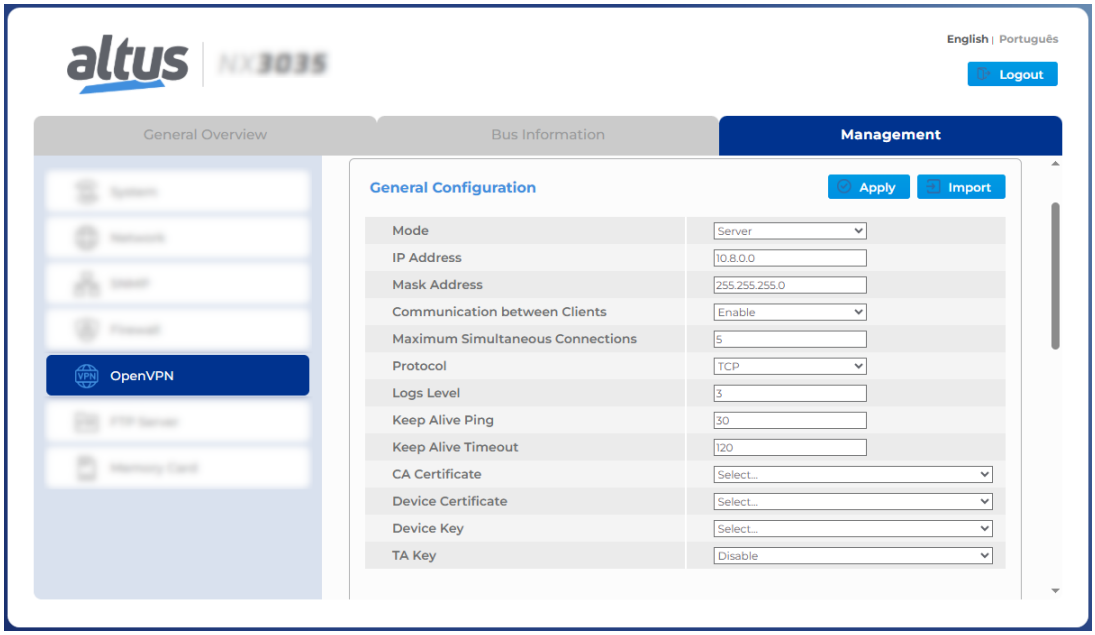

Figure 56: OpenVPN Server Configuration Table

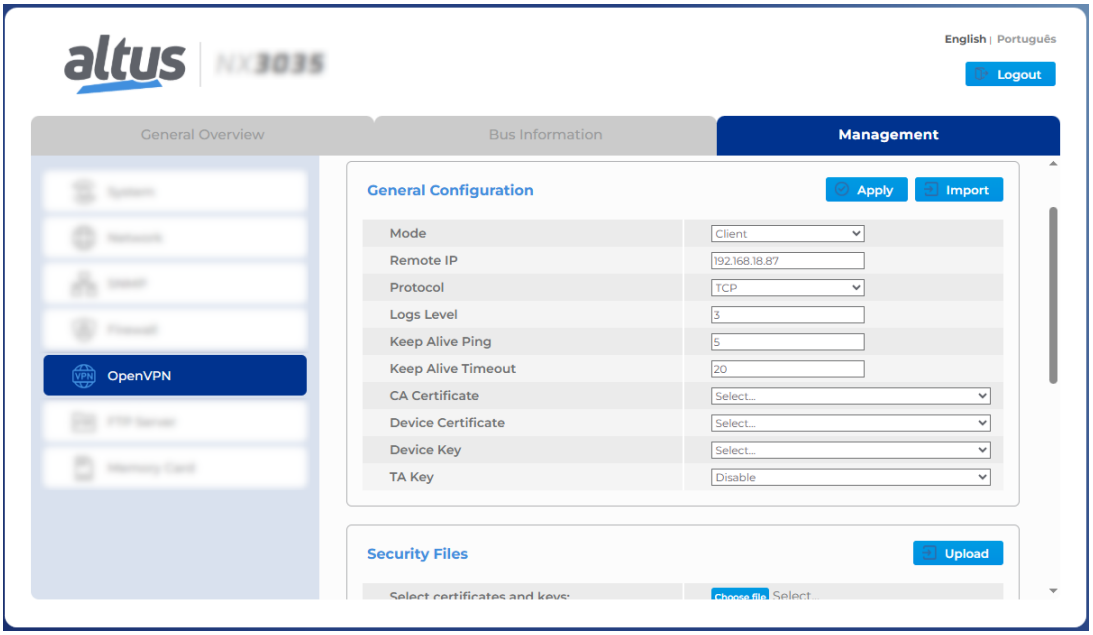

Figure 57: OpenVPN Client Configuration Table

This section shows how OpenVPN configuration is performed. The settings will be divided into three parts: settings common to both operating modes, settings unique to a server, and settings unique to a client.

6.8.2.1. Common Configurations

Looking at the figures with the client configurations, figure 57, and the server configuration, figure 56, you can identify that several parameters are the same for both configurations. These are:

6.8.2.1.1. Mode

With the configuration of the *Mode*, you can select between two options, client or server. When you select one of the two modes, the settings table changes automatically to allow the configuration of the necessary fields for each mode of operation.

6.8.2.1.2. Protocol

This field configures which transport protocol will be used for VPN communication. It can be set between UDP and TCP.

ATTENTION

The configuration of the server and all its clients must be the same. With a divergent configuration, OpenVPN is not able to perform communication.

6.8.2.1.3. Log Level

This field sets the level that the log file will receive. The setting ranges from 0 to 5, 0 being the most basic level and 5 being the most advanced.

Level 0 only displays logs about some critical failure in OpenVPN and levels 4 and above are used for debugging as there is a lot of information being written to the log file. For normal operation, it is recommended to use value 3.

This field only accepts numbers as input. You are not allowed to use letters or special characters.

6.8.2.1.4. Keep Alive Ping

This field sets the time, *in seconds* when the *Ping* request will be forwarded. This request serves to verify the connection between the server and the clients.

This parameter can be set on both the server and the OpenVPN clients, but if this parameter is set on the server, the clients will assume the server's value and not the value set on them. If the server doesn't have such a setting, each client assumes its setting normally. If you want to disable pinging between the server and the clients, set the value to 0.

This field only accepts numbers as input. You are not allowed to use letters or special characters.

6.8.2.1.5. Keep Alive Timeout

This field sets the time, *in seconds* when the timeout of the *Ping* request will occur. After the expiration of this time, without a response from the other VPN device, it will be considered disconnected.

This parameter can be set on both the server and the OpenVPN clients, but if this parameter is set on the server, the clients will assume half of the server's value and not the value set on them. Clients receive half the amount to ensure that they are disconnected in case the server disconnects. If the Server does not have such a setting, each client assumes its setting normally. If you wish to disable this feature, set the value to 0.

This field only accepts numbers as input. You are not allowed to use letters or special characters.

6.8.2.1.6. Security Files

In the fields *CA Certificate*, *Device Certificate*, *Device Key* and *TA Key*, you must select which security file, certificate, or key, will be used to establish the OpenVPN communication. The options in each field, *combobox*, are filtered according to the type of key file or certificate, although there is no differentiation between keys and certificates.

To be possible to select a file, it must first have been imported.

All security files are required for correct communication to be established between clients and the VPN server, except for *TAP Key*. This key is optional for communication, but if it is used on the server, it becomes mandatory for all clients on the server.

See the TLS Certificates and Keys Management section for further information about generating certificates and security keys based on TLS.

6.8.2.1.7. TA Keys

In the field *TA Key* it is set which type of encryption will be applied to the *TA Key*. This field stays hidden until you select a file for the TLS key because it is only used in conjunction with this key. The default value of this parameter is *SHA1*, but you can select from the following values: *SHA256*, *SHA512*, and *MD5*, in addition to the default SHA1.

ATTENTION

This configuration needs to be the same between the clients and the server in the same OpenVPN network. If the value of this field is different between the client and server, the connection will not be established.

6.8.2.2. Server-Specific Configurations

The exclusive server configurations, seen in figure 56, are described below.

6.8.2.2.1. Network Address

The IP range that will be used to assign the server and client addresses for the VPN network is configured by the server by setting the *IP Address* and *Mask Address* fields. All IPs that will be assigned to the clients and the server will be taken from the specified range.

The server's IP address is always the first available value in the configured range, and for IP assignments to clients, the values still available in the range are used, so the first available value is assigned as clients make their connection. For example, if a network is configured with the addresses 10.8.12.4 and mask 255.255.255.248, the server will assume IP 10.8.12.5 which is the first available address in the configured range. However, if mask 255.255.255.255.0 is set, the server will assume IP 10.8.12.1, which is the first available address in the range.

The IP and Mask address fields only accept settings that have the syntax of an IP address and mask address, respectively. If anything out of the standard is configured, an alert message will be displayed, informing you that an error has occurred.

6.8.2.2.2. Communications Between Clients

In this field, you can enable or disable communication between clients in the VPN network. When the option is selected as *Disabled*, only client-server communication can be performed directly. If the option selected is *Enabled*, it will allow communication between the clients themselves in addition to the client-server communication.

6.8.2.2.3. Maximum Connected Clients

In this field, you can set the maximum number of clients that can connect to the server simultaneously. This field accepts only numeric characters, and the minimum value is 1.

6.8.2.2.4. Private Networks

When you select OpenPVN's operating mode as a server, a table will be displayed, normally hidden, which allows the configuration of private networks that can be below the server and each client.

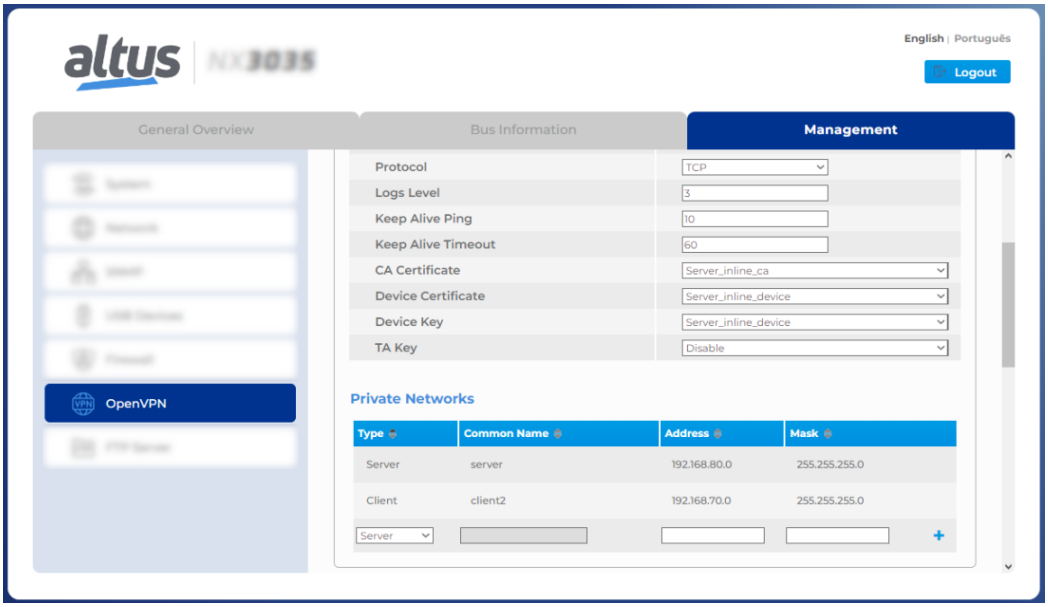

Figure 58: OpenVPN Private Network Configuration Table

To configure a private network that is below the server, simply select the network type as *Server* and configure the network addresses and mask. Configuring a private network for a client requires, in addition to setting the type as *Client*, to enter the *Common Name* of the client that owns the network being configured.

The Common Name of a client is set when generating the *Device Certificate*. This parameter is entered when creating the certificate and is unique for each client and server. The configuration of these private networks creates a routing table that will be checked when receiving or sending packets over the VPN.

