LOGIIC™

Linking the Oil and Gas (O&G) Industry to Improve Cybersecurity

Project 12
Safety Instrumentation

Final Report
April 2021

Prepared for LOGIIC and the Automation Federation

Prepared by SRI International
under contract to the
U.S. Department of Homeland Security (DHS)
Science and Technology (S&T) Directorate

Document Title	Project 12 Safety Instrumentation Final Report
Document Date	April 21, 2021
Version	Version 1.0
Primary Authors	Laura S. Tinnel, Ulf Lindqvist SRI International
Funded By	U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate under Contract No. HSHQDC-16-C-00034
Distribution Category	Public
Approval Status	Approved
Approval Date	April 21, 2021

Contents

Figures

Tables

This page intentionally left blank.

Executive Summary

The Linking the Oil and Gas (O&G) Industry to Improve Cybersecurity (LOGIIC) consortium was established in partnership with the U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate to review and study cybersecurity issues in industrial control systems (ICS) that impact safety and business performance as they pertain to the O&G sector.   Project 12 was conducted during 2020 and focused on the security and management of safety instrumentation. The project revealed numerous consequential and recurring findings that indicate a pervasive industry-wide security problem in safety systems. This report documents key findings and recommendations for asset owners, vendors, and standards bodies.

ICSs use safety instrumented systems (SISs) to monitor operations and take automated actions to maintain a safe state when abnormal conditions occur. Instruments such as transmitters, valve controls, and fire and gas detectors provide critical inputs and controls to safety system function. In recent years, instruments have been modernized to provide smart features such as partial stroke testing for valves.

Smart instruments are typically connected to the SIS using direct cabling and communicate via analog signals. Smart data is superimposed over analog communications using the Highway Addressable Remote Transducer (HART) protocol. This protocol enables systems to read data from instruments and modify their configurations and states as part of normal operations. HART data can be accessed by local handheld devices, through pass-through SIS I/O cards, or with a HART data multiplexer (MUX). In the latter two cases, an instrument or asset management system (IMS/AMS) can interact with and configure safety instruments using the HART protocol over an internet protocol (IP)-based network using HART-IP or SIS proprietary protocols. While the earlier LOGIIC Project 5 focused on wireless HART and handheld devices, Project 12 focused exclusively on wired HART, HART-IP, SIS proprietary protocols, and the use of an IMS/AMS.

The lack of built-in security features in the HART protocol necessitates the use of alternative methods to protect devices from unauthorized modifications. Protections considered under Project 12 included a hardware write-protect switch on the instrument, a software-based write-protect password or pin code on the instrument, password on the IMS/AMS (or its underlying operating system platform) that remotely manages the instrument, and a variety of disparate protections provided by various SIS solutions.

Project 12 defined and used a threat model in which the attacker sought to compromise an IMS/AMS and use that platform to make unauthorized changes to the configuration of safety instruments. Unauthorized changes considered by Project 12 were those that could result in unsafe operating conditions, render the instruments inoperable or unable to perform safety functions, and/or take instrument control away from asset owners. These attacker goals were examined in the context of two architectures: 1) the IMS/AMS controls instruments through a MUX and 2) the IMS/AMS controls instruments through an SIS.

Four individual assessments were planned based on the threat model, industry protection mechanisms and architectures, and a sampling of vendor products typically found in O&G sector operations. Attack avenues considered included malicious and unwitting insiders

and supply chain attacks. Each assessment was conducted as a partial-knowledge test with full cooperation from the vendors.

Concerted adversaries have ample time and resources to analyze vendor products, which enables them to discover undocumented commands and vulnerabilities that can be used in attacks. In contrast, Project 12 was limited in both time and scope. Each assessment was conducted over the course of a few months using publicly available information and several weeks of hands-on testing and was constrained by defined rules of engagement (RoE). Even with these limitations, Project 12 uncovered numerous consequential and recurring exploitable weaknesses across individual assessments that indicate a systemic and pervasive industry-wide problem. This issue is mainly a consequence of four critical findings: 1) some safety system designs allow unchecked HART passthrough, 2) the current HART and HART-IP protocols have no built-in security, 3) devices do not authenticate the sources of received HART commands and many have bypassable write-protections, and 4) the industry uses unverified 3rd party software downloaded from the Internet.

Successfully demonstrated attacks used a number of commonly available attacker tools and exploited common-knowledge architectural weaknesses[1] that were present in all four assessments. These attacks required a low to moderate level of effort to exploit and included effects that can significantly impact device safety function.

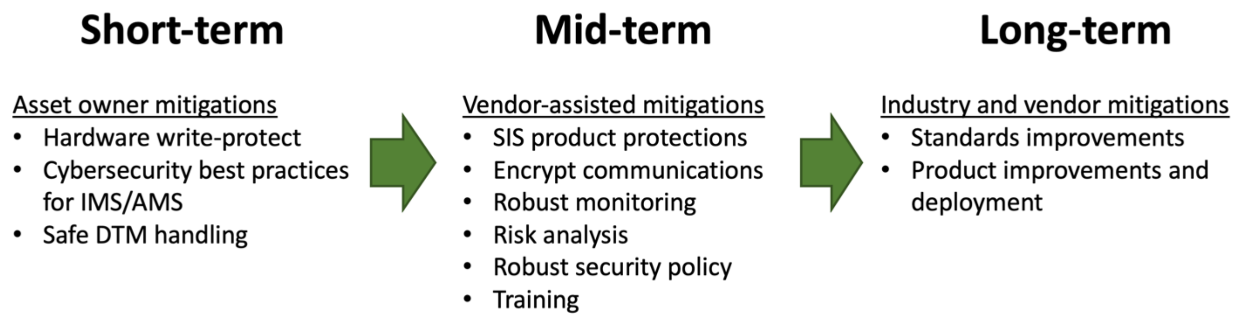
Project 12 also exposed the risks associated with the two architectures and determined the circumstances under which each architecture poses the least risk. Key findings include:

- Attackers can make unauthorized device changes at will and evade detection. Some changes can result in unsafe operating conditions. The risk of cyberattack directly impacts safety and must be considered along with hardware faults and other safety considerations.
- There is no simple and immediate remedy for securing safety systems; risk reduction requires a combination of protection and detection mechanisms.
- Safety systems architectures that mediate IMS/AMS and safety instrument communications using an SIS with enabled protective features pose less risk of unauthorized device modification than do architectures using a passthrough MUX. If SIS protections are not enabled, the risk is equivalent to that of using a MUX.
- Device hardware-based write protections are the only fully protective means to prevent unauthorized device configuration changes. Only 33% of sampled devices had hardware switches.
- Software-based write protections can be bypassed with little effort; therefore, they do not protect against these changes. SIS write protections effectively prevent some, but not all, changes.
- Device write-protect implementation is inconsistent, even across the same vendor products. This can lead to confusion and accidentally unprotected devices.

[1] MITRE Common Weakness Enumeration (CWE) database. https://cwe.mitre.org/data/definitions/1008.html

- HART protocol design deficiencies complicate the prevention of and monitoring for attacks attempting unauthorized changes. The protocol lacks basic security concepts. The HART common command set does not include security-relevant commands which leads to inconsistent implementation across devices using device-specific commands. This hinders the detection of attempts to circumvent device security features. The protocol provides no means to differentiate device-specific read and write commands. This makes it impossible for any SIS to block device-specific write commands without also blocking read commands. Blocking device-specific commands prevents the IMS/AMS from displaying the status of any device-specific commands.
- The practiced method of distributing and installing device type manager (DTM) software opens the door to supply chain attacks and thus poses significant risk to IMS/AMS platforms. These platforms are trusted and can be used as a launch point for device attacks.

Critically, Project 12 concluded that the safety environment is vulnerable to malicious attacks that may be undetectable in practice and that extreme caution should be taken before installing any software, which could introduce malware into the process control network (PCN). ***We cannot sufficiently emphasize the severity of this vulnerability.***



Short-term

Asset owner mitigations

- Hardware write-protect
- Cybersecurity best practices for IMS/AMS
- Safe DTM handling

Mid-term

Vendor-assisted mitigations

- SIS product protections
- Encrypt communications
- Robust monitoring
- Risk analysis
- Robust security policy
- Training

Long-term

Industry and vendor mitigations

- Standards improvements
- Product improvements and deployment

Figure 1. LOGIIC Project 12 Recommended Risk Mitigation Roadmap

LOGIIC recommends a roadmap of mitigations to reduce risk to asset owners over the short-, mid-, and long- terms (Figure 1.) Safety system owners should immediately

- Follow the IEC 61511-1 standard, which requires that all SIS devices have write-protection. Use hardware write-protect switches on all devices that have them. Disable switches only when conducting maintenance.
- Apply security best practices to the IMS/AMS platform to prevent attackers from exploiting the platform’s trust relationship with the SIS to launch attacks. Use network segregation or a host-based firewall (e.g., Windows 10 Security firewall) to prevent remote access.
- Avoid using vendor DTMs in safety-critical applications where possible; instead, opt for device description (DD) files. Where DTMs are currently in use, verify the pedigree and integrity of all DTMs files. Obtain DTM and DD files directly from vendors. Request cryptographic hashes to verify the integrity of all DTM and DD installers. Ask that vendors sign all individual files. Verify DTM and DD integrity before installation on IMS/AMS platforms. Required that all DTMs or DDs downloaded from the Internet use HTTPS.

Based on Project 12 findings, these mitigations will substantially reduce the risk to safety systems. In the midterm, LOGIIC recommends that safety system owners

- Use the SIS to mediate communications between IMS/AMS solutions and safety instruments. Work with the SIS vendor to identify and implement SIS-specific protective measures to reduce the available attack surface and therefore, risk.
- Implement a means to limit allowed SIS network connections only to authorized hosts to prevent unauthorized hosts from making changes.
- Encrypt communications between the IMS/AMS and SIS where possible to avoid network-based attacks that steal passwords and change device commands in transit.
- Implement a robust monitoring system to detect and alert on device changes and on unexpected device states.
- Conduct a consequences-based risk analysis of all operational safety systems using Project 12 findings to identify any residual risk not mitigated by applied countermeasures. Asset owners should identify and implement additional countermeasures based on risk to their own operations.
- Create a robust security policy for their systems. Operators should be trained on the policy and how to avoid inadvertently introducing malware into the environment.

Longer term fixes should address larger issues that require vendor product and industry-level changes. These include implementing the secure HART-IP protocol that was included in the HART Network Management Specification published July 2020.

The full report includes additional findings and recommendations for asset owners, product vendors, and standards bodies. By providing these project outputs, LOGIIC hopes to help improve the overall security posture of all ICS stakeholders.

