Information security aspects of Industrial Control Systems

An introduction for ICS Integrators and Asset owners

Written by: Lars Flå, Eric Törn, Knut Vidar Skjersli and Fredrik Bakkevig Haugli

February 2023

V1.0

Except where otherwise specified, this work is:
Copyright © 2023 SINTEF AS and licensed under a CC BY 4.0 license.

Foreword

With a more connected world, where someone can attack digital industries in distant countries with low risk and cost, cyber security is becoming increasingly important on a nation-wide level. Several recent attacks illustrate this, for example towards the power grid in Ukraine in 2016, the petrochemical plant in Saudi Arabia in 2017, the attack towards a cloud service provider to the train system in Denmark in 2022, and others. These attacks can, if successful, harm important industry and infrastructure such as hospitals, transportation, and industrial manufacturing.

While traditional Industrial Control Systems (ICS) have been mostly isolated, they are increasingly getting connected to the internet, increasing the risks of cyberattacks.

This document intends to look at the state of the art within ICS security and serve as an introduction to ICS security for ICS developers and asset owners, helping them make better choices in their security.

In addition, this document serves as background information for an ICS Guidance document, which aims to provide an easy-to-use security checklist for ICS developers and asset owners.

This document is a deliverable for the project *Ragnarok* at SINTEF, and relates to the *Security for Industrial Control Systems (a guiding document)* document and the IoT checklist document, also a deliverable in this project. In the Ragnarok project we looked at the state of the art in ICS and IoT security, as well as in the future context of Industry 4.0 and the potential collapse of the SCADA pyramid.

Scope

How ICS systems are secured today is the main topic of this document. In addition, we will discuss how ICS attacks are carried out, how ICS security differ from general IT security, what industry 4.0 is, and which security mechanisms and methods that are used today to make ICS more secure. This document does not provide an exhaustive summary of ICS security. It does for instance only provide a very brief discussion of cloud service security, although it could be relevant to industry 4.0. Instead, the document intends to cover the basics and provide ICS developers and asset owners with a foundational understanding.

This document has focused on technical considerations and system architecture regarding security. Aspects of social engineering, while important to IT and ICS security, has been omitted from this work.

Changelog

V1.0

Contents

1 Background

Industrial Control Systems (ICS) is a critical part of modern society. They are distributed computer systems used for managing large production facilities, power plants, factories, petroleum refinement installations, and many other industrial settings where an interruption could result in substantial financial losses, or in the extreme case, human injury, or death.

Traditionally, ICS systems are divided into the control network (often referred to as Operational Technology or OT), the corporate network (information technology or IT), and more recently also a cloud network (the internet). The control network contains the main industrial machines and consists of a diverse set of sensors that monitors the state of the industrial system, a communication network for transmitting states and control commands between devices, control devices such as PLCs or embedded computers for implementing control algorithms, and actuators for implementing the control commands in the physical system. The components in the control network have traditionally been installed in a closed environment with high levels of physical security. An important part of the control network is the control room, presenting the workers with the current state of the system, the history of events, and a user interface for process control.

The corporate network consists of all the office computers, printers, servers, and other IT equipment used by the employees in their day-to-day work. The cloud generally gathers data from the industrial system for use in aggregation with other plants or for third-party applications.

With the flow of data and commands, security is vital to prevent unauthorized people to attack the system, which usually happens through the network or by physical access to a device.

1.1 The evolution of ICS: Industry 4.0

Traditionally, ICS have been running on networks separate from the internet, or at least with several firewalls and DMZs between the internet and the control network. But this is slowly changing. Devices are increasingly having capabilities to connect to internet and wireless communications, giving additional inbound communication vectors to the control system. IT and OT have previously been physically separated by the fact that they involve different equipment, speaking different protocols. However, as industrial equipment adopts modern IT standards for communication and data processing, this separation must instead be actively maintained.

Industry 4.0, sometimes referred to as "the fourth industrial revolution", is a concept that was introduced as part of the industrial strategy of the German government in 2013. It entails an evolution of the third industrial revolution, the digital revolution, which is characterized by using computers and communication technology to control and automate production processes. Industry 4.0 takes this further by increasing interoperability of the different sensors and production equipment and is expected to improve efficiency and flexibility of manufacturing. Connecting devices to the internet and sharing large amounts of data is sometimes referred to as Industrial IoT (IIoT), and are by some seen as a component of or enabler for Industry 4.0 [1] [2].

ENISA [3] identifies a set of cyber security challenges related to Industry 4.0, and categorises them into people, technology, and processes. In the people category, challenges are related to lack of Industry 4.0 cyber security competence, insufficient procedures, and a reluctance to fund cyber security.

In the process category, it is claimed that liability for Industry 4.0 products is poorly defined because of the number of stakeholders and complexity in the ecosystem. The supply chain is another element that is impacted by complexity and the number of stakeholders. The increased complexity can introduce new risks and make existing ones more severe. Another issue is the lack of documents addressing security in a holistic manner.

In the technology category, challenges are related to securing interconnectivity, technical constraints of devices, and to ensuring a common security baseline. Interconnectivity deemed challenging due to diverse types of equipment, a situation that arises when Industry 4.0 solutions are to be interconnected with legacy equipment. This may be further complicated by the use of some proprietary protocols in Industry 4.0. A further challenge is technical constraints of devices, hereunder the lack of security mechanisms and the inability to patch devices over-the-air. Lastly, the document claims that establishing a common security baseline across platforms, devices, protocols, and frameworks is a challenge.

1.2 The evolution of ICS: Cloud Service Risks

An increasing number of Industrial Control Systems are expected to interact with cloud-based systems in several ways. When data can only flow out from the ICS, it will mainly affect the security of the data leaving the ICS, and not the ICS itself (except for the communication interface in use, of course). However, when such a cloud data processing system directly or indirectly can influence the running ICS, e.g., through input/control data from a cloud-based AI data analysis service, then a new set of attack surfaces must be considered, including the ICS input interface and all aspects of the cloud services involved in processing the data used as input to the ICS. A recent example of an ICS that was indirectly influenced by an attacked cloud service, is the shut-down of Danish trains (autumn 2022) due to the (probably economic crime) attack towards a cloud service provider running the back-end of a mobile app needed by the train drivers to access critical operational information.[1] Asset owners relying on cloud services for operating their ICS should therefore take care to also consider the risks of a cyber-attack against the cloud provider.

