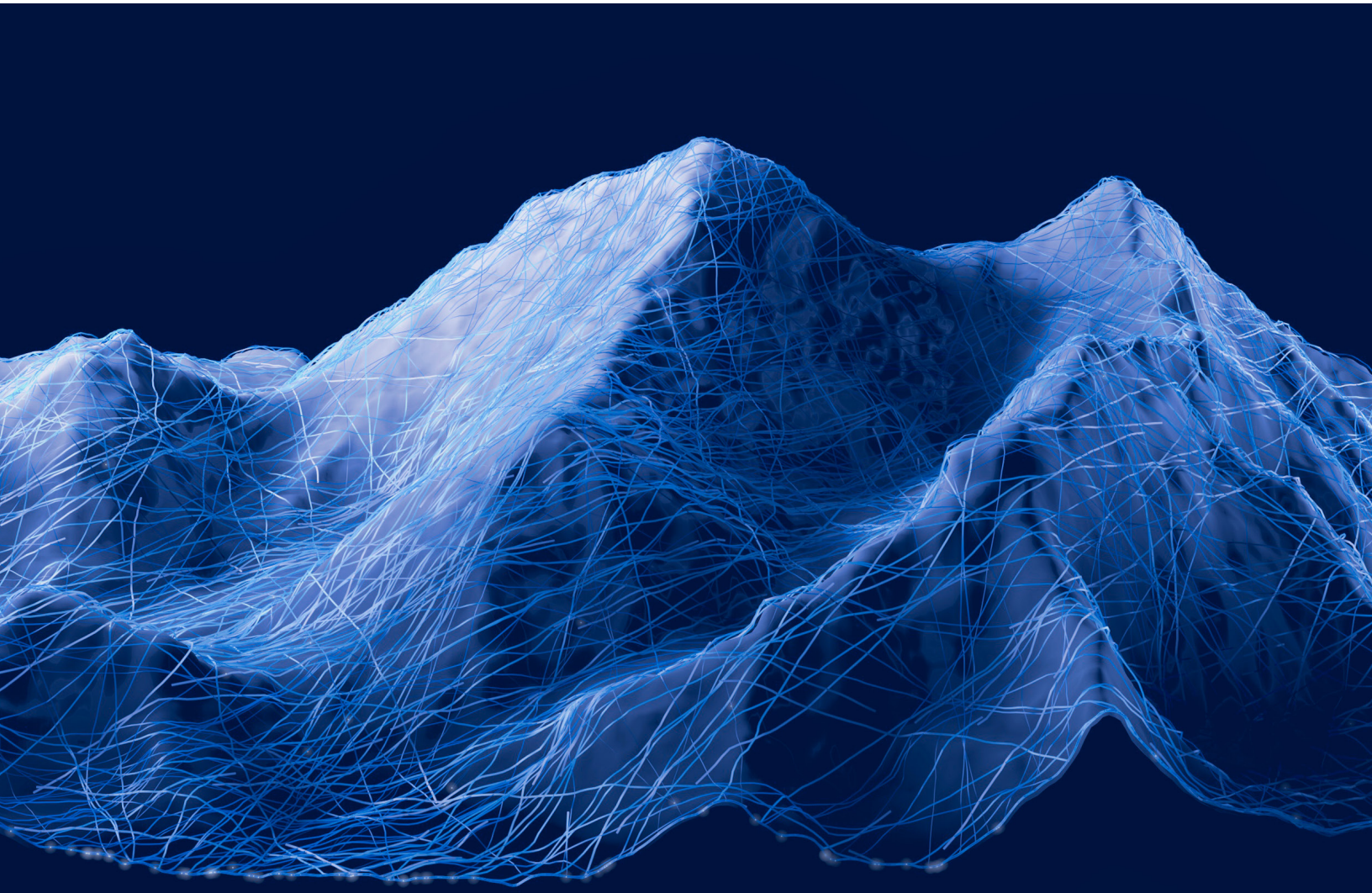
VEDERE LABS



OT:ICEFALL

The legacy of “insecure by design” and its implications for certifications and risk management

Table of Contents

1. Executive Summary

- Vedere Labs has identified a set of 56 vulnerabilities affecting devices from 10 operational technology (OT) vendors that we are collectively calling OT:ICEFALL. The vulnerabilities are divided into four main categories: insecure engineering protocols, weak cryptography or broken authentication schemes, insecure firmware updates and remote code execution via native functionality.
- Exploiting these vulnerabilities, attackers with network access to a target device could remotely execute code, change the logic, files or firmware of OT devices, bypass authentication, compromise credentials, cause denials of service or have a variety of operational impacts. Vulnerabilities allowing for firmware manipulation or remote code execution represent 35% of the total.
- The products affected by OT:ICEFALL are known to be prevalent in industries that are the backbone of critical infrastructures such as oil and gas, chemical, nuclear, power generation and distribution, manufacturing, water treatment and distribution, mining and building automation. Many of these products are sold as 'secure by design' or have been certified with OT security standards.
- Abusing insecure-by-design native capabilities of OT equipment is the preferred modus operandi of real-world ICS attackers (e.g., **Industroyer2**, **TRITON**, and **INCONTROLLER**). This shows the need for robust OT-aware network monitoring and deep packet inspection capabilities.
- This study identifies a shift in the community toward recognizing 'insecure by design' vulnerabilities. Only a few years back, well-known vulnerabilities like some that can be found in OT:ICEFALL would not get assigned a CVE ID because there was the assumption that everyone knew OT protocols were insecure. On the contrary, we believe a CVE is a community-recognized marker that aids in vulnerability visibility and actionability by helping push vendors to fix issues and asset owners to **assess risks** and apply patches.
- In addition to network monitoring, mitigations for OT:ICEFALL include isolating OT/ICS networks from corporate networks and the internet, limiting network connections to only specifically allowed engineering workstations and focusing on consequence reduction where possible.

A QUICK OVERVIEW OF CYBERSECURITY IN OPERATIONAL TECHNOLOGY

Operational Technology. According to **Gartner**, "operational technology is hardware and software that detects or causes a change, through the direct monitoring and/or control of industrial equipment, assets, processes and events." This includes, for instance, industrial control system (ICS) and building automation system (BAS). This type of technology is characterized by (i) long life cycles, since industrial hardware can last for decades; (ii) resource constraints, since these systems rely on embedded hardware that is purposefully limited in memory and processing power; (iii) stringent safety and timing requirements, since they cause changes on the physical world, and their failure can be catastrophic; and (iv) the use of specialized networking communications.

OT architecture and devices. OT systems are usually categorized into either Supervisory Control and Data Acquisition (SCADA), an architecture typically used for geographically dispersed systems that rely on high-level process supervisory management, or distributed control systems (DCS), highly integrated solutions that contain many control loops with autonomous controllers. Both architectures include a variety of specialized embedded devices, such as programmable logic controllers (PLCs) and remote terminal units (RTUs), that are directly connected to sensors/actuators and implement control loops, as well as more traditional computers with specialized software that act as human machine interfaces (HMI) to allow operators to graphically see and effect changes on the industrial process, data historians that store time-stamped data and events collected from the process, or engineering workstations that allow operators to program the field devices. It is traditional to discuss the equipment described above as part

of the **Purdue Enterprise Reference Architecture**, which is a reference model for the different levels of enterprise integration. Figure 1 depicts the types of systems in each level of the Purdue model: level 0 contains sensors and actuators connected to the physical world; level 1 contains the devices that control the physical process (such as PLCs and RTUs); level 2 contains the SCADA and HMI systems used by humans to monitor and control the process; level 3 contains the Historian and other operation management software; level 4 contains business-related devices and software, such as Enterprise Resource Planning (ERP) or financial systems; finally, level 5 contains the remaining enterprise network, with other IT servers. Levels 0-2 are replicated in several cell/area network zones within an organization's site (such as production lines), level 3 tends to be present in several sites within an organization (such as manufacturing plants), while the upper levels are typically part of an organization's global network.

Currently, cybersecurity is fundamental for the safe operation of industrial processes due to an emerging threat landscape, including new malicious actors and targets, and the increased interconnection between OT assets and enterprise networks.

Threat landscape. Threat actors targeting industrial processes include internal and external attackers, such as disgruntled employees, hacktivists, cyber criminals, and state-sponsored actors. These threat actors have different objectives, such as financial gains via espionage or ransomware, industrial sabotage, or damage to property or life. Threats and malicious actors in the OT space have evolved significantly, showing more disruptive and destructive intent over the past decade. OT-targeted attacks nowadays commonly use specific OT protocols and native features to carry out their activities. High-profile malware using these techniques include **Industroyer**, which was used to cause the Ukraine power outage in 2016 and the newer **Industroyer2** variant found in Ukraine in 2022; **TRITON**, which targeted industrial safety systems in the Middle East in 2017; and **INCONTROLLER**, an APT toolkit targeting several OT devices, such as OPC UA servers and PLCs from Omron and Schneider Electric.

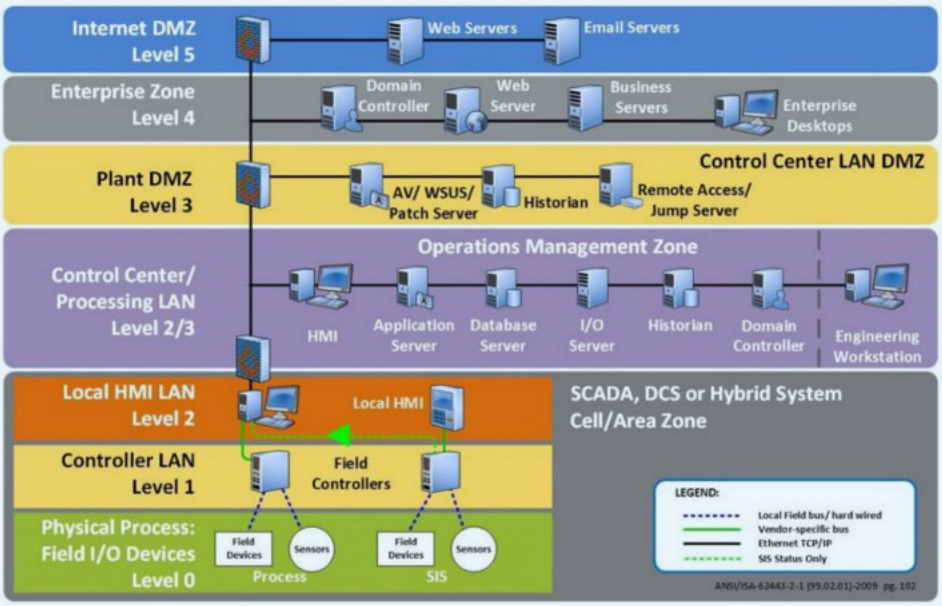


Figure 1 – Purdue Enterprise Reference Architecture – from **ANSI/ISA-62443-2-1**

Asset and networking changes. Several attacks are successful because OT networks have become increasingly interconnected and exposed, which allows threats to breach the internet-connected enterprise network and move between IT and OT networks. Often, enterprise and OT networks, which used to be completely separate (also known as air-gapped networks), are being interconnected to allow remote access, predictive maintenance, integration with ERP systems and other use cases. However, these networks are often not segmented as well as they should be, using firewalls, access control lists, data diodes and other measures, so that unwanted communications are allowed between different segments and threats can move from one domain to another. To complicate things further, once an attacker reaches the OT network, it is usually fairly easy to compromise OT devices. This is the case because OT products, with their long lifespans, often proprietary nature and backward compatibility demands, tend to retain insecure-by-design features for a long time. Patching is also complicated by uptime and safety requirements. A database of incidents – cyberattacks or just malfunctions – affecting operational technology can be found at http://search.infracritical.com/.

The issues in OT:ICEFALL that we describe in the rest of this report affect mostly level 1 and 2 devices and could be used in OT-specific attacks targeting those devices. Some of the vulnerabilities we report affect devices that have been targeted by real-world malware, such as CVE-2022-31206, an RCE affecting Omron NJ/NX controllers that were targeted by INCONTROLLER.