Figure above, shows a configuration of a subnet *80* on the OpenVPN server, then a routing rule will be configured that will forward the data packets that will be received by the VPN to the device interface configured on this network. It also creates a rule, internal to the server, that if a data packet has the subnet *70*, this packet will be routed and forwarded through the VPN tunnel. The same behavior occurs with the *client2* client, but with the subnets switched, because below this client is the *70* subnet and it will forward packets with the *80* subnet to the VPN tunnel.

See the following figure for an example of architecture:

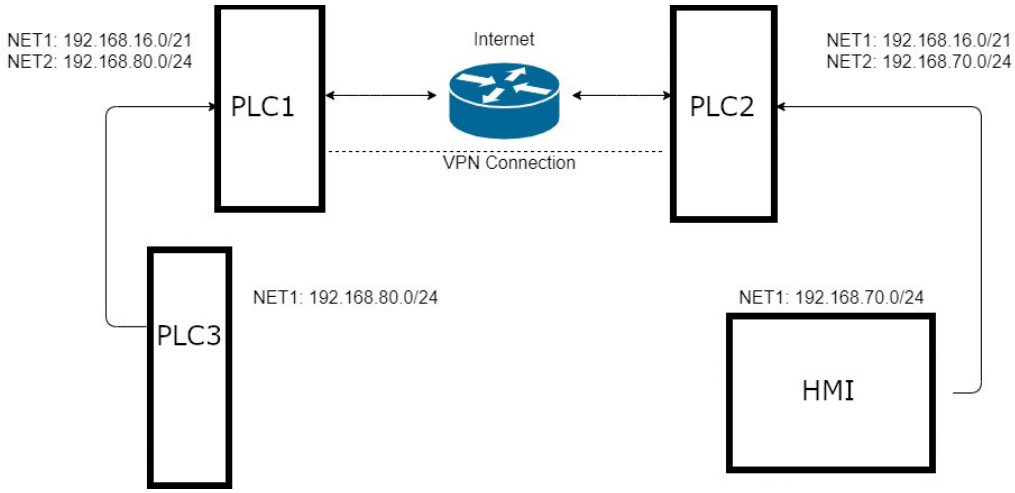

Figure 59: Architecture Example with Private Networks

In Figure 59, the PLC1 on the left has a private network 80 configured on its NET 2, and connected to it is the PLC3 on the same network. The PLC2 on the right has a private network 70 configured on its NET 2 and connected to it is an HMI, on the same network. The example architecture realizes the communication between the PLC3 and HMI devices over VPN by configuring their respective private networks.

After filling the fields, shown in Figure 58, with the desired configuration, you must click on the blue + button that appears on the far right of the configuration fields, so that the rule is added to the table. If you want to delete a rule, drag your mouse over the rule you want to remove, and a red *X* will appear on the right, as shown in Figure 58. By clicking on this *X*, the rule is removed from the table.

For the settings present in the table to be applied to the device you must click the *Apply* button and confirm the operation in the confirmation window that will appear. When the rules are applied, a message will be displayed indicating whether the operation was successful or not.

6.8.2.3. Client-Specific Configurations

There is only one configuration unique to OpenVPN clients on the page, which you can see in the picture 57. This configuration is the *IP Remote*.

6.8.2.3.1. Remote IP

The Remote IP field sets the address where the VPN server is expecting communication from the clients. If an OpenVPN server is established on a computer, the remote IP configuration must be done according to the IP address of this computer. This field also accepts *host names* as the remote address, so you can set an IP or a hostname in this parameter.

ATTENTION

Because of the need to allow for such different parameters, IPs, and host names, the only check that exists in this field is whether or not data exists. Be careful when performing the configuration.

6.8.2.4. Applying Configurations

To enable the functionality, the checkbox *Enabled*, shown in the figure above, must be checked. If you just want to apply the settings you have made and not enable OpenVPN, uncheck this checkbox.

After you have made all the desired settings, the settings must be applied to the device, to do this use the *Apply* button. This button is shown in the figure 57 in the lower right corner. When the settings are applied and the VPN is enabled, the web page will perform an automatic *scroll* to the OpenVPN *status* table, displayed in the Status Table section.

6.8.3. Security Files

Security files are used to establish OpenVPN's communication securely by performing the role of encrypting and decrypting the data packets that will travel through the VPN tunnel. In the TLS Certificates and Keys Management section, it is described how to generate TLS keys and certificates. Here is a screenshot that shows the section responsible for managing the security files:

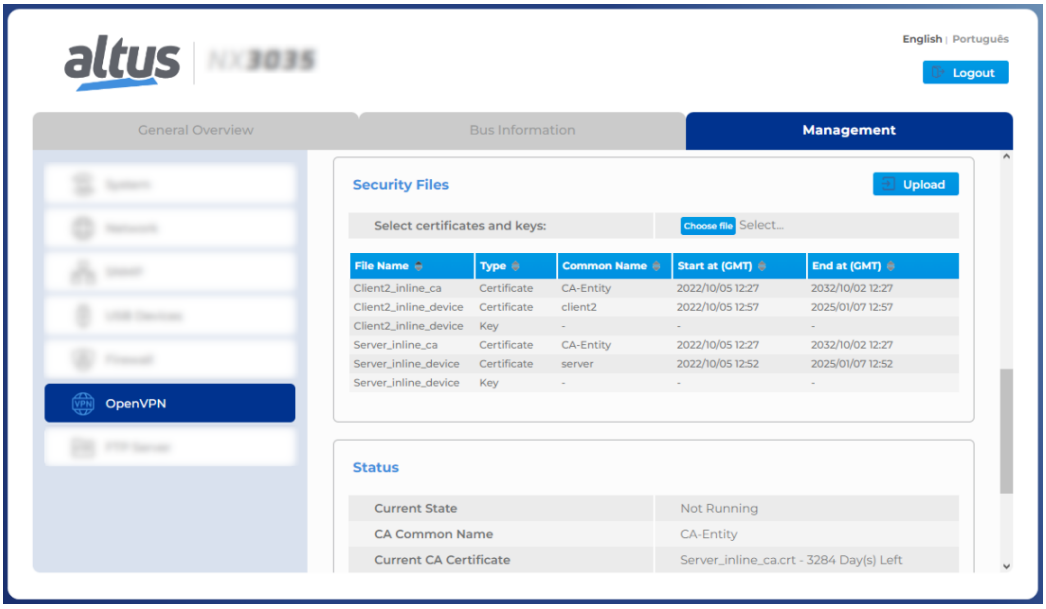

Figure 60: OpenVPN Security Files Table

In this section of the System Web Page, you can manage the security files. You can import files, monitor the validity of certificates, download files uploaded to the device, and delete files that have been uploaded.

By clicking the *Choose files* button, you can import certificates and keys, these files must have the respective extensions *.crt* and *.key*. This button opens a file explorer window and allows the selection of one file, i.e. multiple files.

ATTENTION

There is a limit of 12 files that can be imported into the controller.

The control of the files is done in the table, which is shown in the picture 60. This table adds new items, or removes them, as the import or delete operations occur. You can identify whether the file is a key or a certificate by the second item in the list, the *Type*, which indicates what that file is. For the certificates, their *commons names* and their expiry dates, both start and expiry, are also displayed.

You can recover a file that has been imported into the part and also delete it. When you drag the mouse over a file in the table, two buttons appear, one for downloading and one for deleting. The download button is a black arrow pointing downwards, and the delete button is a red *X*.

6.8.4. Status Table

Designed to allow for data monitoring, OpenVPN's status table automatically expands as you change settings and displays various data about the connection such as the state of the VPN, the VPN IP assigned to that device, the data being transmitted, and the security files being used for communication.

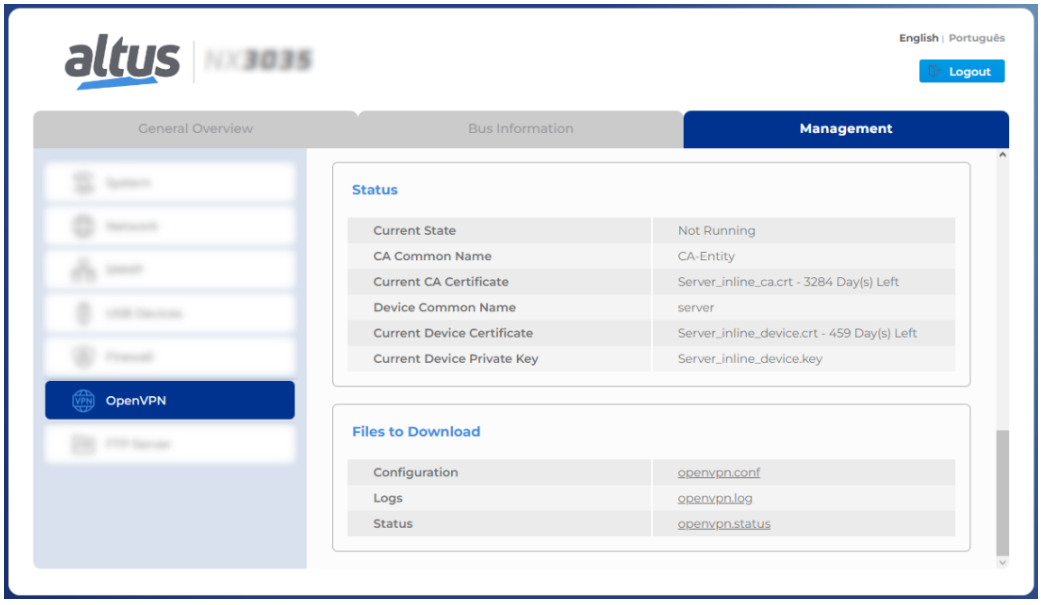

Figure 61: OpenVPN Status Table with Feature Disabled

When VPN is disabled, the table has few parameters. The field *Current State* indicates whether the VPN is enabled or not, and the other fields show which certificates and keys are configured for VPN communication. If one of the security files has not been selected, the character "-" will appear instead of its name, indicating that there is no file configured.

The common name fields for the CA and the device display the names given to the respective certificates, certificate authority, and device.

Next to the file name of each certificate is displayed the remaining time, *in days*, until its expiration date.

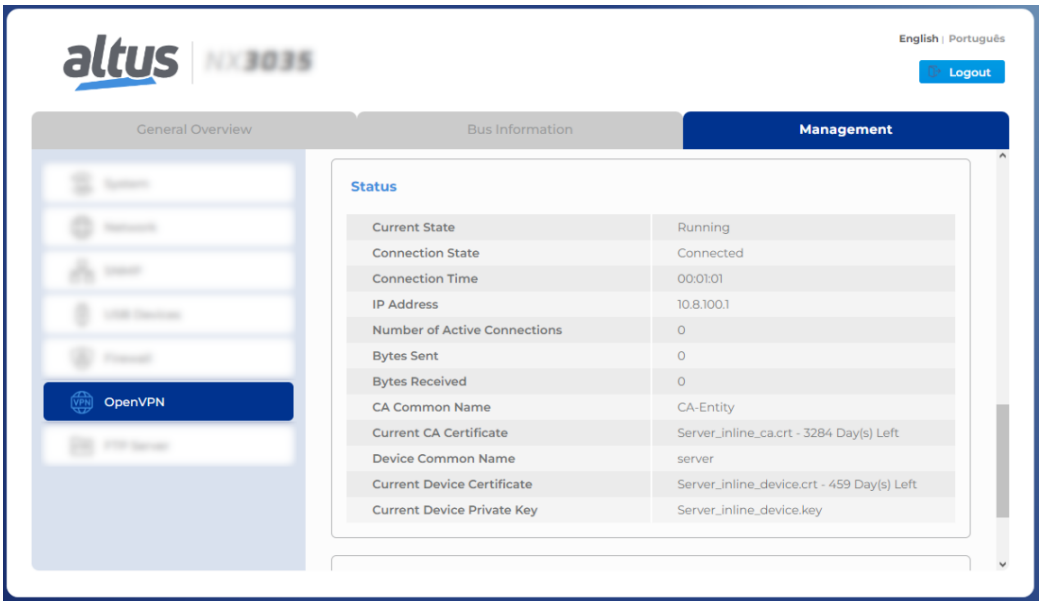

Figure 62: OpenVPN Status Table with Feature Enabled

When the functionality is enabled and the settings are applied on the device, the table has its cells dynamically modified so that the remaining information is displayed. Information about the OpenVPN connection status can be found in the first two topics of the list.