[1] https://www.securityweek.com/cyberattack-causes-trains-stop-denmark and https://www.digi.no/artikler/tog-over-hele-danmark-stod-i-timevis-hackere-hadde-installert-kryptominer-pa-serverne-br/523748 (Norwegian)

2 How are attacks against ICS systems carried out?

This section briefly describes two past attacks in ICS and analyse them using the ICS Cyber Kill Chain. The ICS Cyber Kill Chain was introduced in 2015 by Assante and Lee [4] to "*help defenders understand the adversary's cyber-attack campaign*" The kill chain is based on the more general Cyber Kill Chain™. The idea behind the ICS kill chain is that an attack can be described as a set of steps or parts, constituting a chain, from planning to a successfully executed attack. The different steps of the Kill Chain for ICS are described as follows:

Planning: The first step is to gather data, often through open-source channels such as Google, Shodan and social media. The goal of this step is to obtain information that can aid the attacker in the further steps of exploiting a system. Examples of such data may be information on features, vulnerabilities, network, accounts, and protocol.

Preparation: Preparation is the second step, which may take the form of weaponization and/or targeting. Weaponization is the process of turning an object, for instance a file, into a carrier for the attacker's malicious content. Targeting is the process of identifying how and where to attack.

Intrusion: The third step, intrusion, is about gaining access to the victim's network or system. This further includes delivery mechanism to interact with the victim, i.e., phishing mail, exploitation, the method used to perform a malicious action. The final sub step is to install malicious content in the victim's environment or modify existing functionality. An example may be to install a remote access trojan.

Management & enablement: In the fourth step, attackers typically set up command and control paths. Multiple paths may be set for redundancy.

Sustainment, development & execution: The fifth step is to exploit the capabilities obtained through the preceding steps. This may include discovering new systems, collecting and exfiltrating data, or installing and executing new capabilities.

Develop: After the sustainment, development and execution step, the development of attacks tailored to ICS takes place. This development normally takes place at the attacker premises using exfiltrated data, obtained in earlier steps.

Test: In this step, the developed attacks is validated to ensure that it has the desired impact. This may involve significant testing on components similar to those in the victims ICS.

ICS attack: In the final step, the actual ICS attack is carried out. For the ICS attack to achieve the desired impact, and attack may exhibit behavior that can be categorized as enabling, initiating, and supporting. As an example, certain states may have to be reached to enable the attack to take place, the attack may be initiated by injecting certain values into the communication and lastly, the impact may be amplified (supported) by hiding the true state of the ICS to operators.

We now analyse two past attacks on ICS using the ICS cyber kill chain method described above. The two attacks are the 2015 attack on the Ukrainian power grid and the 2017 attack on a Saudi Arabian petrochemical plant. For additional attacks analysed using the ICS Cyber Kill Chain, interested readers are referred to the whitepaper on the ICS Cyber Kill Chain [4], where Stuxnet and Havex are analysed.

2.1 2016 Ukrainian power grid attack

Information regarding this attack[2] is mainly based on two reports by Slowik [5], [6] and one report by Cherepanov [7]. The attack resulted in circuit breakers in the grid being opened with subsequent loss of power to consumers on December 17th, 2016. The power was restored manually after one hour. However, a later report suggested that the attack additionally planned to disable a protection relay, but that this part of the attack failed. The consequence had this part of the attack succeeded is hard to estimate but may have included physical destruction of equipment.

Although uncertain, it is suspected that the **intrusion** happened via phishing emails. However, little is known about the **planning** and **preparation** steps, but a few things can be assumed based on the suspected phishing attack. In the **planning** step, the attacker likely gathered information on the company, employees, and email addresses. In the **preparation** step, the attacker likely determined what emails to target and how to weaponize the email content. The **intrusion** into the IT network happened sometime between January and October 2016.

Analysis of the malware used in the attack indicated the presence of both a main and an additional backdoor. The main backdoor included functionality to receive and execute commands from a remote server, realizing the **Management & Enablement** step of the attack. Additionally, the malware embedded a secondary backdoor in Windows notepad. When run, the notepad application would attempt to connect to another remote server, unbeknown to the notepad user.

In the **Sustainment, development & execution** step the attacker is believed to have gathered credentials which were later used for the penetration into the ICS network. Most likely, the attackers would also have had to traverse the IT network to get access to the ICS network, which was breached in early December. It is however noteworthy that the malware did not include functionality to extract data, indicating a motive oriented more towards sabotage than espionage.

The **Development** of the ICS specific malware included a module capable of targeting four industrial protocols used in the power grid. Details regarding the development are unknown. There are however hints towards significant reconnaissance activity in the ICS network between early and mid-December. Although an attack on the Ukrainian power grid one year prior was similar in impact, there are few other reported similarities between the two attacks. Whether lessons from the 2015 attack were used to develop the ICS malware for the 2016 attack remains speculation.

As for the **Test** step, the successful opening of circuit breakers points towards some degree of testing of the malware before it was launched. On the other hand, the unsuccessful attempt to disable the safety relay could indicate that this phase was not sufficiently thorough.

The **ICS attack** itself resulted in opened circuit breakers and loss of power. Regarded in isolation, this can be viewed as the *initiating* step of the attack. However, if we include the attempted disabling of safety relays, the circuit breakers may have merely been the *enabling* step of the attack. In this case the disabling of safety relays and subsequent closing of circuit breakers would have been the *initiating* step. After having executed initiating/enabling steps of the attack, a data wiper module was launched based on a timer. The motive for this was likely to hamper restoration and forensics efforts, and this part of the attack can therefore be described as *supporting.*

[2] Known as *Industroyer* or *Crashoverride*

2.2 2017 Saudi Arabia petrochemical attack

Information on this attack[3] is based on analyses by Johnson et al. [8], the US National Cybersecurity and Communications Integration Center [9], the Cybersecurity and Infrastructure Security Agency [10], and talks by Kling and Forney [11] and Gutmanis [12].

This attack resulted in the emergency shutdown of a petrochemical plant in Saudi Arabia in August 2017. The attack is the first known incident where attackers deliberately tried to attack systems tasked with ensuring the safety of an ICS. The attack failed to achieve physical harm as it triggered a shutdown procedure.