2. Main Findings

Below is a summary of the main findings of this research and the sections where they are further discussed:

- **Insecurity by design remains very relevant in OT** (Section 3.1) – The past decade has shown that one of the biggest security problems in OT continues to be the lack of basic controls, and OT-focused attackers have exploited this in practice.
- **Insecure-by-design vulnerabilities abound** (Sections 3.2 and 3.3) – We found 56 vulnerabilities affecting 10 major OT vendors. More than one-third of these vulnerabilities (38%) allow for compromise of credentials, with firmware manipulation coming in second (21%) and remote code execution coming third (14%). The prime examples of insecure-by-design issues are the nine vulnerabilities related to unauthenticated protocols, but we also found many broken authentication schemes, which demonstrates subpar security controls when they are implemented.
- **Vulnerable products are often certified** (Section 3.4) – 74% of the product families affected by the found vulnerabilities have some form of security certification and most issues we report should be discovered relatively quickly during in-depth vulnerability discovery. We list a set of factors contributing to this problem, such as limited scope for evaluations, opaque security definitions and focus on functional testing.
- **Risk management is complicated by the lack of CVEs** (Section 3.5) – It is not enough to know that a device or protocol is insecure. To make informed risk management decisions, asset owners need to know how these components are insecure. Issues considered the result of insecurity by design have not always been assigned CVEs, so they often remain less visible and actionable than they should.
- **There are insecure-by-design supply chain components** (Section 3.6) – Vulnerabilities in OT supply chain components tend to not be reported by every affected manufacturer. We discuss two vulnerabilities with CVEs assigned to the ProConOS runtime that we often encountered in PLCs and RTUs without an associated CVE or public discussion that they were affected.

- **Not all insecure designs are created equal** (Section 3.7) – We investigate three main pathways to gaining RCE on level 1 devices via native functionality: logic downloads, firmware updates and memory read/write operations. None of the systems analyzed support logic signing and most (52%) compile their logic to native machine code. 62% of those systems accept firmware downloads via Ethernet, while only 51% have authentication for this functionality.
- **Offensive capabilities are more feasible to develop than often imagined** (Section 3.8) – Reverse engineering a single proprietary protocol took between one day and two man-weeks, while achieving the same for complex, multi-protocol systems took 5 to 6 man-months. This shows that basic offensive cyber capabilities leading to the development of OT-focused malware or cyberattacks could be developed by a small but skilled team at a reasonable cost.

3. Technical Analysis

3.1. The Insecure-by-Design Debate

More than a decade ago, **Project Basecamp** showed that many OT devices and protocols deployed in a wide range of industries, and critical infrastructure applications were insecure-by-design. Since, it has been common knowledge that one of **the biggest issues facing OT security is not so much the presence of unintentional vulnerabilities but the persistent absence of basic security controls.**

While the past decade has seen the advent of standards-driven hardening efforts at the component and system level, it also has seen impactful real-world OT incidents, such as **Industroyer**, **TRITON** and **INCONTROLLER** abusing insecure-by-design functionality, which has left many defenders wondering just how much has changed.

One major difference is the increased adoption and proliferation of (certifiable) standards applicable to OT environments such as **IEC 62443**, **NERC CIP**, **NIST SP 800-82**, IEC 51408/CC and various sector-, region- or protocol specific (e.g. **IEC 62351**, **DNP3 Security**, **CIP Security**, **Modbus Security**) standards. And while these standards-driven hardening efforts have certainly contributed to major improvements in the areas of security program development, risk management and architecture-level design and integration activities, these efforts have been less successful at maturing secure development lifecycles for individual systems and components.

With OT:ICEFALL, we wanted to disclose and provide a quantitative overview of OT insecure-by-design vulnerabilities rather than rely on the periodic bursts of CVEs for a single product or a small set of public real-world incidents that are often brushed off as a particular vendor or asset owner being at fault. These issues range from persistent insecure-by-design practices in security-certified products to subpar attempts to move away from them. The goal is to illustrate how the opaque and proprietary nature of these systems, the suboptimal vulnerability management surrounding them and the often-false sense of security offered by certifications significantly complicate OT risk management efforts.

3.2. Vulnerabilities Found

We performed in-depth manual and automated analysis of the software, firmware and hardware components of the systems under test (SUTs) and their corresponding network traffic (from both lab setups, part of which are shown in Figure 2, and live environments) to develop a deep understanding of their proprietary protocols and corresponding system functionality. We leveraged this understanding to discover the issues listed in Tables 1-9 and subsequently develop deep packet inspection (DPI) capabilities able to detect possible exploitations.

Figure 2 – Several Devices in our Lab

Our analysis process for the devices in the lab included selecting interesting targets based on their popularity in critical infrastructure, acquiring selected devices and related software, commissioning devices in the lab, capturing packets from normal operations and special functions, reverse engineering software functions when necessary and identifying and disclosing vulnerabilities to the affected vendors.

Since the issues uncovered are the result of insecure design practices affecting core system functionality, many of them will remain unpatched in production environments for a significant amount of time. As such, we have chosen to not disclose full technical details for all issues in question.

Tables 1-9 present the new vulnerabilities we found, including their CVE IDs, short descriptions, affected products and impacts. There are four vulnerabilities still under disclosure. We do not give details about them but include them in our quantitative analysis on the next sections. One of these issues allows for the compromise of credentials due to a protocol transmitting them in plaintext; two allow for firmware manipulation due to an unauthenticated protocol and missing firmware signing; the final issue is an RCE via memory write.

Although the impact of each vulnerability is highly dependent on the functionality each device offers, we tried to summarize it using the following categories:

- **Remote code execution (RCE):** allows an attacker to execute arbitrary code on the impacted device, but the code may be executed in different specialized processors and different contexts within a processor, so an RCE does not always mean full control of a device. This is usually achieved via insecure firmware/logic update functions that allow the attacker to supply arbitrary code.
- **Denial of service (DoS):** allows an attacker to either take a device completely offline or to prevent access to some function.
- **File / Firmware / Configuration Manipulation:** allows an attacker to change important aspects of a device or system, such as operational parameters, files stored within the device, the firmware running on the device or specific configurations of the device. This is usually achieved via critical functions lacking the proper authentication/authorization or integrity checking to prevent attackers from tampering with the device.
- **Compromise of credentials:** allows an attacker to obtain credentials to device functions, usually either because they are stored or transmitted insecurely.
- **Authentication bypass:** allows an attacker to bypass existing authentication functions and invoke desired functionality on the target device.

We recommend readers follow the advisories of each vendor for more details on specific impact and products affected by each vulnerability.

Table 1– Bently Nevada (Baker Hughes) Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-29953	Maintenance interface has undocumented, hardcoded credentials.	Bently Nevada 3701	RCE
CVE-2022-29952	TDI command and data protocols have no authentication.	Products using TDI protocol	File manipulation, D0S

Table 2 – Emerson Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-29957	Several protocols, including firmware upgrade, plug and play, Hawk services, management, cold restart, SIS communications and wireless gateway protocol have no authentication.	DeltaV	Firmware manipulation, configuration manipulation, DoS
CVE-2022-29962	Hardcoded local credentials	DeltaV controllers	Compromise of credentials
CVE-2022-29963	Access to privileged operations on the shell interface is controlled by utility passwords generated using a deterministic, insecure algorithm.	DeltaV controllers	Compromise of credentials

Table 2 Continued – Emerson Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-29964	Hardcoded local credentials	DeltaV controllers	Compromise of credentials
CVE-2022-29965	Access to privileged operations on the maintenance interface is controlled by utility passwords generated using a deterministic, insecure algorithm.	DeltaV controllers	Compromise of credentials
CVE-2022-29966	Several protocols including controller diagnostics, database management, point edit, point register, point query, RPC, sheet loader and SIS communications have no authentication.	Ovation	Firmware manipulation, Configuration manipulation, DoS
CVE-2022-29959	Credentials for various users are stored insecurely in the SecUsers.ini file by using a simple string transformation.	OpenBSI	Compromise of credentials
CVE-2022-29960	DES with hardcoded cryptographic keys is used to protect system credentials, engineering files, and sensitive utilities.	OpenBSI	Compromise of credentials
CVE-2022-29961	The BSAP/IP protocol authenticates based on a MAC/IP whitelist and does not ensure cryptographic binding between subsequent messages and the authentication key material.	ControlWave, Bristol Babcock 33xx	Authentication bypass
CVE-2022-29954	The BSAP/IP protocol transmits passwords in plaintext.	ControlWave, Bristol Babcock 33xx	Compromise of credentials
CVE-2022-29955	The BSAP/IP protocol uses weak encryption to transmit passwords.	ControlWave, Bristol Babcock 33xx	Compromise of credentials
CVE-2022-29956	The BSAP/IP protocol transmits encrypted passwords that can be decrypted with a transmitted key challenge.	ControlWave, Bristol Babcock 33xx	Compromise of credentials
CVE-2022-30260	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	DeltaV M-series/S-series/P-series controllers, IO cards (CIOC/EIOC/WIOC) and DeltaV/Ovation SIS nodes (SLS1508/CSLS/LSNB/LSNG)	Firmware manipulation
CVE-2022-30267	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	Ovation OCR400, OCR1100 controllers and related IO modules	Firmware manipulation
CVE-2022-30262	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	ControlWave	Firmware manipulation

Table 2 Continued – Emerson Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-30261	The ROC protocol transmits passwords in plaintext.	ROC, FloBoss	Compromise of credentials
CVE-2022-30264	The ROC protocol allows for arbitrary file and directory read, write and delete operations.	ROC, FloBoss	RCE
CVE-2022-30266	Communications between an engineer's browser and the PLC Web UI are unprotected. User credentials are hashed by client-side Javascript.	PACsystems PLCs (with the exception of HTTPS-supporting models such as IC695, CPE330, CPE400)	Compromise of credentials
CVE-2022-30263	The SRTP protocol transmits passwords in plaintext.	Fanuc/PACSystems PLCs	Compromise of credentials
CVE-2022-30265	Control logic downloaded to the PLC, which can be either written in one of the IEC 61131-3 languages or written in C and supplied as an ELF binary block, is not cryptographically authenticated.	Fanuc/PACSystems PLCs	RCE
CVE-2022-30268	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	Fanuc/PACSystems PLCs except certain RX3i (CPx330, CPx400, CPx410) and RSTi-EP (CPE100, CPE115) models	Firmware manipulation

Table 3 – Honeywell Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-30312	The Inter-controller (IC) protocol transmits PINs, usernames and passwords in plaintext.	TREND controls products using the IC protocol	Compromise of credentials
CVE-2022-30313	The Honeywell Modbus TCP and Safety Builder protocols have no authentication.	Experion PKS Safety Manager	Configuration manipulation
CVE-2022-30314	Access to the boot configuration is controlled by credentials hardcoded in the Safety Manager firmware.	Experion PKS Safety Manager	Firmware manipulation
CVE-2022-30315	The Safety Builder protocol does not authenticate downloaded logic, allowing an attacker capable of triggering a logic download to execute arbitrary machine code on the controller's CPU.	Experion PKS Safety Manager (SM and FSC)	RCE
CVE-2022-30316	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	Experion PKS Safety Manager	Firmware manipulation

Table 3 Continued – Honeywell Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-30317	The EpicMo protocol has no authentication.	Experion LX	Firmware manipulation, DoS
CVE-2022-30318	Root credentials are hardcoded and not changed automatically upon first commissioning.	ControlEdge	Compromise of credentials
CVE-2022-30319	The S-Bus protocol authenticates functions based on a MAC/IP whitelist.	Saia Burgess Controls (SBC) - PCD controllers	Authentication bypass
CVE-2022-30320	The S-Bus protocol uses insecure hashing algorithm for passwords.	Saia Burgess Controls (SBC) - PCD controllers	Compromise of credentials

Table 4 – JTEKT Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-29951	The CMPLink/TCP protocol has no authentication.	TOYOPUC	File manipulation, DoS
CVE-2022-29958	The logic downloaded to the PLC is not authenticated.	TOYOPUC	RCE

Table 5 – Motorola Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-30276	The IPGW protocol has no authentication.	MOSCAD IP Gateway, ACE IP Gateway (CPU 4600)	Configuration manipulation
CVE-2022-30273	The MDLC protocol offers a legacy encryption mode that encrypts traffic using the Tiny Encryption Algorithm (TEA) block-cipher in ECB mode, which offers no message integrity and reduced confidentiality.	MDLC	Possible authentication bypass
CVE-2022-30270	The device ships with default credentials for five SSH accounts, some of which are undocumented and unlikely to be changed.	ACE1000	Compromise of credentials
CVE-2022-30271	The device ships with a hardcoded SSH private key, which is likely to be used by default.	ACE1000	Compromise of credentials

Table 5 Continued – Motorola Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-30274	Credentials for the XRT LAN-to-radio gateway and authentication to the XNL port are protected with the Tiny Encryption Algorithm (TEA) in ECB mode using a hardcoded key.	ACE1000	Compromise of credentials
CVE-2022-30275	The wmdlcdrv.ini driver configuration file stores passwords in plaintext.	MOSCAD/STS Toolbox, StarControls staRTU	Compromise of credentials
CVE-2022-30269	Application images are not signed and only rely on insecure checksums for regular integrity checks.	ACE1000	RCE
CVE-2022-30272	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	ACE1000	Firmware manipulation