1. Introduction

The Linking the Oil and Gas (O&G) Industry to Improve Cybersecurity (LOGIIC) Project 12 focused on the safety instrumentation and the management and control of safety instruments. Project 12 is built on Project 11, which focused on safety controllers, engineering workstations (EWSs), and human-machine interface (HMI) components within various vendors' safety system offerings.

Project 12's objectives were to identify vulnerabilities in safety instruments and instrument- or asset-management system (IMS/AMS) solutions used within the safety system architectures typically found in the O&G sector and to recommend security design alternatives and configurations that can mitigate the exploitation of found vulnerabilities.

Project 12 focused on three specific types of safety instruments and how attacks can adversely affect their operation: transmitters, smart valve solenoids and positioners, and fire and gas detectors. These instruments have been modernized in recent years, and many of these devices now provide smart features, such as partial-stroke testing for valves.

Smart instruments are typically connected to a safety instrumented system (SIS) by direct cable and communicate via analog signals. Smart data is superimposed over analog communications using the HART protocol, which has no built-in security. This data can be accessed by local handheld devices, through pass-through SIS I/O cards, or with a Highway Addressable Remote Transducer (HART) protocol data multiplexer (MUX). In the latter two cases, an IMS/AMS can interact with and configure safety instruments using the HART protocol. Project 5 focused on wireless HART and handheld devices, which can be used to configure safety instruments. Project 12 focused exclusively on wired HART 5 and 7 and IP-based communications for instrument management, including HART-IP.

The HART protocol supports the ability to read data from safety instruments and modify their configurations and states as part of normal operations. If an instrument parameter is altered or the transmitter is forced into a test mode, the instrument may no longer be able to perform its safety function.

Methods used by industry to protect instruments from unauthorized modifications include a hardware write-protect switch on the instrument, software write-protect password on the instrument, password on the IMS/AMS that remotely manages the instrument, and a variety of unique protections provided by SIS solutions. Using these protections, Project 12 sought to determine if and how the IMS/AMS could be used by an attacker to change instrument configurations and states to create potentially unsafe conditions. Specifically, Project 12 intended to answer the following high-level questions:

- How can the IMS/AMS be compromised?
- How can the IMS/AMS be used in unauthorized ways to adversely affect safety instruments?
- Can unauthorized changes to safety instruments negatively impact the operation of the safety system?

Answering these questions required direct examination of the IMS/AMS platforms and the safety instruments. Once an understanding was gained, the project examined the IMS/AMS and the instruments in the context of two architectures: IMS/AMS control safety instruments through an SIS and IMS/AMS control safety instruments through a MUX. The project also sought to identify safety system configuration options that could potentially mitigate some portion of the attacks.

Project 12 conducted four (4) individual assessments, each using a representative sample of different vendor products. Concerted adversaries have ample time and resources to analyze target products, which enables them to discover undocumented commands and software or firmware vulnerabilities that could be used in attacks. Project 12 assessments were limited in time and scope; each was conducted as a partial-knowledge test with only a few weeks of hands-on testing. Still, the test team discovered numerous concerning and recurring findings, which indicate a systemic industry-wide problem. A large part of this risk can be mitigated by using device hardware write-protections, securing the IMS/AMS platform with cybersecurity best practices, and enabling SIS protective features.

While Project 12 was undertaken for the benefit of the LOGIIC members, LOGIIC is making these results available to the broader industry to

- Help the industry understand the inherent risks and potential impacts associated with different safety-system architectures and where security gaps exist that require additional technology or policy solutions.
- Make recommendations for selecting and implementing a safety-instrumentation architecture, including additional proposed defensive mechanisms needed to mitigate serious threats that may exist due to architectural trade-offs (i.e., no architecture is fully secure.)
- Convey a general knowledge of the efficacy of different protection measures in preventing unauthorized safety instrument modifications.
- Provide general insights to major IMS/AMS and instrument vendors to improve the security of their products.
- Provide recommendations to standards bodies for areas to address to improve the risk posture for the entire industry.

The remainder of this report presents the test details, findings, and recommendations. Section 2 states the project objectives. Section 3 documents the assessment methodology and scope. Section 4 summarizes the results. Section 5 discusses recommendations. Section 6 summarizes the report.

2. Project Objectives

Project 12's objective was to understand an attacker's ability to compromise an IMS or AMS and use that platform to alter the configuration of safety instruments to create unsafe operating conditions, render instruments inoperable, and/or take control away from asset owners. Within this context, the project sought to identify potential vulnerabilities in safety instruments and the IMS/AMS when used in safety system architectures typically found in the O&G industry, evaluate available protections, and identify protection gaps. The project also sought to recommend architectural and configuration changes to help mitigate the exploitation of found vulnerabilities.

Project 12 was based on a hypothesis that an architecture in which an SIS mediates communications between an IMS/AMS and the devices it manages can better mitigate device vulnerabilities than is an architecture in which the IMS/AMS communicates with the devices through a MUX. Using this hypothesis, the test team crafted a series of questions that, if answered, could provide evidence to prove or disprove the hypothesis. The derived questions were