The item *Current State* has the states of *Not Running*, *Starting service...*, and *Running*, which indicate respectively that the VPN is disabled, is starting or is enabled.

The item *Connection State* has the states *Not connected*, *Connecting...*, and *Connected*.

The other information that can be obtained from this table is the total connection time, the device's IP address, and the amount of data sent and received, in bytes. The status of how many clients are currently connected is only displayed when OpenVPN is operating as a server.

6.8.5. Files to Download

You can check the information generated by OpenVPN through status and log files. The list of files to download is only displayed when there is a file to download. If there is none, the message "No file found in the controller!" is displayed. Clicking on any of the links will download the requested file through the browser.

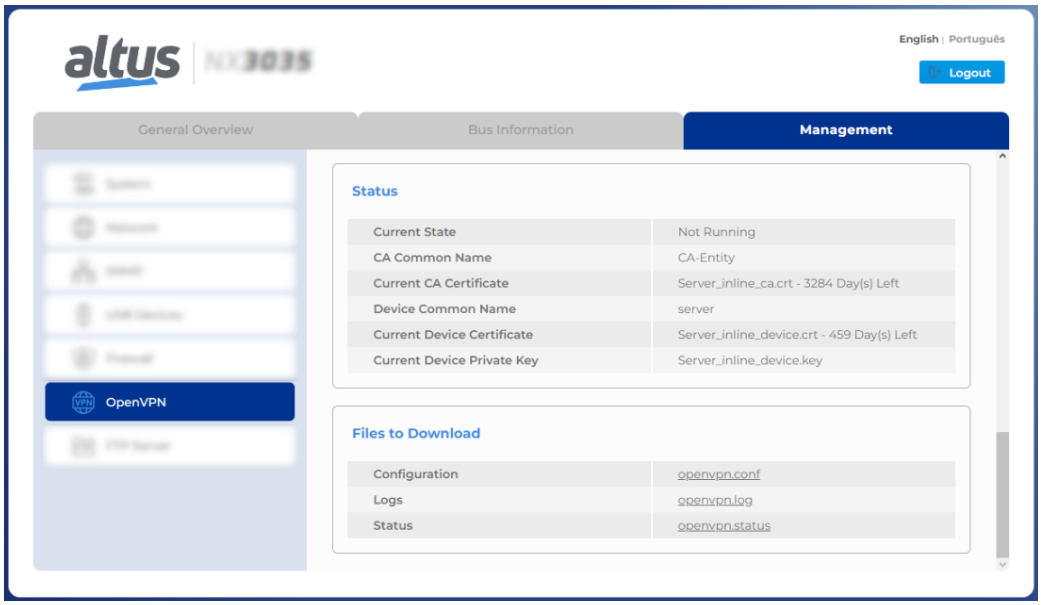

Figure 63: OpenVPN Download Section

6.8.6. Archtectures Configuration

This section will cover some possible configurations for OpenVPN, such as Host-to-Host, Host-to-Site, and Site-to-Site architectures.

6.8.6.1. Host-to-Host

Figure 64: Example of Host-to-Host architecture

This topology allows the connection between two VPN hosts. Both hosts can be chosen to be configured as the server, then the other should be configured as the client, or both hosts can be configured as clients and have a third host that will be the server for the VPN network.

Setting up this type of architecture doesn't require any specific configuration. In other words, there is no restriction on the settings available on the OpenVPN web page.

6.8.6.2. Host-to-Site

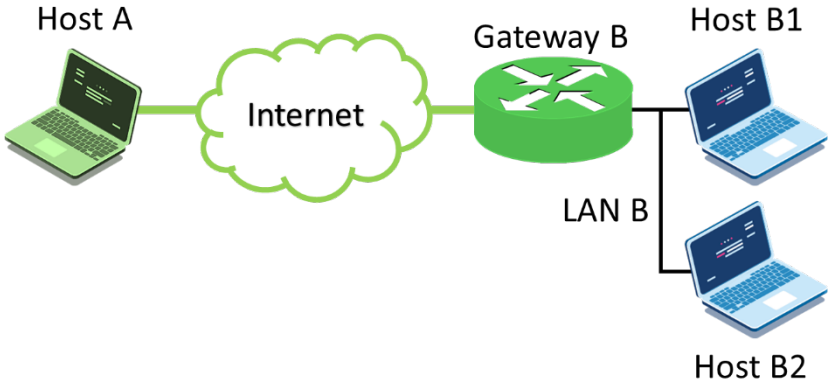

Figure 65: Example of Host-to-Site architecture

This topology allows the connection between two VPN hosts, but one of these hosts also acts as a gateway to the VPN network. Through this gateway, routing is performed to set up communication between hosts A, B1, B2, and Gateway B. In this scenario, either Host A or Gateway B can be the server. When one is the server on the network, the other will be the client.

The hosts, B1 and B2, that are on a private Lan B network below Gateway B, don't need to support OpenVPN to be able to communicate since all communication is handled by the VPN network gateway.

To enable communication between all devices on the network, you need to create routing rules for the VPN tunnel. Please refer to the section Private Networks to see how to create private network rules.

This VPN connection architecture requires some specific configurations. The server must have its topology configuration as a Subnet, this being the default configuration of the controller, to configure the private networks under Gateway B, as seen in the image above.

You also need to enter the address of the private network, Lan B, that will be communicating through the VPN. This configuration is done using the command *push "route Lan_B_IP Lan_B's_Mask"* and is required regardless of whether the private network is located below the client or the OpenVPN server, but if the private network is below the VPN client, you must add, in addition to this command, the following configuration: *route route Lan_B_IP Lan_B's_Mask*. These settings are written to the VPN server's configuration file.

6.8.6.3. Site-to-Site

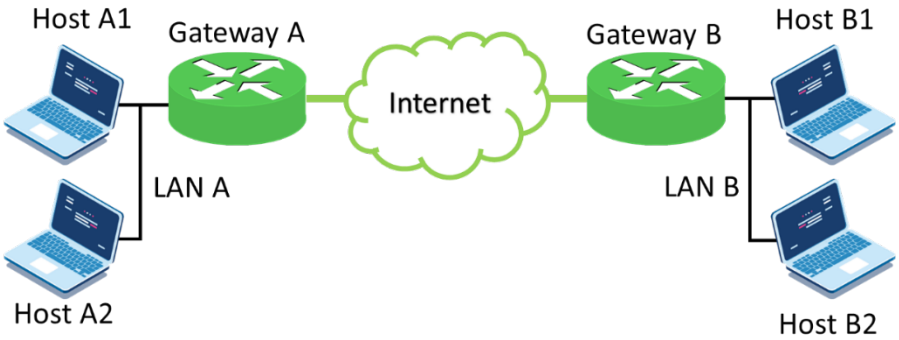

Figure 66: Example of Site-to-Site architecture

This topology allows the connection between two VPN hosts, both of which acts as a gateway to the VPN network. Through these gateways, access is provided to establish communication between hosts A1, A2, B1, B2, Gateway A, and Gateway B. In this scenario, any gateway can assume the role of a server, so the other will be the client.

None of the hosts that are in a private network below one of the two gateways need to support OpenVPN to be able to communicate, since all communication is handled by the VPN network gateways.

To enable communication between all devices on the network, you need to create routing rules for the VPN tunnel. Please refer to the section Private Networks to see how to create private network rules.

The configurations for this architecture need the same specific settings described in section Host-to-Site, with the difference that now, there are two private networks, and both must follow the configuration that has been demonstrated. Assuming that Gateway A is the server on this connection, you should add the following commands to the configuration file: *push "route Lan_A_IP Lan_A's_Mask"*, *route Lan_B_IP Lan_B's_Mask*, and *push "route Lan_B_IP Lan_B's_Mask"*. If the server is Gateway B, in the configuration file it would be added: *push "route Lan_B_IP Lan_B's_Mask"*, *route Lan_A_IP Lan_A's_Mask*, and *push "route Lan_A_IP Lan_A's_Mask"*.

6.9. Secure OPC UA Server

RC 3.1 from IEC 62443-4-2 standard

OPC UA is an industrial communication protocol for interoperability developed by the OPC Foundation. MasterTool is equipped with an OPC UA server functionality to provide access to the controller and its application. Several security measures are provided, including an OPC UA server operating with encrypted communication based on X.509 certificates and acting with access to a specific user-defined set of symbols, ensuring greater confidentiality for the data exchanged with connected clients.

6.9.1. OPC UA Server: User Management Available

RC 2.1 from IEC 62443-4-2 standard

By adding the *Symbol Configuration* object to the project for OPC communication, subsets of all defined symbols (symbolSets) can be created. In the *symbol configuration* window, under the *Settings* menu, symbol sets can be activated, as shown in Figure 67.

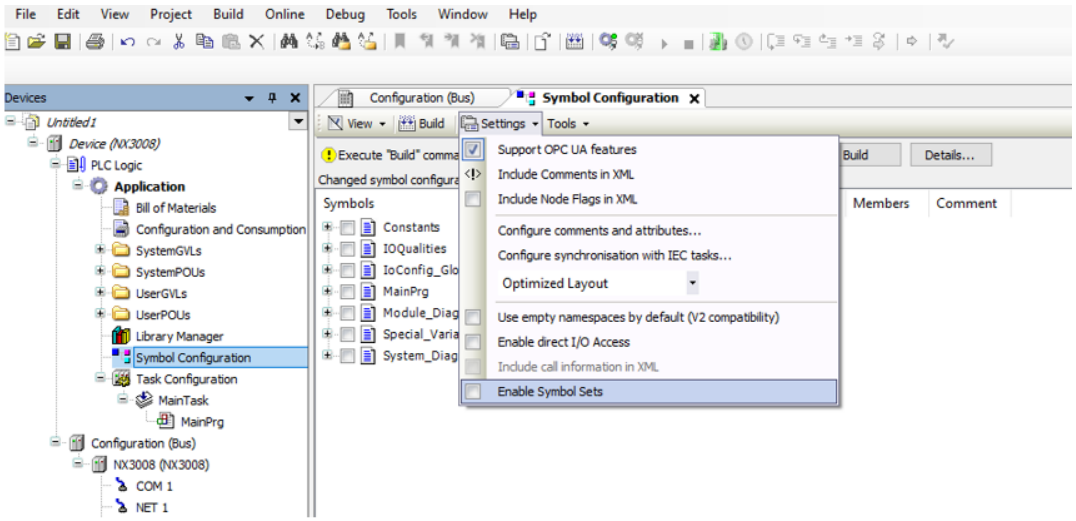

Figure 67: Activating symbol sets.

Once activated, a new symbol set can be created using the "+" button, as shown in Figure 68.

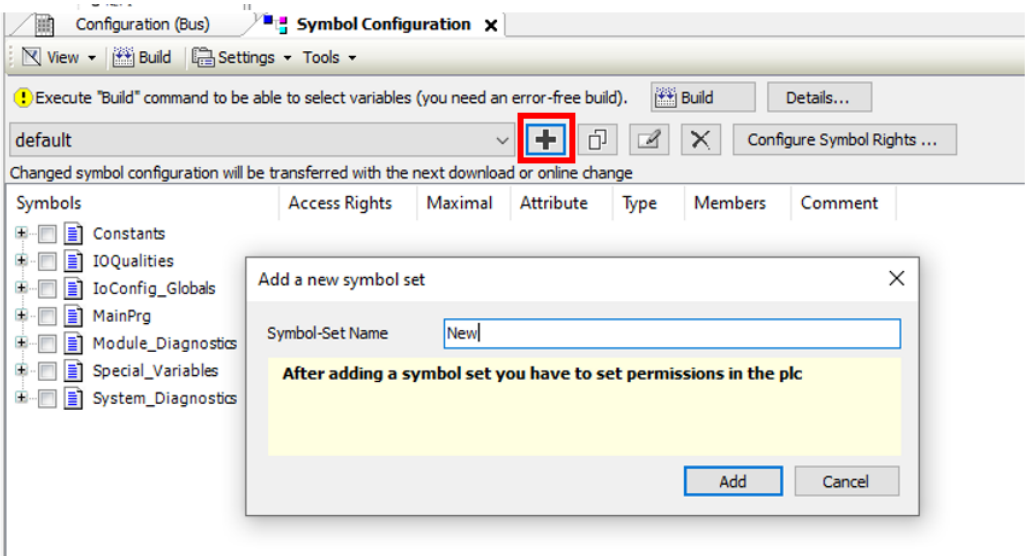

Figure 68: Creating a new symbol set.