Table 6 – Omron Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-31204	The password used to restrict engineering operations is transmitted in plaintext.	SYSMAC CS1/CJ1/ CP1/CP2 series	Compromise of credentials
CVE-2022-31205	The password to access the Web UI can be read from memory using the Omron FINS protocol without any further authentication.	SYSMAC CP series	Compromise of credentials
CVE-2022-31207	The logic that is downloaded to the PLC is not cryptographically authenticated, allowing an attacker to manipulate transmitted object code to the PLC and execute arbitrary object code commands on the defined software logic.	SYSMAC CS/CJ/CP series	Logic manipulation
CVE-2022-31206	The logic that is downloaded to the PLC is not cryptographically authenticated, allowing an attacker to manipulate transmitted object code to the PLC and execute arbitrary machine code on the processor of the PLC's CPU module.	SYSMAC NJ/NX	RCE

Table 7 – Phoenix Contact Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-31800, CVE-2022-31801	The logic that is downloaded to the PLC is not cryptographically authenticated, allowing an attacker to execute arbitrary code."	ProConOS/eCLR Runtime	RCE

Table 8 – Siemens Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-33139	The WinCC OA Dektop UI uses client-side only authentication, when neither server-side authentication (SSA) nor Kerberos encryption is enabled. In this configuration, attackers could impersonate other users or exploit the client-server protocol without being authenticated.	WinCC OA	Authentication bypass

Table 9 – Yokogawa Vulnerabilities

ID	DESCRIPTION	AFFECTED PRODUCTS	IMPACT
CVE-2022-29519	The ResConf protocol transmits usernames, passwords and session tokens in plaintext.	STARDOM	Compromise of credentials
CVE-2022-30997	The maintenance interface on port 23/TCP has undocumented, hardcoded credentials.	STARDOM	Compromise of credentials
FSCT-2022-0039	Firmware images are not signed and only rely on insecure checksums for regular integrity checks.	STARDOM	Firmware Manipulation

3.3. Analysis of Found Vulnerabilities

Figure 3 shows the vulnerabilities of Section 3.2 divided by their impact. Five vulnerabilities were counted more than once because they have various possible impacts (such as CVE-2022-29953, which allows for file manipulation and DoS). More than one-third of vulnerabilities (38%) allow for compromise of credentials, with **firmware manipulation coming in second (21%) and RCE coming third (14%)**. We discuss the RCEs and firmware vulnerabilities in more details in Section 3.7.

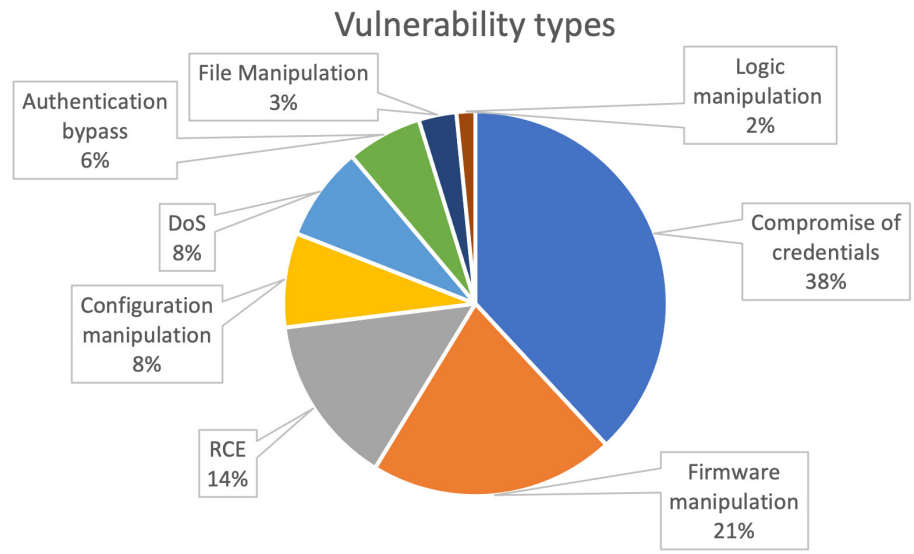


Figure 3 – Vulnerability Types in OT:ICEFALL

The prime examples of insecure-by-design issues in OT are the vulnerabilities related to unauthenticated protocols, which means any attacker with a presence on the network can invoke (potentially very sensitive) functions on a target device. There are nine examples of those in OT:ICEFALL: CVE-2022-29953, CVE-2022-29957, CVE-2022-29966, CVE-2022-30264, CVE-2022-30313, CVE-2022-30317, CVE-2022-29952, CVE-2022-30276 and one vulnerability under disclosure. Most of those allow for firmware and logic downloads (which lead to RCEs in several cases) or shutdown/reboot commands (which lead to DoS); however, there are other native capabilities that are interesting from the attacker point of view, such as direct manipulation of I/O or variables in the PLC and issuing passthrough commands via gateway to devices nested behind.

Although many vulnerabilities are due to the insecure-by-design nature of OT, an important observation of our findings is that **many authentication schemes are broken**, which also demonstrates subpar security controls when they are implemented. Table 10 lists the vulnerabilities in OT:ICEFALL that are related to broken authentication and details how they are broken: **plaintext credentials**, **authentication bypass** or **broken cryptography**. One vulnerability under disclosure, related to plaintext credentials, is missing in the table.

Table 10 – OT:ICEFALL Vulnerabilities Related to Broken Authentication Schemes

PRODUCT	VULNERABILITY TYPE	CVES
Emerson ControlWave	Authentication bypass, plaintext credentials, broken or risky crypto	CVE-2022-29961, CVE-2022-29954, CVE-2022-29955, CVE-2022-29956
Emerson ROC	Plaintext Credentials	CVE-2022-30261
Emerson PACSystems	Plaintext credentials, broken or risky crypto	CVE-2022-30266, CVE-2022-30263
Emerson OpenBSI	Hardcoded keys, broken or risky crypto	CVE-2022-29959, CVE-2022-29960
Emerson DeltaV	Hardcoded credentials, broken or risky crypto	CVE-2022-29962, CVE-2022-29963, CVE-2022-29964, CVE-2022-29965
Honeywell Trend IQ	Plaintext credentials	CVE-2022-30312
Honeywell Safety Manager	Hardcoded credentials	CVE-2022-30314
Saia Burgess PCD	Authentication bypass, broken or risky brypto	CVE-2022-30319, CVE-2022-30320
Motorola MDLC	Broken or risky crypto	CVE-2022-30273
Motorola ACE1000	Hardcoded credentials	CVE-2022-30270
Omron Cx series	Authentication bypass, plaintext credentials	CVE-2022-31204, CVE-2022-31205
Siemens WinCC OA	Client-side authentication	CVE-2022-33139

We compared our findings to prior work and found that this is a recurring pattern not addressed by standards because they focus on functional testing (see Section 3.4). Table 11 shows an illustrative selection of similar previous work. Note that this overview only concerns those products that tried to be secure-by-design and disregards the many cases of non-engineering interfaces (such as regular FTP, Telnet or SSH access) with hardcoded credentials.

Table 11 – Previous Vulnerabilities Related to Broken Authentication Schemes

PRODUCT	VULNERABILITY TYPE	CVES
ABB PGIM	Client-side authentication Plaintext credentials	CVE-2019-18250
AutomationDirect CLICK	Authentication bypass Plaintext credentials	CVE-2021-32980, CVE-2021-32982, CVE-2021-32984, CVE-2021-32986
Baker Hughes Bently Nevada 3500	Broken or risky crypto	CVE-2021-32997
CODESYS	Insufficient randomness Broken or risky crypto	CVE-2018-20025, CVE-2019-9013
Ovarro Tbox	Hardcoded key	CVE-2021-22640, CVE-2021-22644
Rockwell Micrologix 1100, 1400	Plaintext credentials Broken or risky crypto	CVE-2021-32926
Rockwell Logix	Hardcoded key	CVE-2021-22681
Rockwell ISaGRAF	Hardcoded key	CVE-2020-25180
Schneider Electric Modicon M221	Authentication bypass Broken or risky crypto	CVE-2017-6034, CVE-2018-7790, CVE-2018-7791, CVE-2018-7792, CVE-2020-7566, CVE-2020-7568
Schneider Electric Machine Expert Discovery	Hardcoded key	CVE-2019-6820
Schneider Electric Modicon M340, M580	Authentication bypass	CVE-2017-6034, CVE-2017-6032, CVE-2019-6855, CVE-2020-7537, CVE-2021-22779
Schneider Electric TriStation	Broken or risky crypto	CVE-2020-7483
Siemens, S7-300, S7-400	Broken or risky crypto	CVE-2020-15791
Siemens S7-1200, S7-1500	Man-in-the-middle	CVE-2019-10943

3.4. Vulnerable Products are … Certified!

As shown in Figure 4, most product families affected by the issues discussed in this research had achieved one or more of the following certifications:

- **ISASecure Component Security Assurance (CSA)**: Subsumes **Embedded Device Security Assurance (EDSA)**, based on **IEC 62443-4-1** and **IEC 62443-4-2**.
- **ISASecure System Security Assurance (SSA)**: Based on **IEC 62443-4-1** and **IEC 62443-3-3**.
- **GE Achilles Communications Certification (ACC)**: Similar to, but not based on, **IEC 62443-4-2**.
- **ANSSI Certification de Sécurité de Premier Niveau (CSPN)**: Based on **Common Criteria**.

ISASecure Secure Development Lifecycle Assurance (SDLA) and **GE Achilles Practices Certification** (APC) were left out of our figures due to their implied nature in other certifications. In addition, several products claimed to have a security posture "based on IEC 62443" – in some cases "up to SL3/SL4" but were not certified as such.

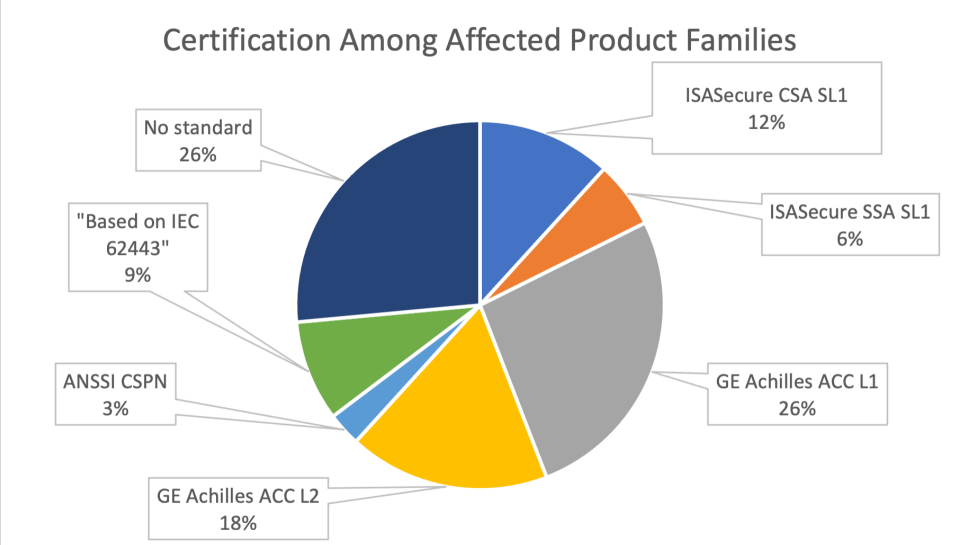


Figure 4 – Security Certification Among Affected Product Families

Considering the vulnerabilities discussed in this research are either the result of insecure-by-design or often-trivial failures of security design, these findings point to some serious issues with security standards and certifications for OT. In particular, the following factors seem to contribute to this situation:

- **(Re)certification effort:** Security certifications are typically either valid for a limited period or a specific hardware and software version, after which a recertification audit is required to ensure all changes are compliant. Since this can be a lengthy and costly process, which grows in scope as the targeted security levels go up, it seems some vendors have opted for certification at the lowest level only while others claim to have developed products 'according to' a standard but have not gone through the effort of actual certification.
- **Limited Target of Evaluation (ToE)**: Security certifications typically have an implicit or explicit ToE, a specific system, or set of components thereof, which is subjected to evaluation for a given set of security requirements according to a given security profile or level. It is not uncommon for ToEs to be

incredibly limited and not cover some of the most relevant attack surface, such as proprietary engineering functionality or third-party networking libraries.