Little is unfortunately known about the initial steps of the attack. Due to the targeted behavior of the malware (i.e. the reverse engineering of a proprietary protocol) we assume a targeted **Planning** step where information on the company, employees and the petrochemical plant was collected.

For the **Preparation** and **Intrusion** steps, a phishing attack may have been a possibility, but this remains unknown. After the attack malware described as unrelated to the ICS malware was discovered in the IT network. However, it remains unknown whether the IT malware originated from the same source or otherwise can have aided in the deployment of the ICS malware. Either way, it does demonstrate the insecurity of the IT network.

The details surrounding the **Management & Enablement** step are unknown. However, the ICS malware did not include functionality to intrude into or propagate through networks. It also does not appear to have had the level of autonomy required to automatically execute attacks, as seen in for instance the Stuxnet malware. One can therefore assume that a command-and-control structure was present.

In the **Sustainment***,* **Development** *&* **Execution** step the attacker pivoted from the IT network to the ICS network via the DMZ. While secure on paper, insecure firewall rules enabled this intrusion. It is also not unlikely that that the attacker would have had to move laterally from the initial IT intrusion point to the DMZ entrance point, this however remains unknown. Once in the ICS network, the attacker may have had to move further through the network to reach the safety controller. To what extent is unknown, but there are indicators hinting towards the safety PLCs being connected to the same network as the control system. If true, this would have made it easier for the attacker.

The details surrounding the **Development** step are unknown, apart from the attacker's successful reverse engineering of a proprietary protocol for safety PLC-communication and exploitation of zero-day vulnerability to elevate privileges on the PLC.

The safe shutdown of the plant could indicate that the attack was not sufficiently **tested**. Analysts believe that the goal of the attack was to reprogram the controller to allow for the plant to enter an insecure state, as a safe shutdown could have been caused in easier ways than the one used by the attack. A few months before the attack, the plant also experienced a safe shutdown initiated by a safety controller, further supporting the claim of insufficient testing.

One can argue that the goal of the **ICS attack** was to *initiate* a shutdown of the plant. However, more likely is that the original plan was for it to have a *supporting/enabling* role, in that a disabled safety system would have greatly amplified the consequences of a later attack on the control system.

[3] Known as Trisis, Triton or HatMan

2.3 Attack Classification

In this section we will discuss attack vectors, and vulnerable components.

Attack vectors of an ICS include [13] :

- **Backdoors and holes in the network perimeter.** This is often an IP:Port connection that is open to outside communication. A hole is an IP:port route that is not blocked by the firewall, and a backdoor is a hidden software process that implements a remote shell.
- **Vulnerabilities in common protocols and software.** This includes both industrial control protocols and higher-level OT/IT protocols. All protocols and their implementations may contain bugs that can be exploited. Classic low-level industrial protocols have low or no security.
- **Direct attacks on field devices**. This includes physically accessing PLCs, embedded devices, and other devices in the control network, and performing various attacks directly (not over a network)
- **Service attacks**. This includes denial of service attacks, gaining access to sensitive data (e.g., from database), passing malicious inputs to services to cause some side-effect, and other.
- **Communications hijacking and 'man-in-the-middle' attacks**. Where the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection.
- **Spoofing attacks**. This is an attack where you are pretending to be someone you are not, e.g., to gain data or send control commands.
- **Attacks on privileged and/or shared accounts**. If a privileged account gets compromised, that is an easy way to access the ICS.

And the vulnerable components that are attacked include (not an exhaustive list):

- **Hardware.** An attacker normally needs physical hardware access to enable unauthorized probing, monitoring and control, but there are also situations where attackers can utilize e.g., electromagnetic fields or sound waves from a distance.
- **FPGA configuration.** Some systems require hardware access (see above) to attack FPGAs, but others enable updates via software and are thereby more vulnerable.[4]
- **CPU microcode.** CPUs with complex instruction sets are often internally implemented with microcode that defines how each CPU instruction behave. Often, such microcode is fixed/immutable by design, but there are CPUs that also support updating sections of microcode, e.g., Intel introduced an option to update the Pentium Pro microcode in the 1990s. Attacking microcode has been demonstrated.[5]
- **Firmware.** Updating firmware has for many years been one of the important security measures against attacks that exploit known flaws. However, this option is then also vulnerable for unauthorized updates to inject malicious code at firmware level.
- **Persistent storage of configuration and/or data**. Any stored configuration or data can be of interest to an attacker. Resource names, usernames, password hashes, encryption keys are all well-known examples on vulnerable information an attacker can exploit to continue or escalate an attack, but other internal information might also be at risk. Both unauthorized reading of such information and unauthorized injection of false information are risks to be aware of.

[4] https://ieeexplore.ieee.org/abstract/document/6461919

[5] https://hackaday.com/2017/12/28/34c3-hacking-into-a-cpus-microcode/

- **Operating system and privileged drivers.** These components often have higher access rights than other parts of the system, and thereby more severe consequences in case of a breach. Modern operating systems are normally set up with automatic security updates, and the admins should maintain its configuration according to the needed security level. However, some systems might deviate from this due to high up-time requirements (inhibiting automatic updates) or restrictions by the system supplier/developer (e.g., some component not yet compatible with the latest OS and driver versions). Encapsulating insecure systems with other countermeasures might then be needed.
- **Application/service software.** Vulnerable software needs much of the same concern as operating systems above, but security updates might be missing or less frequent – depending on the supplier/developer. Admins therefore should scan for outdated software that might need to be replaced or deleted.
- **File system or database storage of configuration and/or data.** As for vulnerable configuration/data mentioned above, such internal information is also vulnerable when stored in file systems and databases. The risk level might even be higher because file systems and databases often can be accessed from a higher number of devices.
- **Removable media.** As with any IT system, there is a risk of executing unknown code when connecting CDs/DVSs, USB drives, SD cards, and similar that can carry unauthorized code to silently help an attacker.
- **Network interfaces.** As with any IT system, only strictly needed interfaces should be allowed. Admins can monitor traffic and/or scan for open interfaces to detect unsecure issues.
- **Communication between components.** This can include unauthorized listening, denying legit messages, and injecting unauthorized messages. Communication via wireless or external systems are of course more vulnerable, but internal communication is also at risk when an attacker already has succeeded with the intrusion step.
- **Users.** Typical social engineering, and convincing naive users to help an attack by clicking links or accepting attachments constructed by an attacker.