- Can an attacker compromise the IMS/AMS platform?
- Can an attacker gain administrative privilege on the IMS?
- Can an attacker gain remote control of an IMS?
- Can an attacker compromise the IMS software and/or system (e.g., modify or install a trojan version) either from the IMS system host platform or by remote means?
- Can an attacker intercept a safety instrument password via keystroke analysis, memory leakage, or network sniffing?
- Can an attacker affect smart instruments by remotely controlling the IMS software using stolen or cached credentials, with or without IMS administrative privilege?
- Can an attacker affect smart instruments using a vulnerability exploit, with or without IMS administrative privilege?
- Can an attacker change an instrument parameter to an unsafe setting while evading detection of the parameter change? (If an attacker has desktop control of the IMS and the appropriate login credentials, they can change any normal operating parameters. The question under consideration is whether such a change can be made undetected.)
- Can an attacker bypass any instrument's physical lock or password to
  - Cause the instrument to give a false reading (e.g., change the range on the instruments to send the wrong analog signal to the SIS)
  - Force the instrument into commissioning mode so it will send any attacker-specified value to the SIS
  - Cause the device to fail to execute authorized parameter or state-update commands
  - Cause the instrument to go offline or otherwise become unresponsive
  - Change any instrument password
  - Lock administrative operators out of controlling the instrument

These questions were examined in the context of the two architectures shown in Figure 2 to expose risks associated with the two architectures and determine whether either architecture posed more or less risk. Section 4.2 answers these questions.

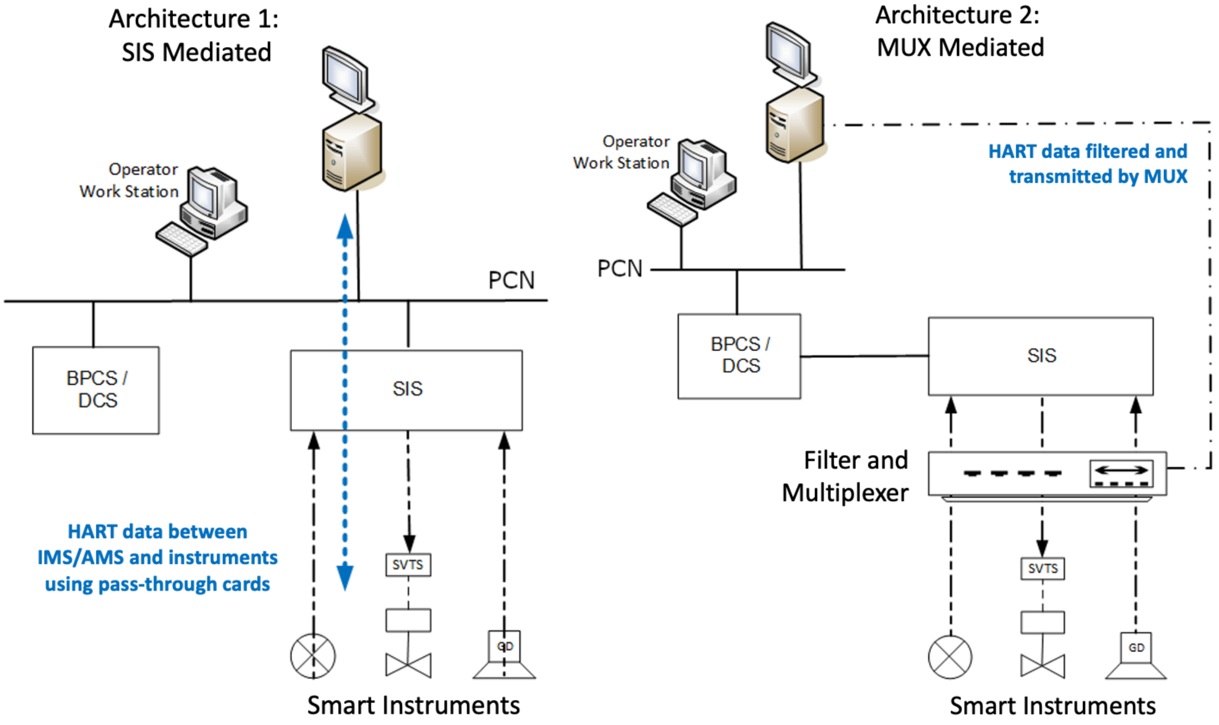


Figure 2. Architecture 1 (left) provides direct network connection to the SIS, which mediates communications between the IMS/AMS and devices. In contrast, the SIS in Architecture 2 (right) is accessed through an interface on the BPCS/DCS and the IMS/AMS communicates with devices through a serial connection with a MUX.

In reference Architecture 1, the SIS and IMS/AMS are on the process control network (PCN). Safety instruments are not directly accessible on this network. HART data is passed between the IMS/AMS and devices through the SIS using pass-through I/O cards. In reference Architecture 2, the SIS is not accessible on the PCN; it is only accessible through the basic process control system (BPCS) or distributed control system (DCS). Because of this, the IMS/AMS cannot communicate through the PCN with the SIS or the instruments. To enable these required communications, the IMS/AMS is connected to a MUX. HART data is passed between the IMS and instruments through the MUX, bypassing the SIS entirely.

Project 12 sought to understand more generally whether different safety system configurations had inherent risks rather than uncovering specific vulnerabilities in specific vendor products. Therefore, the project tested four instances of each reference architecture using a representative set of vendor products in the following categories: safety instrumented systems (SISs), IMSs or AMSs, transmitters, fire detectors, gas detectors, and smart valve positioners. To meet the goal of the effort, the project analyzed results across all four assessments to draw generalized conclusions.

3. Assessment Methodology

The project team engaged vendors to plan four individual assessment activities using a diverse set of industry products representative of those used in the O&G industry. For each assessment, the team designed a series of general and product-specific test cases to answer the assessment questions in Section 2. All test cases were based on a plausible threat and met Project 12 objectives. Test case design was limited by the scope and rules of engagement (RoE) defined in this section.