- **Opaque security definitions**: Many security standards use opaque definitions. For example, the IEC 62443 Security Assurance Levels are defined to correspond to **attacker classes of increasing sophistication**. This sophistication, however, is defined in very generic and opaque terms such as 'moderate resources', 'sophisticated means' and 'IACS specific skills.' These terms, when left vague and unquantified, lend themselves to idiosyncratic interpretations more reflective of the auditor's perceptions and expectations than of a product's security posture. For example, would a decent **Capture the Flag (CTF)**-playing teenager with a few weeks of spare time, a free copy of **Ghidra**, **dotPeek**, **Wireshark** and a Wikipedia-level understanding of Modbus be considered a level 3 or 4 attacker? Because such an adversary would be more than capable of reverse-engineering and exploiting most proprietary Modbus extensions. To complicate matters further, the IEC 62443 security requirements are incremental. That is to say, the requirements for level 1 are a subset of level 2 and so on. As such, a product evaluated for mere protection against 'unintentional misuse' (level 1) will have an authentication requirement that satisfies protection against 'state-sponsored actors' (level 4). Once a product is certified at a certain level and aims to achieve the next, it is unlikely auditors will reevaluate already met requirements with more scrutiny rather than just focus on the missing requirements.
- **Limited security evaluation**s: Many security certification processes limit the evaluation of security requirements to functional testing. That is, features are verified to be present, but no inspection for robustness is made. In addition, only a very limited amount of time, typically between a **few days and two weeks**, is spent on functional security assessments on all relevant interfaces. This type of testing typically excludes any sort of investigation of proprietary protocols. As such, a functional security assessment might conclude authentication is present on an engineering interface while in reality the protocol is unauthenticated, and all authentication is done client-side. The same holds for fuzzing- and conformance-based communication robustness tests that will only be able to assess open protocols for which the specifications are known by the auditors. For those standards with security assessments that go beyond the purely functional but do not operate on a white-box principle or where source code is provided but detailed specifications of proprietary protocols are simply not available, the proprietary nature of many OT products can remain an obstacle since this will require auditors to spend a significant amount of their limited time to reverse engineer functionality, which increases chances these aspects are left out of the ToE.

Particularly illustrative of the limits of functional testing is the wide range of OT products with broken authentication and access control schemes, as shown in Section 3.3. The issues in question (plaintext credential transmission, hardcoded keys, broken or weak cryptography) are typical for low-quality authentication schemes and are discovered relatively quickly during in-depth vulnerability discovery efforts. Functional testing, however, will not uncover such issues and, as such, lead to certified products incorporating these types of flaws.

This is not to say standards-driven security efforts do not have an important role in hardening the OT security landscape, but **it does point to the serious and often downplayed limits of stopping at mere compliance**. In order to avoid standards and certifications from prepping up **Potemkin security**, products with insecure-by-design features nullifying security mechanisms that are part of the ToE ought to be uncertifiable and all parties involved ought to settle on a common understanding of the meaning of being conformant or compliant. That is, if the evaluation of a security requirement is limited to a functional test, this ought to be reflected in the framing of the corresponding security levels.

3.5. The Effect of Opacity on Risk Management

One of the most striking findings of this research is the effects of the opaque nature of OT product on risk management. While it is well known that many OT products are insecure-by-design and vendors often recommend that asset owners treat these products as such by focusing on perimeter hardening and monitoring, the proprietary nature of many components of these systems complicates sound risk management.

After all, it is insufficient to simply know that a protocol or interface is insecure. **To make informed decisions around segmentation, monitoring and hardening efforts, asset owners need to know *in what way* these components are insecure**. There is a huge difference in potential impact between an attacker's ability to change a setpoint, which could be limited by logic-based sanity checks and downstream alarms, and their ability to execute arbitrary code on a controller.

Similarly, while segmentation is often top-of-mind in OT environments, many vendors still require various types of cross-zone communications to function. They will typically recommend system integrators to whitelist certain port ranges in firewalls. Without DPI capabilities and without granular insights into the insecurity of the associated protocols, asset owners will have no idea what they just exposed outside their perimeters.

Since vendors are naturally disinclined to provide such information, the onus has typically been on asset owners or third-party security personnel to bridge this gap. This has been somewhat complicated by the historically sensitive nature of OT security research. For a variety of reasons, issues considered the result of insecure-by-design have not always been assigned CVEs and, as a result, they often remain less visible and actionable than they ought to be. In Section 3.4, we discussed how products affected by these issues can end up security certified, making them seem more secure than they are in practice. Absent explicit CVEs, asset owners could be forgiven for thinking they are in control after procuring OT products certified to have an authentication mechanism. They could be similarly forgiven for prioritizing security efforts geared to addressing minor issues with CVEs over addressing more impactful issues for which there is no CVE or official guidance, particularly in quantitative, KPI-driven security programs working their way through threat feeds and CVE lists in their visibility and control solutions.

3.6. Supply Chain Issues: the ProConOS/eCLR Runtime Case

Unlike **Project Memoria** and its OT study **INFRA:HALT**, this research focused on individual devices and vendors. Nevertheless, we also encountered vulnerabilities on an important supply chain component of OT devices: the **ProConOs runtime system**.

A runtime system is the component responsible for running the logic program of the PLC, which cyclically scans the device's input and produces the desired output. Previous research on PLC runtimes has found critical vulnerabilities on **CODESYS**, **ISaGRAF** and **ProConOS**.

The Phoenix Contact (previously KW-Software) ProConOS/eCLR runtime system provides the backbone for programmable controllers of many different vendors, allowing for the execution of IEC 61131 and C# programs, as well as various controller management functions.

Due to **the lack of Software Bill of Materials (SBOM)** and the complexity of product supply chains, it is often not immediately clear what runtime a particular PLC uses. Runtimes typically have different versions with corresponding protocol differences and are subject to OEM integration decisions. A PLC manufacturer may choose to use the runtime but not the protocols, preferring to use its own, or may choose to use the protocol on a non-default port or may choose to rebrand or modify the runtime altogether. Absent proactive, coordinated efforts by vendors, CVE numbering authorities and CERTs to propagate knowledge of supply chain vulnerabilities to all affected parties, the security community is forced to rediscover them periodically and haphazardly, resulting in CVE duplication and complicating root-cause analysis.

In the past, two CVEs have been assigned to ProConOS protocols: **CVE-2014-9195** and **CVE-2019-9201**. These CVEs, however, were associated only with Phoenix Contact and its own PLCs and did not propagate to other vendors incorporating this software package. This means the various other vendors and controllers using ProConOS/eCLR and the asset owners using them are likely unaware of these security issues. This has already led to a duplicate finding in the past with **CVE-2016-4860**, which seems specific to Yokogawa STARDOM controllers and makes no mention that it is the same issue as CVE-2014-9195.

During our research into PLCs and RTUs by various vendors, we kept encountering this runtime and its associated protocols but could find no associated CVEs for these products or public discussion that they were affected. As such, we compiled a more extensive overview of vendors and products using a variant of the ProConOS/eCLR runtime, possibly in combination with one of its protocols, from our internal research and open-source intelligence. We coordinated with Phoenix Contact, **CERT VDE** and **CISA** to contact these downstream vendors individually and hopefully include them in the original CVEs. In addition, while **prior work** has pointed out the possibility for remote code execution on ProConOS systems, no CVE was assigned for this. Since we confirmed this possibility against many other devices, we requested CVE-2022-31800 cover this issue for all affected vendors. It is important to note that not all products affected by CVE-2022-31800 are also affected by CVE-2014-9195 or CVE-2019-9201. For example, the Emerson ControlWave family of PLCs/RTUs uses the ProConOS runtime but performs engineering operations using its own proprietary BSAP/IP protocol.

Table 12 – Products Using the ProConOS/eCLR Runtime and/or the SOCOMM, ADE or DDI Protocols.

VENDOR	QUERY
Phoenix Contact	AXC 1xxx, AXC 3xxx, RFC 4xx, ILC 1xx ETH, ILC 3xx, FC 200, FC 350
Emerson	ControlWave 'Next Generation': including but not limited to CWM, CWP, CWX, CMR, CME, GFC, XFC, EFM, PAC and LP
ABB	RTU 5xx (RTU520/RTU540/RTU560)
Advantech	ADAM-3600, ADAM-5xxx, APAX-5xxx, APAX-6xxx, AMAX-2050, UNO-2171
KUKA	KUKA.PLC
ICP DAS	KinCon-8xxx
Yaskawa	MPiec
Schleicher	XCx (300/400/500/700/800/1100/1200)
Hilscher	netPLC
Luetze	DIOLINE PLC
Delta	DMXC
ISH	SIS800, SIC400, uPLC iSOC300P
Yokogawa	STARDOM FCJ, FCN-RTU, FCN

3.7. Not All Insecure Designs Are Created Equal

While Ralph Langner's observation that "***the pros don't bother with vulnerabilities; they use features to compromise the ICS***" has become almost cliché when talking about insecure-by-design in OT, an often-overlooked nuance is that not all these features are equally impactful. After all, there's a big difference between being able to change a variable in PLC memory, modifying control logic and getting remote code execution. In the latter case, an attacker typically has full control over the level 1 device and can use this to execute more **advanced attack scenarios** such as the TRITON implant or turning a PLC into a beachhead for lateral movement into otherwise unreachable fieldbus networks for direct manipulation of sensors, actuators and variable-frequency drives. Typically, there are three main pathways to gaining RCE on level 1 devices by relying on native functionality, which we discuss in this section.

3.7.1. Logic Downloads

Most level 1 devices are typically programmed using one of the **IEC 61131-3** languages or a proprietary equivalent. As shown in Figure 5, programming is done in an IDE where the source code is compiled to either native machine code, an FPGA bitstream or proprietary bytecode or scripting language of some sort. The resulting project is then downloaded to the controller, which will load the logic into its runtime environment for scheduled execution or interpretation. Due to a common lack of logic signing and runtime security measures, the ability to download logic typically grants an attacker the ability to execute native machine code on a controller with capabilities way beyond mere logic modification.

We examined the logic generation and runtime mechanisms used in **36 product families** from our research, as well as **nine product families** covered in prior work, spanning over **10 different runtime systems** in total. As shown in Figure 6, the vast majority compile their logic to native machine code for direct execution on the CPU module's microprocessor.

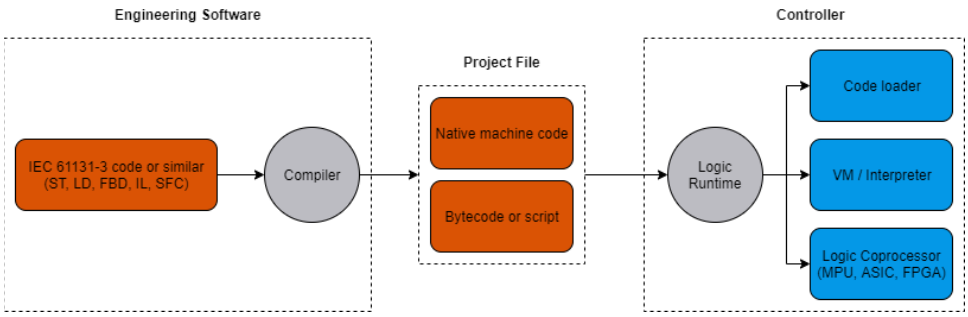


Figure 5 – Typical Programmable Logic Ecosystem

A second class compiles their logic to a proprietary bytecode, which is either interpreted by a virtual machine or a dedicated ASIC. Finally, certain DCSs seem to prefer handling controller strategy through interpreted scripts. The dominance of native machine code execution is worrying given it presents by far the easiest target for achieving RCE.

This is compounded by the fact that none of the examined systems signed their logic, and with the exception of one product family, no sandboxing was used for those systems executing native machine code. In addition, many of the products examined use hardware and OS combinations that do not allow for memory and privilege separation, resulting in attacker code execution with the highest possible privileges.

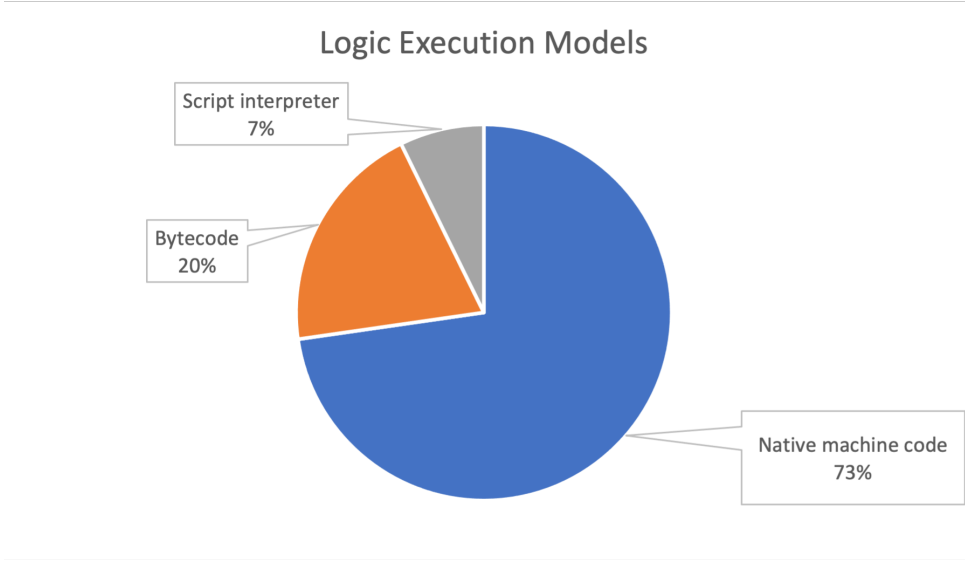


Figure 6 – Logic Execution Models Of The Analyzed OT Equipment

One of the most prominent defenses against unauthorized logic downloads is relying on physical mode switches governing controller operating modes. Many controllers need to be in a specific operating mode (such as RUN or PROGRAM) before a download can be initiated, and in some cases, these modes are set by a physical switch (a key, toggle, rotary switch or combination of panel buttons). The switch is typically connected to some pins on the controller microprocessor that are actively polled or trigger an interrupt of some sort.