3 How does ICS security differ from general IT security?

An important difference between ICS and IT security is that the high-level objectives of cyber security in OT are different compared to those in IT. While IT tend to prioritize confidentiality, integrity and lastly availability, ICS typically regard availability and integrity as more important than confidentiality. Some add the additional property of control and gives this the top priority for ICS security. Control refers to the ability to change the state of the system in a safe and secure manner [14].

A consequence of these different priorities is that ICS protocols and equipment traditionally were designed with reliability as the focus. What we however observe is that many legacy devices and protocols are insecure by design. This was initially acceptable as ICS systems used to be isolated and inaccessible. With the increases in digitalization this isolation faded and is expected to disappear with the introduction of industry 4.0. Another trend, according to NIST [13], is the use of more low-cost devices supporting the Ethernet and IP protocols instead of the proprietary devices and protocols traditionally in use.

NIST [13] has created a list of areas where ICS and IT security differ, most of which are rephrased here:

- **Timeliness and availability requirements:** As ICS control the physical world such systems often have more stringent requirements on delay, jitter, and availability than IT systems. This must be considered when securing an ICS, both with regards to threats and countermeasures. The threats of denial-of-service attack by generating large volumes of traffic may be even more critical to an ICS given their lower tolerance for delay and jitter. Regarding countermeasures, application-level firewalls may introduce unacceptable delay to critical communication. Requirements for high availability of ICS systems may result in patches being deployed later as downtime may have to be planned days or weeks in advance. Even the shutdown and restart procedures may in some industries take days [14].
- **Physical effects and safety:** ICS control processes in the real world and cyber-attacks can result in physical damage to personnel and property. These processes can be complicated and understanding these consequences may require communication with experts in control systems and the respective physical domain. Due to the risk of physical damage ICS often include dedicated systems for process safety. Security countermeasures must not negatively affect safety systems in any way.
- **Constrained resources:** ICS often lack common security solutions found in IT systems, such as cryptographic capabilities, error logging, and protection of passwords. Current resources may not be sufficient to retrofit such solutions and adding new resources may not be feasible.
- **Communication:** Protocols are generally different than the protocols found in the IT domain, and some protocols may be proprietary. Proprietary protocols may make it hard to determine what is legitimate traffic, which can be a problem for Intrusion Detection Systems [15].
- **Change management:** The update of security issues (patching) is among the most important measures to ensure a secure system. Due to the nature of ICS, several challenges to the timely patching of software arises. Patches must be thoroughly tested before they can be deployed, sometimes by both the patch provider and the operator (This is an approach suggested by IEC 62443-2-3). Patching could prove difficult if the operating system used in the ICS is no longer supported by the vendor. While IT components typically have a lifetime of 3-5 years, ICS components can be in use for 10-15 years, and sometimes longer. Lastly, after applying a new patch, parts of the ICS may have to be revalidated to ensure compliance with relevant standards. Because of the longer time it takes to patch ICS, exploits in ICS can have a lifetime of months or years, according to GE [14].

- **Managed support:** ICS support is sometimes through a single vendor, and solutions may not be interoperable with that of other vendors. In some cases, installing third-party security solutions may not be allowed due to license and service agreements.
- **Unique solutions:** SINTEF [16] argues that ICS are often unique, and that the vendor of the system is often the one who is best suited to maintain it.

While there are differences related to securing ICS and IT, there are also similarities. In 2021, SINTEF [16] released a report evaluating the use of the Norwegian National Security Authority's foundational principles for IT security in ICS for petroleum. This document groups 118 controls into the four categories of identify, protect, detect, and manage and restore. Of these, 96 controls were classified as fully relevant for ICS, 18 were classified to be relevant, but in need of special care, and only 4 were classified as not relevant. This illustrates that even though ICS and IT is different, many good practice controls from IT can still be applied to ICS. Further supporting this, according to General Electric [14] the company Gartner has a rule of thumb that 80% of security issues faced by ICS are almost the same as those in IT.

SANS [4] has adapted the general Cyber Kill chain for ICS and highlight the difference between attacks in ICS and IT. ICS are often designed and configured in unique ways based on the process it controls and an attacker needs detailed knowledge of the system to impact it in a desired way. Furthermore, properly designed ICS have several layers that an attacker must traverse to access the components. Increased use of Internet connectivity, as can be expected in Industry 4.0, reduces this advantage.

4 Existing security mechanisms and methods

This chapter starts by categorizing ICS security into different areas of security measures. Then we look at what mechanisms and methods that are used to implement security within each of these areas. These areas are:

- **Segmentation and Segregation.** Setting up isolated or semi-isolated networks with different security clearances.
- **Boundary protection**. Controlling what types of communication that can transfer between different security domains.
- **Authentication and authorisation.** Ensuring that only certain users can access and do certain things within the ICS network.
- **Confidentiality**. Protecting data, making it sure it is not leaked.
- **Data integrity.** Detecting or making sure that the data is not tampered with.
- **Data and service availability.** Making sure that data and services is available in the face of attacks.
- **Communication protection.** Protecting the communication that goes over and between networks. Protecting against attacks like man-in-the middle, spoofing, eavesdropping, or other protocol and communication exploits.
- **Monitoring and auditing**. Recording and checking events to detect unexpected events and being able to backtrack attacks.

We will now look at the security mechanisms that exists under these areas. At the end of this chapter, we will look at security features on PLCs and microcontrollers.

4.1 Segmentation, Segregation, and Boundary Protection

Properly segmenting and segregating your network is fundamental and the first step to a secure ICS system. System architects needs to decide which network domains are permitted direct communication, what policies that govern that communication, what devices/mechanisms that implement the policies, as well as the trust relation between domains.

Typically, an ICS network consists of at least one control network and one corporate network, where the control network requires high amounts of security. The control network is where robots, machines, PLCs and similar devices lives and communicate. Internet access by devices on the control network should be strongly discouraged.

The network setup should use the principle of least privilege[6]: if a system doesn't need to communicate with another system, it should not be allowed to. As a rule, enable communication routes only as it becomes needed. In your firewalls, use whitelisting instead of blacklisting.

The security of a network can greatly be improved by using a DMZ. If this is used, all traffic should terminate in the DMZ. Improve security further by using different protocols between the corporate network and the DMZ, and the DMZ and control network. Place computers and systems that is accessed by both corporate and control network within the DMZ (e.g., a historian). If you don't use a DMZ, at the very least you should only allow connections to be established from control network to the corporate network.