3.1 Roles

The Project 12 assessment methodology was designed to be rigorous, highly collaborative, and conducted with support from relevant vendor staff. The assessment team included a Test Director, a red team, a green team, and test observers.

The Test Director monitored progress and kept a record of all test activities, including test techniques, steps launched, results, and observations. The red team was composed of ICS-knowledgeable penetration testers, subject matter experts (SMEs), who planned and executed the hands-on testcases. The Test Director oversaw and conferred with the red team to dynamically change course, as necessary, to meet the overall assessment objectives.

The green team was composed of the vendors participating in each individual assessment. Device vendors provided telephone and email support to SMEs, who performed tests in a closed laboratory environment. Architecture 2 was also tested in this closed laboratory. SIS and IMS/AMS vendors assisted in setting up the SIS testbeds and were present for and allowed to observe Architecture 1 testing. All issues found were reported to the appropriate vendors. Vendors were provided results specific to the performance of their own products. Participating vendors and products are protected under the LOGIIC confidentiality agreement.

One or more LOGIIC members attended and observed Architecture 1 testing. A LOGIIC safety system expert provided technical advice on the potential operational impacts of the test attacks. The Test Director used this input to guide the testing activity.

3.2 Threat Model

Project 12 used a threat model to help define the assessment RoE. The Project 12 threat scenario involved attackers with insider-sourced knowledge from an O&G company regarding the specific vendor products and versions used in an operational safety system. The O&G insider also provided limited physical access to some systems. The choice of an insider obviated the need to expend project resources attacking the IMS/AMS platform operating system (Microsoft Windows), which was not a focus of Project 12.

Attackers did not have inside access to any product vendor companies. They had access to publicly available product information but not to detailed schematics and code. Attackers had no ability to perform lifecycle attacks by injecting malware into vendor firmware. However, they could create and distribute trojan versions of product software through any of a number of commonly used methods (e.g., supply chain or social engineering.) Specific attacker access is shown in Table 1.

Table 1. Summary of Project 12 Threat Model Attacker Assets and Accesses

Source	Asset and/or Access Provided
O&G company insider	List of specific safety system products and versions in use and how they are used within the system
	Network switch access, including the ability to insert a network sniffer
	Physical access to IMS/AMS that is connected to the PCN
	Copies of IMS/AMS, device type manager (DTM), and device description (DD) software installed on IMS/AMS platform
	Ability to install IMS/AMS patches and DTMs on an IMS/AMS platform (i.e., administrator access)
buyusedICSstuff.com	Used industrial control system (ICS) instruments for probing and analysis
Product vendor public websites	Product sales literature, user manuals, and other documentation
	HART protocol specification
	Product DTMs, software updates and/or patches (only available publicly)
Public web site	ICS-CERT and other advisories
	Other public information (e.g., from product resellers)
Dark web	Working product exploits

Attackers did not have physical access to the SIS system for attack analysis, but they had access to the IMS/AMS software and the DTMs used to communicate with and control the field devices. The O&G insider had logical and physical access to the IMS/AMS, but they did not have physical access to the field devices. Since the IMS/AMS communicates with and controls the devices, and the insider had access to this system, the attackers chose to target this platform as the launch point for device attacks. Attackers wanted to maintain long-term access, so evading detection was important.

Devices. Instrument testing modeled an outsider threat with respect to the instrument vendor and an insider threat from the perspective of the O&G operations environment. The threat model assumed a launch point of an AMS/IMS platform within an operational environment. Detailed device architecture specifications, schematics, firmware revisions, and source code that a vendor insider could obtain were unavailable. Available information sources were 1) public information available on the Internet (e.g., user guides, HART protocol specification, CERT advisories); the dark web (e.g., vulnerabilities and working exploits); and O&G insider information about how devices are used and configured in the operational environment. This information was combined to design and craft device-specific attacks to be launched by an insider at the O&G facility.

IMS/AMS. IMS/AMS testing modeled an outsider threat with respect to the IMS/AMS vendor and an insider threat from the perspective of the O&G operations environment. The threat model assumed the attacker had access to the IMS/AMS software binary files and DTM plug-ins for attack analysis and planning and could install modified versions of the software.

3.3 Rules of Engagement

Project 12 defined and followed the RoE, which detailed the assessment scope and boundaries.

3.3.1 Scope

Project 12 focused on data flow, stability, connectivity, security controls, and architectural and configuration vulnerabilities. It addressed threat vectors relevant to the questions outlined in Section 2. As Project 12 sought to learn how attackers, using the IMS/AMS platform as an attack launch point, could modify device configurations to create unsafe conditions, alternate platforms were out of scope for assessments with one exception: determining if IMS/AMS-to-SIS communications could be hijacked from a network-connected device.

The IMS/AMS communicates with field devices using the HART protocol and, for ethernet communications. LOGIIC Project 5 examined the Wireless HART protocol. Wireless HART was out of scope for Project 12 and not used in the test environment.

Attacks based on physical access to instruments during the architecture assessment were out of scope; however, the detailed instrument assessment was conducted with physical access. A representative set of vendor products included SISs, IMSs or AMSs, transmitters, fire detectors, gas detectors, and smart valve positioners. A single representative HART MUX was used. All other products were out of scope for this project.

Supply-chain attacks were in scope. The scope was additionally constrained by the threat model discussed in Section 3.2.

3.3.2 Test Case Construction

Project 12 testcases were planned using partial product knowledge. All testcases were required to be based on plausible threats and be traceable and reproducible. Testcases were documented step-by-step so vendors could rerun the tests in their own labs.