However, not all mode switches offer this kind of protection. What is crucial is that these switches make a distinction between RUN and PROGRAM modes so that operators and engineers can follow explicit policy to only ever switch to PROGRAM mode during work order-initiated changes. Many mode switches do not make this distinction, however, supporting only a combined RUN/PROGRAM mode that offers no protection. In addition, one should keep in mind that even in the case of switches that do distinguish between these modes, the absence of logic signing means an attacker can still strike during legitimate maintenance windows.

In Figure 7, we mapped the support of the examined systems for either full (distinct RUN and PROGRAM modes), partial (no distinct RUN/PROGRAM modes) or no mode switch support, while "Various" indicates that different products within the same family had different types of switches. Worryingly, only a minority had full mode switch support, although this is somewhat offset by most safety systems investigated falling within this group. One mitigating factor is that some vendors have started to offer plug-and-play solutions at a network level where engineering actions require physical interaction with a button on a switch or firewall.

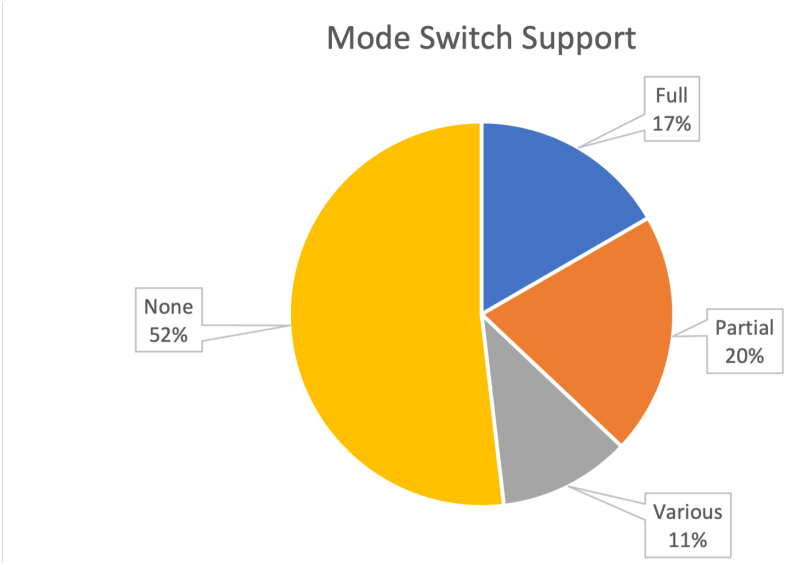


Figure 7 – Mode Switch Support

As such, it seems that in a non-trivial number of cases, attackers who merely figure out how a logic download is performed against a system can achieve unconstrained code execution unhindered by any physical security measures.

3.7.2. Firmware Updates

In order to apply bugfixes, security patches and introduce new functionality, most OT devices have a way to receive firmware updates. Typically, there are four main channels to transmit firmware updates on OT devices: Ethernet, serial connections, USB and SD cards.

We investigated the firmware updating mechanisms of 31 product families from our dataset and 9 from prior work. Figure 8 shows that the vast majority of those do updates over Ethernet.

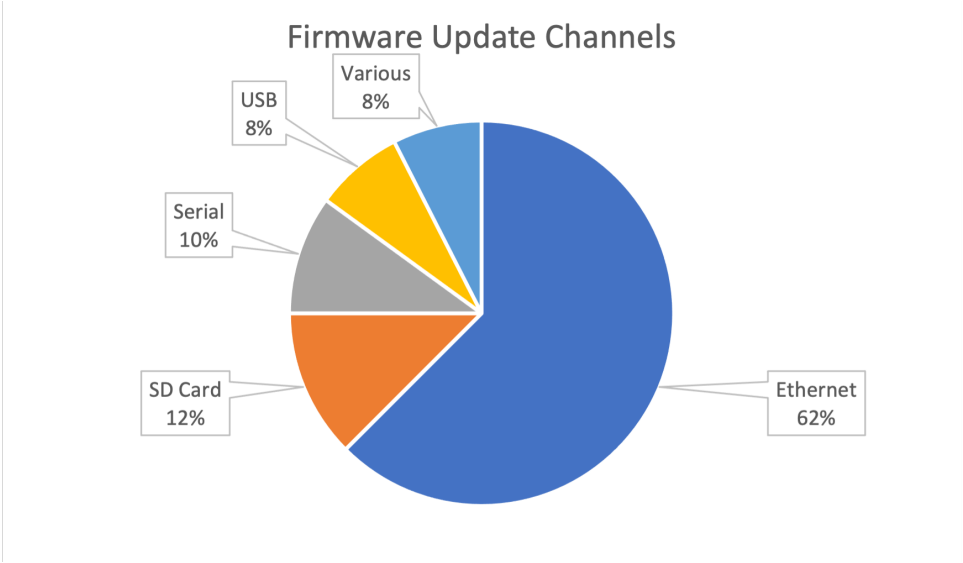


Figure 8 – OT Firmware Update Channels

In addition, we found that only 51% of the examined devices had some sort of authentication for firmware updates, even if this was in the form of hardcoded credentials in some cases, while only 22% performed some sort of cryptographic firmware signing.

These findings are troubling since they indicate that not only are many OT device firmware update mechanisms insecure (either by design or by implementation), they expose these mechanisms to the network more often than not. Only a small amount of these devices requires some sort of mode switch operation involving physical operator intervention.

Finally, while serial, USB and SD card transmission channels are less at risk than Ethernet-based channels, it is important to keep in mind that without proper firmware signing, these devices are still at risk of compromised engineering workstations and attackers piggybacking on legitimate firmware updates. It is also worth mentioning that we have observed several cases where asset owners permanently connected serial or USB interfaces used for firmware updates to engineering workstations or Ethernet media converters, rendering these interfaces permanently accessible to attackers. As such, following **secure firmware management practices** is of paramount importance.

3.7.3. Memory Read / Write Operations

One relatively underexamined aspect of many proprietary OT protocols is that they tend to be full of (undocumented) commands for performing memory read or write operations. In some cases, these operations are restricted to manipulating operational variables, but in others they operate directly on a device's internal memory organization in the form of some object- or block-based scheme.

An attacker with protocol and internal memory layout knowledge can typically retrieve and modify configuration information and, in some cases, runtime executable code held in user program or firmware-native functional block areas. In cases where such memory areas are not directly exposed for modification, attackers can sometimes abuse the lack of bounds-checking on these operations. For example, if writing to code areas is disallowed but an attacker knows their offset from writable data areas, they can attempt to write enough data to those areas to essentially overflow into the executable ones. The typical lack of memory and privilege separation on controllers makes this kind of endeavor even easier.

Memory read and write operations played a role in authentication bypasses and RCE such as in this work's CVE-2022-31205 against Omron Cx controllers and one of the vulnerabilities currently ongoing disclosure, as well as in previous vulnerabilities in **Opto 22 energy monitoring devices** and the **Schneider Electric M340/M580** PLCs. This type of vulnerability illustrates that security controls cannot be bolted onto a product that is otherwise still riddled with insecure-by-design features.

3.8. Offensive Cyber Capability Development for OT

Offensive Cyber Capabilities (OCC) are the whole of "***resources, skills, knowledge, operational concepts and procedures [required] to be able to have an effect in cyberspace***" and underpin the execution of **Computer Network Operations** (CNO). While technical activities such as vulnerability research, exploit and malware development and their resulting artifacts are only one aspect of OCC, insights into the corresponding effort required can often aid defenders. (e.g., if one wants to determine what exactly constitutes a 'sophisticated and well-resourced attacker' or attempts to get an idea of baseline capability development cost to determine where, how and how much additional cost to impose).

While generally speaking it is hard to say anything sensible about the **cost of developing a 'cyber weapon'**, there have been some efforts into obtaining quantitative data on zero-day **vulnerability research and exploit development** activities. However, this data is not broken down by target type, and capability development for hardened, IT-oriented general-purpose systems differs significantly from insecure-by-design, embedded OT systems. For one, the absence or low-quality nature of OT security controls typically reduces vulnerability research efforts to reverse engineering to reproduce native features or extract hardcoded key material. Secondly, once a capability is developed (or repurposed from another attacker's malware), it typically has a long shelf life and low maintenance cost since patches are unlikely to be developed and rolled out quickly (or sometimes, ever). As such, most of the narrow technical cost associated with OT OCC development comes from acquiring and developing a deep understanding of a given OT product, producing tooling for interacting with it and testing this tooling for reliability. In the case of **TRITON**, for example, this meant reverse engineering the TriStation protocol and parts of the Triconex ETSX runtime to develop the capabilities to download a program, run arbitrary machine code, escalate to obtain supervisor privilege, and patch firmware memory to install an in-memory implant.

In order to get some quantitative insights into the associated cost, we ran some numbers on the reverse engineering efforts required to understand the proprietary protocols covered in this research. Since these efforts mainly served to develop DPI capabilities, they typically result in a far more extensive understanding of the protocols than strictly necessary for OCC development. This is particularly true if one is not concerned with advanced, in-controller implants but with simpler attacks relying on starting or stopping controllers and changing operational parameters.

Out of **17 different product families**, thoroughly reverse engineering a single, simple proprietary protocol took between **one day and two man-weeks**. Achieving a similar understanding for complex, multi-protocol systems (such as DCSes) typically took **five to six man-months**.

Of course, not all reverse engineering efforts are equal. Taking apart a 5GB software package written in C++ is not the same as a single 400 KB .NET binary, just as analyzing a Linux-based router is not the same as doing digital archeology on a deeply embedded system with an obscure microcontroller and in-house RTOS. To determine if the above indications are representative, we gathered data on the technical composition of the software and firmware involved (restricting ourselves to relevant components only). As shown in Figure 9, the vast majority of software was written in C++ which is typically **more tedious and involved** than C or .NET. In addition, many software packages made heavy use of MFC, ATL, COM, RPC and Qt, while in some cases a single subsystem implementation (such as a protocol parser and dispatcher) was spread out across multiple DLLs loaded by different processes interacting through IPC.

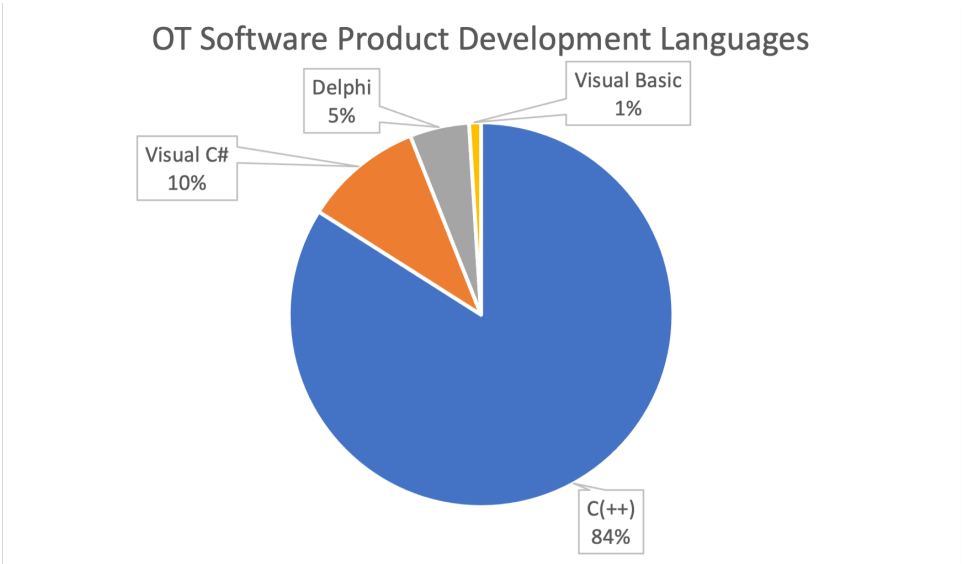


Figure 9 – OT Software Product Development Languages

Similarly, when reverse engineering embedded systems firmware, one might run into CPU architectures and operating systems not supported by the capabilities of your current toolchain (e.g., code and data topology reconstruction, function recognition and identification, among others). We enumerated the different CPU architectures and OSes of relevant firmwares for **32 different product families**, as shown in Figure 10 and Figure 11. While the most popular architectures and OSes are typical of non-consumer embedded systems, we noticed some significant outliers based either on region (e.g., SuperH with OS-9 or ITRON in Asia) or market share (many smaller vendors opted for free or obscure/legacy RTOSes). All investigated firmwares were exclusively written in a mix of assembly and C(++) with virtually no encryption or obfuscation used other than heavily proprietary file formats.