[6] https://en.wikipedia.org/wiki/Principle_of_least_privilege

Mechanisms and methods:

- **Separation of networks** based on security needs**.** E.g., between the internet and the corporate network, and the corporate and control network. The following mechanisms and methods are often used:
    - Whitelisting
    - Network traffic filtering (IP, ports, states, and protocol)
    - Application-level filtering and packet inspection. E.g., email format and protocol checker.
    - Application-proxy gateway firewalls. A proxy gateway translates the full protocol message, up to application format/protocol, and makes a new connection on behalf of the client.
    - Enforcing secure authentication between two parts using certificates.
    - Monitoring of network traffic. Intrusion detection and alarms.
- **Disabling protocol feedback to senders**. Feedback on protocol communication to the sender can give information to an attacker. For example, it can give the attacker information on the types of protocols the target device supports, the services it supports, and what ports that is open for communication. The disadvantage with disabling feedback is that feedback is useful also for real users in the network. An option is to configure feedback only to apply to connections coming from within the network
- **DMZ**[7]**.** For limiting direct connections between two networks.
- **Proxy server** for incoming and outgoing connections.
- **Unidirectional gateways.** This is a solution that only allows data to flow in one direction. Using it can make it harder for an attacker to gain access to a network.
- **VLAN.** To separate networks further, requiring the traffic to go through a gateway with a firewall and authentication. Users can be separated into different security levels.
- **VPN.** For accessing a secure network remotely.
- **Concealing network addresses of ICS components from discovery**. This requires prior knowledge for access. E.g., not having a device's address in domain name systems.
- **MAC address locking** and static ARP tables. To counter man in the middle attacks.

4.2 Authentication and Authorization

An authentication and authorization system is equally fundamental as segmentation and segregation. Only users with proper authority should be able to access certain parts of the ICS.

It is common to use a centralized authentication server. But while a centralized approach provides substantially improved scalability, it also presents numerous additional concerns that may impact its use in ICS environments. The following considerations apply [13].

- A single system that is responsible for managing all system accounts and must be highly secured.
- The authentications server system requires high availability because its failure may prevent users from authenticating to a system during an emergency. Redundancy may be required.
- Networks used to support the authentication protocol must be reliable and secure to ensure

[7] The DMZ functions as a small, isolated network positioned between two other networks, such that all traffic between the two networks must go through the DMZ.

authentication attempts are not hindered.

Mechanisms and methods:

- **Authentication server** with digital signatures and public/private key encryption.
- **Multi-factor authentication**
- **Authentication and access on the middleware level**: e.g., DDS Security or OPC UA Security
- **Remote access solutions should have stringent security requirements** multi factor authentication, re-authenticate when accessing OT, access only allowed when a work permit is issued, have an additional person monitor the actions being taken.

4.3 Confidentiality

Getting access to sensitive data should be hard, and even if someone gets unauthorized access to data or devices, there should be measures such as encryption in place to make it harder to use the data in a meaningful way.

It is helpful to separate data into different security levels, where sensitive data has limited access. Accesses are then given for limited periods of time, with documented details about who and what was accessed. There should be regular reviews to close access when it is no longer needed.

Mechanisms and methods:

- **Encryption of data in databases**
- **Encryption of hard drives**. E.g., BitLocker for Microsoft, or similar software for other OSes, where the hard drive is continually encrypted.
- **Using authentication and access roles** to limit access to data.
- **Using time limits on access**. Access should have to be renewed periodically.
- **Protect data in transit** by using secure protocols/middlewares such as DDS Security or OPC UA Security, and TLS for TCP/IP.

4.4 Data integrity

There should be mechanisms in place to check for data integrity, to make sure that the data that is used for system functionality or other uses have not been tampered with. Tampered data can, for example, change how a system behaves. To counter this, checksums can be used to check for unauthorized changes to files and binaries. During development and commissioning of the ICS, acceptance test can also be used to check if the system has the correct behaviour.

Mechanisms and methods:

- **Checksums**
- **Checking data for coherence with acceptance tests**
- **Trusted zones** on processors/microcontrollers (only processes within the trusted zone can change the data)

4.5 Communication protection

Communication over a network can be exposed to several threats, including:

- Unauthorized subscription to sensitive messages.
- Unauthorized publication of harmful messages.
- Unauthorized access to data or services
- Message flooding
- Eaves dropping
- Message spoofing
- Message alteration and replay
- Malformed messages
- Man in the middle

To counter these threats, it is important to use a secure protocol. Popular protocols for ICS are OPC UA (with security features) and DDS (with security plugins). Transport Layer Security is also used for providing encryption and certificate security to TCP (and DTLS for UDP).

In addition to using a secure protocol, there are techniques such as MAC address locking and static ARP tables that counters man in the middle attacks. Without these, it is possible for a skilled attacker to perform such attacks by plugging into the network through a switch and spoofing ARP messages to manipulate ARP tables.

Mechanisms and methods:

- **User authentication and Access control**. Making participants on the network authenticate themselves and give users access to different types of messages depending on its security clearance level. Either with certificates or by shared secrets.
- **Encrypted communication**.
- **Logging and monitoring**. It should be possible to monitor all the traffic over the network in each protocol. E.g., ARP monitoring with ARPwatch.
- **MAC Address locking**. Locks a specific MAC address to a specific physical port on a switch. This prevents someone to plug in devices with unauthorized MAC addresses.
- **Static ARP tables**, to prevent spoofing in ARP resolution, when an attacker links a servers IP address to its own MAC address.

4.6 Monitoring and auditing

Being able to monitor and audit traffic is important not only for detecting on-going attacks, but also to be able to backtrace attacks after the fact so that attack can be understood and security weaknesses updated.

Monitoring systems should be tested periodically.

Mechanisms and methods:

- Using a monitoring software system to detect abnormal situations, alert security personnel, and even shut down certain services. There exist monitoring solutions for many protocols.
- Log traffic for auditing and back-tracing of attacks.
- Watchdogs to detect compromised devices or intruders.
- A network Intrusion Detection System (IDS) is often placed on a subnet that is directly connected to a firewall so that it can monitor the traffic that has been allowed and look for suspicious activity.