Devices. SMEs had hands-on access to devices that allowed them to analyze device behavior, extract and analyze firmware for vulnerabilities, and plan and craft attacks. Abusing valid device commands and input fuzz testing were in scope. Crafting and loading malicious firmware were out of scope due to time constraints. Testcases were prepared to sample available HART common, universal, and device-specific commands for each device. Some testcases used multiple commands to achieve a higher-level goal.

IMS/AMS. SMEs had access to the IMS/AMS software binary and DTM plug-ins for attack analysis and planning. Testing could include 1) remote attacks that could occur from any other point on the PCN; 2) physically initiated attacks through insertion of malicious removable media (e.g., USB, CDs); 3) password cracking attacks; 4) use of trojan components; and 5) any other attacks that could be launched from co-resident software (e.g., memory-based attacks, hijacking) and used to take control of the IMS/AMS software or modify device settings.

SIS. The SIS was considered a black box, so testing it for vulnerabilities was out of scope; therefore, SMEs did not craft testcases specifically aimed at the SIS. Testing SIS features that could mitigate device vulnerabilities was in scope as part of the Architecture 1 testing.

MUX. The MUX was considered a black box, so testing it for vulnerabilities was out of scope; therefore, SMEs did not craft testcases specifically aimed at the MUX.

Architecture. Architecture testcases were built on device and IMS/AMS testcases. These testcases considered various IMS/AMS supply-chain attacks (e.g., malicious DTM, trojan IMS/AMS software) that could compromise the platform and launch device attacks.

For each assessment, testcases were formulated to compare the ability of available protections to mitigate vulnerabilities found in device testing. Tested protections included those provided by devices (e.g., physical switches, passwords, pin codes), IMS/AMS (e.g., access control), and SIS (e.g., program/run mode key).

3.3.3 Architecture 1 Testing

Architecture 1 testing was conducted in an environment representative of the safety system portion of an O&G organization's process control system. This environment was instrumented with a test harness as shown in Figure 3.

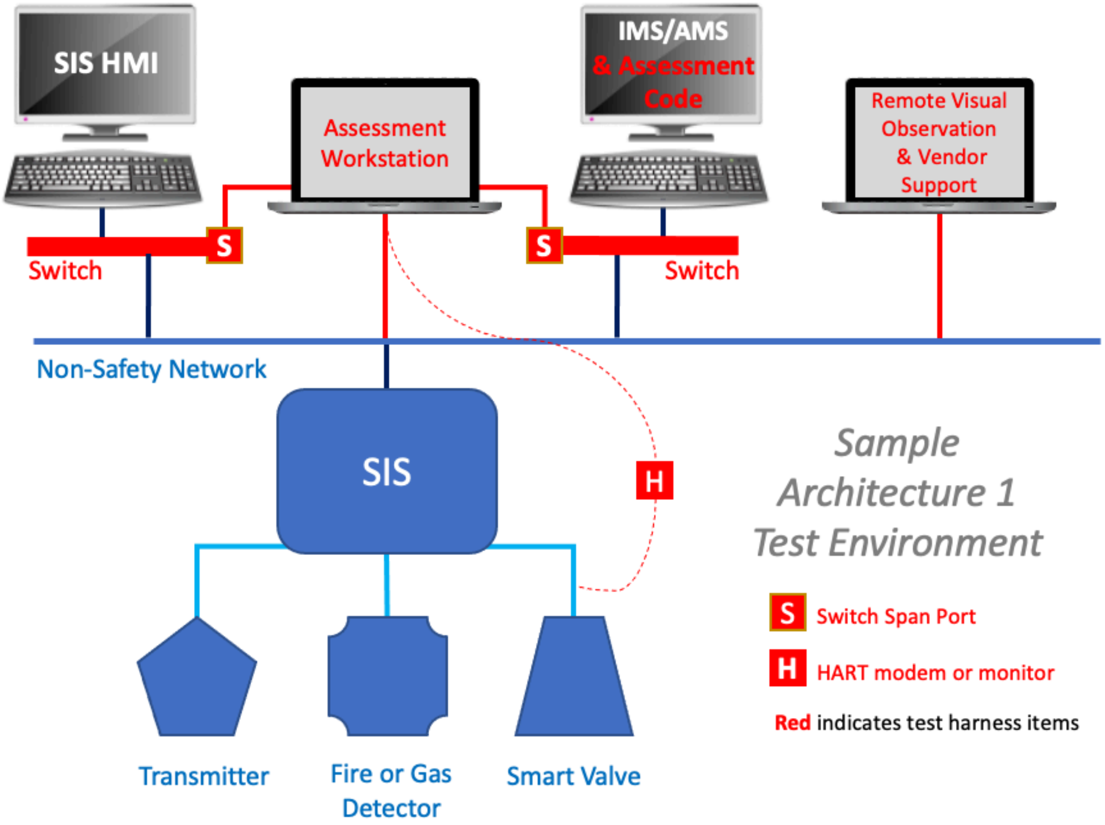


Figure 3. The Architecture 1 test environment consisted of multiple instruments, an IMS/AMS, an SIS, an SIS HMI, and test harness components (shown in red). Each environment differed slightly based on vendor product solutions.

Testing considered both insider and outside threats with respect to the O&G company and included supply chain attacks against device DTMs. Testcases covering all available protections were executed in this environment. A subset of representative device testcases were used for this purpose. All IMS/AMS, DTM, and DD testcases that were of architectural relevance were conducted in this environment.