In the symbol list, the desired symbols can be selected for inclusion in the group. With user management enabled, these symbol sets can be assigned to specific users for visibility and the determination of read/write access rights, ensuring the confidentiality of the data exchanged with connected clients. This is done after creating the symbol subset, using the *Configure Symbol Rights...* button, in the *Users and Groups* and *Access Rights* tabs. For more information on *User Management* and *Access Rights*, refer to Chapter 6.1.

6.9.2. OPC UA Server: Support for X.509 Certificate-based Communication RC 3.1 from IEC 62443-4-2 standard

One of the features of the OPC UA Server is to operate with encrypted communication based on X.509 certificates. Different security profiles are defined by the OPC Foundation.

Depending on the profile, this protects the integrity (for signed profiles only) or the integrity and confidentiality (for signed and encrypted profiles) of the data exchanged with connected clients.

This measure safeguards the confidentiality of the data exchanged with connected clients.

To verify centificate configurations, access *View* > *Security Screen*. If desired, the user can configure encryption for OPC UA communication using the *Basic256SHA256* profile, for a secure connection (cyber security).

To configure encryption on an OPC UA server, you must create a certificate for it using the following steps in the Mastertool programmer:

1. Define an active path for communication with the controller (no login required);
2. From the *View* menu, select *Security Screen*;
3. Click the *Devices* tab on the left side of this screen;
4. Click the icon to perform a refresh;
5. Click on the *Device* icon, below which will open several certificates (*Own Certificates, Trusted Certificates, Untrusted Certificates, Quarantined Certificates*);
6. Click the icon to generate a certificate and select the following parameters:
   - *Key length* (bit): 3072
   - *Validity period* (days): 365 (can be modified if desired)
7. Wait while the certificate is calculated and transferred to the controller (this may take a few minutes);
8. Reboot the controller.
9. On the OPC UA client, perform the necessary procedures to connect to the OPC UA server and generate a certificate with the *Basic256Sha256* profile (see specific OPC UA client manual for details);
10. Back to Mastertool, click on the icon of the *Security Screen* to perform a refresh;
11. On the *Security Screen*, select the "*Quarantined Certificates*" folder under the *Device*. In the right panel you should observe a certificate requested by the OPC UA client;
12. Drag this certificate to the folder "*Trusted Certificates*";
13. Proceed with the settings in the OPC UA client (see specific OPC UA client manual for details).

To remove encryption previously configured on a controller, you must do the following:

1. Define an active path for communication with the controller (no login required);
2. From menu *View*, select *Security Screen*;
3. Click on the *Devices* on the left side of this screen;
4. Click the icon to perform a refresh;
5. Click on the *Device* icon, below which will open several certificates (*Own Certificates, Trusted Certificates, Untrusted Certificates, Quarantined Certificates*);
6. Click the folder "*Own Certificates*" and in the right panel select the certificate (OPC UA Server);
7. Click the icon to remove this project and driver certificate;
8. Reset (turn off and on) the controller.

6.10. Resource Management

RC 7.2 from IEC 62443-4-2 standard

For equipment with multiple communication interfaces, it is possible to disable some of them through Mastertool, if desired. However, it is essential that at least one interface remains active to ensure communication between the user and the controller.

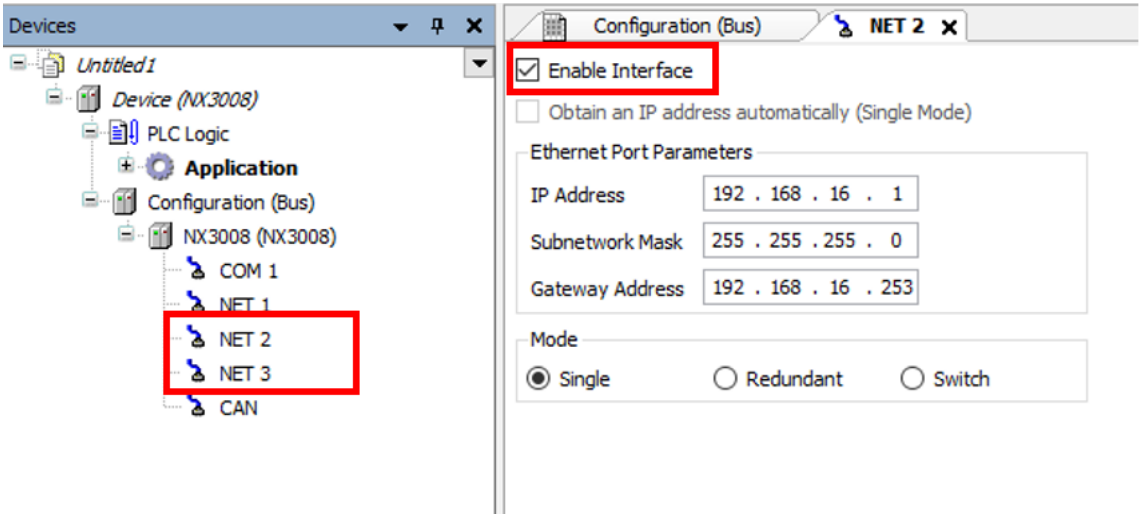

Figure 69: Disabling Network Interfaces

Table 4 indicates the reserved TCP ports of the equipment. These ports remain open only when the corresponding protocol is in use; otherwise, they remain closed.

6.11. System Recovery

RC 7.4 from IEC 62443-4-2 standard

The components have the ability to recover to a known state after a failure by performing procedures that aim to save data and configurations. This must be done by the user when the system is fully operational, thus creating a complete backup of the project.

6.11.1. User Settings

The export of user permissions and access rights for the project is done from the menu *Project* > *User Management* > *Permissions*. By clicking on *Export/Import* and then *Export all permissions...*, Mastertool will generate a file containing the settings.

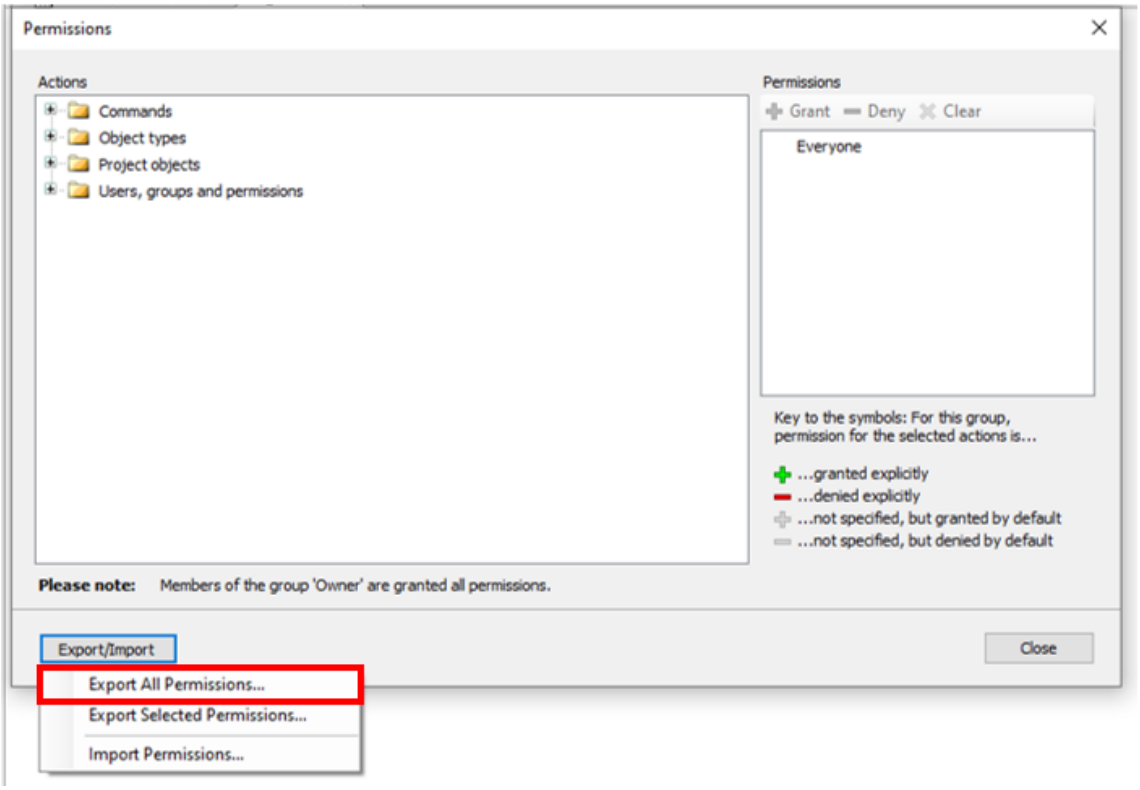

Figure 70: Exporting user permissions

To import these permissions into the project, go to the same menu, but click on *Import Permissions*.

6.11.2. Online Variables Export

It is possible to export the values of the online variables from the *Online* > *Export online variables* menu. This command will create a file (saved in the same directory as the project) containing all the values of the online variables.

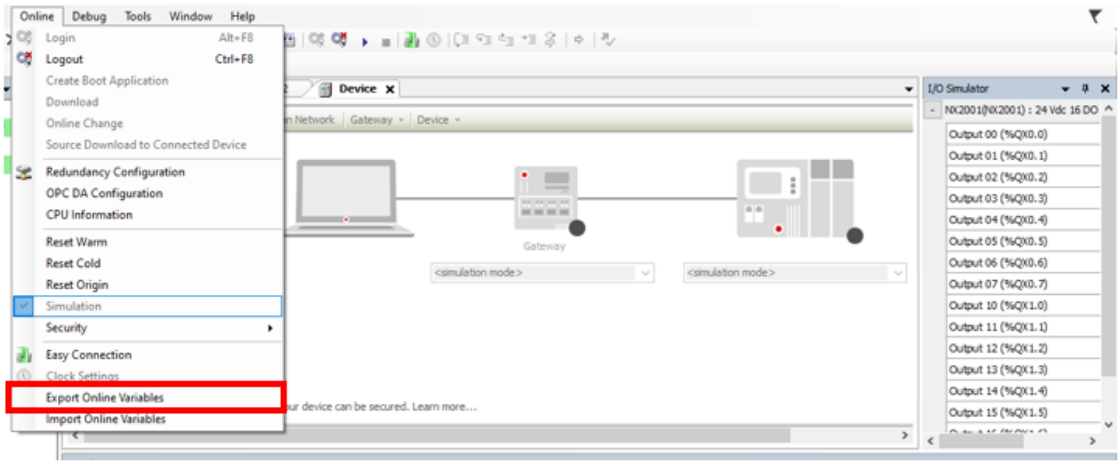

Figure 71: Exporting online variables

6.11.3. Configuration Data Export

It is possible to export certain configurations from the device's web page. By clicking the *Export* button available on the page, this will generate a *.txt* file containing the settings for that functionality. This can be stored as a backup and, if necessary, imported back into the PLC.