4.7 Security mechanisms in control devices

4.7.1 PLC

Traditional PLCs have limited security features, as they were made with the presumption that the control network is secure. However, they do have features such as username/password login.

However, next generation IIoT PLCs, such as the PLCNext[8], are running general-purpose OSes like linux with a TCP/IP stack, giving capabilities to connect to the internet. This gives attackers additional attack vectors for accessing the ICS.

With these new abilities comes a need for being more careful about firewalls, frequent patching, monitoring, and disabling unused features.

4.7.2 Microcontrollers

Microcontrollers are used in many settings and are more general-purpose than PLCs. While the main theme of this document is about making ICS secure from attacks, security features on microcontrollers can help limiting the effects of an attack if a device gets compromised. For example, by using secure zones for sensitive data that is vital for operation, secure boot for limiting backdoor malware, and MPU and internal firewalls to limit the memory a process can access.

Here are some examples of security features that can be used on microcontrollers to make them more secure (not an exhaustive list):

- **Secure zones** within a single core. For example, Arm cortex processors with TrustZone run a secure OS and normal OS from a single core. Non-trusted software is blocked from access to the secure side and the resources that resides there.
- **Secure boot**. This protects your device from malware by, before booting the OS, verifying that a trusted authority has signed the software you are running in the OS.
- **Internal customizable firewalls** between components on the microcontroller. For example, the TI DRA821U and its siblings are able to configure what parts of the microcontroller that can access certain features and locations on the board.
- **Memory Protection Unit (MPU)**. With MPU you can assign specific accessibility rules to processes and certain ranges of memory.
- **Hardware-based encryption**: Random Number Generator (RNG), AES, SHA-2, SHA-3, Public Key Accelerators (PKA)
- **True random number generation** (TRNG)

Different types of microcontrollers have different security features, and security needs should be evaluated before choosing the microcontroller.

Microcontrollers are often used for IoT purposes. For a more in-depth look at security measures for IoT, take a look at the *The SINTEF IoT Security Checklist*[9].

[8] https://www.phoenixcontact.com/en-pc/industries/plcnext-technology

[9] https://www.sintef.no/contentassets/8fa5c7e3a81749b8952979000ee34c31/iot-security-checklist-v1.1.0.pdf

5 Analysis of relevant protocols

5.1 DDS

We chose to feature Data Distribution Service (DDS)[10] because it is a widely used and growing protocol/middleware used for communication in safety-critical systems. It has strict requirements regarding security. DDS can provide fine-grained security control both within an internal ICS system and for systems/networks connected with a router or gateway.

DDS is a data-centric peer-to-peer middleware that is used in mission- and safety-critical real-time systems, as well as systems with many devices that may jump in and out of the system network. The middleware handles all connections and addresses, connecting an application that publishes data to the applications that want to consume that data.

Security is built into DDS and the DDS Security specification[11] defines the security model for DDS, which includes five security plugins that can be implemented:

- **Authentication:** Provides the means to verify the identity of the application and/or user that invokes the operations on DDS. Includes facilities to perform mutual authentication between participants and establish a shared secret.
- **Access Control:** Provides the means to enforce policy decisions on what DDS related operations an authenticated user can perform. For example, which domains it can join, which topics it can publish or subscribe to, etc.
- **Cryptography:** Implements (or interfaces with libraries that implement) all cryptographic operations including encryption, decryption, hashing, digital signatures, etc. This includes the means to derive keys from a shared secret.
- **Logging:** Supports auditing of all DDS security-relevant events.
- **Data Tagging:** Provides a way to add additional labels to data. This can be used for e.g.: specifying classification levels of the data (as a complement to access control) or message prioritization.

5.1.1 DDS Threat model and counteractive measures

The DDS specification lists the threats that impacts an application that uses DDS and its underlying wire protocol Real-Time Publish Subscribe (RTPS). There are mainly four categories of threats:

- **Unauthorized subscription:** An eavesdropper who is connected to the same network and tries to see or subscribe to data that she is not authorized for.
- **Unauthorized publication:** An intruder that is connected to the same network and tries to send data over the network without being authorized to do so.
- **Tampering and replay:** Someone who is authorized to subscribe to a topic, but not allowed to publish to that topic. This person will try to use the data gained from the subscription to publish malicious data to the network (and to convince subscribers that she is a legitimate publisher).
- **Unauthorized access to data by infrastructure services:** This can be gateways that relays messages from one network to another. These gateways should not be able to understand the contents of the message.

The DDS Security specification lists the following measures for securing against these threats:

[10] https://www.dds-foundation.org/

[11] https://www.omg.org/spec/DDS-SECURITY/

- **Measures against unauthorized subscription:** Encryption of messages using a secret key that is only shared with authorized receivers.
- **Measures against unauthorized publication:** Require that the messages include either a hash-based message authentication code (HMAC) or digital signature. An HMAC creates a message authentication code using a secret key that is shared with the intended recipients. When an unauthorized publisher sends a message, the HMAC of that message will not be recognized by the receivers. A digital signature is based on public key cryptography. To create a digital signature, a digest of the message is encrypted using a private key (by the publisher). Everyone has access to the publisher's public key**.** The recipients can identify messages from an authorized publisher by interpreting the encrypted message with the sender's public key. HMACs are often preferred because their computation is about 1000 times faster than the computation/verifying of digital signatures.
- **Measures against tampering and replay:** In this case, a person has the secret key to an HMAC and can read messages but should not be able to publish any messages. If the malicious person gets a message from Alice, this person could pretend to be Alice by using Alice's secret key to encrypt a new message and manipulate the headers to make it look like Alice is the source of this new message. To counteract this, *a different secret key is shared with all recipients,* such that each message to each recipient is encoded with a different HMAC*.* If a digital signature is used, this problem does not arise. However, due to the computation cost, it is often better to compute and send different HMACs to all recipients than to use digital signatures.
- **Measures against unauthorized access to data by infrastructure services:** A relay device needs to be authorized as a valid destination for a message, but not to be able to read the message itself. A measure for this is to use two secret keys, one to authorize a recipient and one to encrypt the message/data. Only the first secret key is shared with the relay device. The device also needs to be accepted by others as a source of messages, and thus needs its own secret key for an HMAC, that is shared with recipients of relayed messages.

Read the DSS Security[12] specification for more information about how these security plugins work and how they are implemented.