Testcases executed individual steps in end-to-end attacks, which simplified the testing. Testcases that attempted to change device settings were launched from the IMS/AMS platform as co-resident processes with the IMS/AMS software. Separately, IMS/AMS and/or DTM/DD testcases demonstrated gaining access to and compromising the platform. The assessment workstation was used to run other testcases and collect and analsyze data.

When executing a test case resulted in an unexpected or undesirable outcome, the testcase was analyzed and rerun to ensure a repeatable result. At that point, test execution stopped, and the results were discussed with the LOGIIC safety system expert to understand the potential impact of such an attack. Then the vendor was engaged to discuss findings.

Testcases with the potential to create significant damage were addressed at a time that facilitated rebuild or restoration, or they were conducted last.

3.3.4 Device and Architecture 2 Testing

Architecture 2 testing was conducted concurrent with device testing using a test environment as shown in Figure 4. The complete set of device-specific protection testcases were executed in this environment. A single IMS/AMS solution was used in all tests and its host platform was used as a launch point for all device testcases. IMS/AMS solution testing was conducted in the Architecture 1 environment (Figure 3).

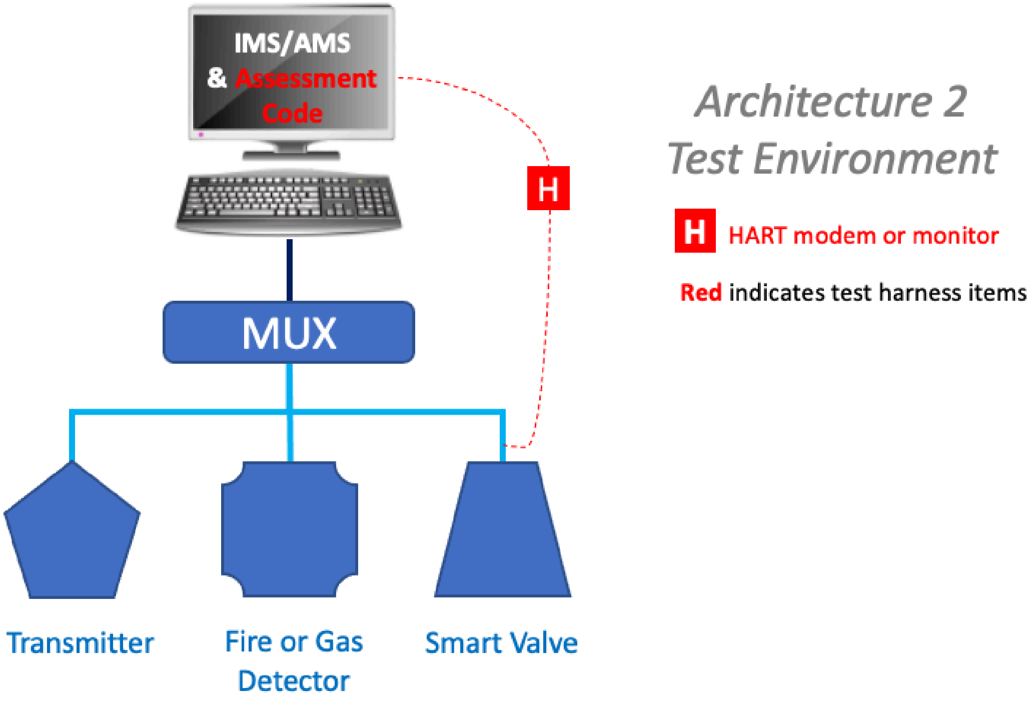


Figure 4. The Architecture 2 test environment consisted of multiple instruments, an IMS/AMS, a MUX, and test harness components. Testcases were launched from the IMS/AMS platform as co-resident processes with the IMS/AMS software.

3.3.5 Data Capture

Data was captured throughout Project 12 activities. Screen shots, logfiles, network packets, and other evidence were collected and preserved for subsequent inspection and review. Collected data was shared with appropriate vendors, SMEs, and the Technical Lead. All captured data was protected as LOGIIC Confidential.

3.4 Human Subjects

No human subjects were used in any part of this test.

4. Results

This section details Project 12 results. Overall findings listed by safety system component class are discussed first, followed by answers to the hypothesis-derived questions posed in Section 2. This section concludes with an architecture discussion.

For inclusion in this report, findings must have been observed, be reproducible across multiple vendor products, and be common-knowledge architectural weaknesses[2]. Product-specific vulnerability information was provided directly to the appropriate product vendors and will not be disclosed by LOGIIC.

4.1 Findings by Safety Component Class

4.1.1 Safety Instruments

All devices implemented a combination of common, universal, and device-specific HART commands. Not all required HART common commands were implemented on every device, and many of the devices implemented undocumented device-specific commands. A number of the undocumented commands appeared to operate as a toggle, meaning an attacker does not need to determine valid parameters to execute the commands.

All devices tested appeared to operate under the assumption that any HART packet received was legitimate, regardless of the source of the packet. None implemented authentication or authenticated sessions. In the absence of device write protection or other external protective measures, attackers are able to execute any device-supported HART command at will from the IMS/AMS host platform. Therefore, safety instruments, in general, are subject to configuration integrity attacks.

Successful instrument attacks demonstrated during Project 12 include those shown in Table 2. Individual devices are subject only to the attacks associated with the HART commands they support and may be subject to additional device-specific attacks.