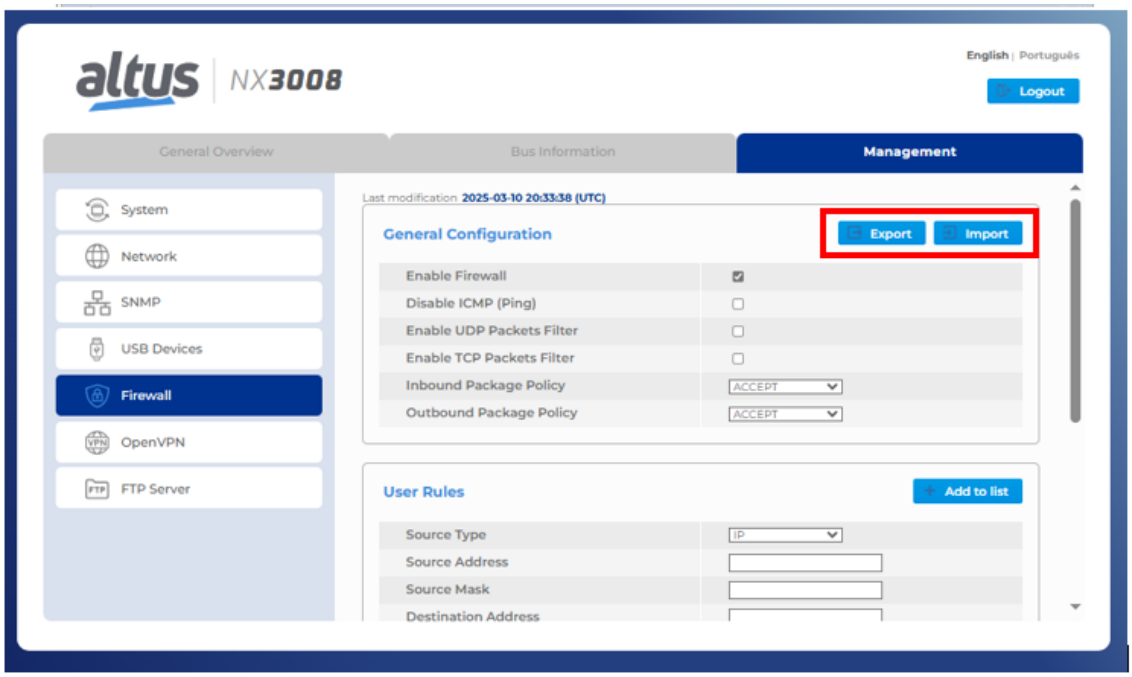

Figure 72: Exporting User Permissions

It is also possible to perform the same procedure for *USB Devices*.

6.11.4. Export Firmware

Information regarding the device firmware backup is contained in Chapter 5.10.

6.12. Possible Sources of Risks

Connecting the device to the internet without proper Firewall and VPN configuration poses significant risks. The USB Host port present in the controllers of some series allows you to expand the controller's capabilities using various types of USB dongles, including SIM chip modems and WiFi adapters. For devices in bridge mode or routers with enabled external access (port forwarding), once connected to the internet, anyone who knows the modem's IP address can remotely access the controller. Therefore, for security reasons, it is extremely important and recommended to configure User Rights on the controller to restrict online operations of MasterTool IEC XE with login and password. Through the management Web page, you can even stop the controller, which is a risk not only to cybersecurity but also to the physical safety of employees and assets.

6.13. Reserved TCP/UDP Ports

The following TCP/UDP ports of both local and remote Ethernet interfaces are typically used by CPU services (subject to availability as per the PLC manual) and are therefore reserved and should not be used by the user.

Service	TCP	UDP
System Web Page	80	-
SNTP	-	123
SNMP	-	161
MODBUS TCP	502*	-
Mastertool	1217*	1740:1743
SQL Server	1433	-
MQTT	1883* / 8883*	-

Service	TCP	UDP
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
IEC 61850	102*	-
DNP3	20000* / 20005*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964
Portainer Docker	9000	-
SysLog	-	514
LibHART	1234	-

Table 4: Reserved TCP/UDP ports

* Default port, but user changeable.

7. Compliance with IEC 62443-4-2

The IEC 62443-4-2 standard defines cybersecurity requirements for components in industrial control and automation systems. It covers all system components, including software applications, network devices, embedded devices, and host servers.

The components are compiled in four groups, with progressive countermeasures that seeks protecting the application against different levels of attacks. The first group, named SL-1, seeks protecting the application against accidental misuse, while the fourth level, SL-4, seeks protecting the infrastructure against directed and sophisticated attacks.

Throughout this document, only the requirements applicable to embedded devices (CR and EDR), such as PLCs and other Altus products, are described.

However, the table below lists all the standard's requirements, including those that do not apply or are not met. The requirements that are met include a reference to the section where they can be found.

Component Requirement	Security Level	Chapter
FR 1 – Identification and authentication control (IAC)		
CR 1.1 Human user identification and authentication	1	5.1.1, 6.1
RE (1) Unique identification and authentication	2	5.1.1, 6.1
RE (2) Multifactor authentication for all interfaces	3	
CR 1.2 - Software process and device identification and authentication	2	5.2
RE (1) Unique identification and authentication	3	5.2
CR 1.3 - Account management	1	5.1.1, 6.1
CR 1.4 - Identifier management	1	5.1.1, 6.1, 6.1.3
CR 1.5 - Authenticator management	1	5.1.1, 6.1, 6.1.3
RE (1) Hardware security for authenticators	3	
NDR 1.6 - Wireless access management	1	N/A
RE (1) Unique identification and authentication	2	N/A
CR 1.7 - Strength of password-based authentication	1	5.1.1, 6.1
RE (1) Password generation and lifetime restrictions for human users	3	
RE (2) Password lifetime restrictions for all users (human, software process, or device)	4	
CR 1.8 - Public key infrastructure certificates	2	5.4
CR 1.9 - Strength of public key-based authentication	2	
RE (1) Hardware security for public key-based authentication	3	
CR 1.10 - Authenticator feedback	1	5.1.1
CR 1.11 - Unsuccessful login attempts	1	5.1.1
CR 1.12 - System use notification	1	
NDR 1.13 - Access via untrusted networks	1	N/A
RE (1) Explicit access request approval	3	N/A
CR 1.14 - Strength of symmetric key-based authentication	2	
RE (1) Hardware security for symmetric key-based authentication	3	
FR 2 - Use control (UC)		
CR 2.1 - Authorization enforcement	1	5.1.1, 5.1.2, 6.9.1, 6.1, 6.9.1
RE (1) Authorization enforcement for all users (humans, software processes and devices)	2	5.1.1, 5.1.2, 6.9.1, 6.1, 6.9.1
RE (2) Permission mapping to roles	2	5.1.1, 5.1.2, 6.9.1, 6.1, 6.9.1
RE (3) Supervisor override	3	
RE (4) Dual approval	4	
CR 2.2 - Wireless use control	1	5.1.2, 6.9.1,6.9.1

Component Requirement	Security Level	Chapter
CR 2.3 - Use control for portable and mobile devices	-	N/A
SAR 2.4 - Mobile code	1	N/A
RE (1) Mobile code authenticity check	2	N/A
EDR 2.4 - Mobile code	1	N/A
RE (1) Mobile code authenticity check	2	N/A
HDR 2.4 - Mobile code	1	N/A
RE (1) Mobile code authenticity check	2	N/A
NDR 2.4 - Mobile code	1	N/A
RE (1) Mobile code authenticity check	2	N/A
CR 2.5 - Session lock	1	5.1.1
CR 2.6 - Remote session termination	1	5.1
CR 2.7 - Concurrent session control	3	
CR 2.8 - Auditable events	1	5.7
CR 2.9 - Audit storage capacity	1	5.7, 6.3, 6.4
RE (1) Warn when audit record storage capacity threshold reached	3	5.7, 6.3, 6.4
CR 2.10 - Response to audit processing failures	1	5.7
CR 2.11 - Timestamps	1	5.7
RE (1) Time synchronization	2	5.7
RE (2) Protection of time source integrity	4	5.7
CR 2.12 - Non-repudiation	1	5.7
RE (1) Non-repudiation for all users	4	
EDR 2.13 - Use of physical diagnostic and test interfaces	2	
RE (1) Active monitoring	3	
HDR 2.13 - Use of physical diagnostic and test interfaces	2	N/A
RE (1) Active monitoring	3	N/A
NDR 2.13 - Use of physical diagnostic and test interfaces	2	N/A
RE (1) Active monitoring	3	N/A
FR 3 - System integrity (SI)		
CR 3.1 - Communication integrity	1	5.3, 6.9, 6.9.2
RE (1) Communication authentication	2	
SAR 3.2 - Protection from malicious code	1	N/A
EDR 3.2 - Protection from malicious code	1	5.12
HDR 3.2 - Protection from malicious code	1	N/A
RE (1) Report version of code protection	2	N/A
NDR 3.2 - Protection from malicious code	1	N/A
CR 3.3 - Security functionality verification	1	5.1.1, 5.7
RE (1) Security functionality verification during normal operation	4	5.7
CR 3.4 - Software and information integrity	1	5.7
RE (1) Authenticity of software and information	2	
RE (2) Automated notification of integrity violations	3	
CR 3.5 - Input validation	1	5.7
CR 3.6 - Deterministic output	1	5.8
CR 3.7 - Error handling	1	5.9
CR 3.8 - Session integrity	2	
CR 3.9 - Protection of audit information	2	
RE (1) Audit records on write-once media	4	6.3
EDR 3.10 - Support for updates	1	6.5.1
RE (1) Update authenticity and integrity	2	6.5.1
HDR 3.10 - Support for updates	1	N/A
RE (1) Update authenticity and integrity	2	N/A
NDR 3.10 - Support for updates	1	N/A
RE (1) Update authenticity and integrity	2	N/A
EDR 3.11 - Physical tamper resistance and detection	2	
RE (1) Notification of a tampering attempt	3	
HDR 3.11 - Physical tamper resistance and detection	2	N/A
RE (1) Notification of a tampering attempt	3	N/A
NDR 3.11 - Physical tamper resistance and detection	2	N/A
RE (1) Notification of a tampering attempt	3	N/A

Component Requirement	Security Level	Chapter
EDR 3.12 - Provisioning product supplier roots of trust	2	
HDR 3.12 - Provisioning product supplier roots of trust	2	N/A
NDR 3.12 - Provisioning product supplier roots of trust	2	N/A
EDR 3.13 - Provisioning asset owner roots of trust	2	
HDR 3.13 - Provisioning asset owner roots of trust	2	N/A
NDR 3.13 - Provisioning asset owner roots of trust	2	N/A
EDR 3.14 - Integrity of the boot process	1	
RE (1) Authenticity of the boot process	2	
HDR 3.14 - Integrity of the boot process	1	N/A
RE (1) Authenticity of the boot process	2	N/A
NDR 3.14 - Integrity of the boot process	1	N/A
RE (1) Authenticity of the boot process	2	N/A
FR 4 – Data confidentiality (DC)		
CR 4.1 - Information confidentiality	1	5.4, 5.5, 5.6
CR 4.2 - Information persistence	2	
RE (1) Erase of shared memory resources	3	
RE (2) Erase verification	3	
CR 4.3 - Use of cryptography	1	5.4, 5.6 5.13
FR5 - Restricted data flow (RDF)		
CR 5.1 - Network segmentation	1	6.8, 6.7
NDR 5.2 - Zone boundary protection	1	N/A
RE (1) Deny all, permit by exception	2	N/A
RE (2) Island mode	3	N/A
RE (3) Fail close	3	N/A
NDR 5.3 - General-purpose person-to-person communication restrictions	1	N/A
FR6 - Timely response to events (TRE)		
CR 6.1 - Audit log accessibility	1	5.7
RE (1) Programmatic access to audit logs	3	
CR 6.2 - Continuous monitoring	2	
FR7 - Resources Availability (RA)		
CR 7.1 - Denial of service protection	1	6.2
RE(1) Manage communication load from component	2	
CR 7.2 - Resource management	1	6.10
CR 7.3 - Control system backup	1	5.10
RE (1) Backup integrity verification	2	
CR 7.4 - Control system recovery and reconstitution	1	6.11
CR 7.5 - Emergency power	-	
CR 7.6 - Network and security configuration settings	1	6.5
RE (1) Machine-readable reporting of current security settings	3	
CR 7.7 - Least functionality	1	6.10
CR 7.8 - Control system component inventory	2	5.11

Table 5: IEC 62443-4-2 Standard Compliance Table

7.1. Security Level 1

The security level 1 (SL-1) defines basic components to prevent the unauthorized disclosure of information via eavesdropping or casual exposure.

One of the requirements of this level that is not met is CR 1.12, which specifies that the system must display a notification before user authentication and, in addition, allow the administrator to configure the message. Currently, the products display a login confirmation screen, but it is not customizable as required by the standard.

Another unmet requirement is EDR 3.14, which requires the device to perform an integrity check of the firmware, software, and configuration files before startup. This tool is currently under development but has not yet been implemented.

The compliance rate for this security level is 94% (34 out of 36 requirements).