5.2 OPC Unified Architecture (OPC UA)

Another protocol we decided to include is OPC UA, which is widely used in industrial automation plants in Europe. OPC UA can be used to communicate from PLCs to OT/IT networks and all the way up to cloud.

In addition, OPC UA it is often touted by OPC UA ambassadors as a secure middleware that is the best alternative for IIoT and Industry 4.0. The German state-funded organization *Platform Industrie 4.0*, which develops an Industry 4.0 reference architecture model (RAMI), cites OPC UA as the approved standard for RAMI's communication layer. However, opponents to OPC UA typically points out that OPC UA is too large and complex, and that most implementations of OPC UA only implements parts of the standard. And that two implementations aren't guaranteed to be able to work together, even if they both pass the OPC Foundation compatibility test. This may not be a concern for large industry players who have the resources required to implement the standard.

OPC UA security operate on three main concepts:

[12] https://www.omg.org/spec/DDS-SECURITY/

- **Trusted information (CIA Triad)**
    - *Confidentiality*, by encrypting messages on the transport layer
    - *Integrity and authenticity*, by signing messages on the transport layer
    - *Availability*, by minimizing the message processing done before authentication.
- **Access Control (AAA Framework)**
    - *Authentication* by username and password or X.509 certificate on the application layer
    - *Authorization* to read, write values of a node or to browse the information model based on the access rights of the information model, access rights of the user or of the user's role
    - *Accountability*, by generating audit events for security related operations.
- **Defense in depth**
    - By using multiple security mechanisms instead of relying on one.

5.2.1 OPC UA security model

The OPC UA security architecture works on three layers, designed with the defense-in-depth concept in mind:

- **User security.** This can be a simple username and password or multi-factor authentication.
- **Application security.** This is part of the communication session and authenticates the application itself. It includes the exchange of digitally signed X.509 certificates and establishes a secure channel.
- **Transport security.** This can be used to sign and encrypt each message during a communication session. Signing ensures the message's integrity and authenticity, while encryption prevents eavesdropping. Transport Layer Security (TLS) is used for this. The encryption key is rotated after a certain amount of time.

For pub/sub security, OPC UA uses session keys that are shared between publishers and subscribers, where the keys are managed for a security group and messages are sent in the context of a security group. Key distribution is done with OPC UA client-server security mentioned above. Authentication and authorization during access to a security group is done at a key server. The payload may be encrypted.

There are different attacks that has been defined and counteracted by OPC UA:

- **Message flooding** is countered by minimizing processing of packets before they are authenticated.
- **Eavesdropping** is countered by encryption.
- **Message spoofing** is countered by message signing, valid session IDs, channel IDs, and timestamps
- **Message alteration and replay** is countered by session IDs, channel IDs, timestamps, sequence numbers, and request IDs
- **Attacks relying on malformed messages** are countered by validating message structure and valid parameter values

5.3 MQTT and SparkPlug

MQTT Sparkplug is an interoperability specification that uses MQTT as the transport middleware. SparkPlug was made for IIoT, smart manufacturing, and industrial automation use cases, where large amounts of devices exchange data. Sparkplug provides "plug-and-play" interoperability by defining a consistent way for equipment manufacturers and software providers to share contextual data and states. It is a more lightweight OT/IT middleware than OPC UA or DDS, but it is currently not as widespread, and fewer device implementations exists.

MQTT Sparkplug defines no security protocol itself, and instead lets you handle security at the TCP/IP level. So, you can choose to implement TLS, for example, and then run MQTT with SparkPlug over it. With MQTT, there is no need to open inbound communication ports for devices in the control network, so control/edge devices can publish data to the MQTT broker without exposing themselves for incoming attacks.

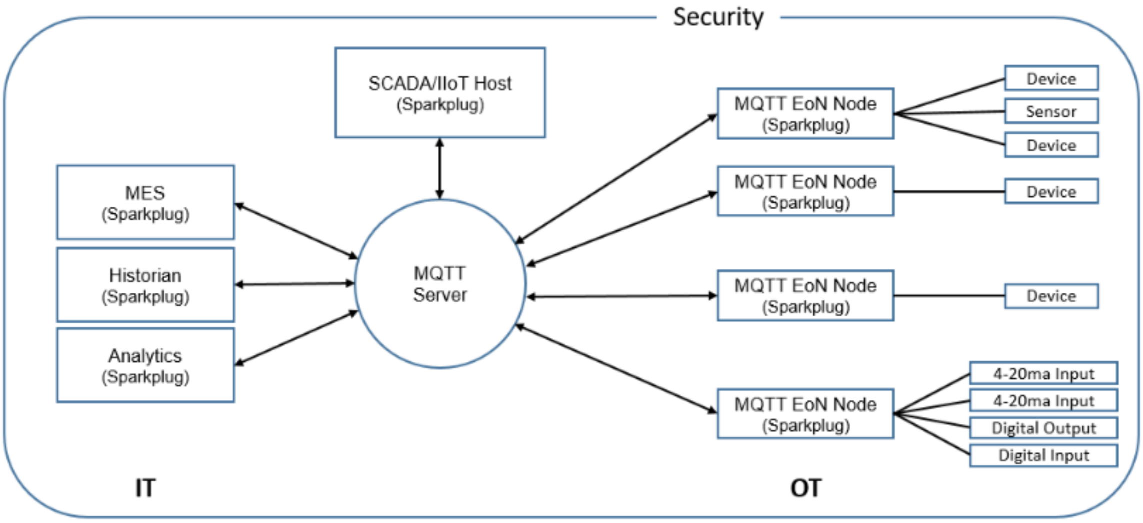

Figure 1: MQTT Sparkplug uses a broker and is secured by using TLS. Image source: Sparkplug Specification V 2.2[13] *by Eclipse Foundation (Copyright © 2019 Eclipse Foundation, Inc. https://www.eclipse.org/legal/efsl.php)*

5.4 Robot Operating System (ROS)

ROS (also known as ROS 1) is a distributed system providing mechanisms for distributing information between nodes in a network. It is a popular choice for connecting all components when building a robot system. Any unauthorized agent (potential attacker) that can access the ROS master, is able to both read/monitor any value and even inject false information.[14] Countermeasures against this, are:

- Restricting network access using a firewall or full network isolation.
- Use extensions to ROS that restrict the access to ROS resources.