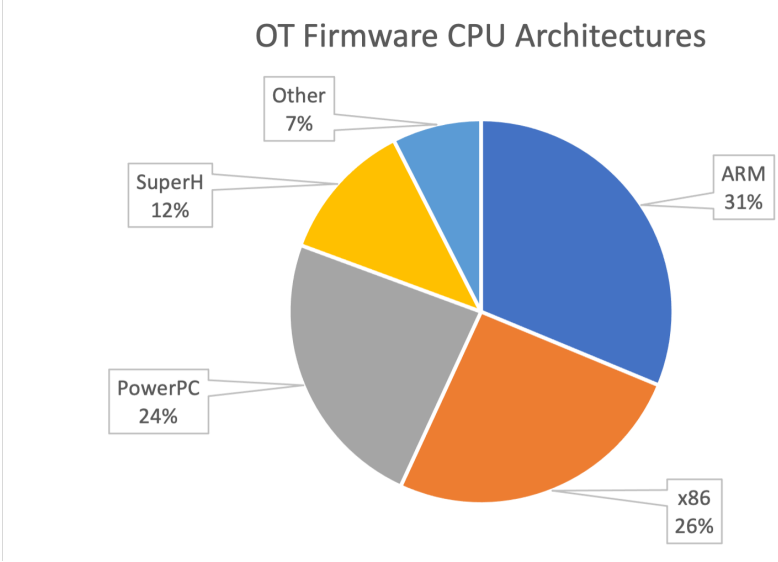


Figure 10 – OT Firmware CPU Architectures

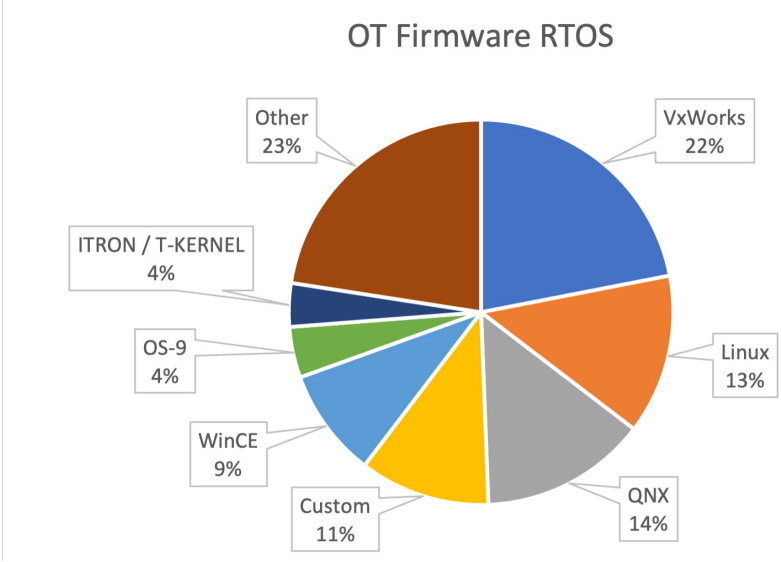


Figure 11 – OT Firmware Real-time Operating Systems

Given these figures, we can safely say that our indications regarding attacker efforts do not represent the most trivial systems to reverse engineer, and they are likely generalizable to most OT products. As such, we think it is reasonable to assume basic OT OCC could be developed by a small but skilled team at surprisingly reasonable cost given the right incentives.

4. Attack Scenarios

The vulnerabilities in OT:ICEFALL affect products commonly used in many critical infrastructure sectors (see Section 4 for more details). To illustrate some potential attack scenarios leveraging these vulnerabilities, we will describe scenarios for three different sectors: natural gas transport, wind power generation and discrete manufacturing. Of course, these scenarios are merely illustrative and other sectors are at risk in similar fashion.

4.1. Natural Gas Transport

When natural gas is being transported through a pipeline, it needs to be periodically repressurized, something which is handled by compressor stations. These stations consist of one or more gas compressors that compress incoming gas, thereby increasing its pressure and providing the energy necessary to move it through the pipeline. After filter separation, incoming natural gas is fed to the compressor unit, which is typically powered either by fuel gas taken from the pipeline or a separate electric motor. Lubrication systems protect and cool the compressor engine, while gas cooling systems cool discharged gas before it is returned to the pipeline for protection and improved flow rate. Compressor stations are typically managed at the site level by a DCS solution, possibly interfacing with a separate safety system. Stations are then managed remotely through RTUs connected to a central SCADA system. Figure 12 summarizes the scenario above.

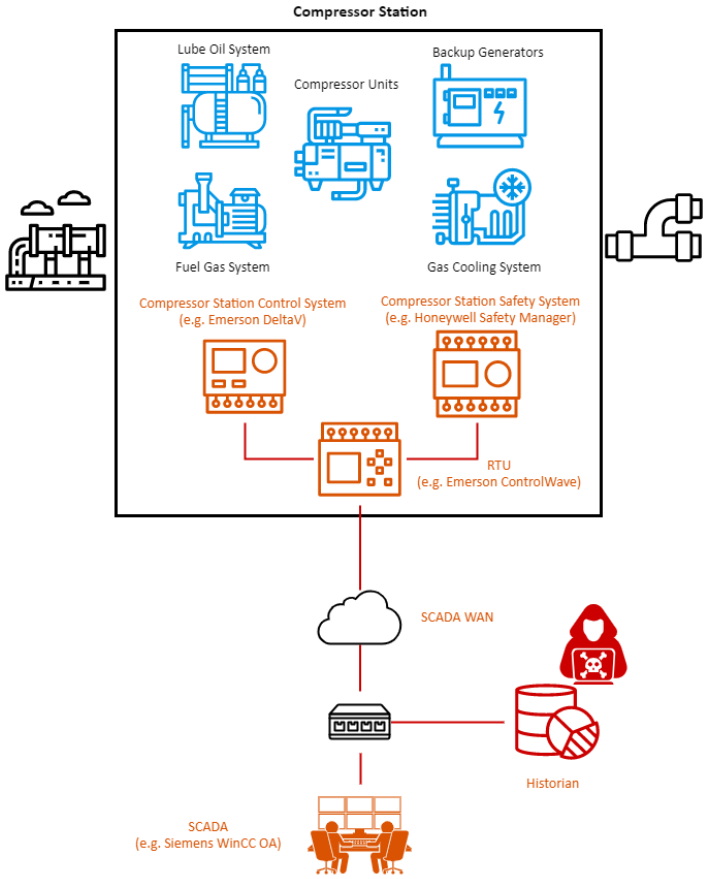


Figure 12 – Natural Gas Transport System

Attack Scenarios: Assuming an attacker compromises the historian in Figure 12, typically a prime target for attackers due to their interconnected nature, there are several attack scenarios an attacker could execute by leveraging the vulnerabilities disclosed in this report:

- **Manipulation of Control** / **View**: An attacker capable of communicating with the WinCC OA SCADA server could leverage CVE-2022-33139 to **bypass authentication** and subsequently **manipulate setpoints and monitoring values** relating to compressor stations to overwhelm operators with false alarms or change flow setpoints to disrupt transport.
- **Denial of Control** / **View**: An attacker capable of communicating with the Emerson ControlWave Micro RTUs could leverage either CVE-2022-29961, CVE-2022-29954 or CVE-2022-29955 to **bypass authentication** to the RTUs and **issue commands** that would halt the logic being executed, as well as sever communications between SCADA and RTUs, denying operators the ability to control and monitor the compressor stations, thus amplifying manipulative actions.
- **Loss of Control** / **Safety**: From the point of view of the SCADA system, the compressor stations are typically treated as package units exposing only a minimal monitoring and control interface through the RTU. As such, an attacker wishing to cause more serious disruptive or even destructive scenarios will have to achieve more granular control over the systems governing the compressor station itself through downstream hacking. Here, an attacker could leverage CVE-2022-30262 or CVE-2022-31801 to **gain code execution** on the ControlWave RTU and communicate on its various network interfaces without restrictions.

Crossing this bridge into the compressor station network itself, the attacker could try to move laterally to the DCS application workstation, or even into the Area Control Network (ACN) if it is improperly segmented, and then leverage CVE-2022-29957 and CVE-2022-30260 to **manipulate the configuration, settings and controller firmwares** of the Emerson DeltaV DCS. This could give an attacker the capability to cause pressure drops in suction lines, disrupt the lubrication and cooling systems, close discharge valves and disable antisurge controls to cause a variety of dangerous pressure-related surge and stalling conditions which, in turn, can lead to mechanical and temperature-related damage. If the attacker then manages to cross over into the safety network, they could leverage CVE-2022-30313 and CVE-2022-30315, the attacker could **manipulate settings** or even **gain code execution** (assuming the right physical keyswitch settings) on the Honeywell Safety Manager to disable the Emergency Shutdown (ESD) and Fire & Gas safety systems.

4.2. Wind Power Generation

Wind turbines convert the wind's kinetic energy into electrical energy by means of blades connected to a horizontal or vertical rotor shaft connected to a gearbox driving an electrical generator. Turbines also come with a cooling system, condition monitoring and safety systems, and a wind vane and/or anemometer which serve as input for a controller driving yaw and pitch systems responsible for positioning the rotor blades and the rotor itself based on wind measurements. The turbine as a whole is governed by a DCS- or PLC-based control system. A battery energy storage system (BESS) monitors and controls an array of battery units (based on ultracapacitors, flywheels or Na-S or Li-ion batteries) meant to aid in supply/demand balancing and grid stabilization. Finally, there is an optional on-site SCADA and typically some sort of RTU or gateway connecting the wind park as a whole to the central SCADA system.

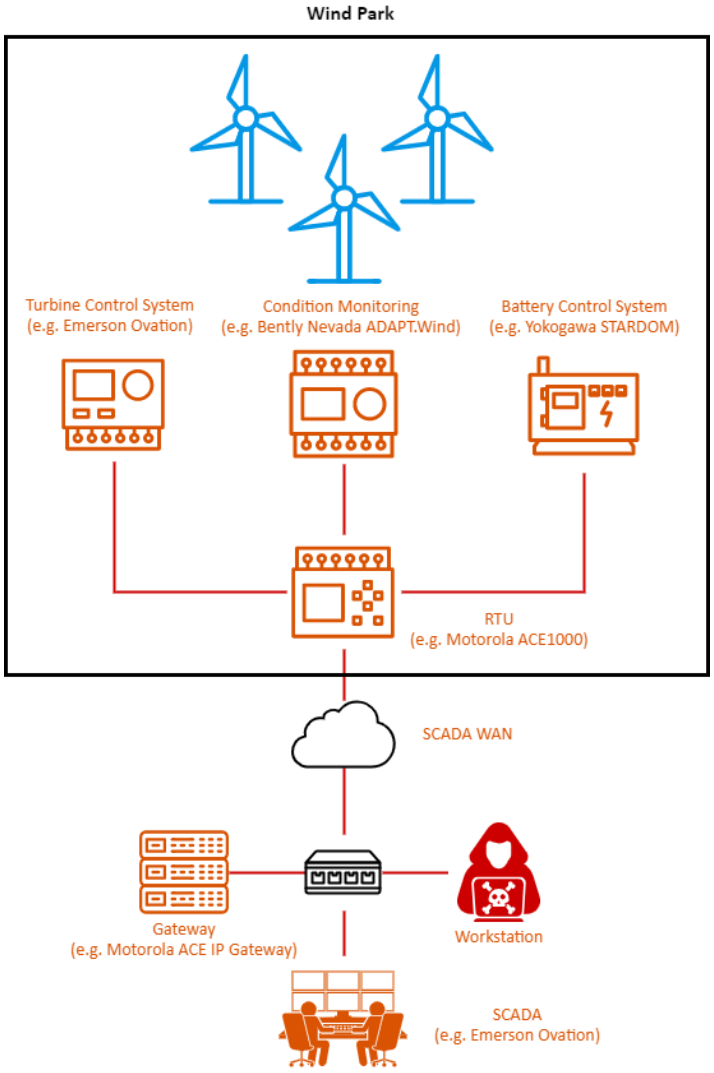


Figure 13 – Wind power generation system

Attack Scenarios: Assuming an attacker compromises the workstation in Figure 13, there are several attack scenarios an attacker could execute by leveraging the vulnerabilities disclosed in this report:

- **Manipulation and Denial of Control / View**: An attacker capable of communicating with the Emerson Ovation SCADA server could, depending on system setup, leverage CVE-2022-29966 to **manipulate system configurations, files or critical values** relating to wind park operations.