7.2. Security Level 2

The security level 2 (SL-2) stack the SL-1 countermeasures and define additional components to prevent the unauthorized disclosure of information to an entity actively searching for it using simple means with low resources, generic skills and low motivation.

To fully meet the level 2 security requirements of the standard, 22 requirements are proposed, of which 7 are met, resulting in a compliance rate of 32%.

The unmet requirements primarily concern functionalities related to public key certifications and the integrity of internal component files, such as firmware and configurations.

In general, to fully comply with this security level, more advanced PKI policies should be implemented, ensuring that components perform more robust checks of digital certificates. This includes the use of encryption to validate the authenticity and integrity of certificates, enhancing protection against attacks.

The unmet requirements are:

1) CR 1.9
2) CR 1.14
3) EDR 2.13
4) CR 3.1 RE(1)
5) CR 3.4 RE(1)
6) CR 3.8
7) CR 3.9
8) EDR 3.11
9) EDR 3.12
10) EDR 3.13
11) EDR 3.14 RE(1)
12) CR 4.2
13) CR 6.2
14) CR 7.1 RE(1)
15) CR 7.3 RE(1)

7.3. Security Level 3

The security level 3 (SL-3) stack the SL-1 and SL-2 countermeasures and define additional components to prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with moderate resources, IACS specific skills and moderate motivation.

To meet the level 3 security requirements of the IEC62443-4-2 standard, the component must satisfy 16 requirements. Since these are more stringent requirements, only 2 of them are met, resulting in a compliance rate of 12.5%.

7.4. Security Level 4

The security level 4 (SL-4) stack the SL-1, SL-2 and SL-3 countermeasures and define additional components to prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with extended resources, IACS specific skills and high motivation.

Of the 7 requirements presented by the standard for this security level, 3 are currently met, which is 43%.

8. Compliance with the Operation Procedures Manual - ONS

This chapter demonstrates the relationship between Altus products and the requirements of the "Operation Procedures Manual: Minimum Cybersecurity Controls for the Regulated Cyber Environment", issued by Operador Nacional do Sistema, the entity responsible for coordinating and controlling the operation of electricity generation and transmission facilities within Brazil. This document is found in Module 5 - Submodule 5.13 of the Operation Procedures Manual.

The objective of this manual is to establish the minimum cybersecurity controls to be implemented by agents and ONS in the Regulated Cyber Environment (ARCiber). The table below presents the guidelines proposed by ONS, along with explanations of their application in Altus products.

REQUIREMENT	COMMENT
4.1 Technological Architecture for the environment	
4.1.1 Networks must be segregated into security zones according to their function. The agent must define an architecture that segments the networks at a minimum into: a) Supervision Zone b) Operational DMZ Zone c) Corporate Zone	Considering the separation into Zones and Conduits from ISA99 (IEC 62443) and the Purdue Reference Model, Altus devices will be installed in the Supervision Zone. Altus devices have the necessary segregation features indicated in the specification.
4.1.2 The ARCiber should not be directly accessible through the internet, even if protected by one or more firewalls, nor should its assets.	The entire network can be accessed through a single VPN without the need for specific component characteristics. It is on Altus' roadmap to implement VPN for its products next year. More information regarding the use of VPN in Altus products can be found in section 6.8
4.1.3 Access to ARCiber from networks external to the organization (such as, for example, the internet) should only be allowed for the performance of authorized activities. This access must be carried out through a Virtual Private Network (VPN), or similar technology, via a gateway or service that provides security controls. a) They should not be visible or accessible from the internet. b) They should not be able to connect to the internet.	
4.1.4 Antimalware solutions must be implemented in ARCiber and kept up to date. a) Application whitelisting solutions can be implemented as an alternative or complement to antimalware solutions.	Not applicable to the products.
4.2 Information Security Governance	
4.2.1 At least one manager and one deputy must be appointed, responsible for the cybersecurity of ARCiber and acting as external points of contact.	These requirements relate to processes and do not necessarily have a connection with product functionalities.
4.2.2 A policy must be established to define roles and responsibilities regarding the cybersecurity of ARCiber.	

REQUIREMENT	COMMENT
4.3 Asset Inventory	
4.3.1 All assets, both software and hardware, connected to ARCiber must be inventoried at least every 24 months and must include at a minimum: a) Device type; b) Equipment manufacturer; c) Function; d) IP address or MAC address; e) Application protocol and/or service port; f) Firmware and/or operating system version;	All information is accessible through the equipment's diagnostics.
4.3.2 The asset inventory must be stored securely, with well-defined storage policies, and access should be restricted to individuals who need the information to perform their duties.	
4.3.3 Secure configuration standards (hardening) must be created according to the agent's security policy for operating systems, firmware, databases, and other software versions present in ARCiber:	As mentioned, our product is designed to not open unused ports, interfaces, and protocols in the system. Resources that can be used but are not, are disabled. The file system of the Operating System and RTS (CoDeSys) are also not accessible to the user. Section 6.10 presents more information on the subject.
4.4 Vulnerability Management	
4.4.1 The organization's security policy should include the management of security patch packages for all technologies connected to ARCiber, covering at least: a) Implementation schedule for patches; b) Mapping of inventoried assets to updates provided by manufacturers.	Altus products provide a firmware update mechanism (containing all necessary files for the system) that allows for the correction of vulnerabilities found in the equipment. The updates are available on the Altus website along with the product revision history (more information in section 6.5.1). A security page on the Altus website is planned, where mapped vulnerabilities will be published.
4.4.2 New assets should only be connected to ARCiber after all available security patch packages have been applied. a) If the new equipment is replacing an existing equipment that has failed, the application of security patch packages may be postponed, but with a predefined deadline.	The possibility of updating the equipment allows new components to be reviewed and updated on the bench before installation.
4.5 Access Management	
4.5.1.1 Access credentials must be individual and approved by the competent authority. In cases where it is not possible to implement individual credentials, the following actions should be taken: a) Generate and maintain a list of individuals authorized to use shared accounts. b) Implement the controls outlined in 4.5.1.6.	Information regarding user management, login, and password is contained in section 5.1.

REQUIREMENT	COMMENT
4.5.1.2 Password policy should include: minimum length, complexity, the requirement to be different from the manufacturer's default password, actions to be taken if the maximum number of failed login attempts is reached, and criteria for change management (timeline, incident occurrence, etc.). a) The password policy can be implemented through technological controls or procedures. If the password characteristics outlined in the policy cannot be implemented on certain assets due to technological limitations, the maximum level supported by the asset should be implemented.	Information regarding user management, login, and password can be found in section 5.1 and 6.1.
4.5.1.3 In constructing access profiles, the principle of least privilege must be followed (only the minimum necessary access should be granted)..	
4.5.1.4 The maximum deadline for canceling/removing credentials of terminated users and credentials that have not been used after a certain period.	
4.5.1.5 Privileged access credentials must be subject to specific controls, including: a) Appropriate approval level, with periodic review by the ARCiber manager; b) Exclusive use during the execution of administrative tasks; c) Monitoring through audit trails; d) Use of multi-factor authentication, such as OTP (one-time password) tokens or biometric recognition.	Item D is not satisfied due to the absence of multi-factor authentication in Altus products.
4.5.1.6 The special characteristics of embedded (local) default access credentials in operating systems and software must be considered in the access and identity management policy: a) Access to the password of embedded accounts should be restricted to a limited number of people; b) Each asset with an embedded credential should have a distinct password. The same password should not be assigned to more than one asset.	There are no restrictions regarding the requirements for access management. The only functionality required in Altus products that has not yet been implemented is MFA (multi-factor authentication), but this is mapped to be implemented until the first semester of 2027. Users will have the option to enable MFA. If they so choose, they will need to enter a password and a token (a temporary password, like in internet banking). The token will be generated every minute and displayed on the equipment's screen menu.
4.6 Incident Monitoring and Response	
4.6.1 ARCiber assets must be configured to generate appropriate security logs to support investigations and the reconstruction of potential security incidents. These logs should be stored for a period defined in the organization's cybersecurity policies.	The components generate logs for this purpose. More details about log generation can be found in section 5.7.

REQUIREMENT	COMMENT
4.6.2 Security devices such as Firewalls, IDS/IPS, Anti-malware, and authentication subsystems must be configured to generate alerts if they detect suspicious activities. a) The rules for generating alerts should be reviewed periodically; b) All alerts must be immediately reported to the responsible team defined in the agent's security policy; c) Generated alerts must be analyzed and responded to within the timeframe defined by the agent's security policy.	There are no restrictions on the use of this type of functionality in Altus products.
4.6.3 Mechanisms must be established for the timely identification and response to cyber incidents.	
4.6.4 A cyber incident response plan must be implemented, covering at least the following requirements: a) Identification of applicable cyber risk scenarios for ARCiber and treatment strategies for each scenario; b) Impact classification; c) Involved teams, with their respective roles and responsibilities before, during, and after the crisis; d) Criteria for activating the cyber incident response plan.	
4.6.5 Cyber incident response plan activation tests must be conducted periodically, at intervals defined in the organization's cybersecurity policy, covering at least the activation lists (call tree) and review of the described procedures. The exercises should generate documents outlining lessons learned and the corresponding corrective and improvement actions.	
4.6.6 Cyber incidents affecting ARCiber assets must be reported to the ONS.	
5.1 Exception Handling	
5.1.1 Cases where requirements cannot be implemented must be treated with an exception. Each exception generated must be created: a) Documented in detail, including the date it was identified, the reason it needs to be treated as an exception, the items from this RO that will not be met, and the expected impacts; b) Approved by the manager responsible for the cybersecurity of ARCiber;	Among the topics described in this manual, the only requirement that cannot be met is item 4.1.5.1-d), where multiple factors of authentication are required.

REQUIREMENT	COMMENT
5.2 Adoption of Complementary Controls	
It is up to each organization to adopt controls: a) Complementary controls on assets that integrate ARCiber, according to their own policies, guidelines, and risk assessments. b) Cybersecurity controls on assets that do not integrate ARCiber, according to their own policies, guidelines, and risk assessments.	There are no restrictions regarding this type of control in Altus products.

Table 6: Requirements from the Operation Procedures Manual - ONS

9. CODESYS Components and Products

The CVE (*Common Vulnerabilities and Exposures*) is an important tool for tracking vulnerabilities in products used in systems around the world. There is a common database with all entries for any product with a known vulnerability. This database can be accessed at the following links:

https://www.cvedetails.com/

https://cve.mitre.org/

Currently, Altus does not have its own database to record known vulnerabilities in its products. In this case, known vulnerabilities can be found in common databases through the use of keywords such as Altus, Hadron Xtorm, HX3040, among others. Many Altus products use CODESYS components and products in their development, and these parts also have vulnerabilities. More details about these vulnerabilities, security procedures, and security advisories can be found at:

https://www.codesys.com/ecosystem/security/latest-codesys-security-advisories/

However, Altus products do not use all CODESYS products and components. Therefore, to determine if a CVE related to CODESYS represents a vulnerability for Altus products, it is necessary to know which CODESYS components are integrated into Altus products.

Table 7 shows the components present in the implementation of each Altus product. All the components used are part of CODESYS V3, so only vulnerabilities related to this version should be considered.

CODESYS Components	MasterTool	Nexto	Xpress	HX3040	NL717
CODESYS OPC DA Server SL	✓				
CODESYS Control for Linux ARM SL		✓ (Just NX3008)	✓		✓
CODESYS Control for Linux SL		✓ (Except NX3008)		✓	
CODESYS Scripting	✓				
CODESYS Visualization	✓				
CODESYS WebVisu	✓	✓ (Just NX3005 e NX 3008)	✓ (Just XP340)		
CODESYS Git	✓				
CODESYS PROFINET	✓	✓ (Except NX3003 e NX3004)	✓	✓	
CODESYS EtherNetIP	✓	✓ (Except NX3003 e NX3004)	✓		
Web Server (part of CODESYS runtime system)		✓	✓	✓	✓
CODESYS OPC UA Server		✓	✓	✓	✓
CODESYS SOFTMOTION CNC+ROBOTICS			✓ (Just XP351)		
CODESYS SOFTMOTION			✓ (Just XP350)		
Communication via Standard Ethernet		✓	✓	✓	✓
Package Manager	✓				
Alarm Configuration	✓	✓ (Just NX3005 e NX3008)	✓ (Just XP340)		
CODESYS Runtime Toolkit		✓	✓	✓	✓
CODESYS Development System or CODESYS Development System V3	✓				
CODESYS Control Runtime System Toolkit		✓	✓	✓	✓
CODESYS V3 Simulation Runtime (part of the CODESYS Development System)	✓				
CODESYS Gateway	✓				
Trace Manager	✓				

Table 7: CODESYS components present in Altus products

10. Final Considerations

Security in control systems is of utmost importance in an increasingly interconnected and digitized industrial automation landscape. Security incidents have been on the rise, demanding that integrators and users remain vigilant and proactive in observing and mitigating these risks.