Examples on the latter are:

- **Rosbridge.** This is a WebSocket interface to ROS and a server that enables applications in an insecure network zone (e.g. Internet) to access a limited set of topic values in a ROS environment without exposing everything else.
- **SROS.** This is an experimental extension that adds TLS support for ROS transport mechanisms, x.509 certificates for chains of trust, etc.

[13] https://www.eclipse.org/tahu/spec/Sparkplug%20Topic%20Namespace%20and%20State%20ManagementV2.2-with%20appendix%20B%20format%20-%20Eclipse.pdf

[14] http://wiki.ros.org/Security

- **ROS 2.** This is the second generation of ROS. It has a middleware based on DDS[15] that uses the DDS-Security standard for authentication, authorization, and encryption mechanisms, etc.

6 Conclusion

This document looks at current ICS security: how attacks are executed, mechanisms that are used to secure ICS, and highlights to a minor degree potential challenges for industry 4.0.

With the increase of ICS connected to the internet, there is a clear need for having a plan for security, and to architect the system with security in mind. While this document is not as comprehensive as reports such as NIST Guide to Industrial Control Systems or standards such as IEC 62443, it provides an introductory of security in ICS, and will hopefully give a better understanding of this issue to integrators and asset owners.

Please read the ICS Guidance document for making a simple initial assessment of the security in your ICS.

[15] See Section 5.1 for more details.

7 Further reading

Standards:

- IEC 62443 *(A set of standards about Cybersecurity for industry, very heavy and is mostly used by large organizations).*

Whitepapers and reports:

- NIST Guide to Industrial Control Systems (ICS) Security *(An easy and recommended read)* [13]
- *SINTEF - Grunnprinsipper for IKT-sikkerhet i industrielle IKT-systemer* [16]
- *ENISA - Industry 4.0 Cybersecurity: Challenges & Recommendations* [3]
- *ENISA - Good Practices for Security of IoT in the context of Smart Manufacturing*
  ENISA- Protecting Industrial Control Systems: Recommendations for Europe and Member States
- *CPNI – Firewall Deployment For SCADA and Process Control Networks: Good Practice Guide (2005)*
- *GE - An Executive Guide to Cyber Security for Operational Technology*
- *ISA-TR84.00.09-2017, Cybersecurity Related to the Functional Safety Lifecycle*
- *21 Steps to Improve Cyber Security of SCADA Networks*
- *SANS Institute – The Industrial Control System Cyber Kill Chain* [4]

Cloud Service Security Resources (not ICS specific):

- NSA – *Cloud Security Basics*, 2018-08-29, U/OO/189300-18, PP-18-0571, https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/professional-resources/csi-cloud-security-basics.pdf [Accessed 2023-01-04]
- EC-Council – *What Are the Top 5 Cloud Computing Security Challenges?*, 2022, https://www.eccouncil.org/cybersecurity-exchange/cloud-security/what-are-the-security-challenges-in-cloud-computing/ [Accessed 2023-01-04]
- Kizhakedath Media Services Pvt. Ltd. – *How to perform a cloud security assessment: 5 key steps*, 2022-12-29, https://infotechlead.com/cloud/how-to-perform-a-cloud-security-assessment-5-key-steps-76088 [Accessed 2023-01-04]
- Surkay Baykara – *How to Conduct a Cloud Security Assessment*, 2022-02-21, https://www.pcidssguide.com/how-to-conduct-a-cloud-security-assessment/ [Accessed 2023-01-04]

8 Bibliography

[1] G. Immerman, “INDUSTRY 4.0 VS. INDUSTRIAL IOT: WHAT'S THE DIFFERENCE?,” Machinemetrics, 05 Jan 2018. [Online]. Available: https://www.machinemetrics.com/blog/industry-4-0-internet-of-things-what-s-the-difference. [Accessed 04 Jan 2023].

[2] "IIoT Vs. Industry 4.0," ATS, [Online]. Available: https://www.advancedtech.com/blog/iiot-vs-industry-4-0/. [Accessed 04 Jan 2023].

[3] ENISA, «INDUSTRY 4.0 CYBERSECURITY: CHALLENGES & RECOMMENDATIONS,» ENISA, 2019.

[4] M. J. Assante og R. M. Lee, «The Industrial Control System Cyber Kill Chain,» SANS, 2015.

[5] J. Slowik, "Anatomy of an attack: Detecting and defeating crashoverride," in *Virus Bulletin*, Montreal, 2018.

[6] J. Slowik, «Crashoverride: Reassessing the 2016 ukraine electric power event as a protection-focused attack,» Dragos, 2019.

[7] A. Cherepanov, «WIN32/INDUSTROYER: A new threat for industrial control systems,» ESET, 2017.

[8] B. Johnson, D. Caban, K. Marina, D. Scali, N. Brubaker and C. Flyer, "Attackers Deploy New ICS Attack Framework “TRITON” and Cause Operational Disruption to Critical Infrastructure," Mandiant, 14 Dec 2017. [Online]. Available: https://www.mandiant.com/resources/blog/attackers-deploy-new-ics-attack-framework-triton. [Accessed 08 Sep 2022].

[9] NCCIC, «MAR-17-352-01 HATMAN—SAFETY SYSTEM TARGETED MALWARE,» National Cybersecurity and Communications Integration Center (NCCIC), 2017.

[10] CISA, «MAR-17-352-01 HatMan—Safety System Targeted Malware (Update B),» Cybersecurity and Infrastructure Security Agency (CISA), 2019.

[11] A. Kling og P. Forney, *TRITON - Schneider Electric Analysis and Disclosure,* Recorded presentation given at he S4x18 security conference, 2018.

[12] G. Julian, *Triton - A Report From The Trenches,* Recorded presentation given at the S4x19 security conference, 2019.

[13] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams og A. Hahn, «Guide to Industrial Control Systems (ICS) Security,» NIST, 2015.

[14] GE, «An Executive Guide to Cyber Security for Operational Technology,» GE, 2017.

[15] C. W. Johnson, "Barriers to the use of intrusion detection systems in safety-critical applications," in *International Conference on Computer Safety, Reliability, and Security*, 2014.

[16] M. G. Jaatun, E. Wille, K. Bernsmed og S. S. Kilskar, «Grunnprinsipper for IKT-sikkerhet i industrielle IKT-systemer,» SINTEF, 2021.