  Alternatively, or in combination with the previous scenario, an attacker capable of communicating with the Motorola ACE IP Gateway could abuse the fact that its communications are **unauthenticated** (CVE-2022-30276) and **issue encapsulated MDLC commands** to the RTU, which could halt the logic being executed, as well as sever communications between SCADA and RTUs, denying operators the ability to control and monitor the compressor stations, thus amplifying manipulative actions.

- **Loss of Control / Safety**: As with the previous scenario, an attacker could attempt to gain access to the wind park network by compromising the Motorola ACE1000 RTU leveraging CVE-2022-30270, CVE-2022-30271 and CVE-2022-30269 to **compromise credentials** and **gain code execution** on the RTU. From there, the attacker

could attempt to pivot to the Emerson Ovation control systems of individual turbines. Depending on the segmentation robustness of the DCS, the attacker could laterally move into the DCS internal networks and CVE-2022-29966 and CVE-2022-30267 to **manipulate system configuration, operational settings and controller firmware**. With these capabilities, the attacker could attempt to degrade performance through manipulated yaw and pitch control or could even attempt to disable the overspeed protection functionality integrated into the DCS and disconnect the load to cause damage to the turbine. By leveraging CVE-2022-29952 and CVE-2022-29953, the attacker could **compromise credentials** and possibly **disable** the Bently Nevada condition monitoring systems that would otherwise provide an early warning of dangerous conditions.

In addition, the attacker could target the Yokogawa STARDOM PLC controlling the BESS and utilize CVE-2022-31240 and CVE-2022-31241 to **compromise credentials**, and they could use either FSCT-2022-0039 or CVE-2022-31801 to **gain code execution** on the PLC. With this level of control, they could easily manipulate connected circuit breakers and other battery management functions leading to system downtime and potentially destabilizing conditions.

4.3. Discrete Manufacturing

Figure 14 shows a remotely controlled bottle filling plant, which could be used for instance for medicine or beverage production. Bottles on a belt encounter two liquid filling stations, a mixer and a quality check at the end. Each station has a sensor to detect the presence of the bottle. The filling stations also have actuators to open and close the filling tube valves. At the quality check, the volume of liquid is measured with an ultrasonic sensor. The process is remotely controlled and monitored from the company headquarters through an internet connection (the M2RTU communication channel in the figure) to a Remote Terminal Unit (RTU) at the plant, which connects (using the RTU2PLC channel) to a Programmable Logic Controller (PLC) that drives the actuators and sensors in the physical process (using the PLC2PP channel).

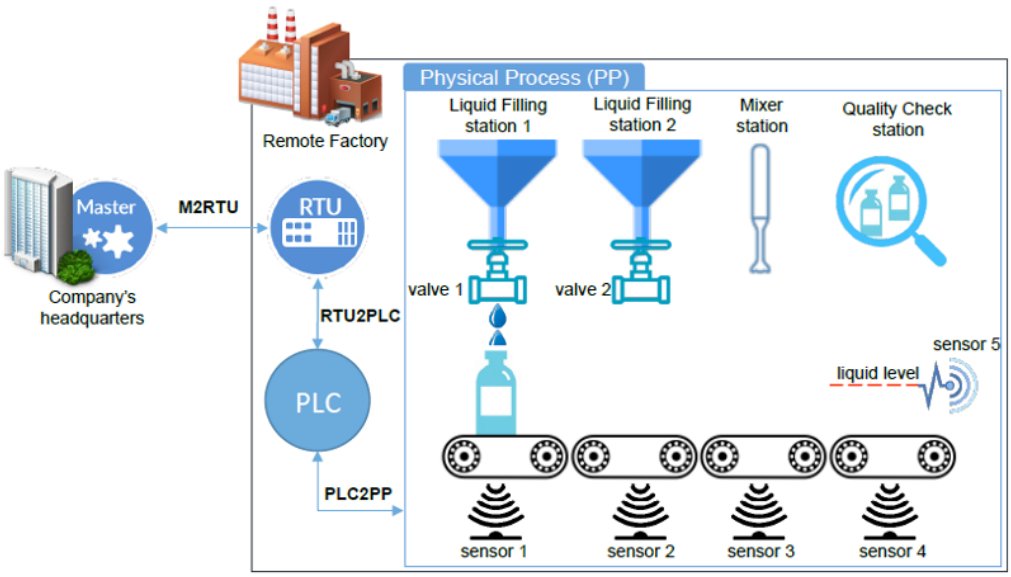


Figure 14 – Bottle Filling Factory

This industrial process comprises the following steps:

- A system operator sitting at the headquarters connects to an HMI such as the one shown in Figure 15, where a process window displays a picture of the system and the states of sensors and actuators. The operator can define the setpoints of the actuators (milliliters for ingredients and seconds for the mixer) and start or stop the remote operation. When the operator sends the setpoints to the RTU and presses the start button, the PLC starts the process.
- When a bottle activates a position sensor, the PLC stops the belt and starts the corresponding actuator (valve or mixer). When the bottle is at an ingredient position, the PLC opens the ingredient's valve and fills the bottle with the quantity defined by the setpoint. When the bottle is at the mixer position, the PLC activates the mixer for as many seconds as defined by the setpoint. When the bottle is at the quality check position, an ultrasonic sensor measures the quantity of liquid in the bottle. The measurement is sent to a SCADA system, which decides whether or not to keep the bottle, according to the setpoints defined by the operator. For instance, if the setpoints for the two ingredients are 40ml and 50ml, then the acceptable measurement is 90ml.
- The process continues indefinitely. Employees go to the remote plant every three days to collect all the bottles produced.

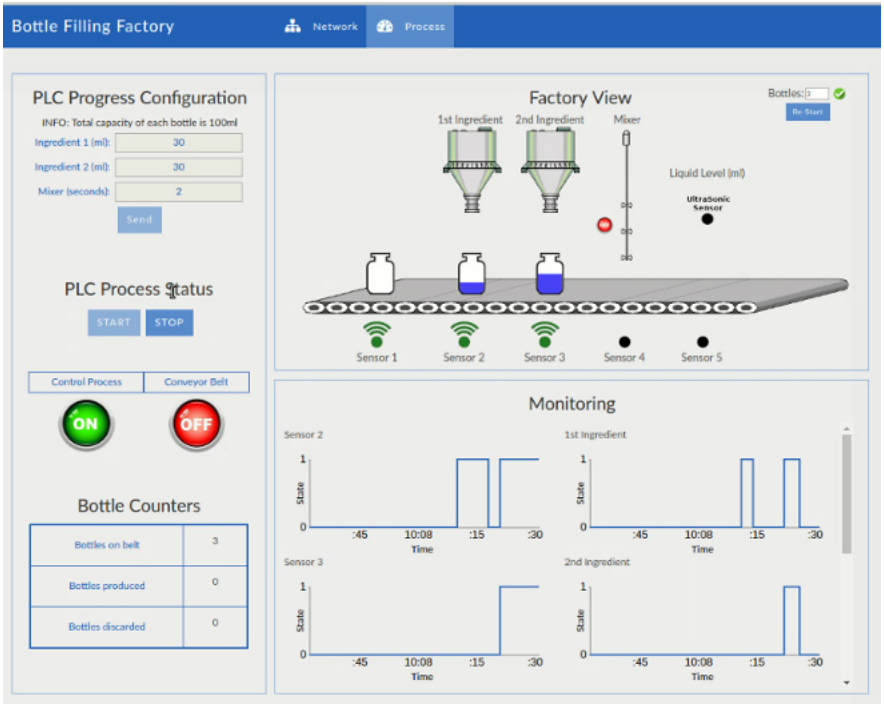


Figure 15 – HMI of the Bottle Filling Factory

Attack Scenarios: The three days unsupervised time span gives an attacker the opportunity to exploit vulnerabilities of the system and to launch attacks that may disrupt the plant production. An attacker can, for instance, stop the running process, subvert the process to produce bottles with too much or too little liquid, or change the ratio of the ingredients since incorrect mixtures could be poisonous. These attacks could be achieved as follows[1]:

- **Loss of Productivity and Revenue**: Exploiting a vulnerability that allows for **Denial of Service** on the PLC can stop the production process until the PLC is active again, since it would not be able to start/stop the conveyor belt or start/stop the actuators.
- **Loss of Control**: Exploiting a vulnerability that allows for **Firmware or Configuration Manipulation** on the

[1] We do not detail the specific devices and vulnerabilities in the manufacturing attack scenario as we did for the previous two, but examples of devices affected by OT:ICEFALL that are popular in this sector include the Omron Nx and Cx series, which are vulnerable to CVE-2022-31206 (RCE) and CVE-2022-31207 (logic manipulation), respectively.

RTU can ignore the setpoints defined by the operator and instead use others silently defined by the attacker. As a result, the PLC receives incorrect setpoints, and the actuators that control the valves of the two ingredients stay open for shorter or longer than they are supposed to. The attacker also can ignore information sent by the PLC to the RTU and always report normal values so that the SCADA system receives wrong process information, and the operator is unaware of the attack.

5. Impact

In this section, we try to estimate the impact of OT:ICEFALL based on the evidence collected during our research using three main sources:

- **Open-source intelligence**. We looked at product documentation, datasheets and marketing information that mention where devices are used and for what purposes.
- **Shodan queries**. Shodan is a search engine that allows users to look for devices connected to the internet. Estimating the number of affected devices based on public data is difficult because these devices are not supposed to be accessible via the internet. However, we are still able to see a few thousand exposed online via Shodan, as shown in Table 13, together with the top countries where they are located and the query we used to find them.
- **Forescout Device Cloud**. Forescout Device Cloud is a repository of information of 18+ million devices monitored by Forescout appliances present in customer networks. We queried Device Cloud for the vulnerable devices and found close to 30 thousand results. Figure 16 shows in which sectors the affected devices are most popular. Manufacturing comes at the top with almost one-third of observed devices. After that, we see healthcare, retail and government mainly because of the presence of building automation controllers since these are industries with many large facilities. We see only a small presence in the OT-intensive oil and gas and utilities sectors, but that is likely because many of those types of customers do not share device information with Forescout's Device Cloud.

Table 13 – Shodan Results for Some of the Vulnerable Devices

VENDOR/DEVICE	QUERY	#RESULTS	TOP COUNTRIES
Honeywell Saia Burgess	http.favicon.hash:-1547576879	2924	Italy (954), Germany (326), Switzerland (263), Sweden (244), France (192)
Omron	port:9600 response code	1305	Spain (321), Canada (113), France (110), US (95), Hungary (70)
Phoenix Contact DDI	port:1962 PLC	705	Italy (285), Germany (104), India (68), Spain (55), Turkey (43)
ProConOS SOCOMM	port:20547 PLC	236	China (65), United States (60), Germany (10), Singapore (10), Hong Kong (8)

Table 13 Continued – Shodan Results for Some of the Vulnerable Devices

VENDOR/DEVICE	QUERY	#RESULTS	TOP COUNTRIES
Honeywell Trend Controls	“trend control”	162	France (74), Denmark (27), Italy (16), Spain (14), UK (14)
Emerson Fanuc / PACSystems	port:18245,18246 product:"general electric"	60	United States (22), Canada (5), Poland (4), Taiwan (4), Spain (3)
Stardom	“stardom”	5	Thailand (2), Egypt (1), Netherlands (1), US (1)
Siemens WinCC OA	"WinCC OA"	1	Austria (1)
Motorola MOSCAD	“moscad”	1	Korea (1)

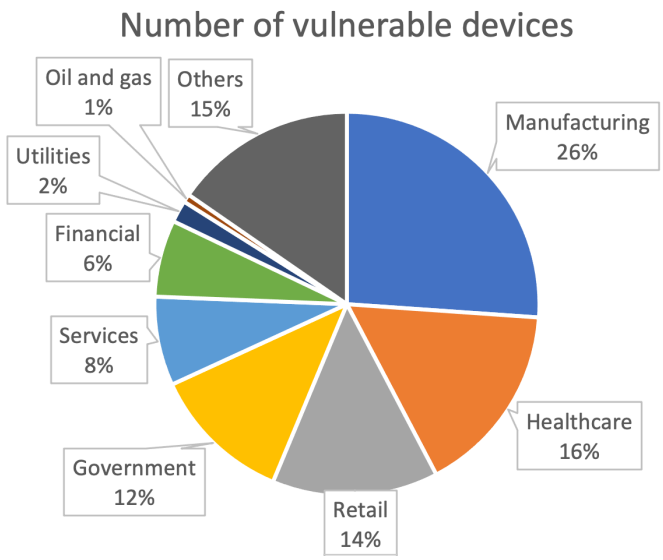


Figure 16 – Number of Vulnerable Devices in Each Vertical as Seen on Device Cloud

6. Mitigation Recommendations

Complete protection against the type of issues uncovered in OT:ICEFALL requires that vendors address these fundamental issues with changes in device firmware and supported protocols, and that asset owners apply the changes (patches) in their own networks.