While it's true that cybersecurity can never be guaranteed at 100%, it's essential to understand that the adoption of security measures and proper precautions can significantly elevate the level of protection for a specific application. Awareness of potential threats and the implementation of preventive measures can create a strong barrier against potential hazards.

Therefore, collaboration among suppliers, integrators, operators, and users is crucial in fostering a robust and effective security culture. By investing in training and education, as well as adopting appropriate security technologies, it's possible to mitigate significant risks and ensure the resilience of control systems in industrial environments.

In this ever-evolving environment, it's crucial to recognize that security is a continuous effort. We must stay attuned to the latest trends and developments in cybersecurity, regularly updating and enhancing our security practices and protocols. This way, we can tackle security challenges in an increasingly digital world, safeguarding our operations and ensuring a safer and more reliable industrial automation environment.

11. Appendices

11.1. TLS Certificates and Keys Management

This section covers the generation of security files, certificates, and keys using TLS. The certificates commented on below are signed by CA. This type of certificate considers an entity, called Certificate Authority (CA), to generate the certificates. This entity can be an official authority service or a simple computer. It is only necessary to restrict access to the CA to avoid any security breach since this entity can generate certificates for any device. The image below shows how each device interacts with the files.

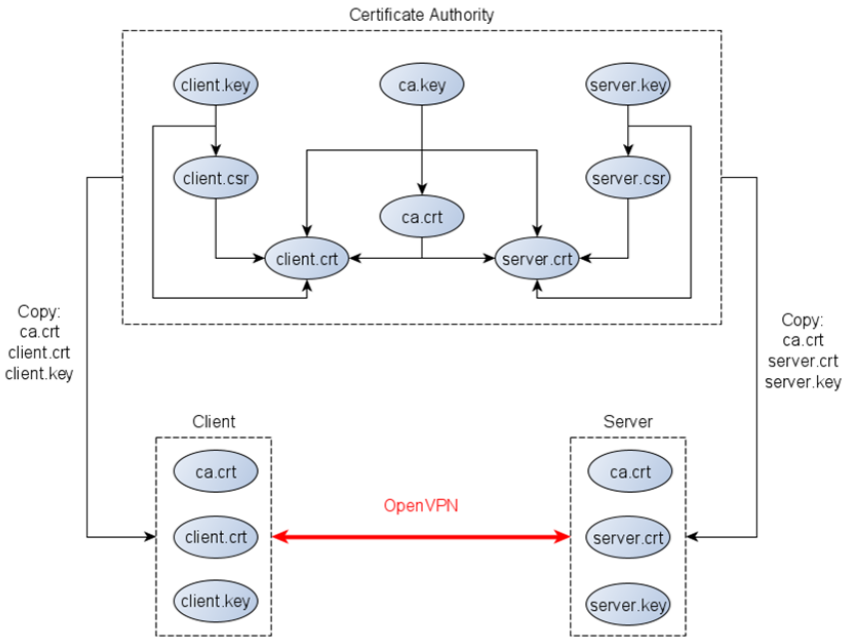

Figure 73: TLS Certificate Generation Flow

First of all, the generated files are private keys. Each device has its key file, created either by the CA entity or the device itself. The most important file is the CA private key *ca.key*, which must not leave the entity. The CA entity generates its certificate based on its private key *ca.crt*. This certificate is a public file used by the devices to validate the VPN connection. Generating certificates from the device first requires a request file (*.csr* or *.req* depending on the tool) based on the device's private key. This document presents two possible tools for generating certificate files: Easy-RSA and OpenSSL.

Make sure you have the date and time set correctly in the CA entity so that the generation of the certificates is based on a current setting.

11.1.1. Certificate Generation with Easy-RSA

The OpenVPN project provides this tool to help with the certificate and keys. Easy-RSA is available for Windows and Linux. See below for step-by-step instructions to generate the files in a Windows configuration:

1- Open a Windows prompt in the Easy-RSA folder and run the following command to enter the tool shell:

```
.\EasyRSA-Start.bat
```

```
C:\Users\igor.franco\Downloads\EasyRSA-3.0.8-win64\EasyRSA-3.0.8>.\EasyRSA-Start.bat

Welcome to the EasyRSA 3 Shell for Windows.
Easy-RSA 3 is available under a GNU GPLv2 license.

Invoke './easyrsa' to call the program. Without commands, help is displayed.

EasyRSA Shell
# _
```

Figure 74: Certificate Generation using Easy-RSA (step 1)

2 - Copy the file *vars.example* and rename it to *vars* in the tools folder.

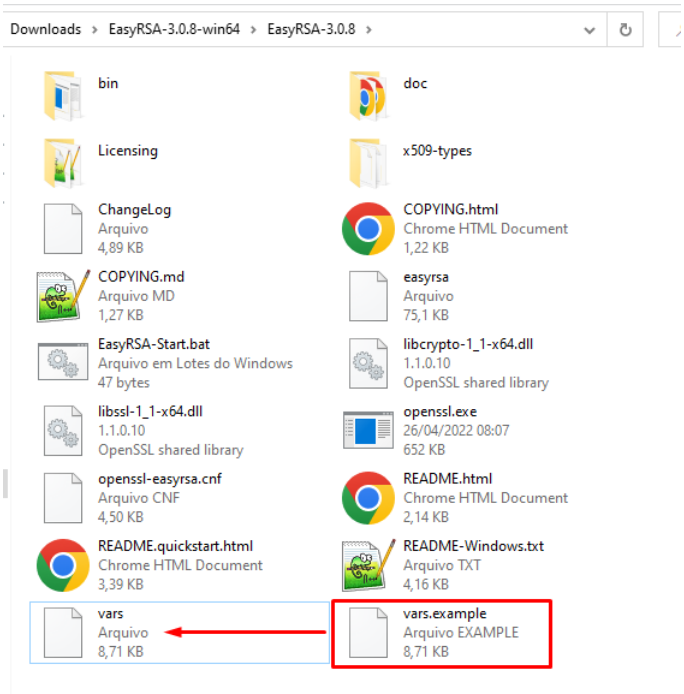

Figure 75: Certificate Generation using Easy-RSA (step 2)

3- Open the file *vars* with a text editor and change the Certification Authority information.

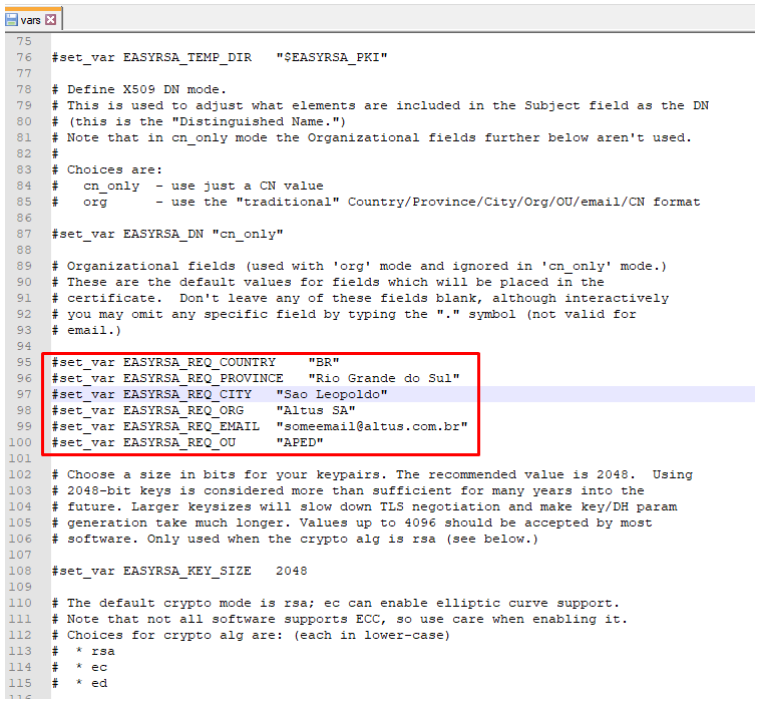

Figure 76: Certificate Generation using Easy-RSA (step 3)

4- Use the following command to prepare the configuration.

```
./easyrsa init-pki
```

Figure 77: Certificate Generation using Easy-RSA (step 4)

5- Then type the following to generate the CA certificate.

```
./easyrsa build-ca nopass
```

Remove the *nopass* argument if you want to set a password for the file. Enter the common name of the CA certificate when prompted (press enter to use the default *Easy-RSA CA* as the common name).

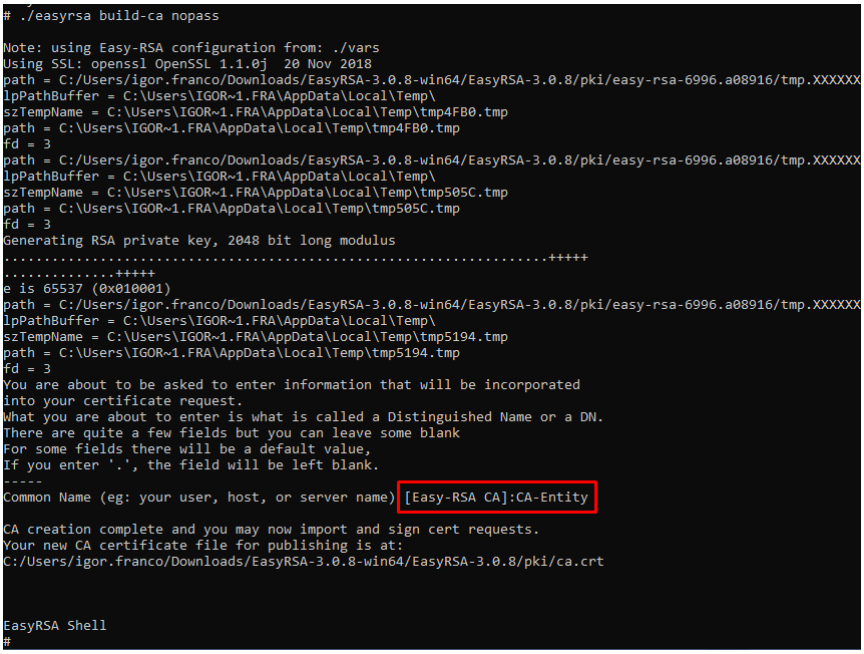

Figure 78: Certificate Generation using Easy-RSA (step 5)

6 - Generate the device key and request files using the following command (change the *DeviceName* with the desired common name):

```
./easyrsa gen-req DeviceName nopass
```

Again, remove the *nopass* argument to use a password for the certificate file. When entering the Common Name as an argument, simply press Enter when prompted (red square).