Realistically, that process will take a very long time. Therefore, below we discuss immediately actionable mitigation strategies for asset owners and system integrators:

- **Discover and inventory vulnerable devices**. Network visibility solutions enable discovery of vulnerable devices in the network and apply proper control and mitigation actions.
- **Enforce segmentation controls and proper network hygiene** to mitigate the risk from vulnerable devices. Restrict external communication paths and isolate or contain vulnerable devices in zones as a mitigating control if they cannot be patched or until they can be patched. Review firewall rules, especially whitelisted OT protocols, against SME knowledge. Some vendors offer dedicated firewalls and switches with protocol-aware security features.
- **Monitor progressive patches released by affected device vendors** and devise a remediation plan for your vulnerable asset inventory, balancing business risk and business continuity requirements.
- **Monitor all network traffic for suspicious activity** that tries to exploit insecure-by-design functionality. Use monitoring solutions with DPI capabilities to alert security personnel to these activities so appropriate actions can be taken.
- **Actively procure for secure-by-design products** and migrate to secure-by-design variants of products where available and when possible. Evaluate device security posture by including security evaluations in procurement requirements.
- **Make use of native hardening capabilities** such as physical mode switches on controllers which require physical interaction before dangerous engineering operations can be performed. Some vendors offer plug-and-play solutions to simulate these capabilities at a network level for multiple controllers. Where possible, activate alerts on operational mode switches into monitoring solutions.
- **Work toward consequence reduction** by following Cyber-PHA and CCE methodologies. This is important to address not only likelihood but also the impact of incidents.

Further general recommended mitigation is available on CISA's Shields Up initiative, Securing Industrial Control Systems publication and list of Recommended Practices. Specific mitigation for each vulnerable device will be provided by the respective vendors.

7. Conclusions and Takeaways

In this report, we discussed OT:ICEFALL, a set of more than 50 vulnerabilities affecting several critical OT devices from major vendors. Exploiting these vulnerabilities, attackers with network access to a target device could remotely execute code, change the logic, files or firmware of OT devices, bypass authentication, compromise credentials, cause denials of service or have a range of operational impacts.

The development of recent malware targeting critical infrastructure, such as Industroyer2, TRITON and INCONTROLLER, has shown that threat actors are aware of the insecure by design nature of operational technology and are ready to exploit it to wreak havoc.

Based on a quantitative analysis of our research and contrary to common perception, we consider it reasonable to assume a small but skilled team with the right incentives would be able to develop at least basic OT Offensive Cyber Capabilities at surprisingly reasonable cost.

Project Basecamp, which initially discussed insecurity by design in OT, was modeled after **Firesheep**, a successful research effort that led many websites to move away from HTTP in favor of the secure HTTPS. OT:ICEFALL shows that despite Project Basecamp's success in raising awareness of the scale and severity of insecure-by-design practices in OT and the past decade of effort by the OT security community, insecure-by-design practices are still very much the norm, and those security controls that have been implemented are often of subpar quality.

In this report, we have also pointed out that despite the important role that standards-driven hardening efforts play in OT security, products with insecure-by-design features and trivially broken security controls continued to be certified. In addition, we have pointed out how the opaque and proprietary nature of many OT systems, coupled with the absence of CVEs, renders many lingering issues invisible and unactionable, leading to unnecessary risk blindness.

As such, we renew the call to action for device manufacturers to properly secure OT devices and protocols, for asset owners to actively procure for secure-by-design products and for the wider security community to ensure that security controls are robust rather than merely functional.

Vedere Labs is committed to finding and disclosing even more vulnerabilities in critical software and devices in the near future. As we did in previous studies, we invite other researchers, device vendors and the cybersecurity community at large to continue this work and collaborate with us in future research.

8. Addendum: Three New Vulnerabilities Affecting Festo and CODESYS

After the publication of the original OT:ICEFALL report, Vedere Labs identified and disclosed three new vulnerabilities affecting OT products from two German vendors: Festo automation controllers and the CODESYS runtime, which is used by hundreds of device manufacturers in different industrial sectors, including Festo.

As in the original 56 OT:ICEFALL vulnerabilities, these issues exemplify either an insecure-by-design approach – which was usual at the time the products were launched – where manufacturers include dangerous functions that can be accessed with no authentication or a subpar implementation of security controls, such as cryptography.

The table below summarizes the newly disclosed vulnerabilities. The disclosure involved the affected manufacturers and the CERT@VDE, a German security platform for small and medium-sized automation companies.

CVE ID	AFFECTED PRODUCTS	TESTED VERSION	DESCRIPTION	CVSS	POTENTIAL IMPACT
CVE-2022-4048	CODESYS V3	3.5.18	CODESYS V3 before 3.5.18.40 uses weak cryptography for download code and boot applications.	7.7	Logic manipulation
CVE-2022-3079	Festo CPX-CEC-C1 and CPX-CMXX Codesys V2 controllers. CPX-CMXX phased out since 2015	CPX-CEC-C1 1.3.10.0.1672	The affected products allow unauthenticated, remote access to critical webpage functions, which may cause a denial of service.	7.5	Denial of Service
CVE-2022-3270	Festo controllers using the FGMC protocol		The Festo Generic Multicast (FGMC) protocol allows for unauthenticated reboot of controllers over the network.	9.8	Denial of Service

These issues are similar to the original 56 in OT:ICEFALL. CVE-2022-4048 is an example of weak cryptography, CVE-2022-3079 exemplifies lack of authentication and CVE-2022-3270 falls in the category of insecure engineering protocols.

We also found that Festo CPX-CEC-C1 controllers and several other Festo devices are affected by known CODESYS V2 vulnerabilities CVE-2022-31806 and CVE-2022-22515 because they are shipped with an unsafe configuration of the CODESYS runtime environment. This is yet another example of a supply chain issue where a vulnerability has not been disclosed for all the products it affects, as we have seen with CVE-2014-9195 affecting ProConOS.

Below, we discuss each new issue in more detail.

8.1. Weak cryptography on CODESYS V3

The CODESYS V3 runtime environment offers application encryption to ensure download code and boot applications are encrypted. This encryption can be facilitated by using the Wibu-Systems CodeMeter security dongle (a popular code protection and license management system among OT vendors), in which case the `EncryptDownloadCode` routine will generate a session key, encrypt it with the dongle key, encrypt the download code with the session key and finally package the encrypted session key and code ciphertext together. This approach suffers from the following issues:

- **Session keys are generated using an insecure pseudo-random number generator (PRNG) working off a small and predictable seed.** `EncryptDownloadCode` generates session keys using the dotnet **Random Class** – which is not a secure PRNG – seeded with the current tick-count, which is predictable and furthermore reduced to 31 bits through a signed integer cast. This session key is then encrypted through interfacing with the dongle component. Next, the session key is used, together with a NULL-IV, to encrypt the download code using AES in ECB mode on a block-by-block basis. Even though the session key is encrypted using the dongle before being shipped with the ciphertext, the session key itself is generated insecurely. Since the effective keyspace is only 31 bits, an attacker can simply brute force the session key to decrypt the downloaded code for manipulation.
- **The encryption scheme uses an insecure mode of operation.** The code is encrypted in ECB mode without additional cryptographic authentication and integrity over the ciphertext as a whole. This means that both confidentiality and integrity are compromised regardless of session key strength. An illustrative example of the weakness of ECB can be found **here**.

The impact is that an attacker can trivially decrypt and manipulate protected code.

8.2. Up to three ways to reboot Festo controllers

We identified and reported three ways to reboot Festo programmable logic controllers (PLCs) CPX-CEC-C1 V2 without authentication. By invoking any one of these functions, an attacker with network access to the target device can cause a denial of service condition:

1. **Accessing the hidden web page cec-reboot.php on Festo CPX-CEC-C1 and CPX-CMXX PLCs.** The web page is not documented but can be found on the controller's filesystem and anyone with network access to a controller can browse to this page, which causes the controller to reboot immediately. This was assigned CVE-2022-3079.
2. **Sending a specific UDP message to multicast group 239.255.2.3 on port 10002 using the Festo Generic Multicast (FGMC) protocol.** This affects several devices that support this protocol, as **listed in Festo documentation**. The same effect can be obtained with the Festo Field Device Tool, which uses FGMC to communicate with the controller. This was assigned CVE-2022-3270.
3. **Using the reboot command via the PLC Browser tool.** **PLC Browser** is a text-based monitor for controllers running CODESYS, which allows an operator to issue several commands to the monitored controllers. Reboot is one of the allowed commands sent via a TCP stream when using PLC Browser. The PLC Browser can issue such commands without authentication by default. This vulnerability was already known and previously described by CODESYS advisories as CVE-2022-31806 and CVE-2022-22515.
   As usual with vulnerabilities on software components ("supply chain" vulnerabilities), there was no indication of which devices were affected by it. Through coordination with Festo and the CERT@VDE, an advisory was published about the effects of those vulnerabilities on Festo controllers and how to mitigate them.

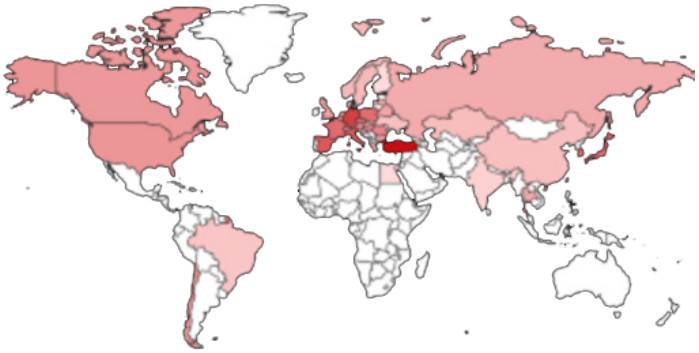
8.3. Distribution of CODESYS and Festo devices

The official website of CODESYS **describes** it as the leading **IEC 61131-3** automation suite, running on several million devices of approximately 1,000 models from over 500 manufacturers. Examples of manufacturers using the technology on their products can be found on this **link**. These devices are **used in industries** such as manufacturing, energy automation and building automation. Although these devices are typically not supposed to be exposed online, we see almost 3,000 devices running CODESYS when querying the **Shodan** search engine ("**port:2455 operating system**"), as shown in the figure below.

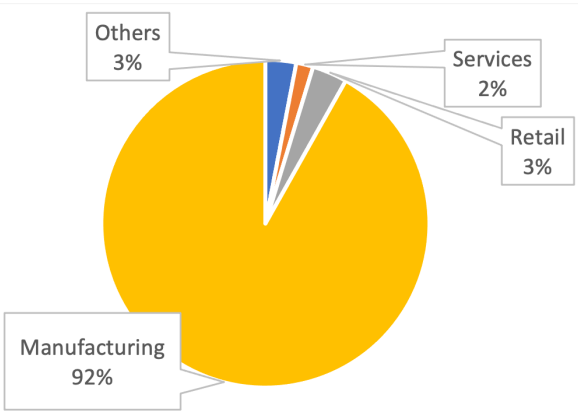
TOTAL RESULTS

2,713

TOP COUNTRIES



Festo CPX is an automation platform for electric and pneumatic systems. CPX-CEC-C1 and CPX-CMXX controllers ran CODESYS V2, while **newer versions run CODESYS V3** and provide capabilities for Industry 4.0, such as remote I/O and cloud connection. On the Forescout Device Cloud – a repository of data from 19 million devices monitored by Forescout appliances – we see close to 1,000 Festo controllers, used overwhelmingly within manufacturing.



8.4. Mitigation strategy: discover, segment and monitor

Complete protection against these vulnerabilities requires patching devices running the vulnerable firmware (CVE-2022-4048), replacing with newer equipment (CVE-2022-3079) or changing default configurations (CVE-2022-31806 and CVE-2022-22515). Vendor-specific mitigations can be found on the published security advisories. General recommended mitigations are the same as those discussed for the original 56 vulnerabilities in Section 6 of this report.

FORESCOUT

VEDERE LABS

Learn more at Forescout.com