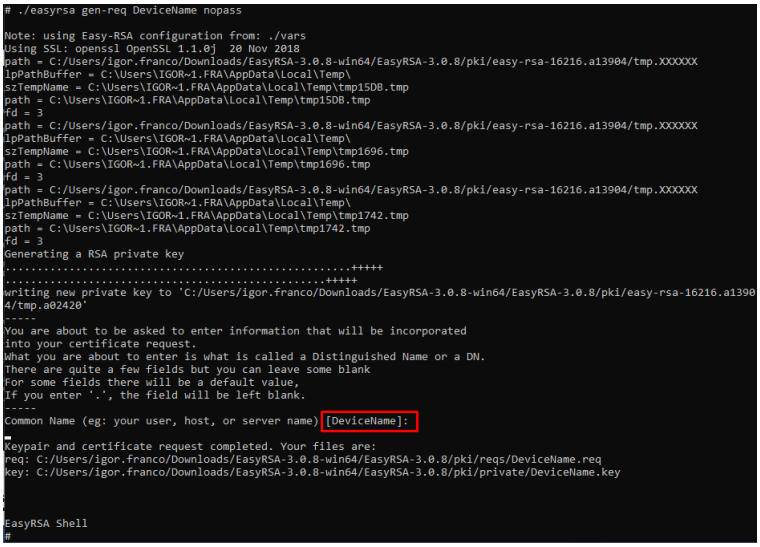

Figure 79: Certificate Generation using Easy-RSA (step 6)

7- Finally, type the following command to generate the device certificate (the *DeviceName* is the desired common name, and the *server* is the type; use *client* if you are generating for a VPN client).

```
./easyrsa sign-req server DeviceName
```

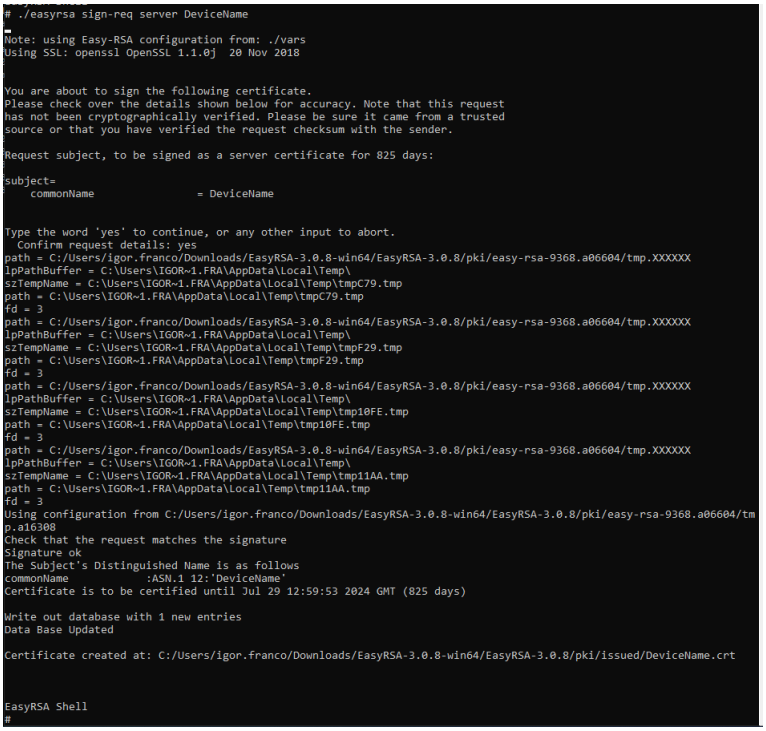

Figure 80: Certificate Generation using Easy-RSA (step 7)

8- Repeat steps 6 and 7 to generate more device certificates.

9- Find the *ca.crt* in the *pki* folder, the device private keys in the *pki/private* path, and the device certificates in the *pki/issued* directory.

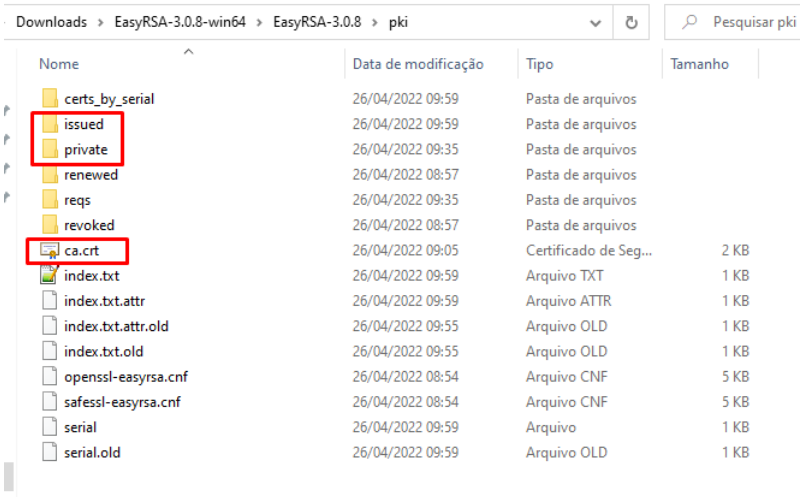

Figure 81: Certificate Generation using Easy-RSA (step 9)

11.1.2. Certificate Generation with OpenSSL

OpenSSL is an open-source package with tools that help generate many files and security features. This package is native to most Linux distributions and is available for Windows. Just remember to set the OpenSSL folder in the PATH (environment variable) to allow you to use the command from anywhere via the prompt. Find below the step-by-step using this feature (all files can have any name as desired, the steps consider only an example):

1- Open a prompt in the certificate folder (where you will create the files).

2- Generate the CA private key with the following command:

```
openssl genrsa -out ca.key 4096
```

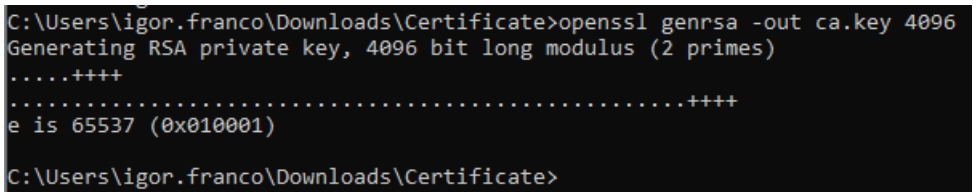

Figure 82: Certificate Generation using OpenSSL (step 2)

3- Then generate the CA certificate based on the private key, using the following command.

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

The parameter *-days* represents the expiration time for the certificate. Set it as desired. In this example, the certificate is valid for one year. Fill in the values requested at the prompt as needed (press enter to use the default, which is enclosed in square brackets []). It is mandatory to define a Common Name for the certificate work.

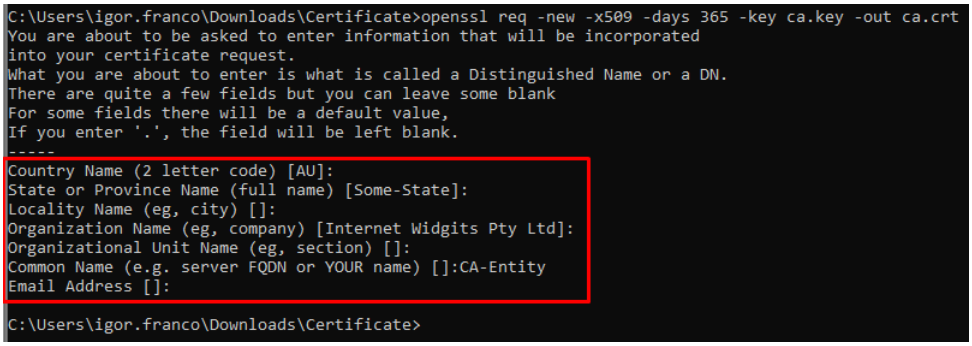

Figure 83: Certificate Generation using OpenSSL (step 3)

4- Now generate the device's private key, similar to step 2, using the following command:

```
openssl genrsa -out DeviceName.key 2048
```

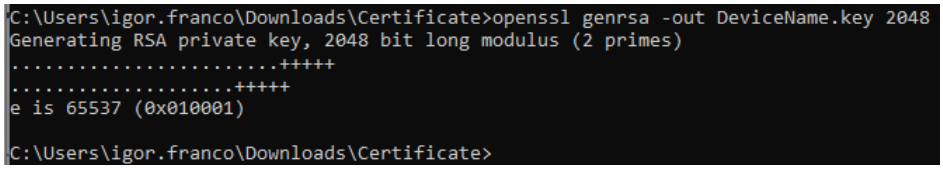

Figure 84: Certificate Generation using OpenSSL (step 4)

5- After that, generate the certificate request file based on the private key using the following command:

```
openssl req -new -key DeviceName.key -out DeviceName.csr
```

Enter the desired information, and remember to use a common name other than CA.

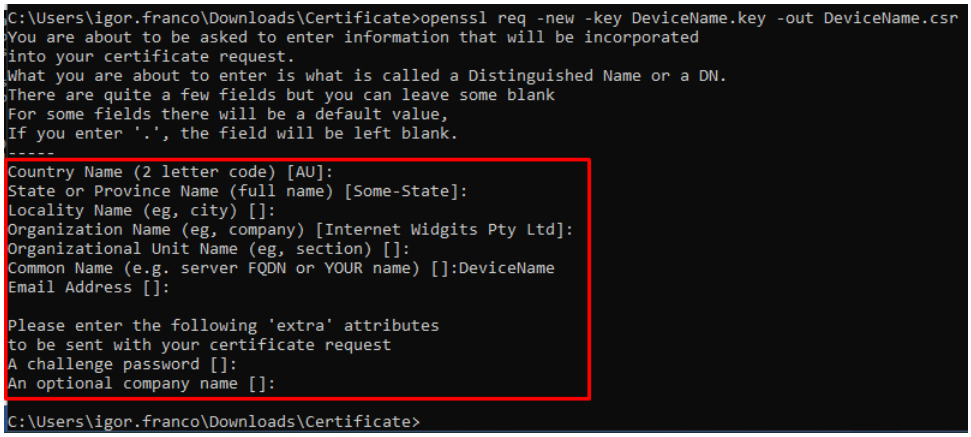

Figure 85: Certificate Generation using OpenSSL (step 5)

6- Finally, generate the device certificate using the CA private key, the CA certificate, and the device certificate request file using the following command:

```
openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out
```

Set the expiration date as desired with the parameter *-days* and the serial number of the certificate with the argument *-set_serial*.

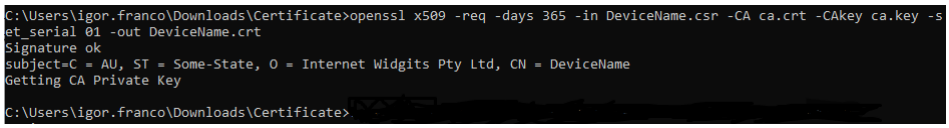

Figure 86: Certificate Generation using OpenSSL (step 6)

7- Repeat steps 4 to 6 for any new device.

8- (Optional) OpenSSL provides a tool to verify that the device certificate works with CA:

Use the following command:

```
openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt
```

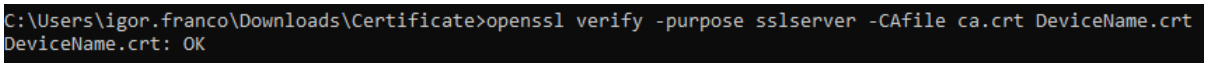

Figure 87: Certificate Generation using OpenSSL (step 8)

11.1.3. TA Key Generation by OpenVPN

The OpenVPN project provides a tool for generating a TLS key, commonly called ta.key. This key is an extra layer of protection on OpenVPN's UDP/TCP communication ports, so the use of this key can be interpreted as an HMAC Firewall for VPN communication, requiring the existence of the parameter on both sides of the communication for it to be established.

The key generation in Windows can be done with the following command:

```
openvpn --genkey secret ta.key
```

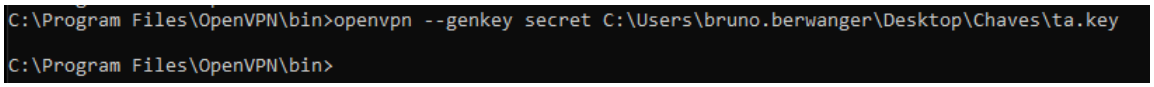

Figure 88: TA Key Generation in Windows example

To execute the command, we used the executable installed with the OpenVPN package. The directory used in the image above is an example and is optional. You can use only the desired file name too.

The command can be used to generate the key from Linux, but there is a minor change in the command compared to Windows. To generate the key on Linux, use the following command in the terminal:

```
openvpn --genkey --secret ta.key
```

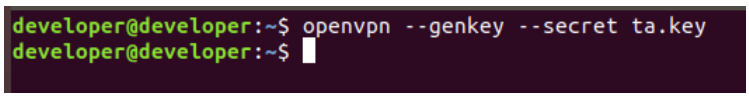

Figure 89: TA Key Generation in Linux example

This parameter is not mandatory for VPN communication, but if the server uses it, all its clients must also use it, and the key for the server and the clients must be the same.