Table 2. Project 12 Testcases demonstrated the ability to modify a wide range of device configurations and states.

Configurations	States	Reset/Evasion
Password and pin code values	Disable write protect	Wipe device alert logs
Alarm settings	Enable write protect	Wipe device history
Valid range limits	Force offline	Reset device change bit
Scaling factors	Put in firmware upgrade mode	
Valve high-low cut off values	Conduct partial stroke test	
Valve positioner feedback values	Put in fixed current mode	
Relay latching behavior	Put in loop current mode	
Partial stroke values	Reset device repetitively	
Positioner calibration	Value position (override)	
Polling address		

Most Project 12 testcases executed 1-3 device commands to achieve some effect. An attacker could issue any combination of commands on one or more devices to achieve

[2] https://cwe.mitre.org/data/definitions/1008.html

some overall safety system effect, including resetting device change bits to dampen potential alarms and changing polling addresses so devices become unresponsive to the IMS/AMS. This would hamper response to any created safety situation.

Device Write Protection. Device protections vary widely with some devices having hardware switches, some having software passcodes, and others having a combination of protections. Only 33% of sampled devices had physical write-protect switches. Protection implementations on same-vendor products are generally inconsistent, possibly leading to devices left unprotected due to customer confusion.

The means to control software-based device write protections is inconsistent. Each device uses a unique HART device-specific command to enable and disable write-protection. This makes it difficult to implement a standard monitoring system to alert when device write-protection is disabled.

All tested device software-based protections (i.e., passwords and pin codes) were bypassed during the test. The password/pin code strength was weak, ranging from 4 to 8 characters, and in some cases, consisting only of digits. No devices examined had any lockout period, which allowed the attacker to quickly crack the password or pin code. Furthermore, these codes are transmitted in clear text and may be intercepted on the network, except in Architecture 1 cases where the SIS supports encrypted communications between the SIS and IMS/AMS.

Attackers are able to change software-based passwords and pin codes to lock out authorized users. Furthermore, in some cases, passwords could be set to strings that are not typable using a standard keyboard.

All hardware-based write protections worked as designed and were not bypassed during the test.

Input Parsing. Most of the devices tested did not respond inappropriately to device fuzzing. They either ignored malformed packets or returned an error code. A relatively small number of devices responded in a way that could indicate a parsing error and hence, a potential buffer overflow.

4.1.2 Instrument and Asset Management Solutions

The IMS/AMS solution is a trusted platform in safety systems architectures. Because of this, the platform can be used to launch any of a number of attacks against the safety system and anything connected to the PCN. It is therefore important to protect this platform using security best practices and limit exposures to potential attacks.

Authentication and Authorization. IMS/AMS solutions tested used host-platform, Windows-domain, or custom-implemented authentication to authorize users. For those that relied on host platform authentication, once a user logged onto the platform, no additional authentication was required to make device changes. Most solutions required additional user authentication to use the IMS/AMS application. This prevented unauthenticated or unauthorized users from affecting the systems through direct system console access. However, the test team was able to bypass all implemented authentication mechanisms and affect instruments through malware insertion on the platform.

Alerts and Logs. Alerts and logs can provide valuable clues to determine when attacks have occurred. IMS/AMSs alert on device changes when the device DTM implements alerting. Other visual indicators can also provide clues when a device state changes unexpectedly. IMS/AMS logging capability varies widely with some solutions providing no logging and others providing comprehensive logs.

Software Integrity. Some solutions apply strict file system permissions to prohibit non-administrative users from modifying IMS/AMS components. However, because installing DTM files requires administrative privilege, malware can be embedded in and installed alongside DTM software and run as co-resident processes (see Figure 5.) Once malware is installed, the platform can then be used to launch any of a number of attacks against the safety system and anything connected to the PCN. Therefore, protecting IMS/AMS platform integrity is of the utmost importance.

Most IMS/AMS solutions tested did not have digitally signed components, and most code was subject to reverse engineering. Creating and installing trojan IMS/AMS components is possible. This weakness allows an attacker to run malware inside the IMS/AMS process space (see Figure 5.)

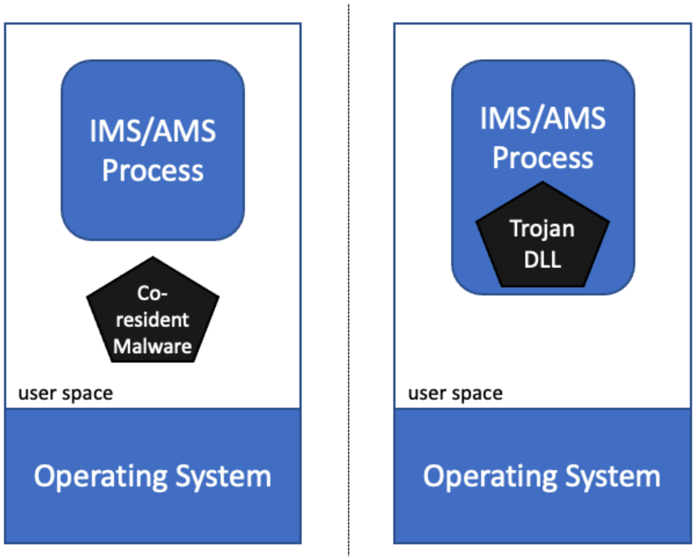


Figure 5. Co-resident malware (left) runs as a separate process on the computer. Injected code through DLL loading (right) runs as part of the IMS/AMS process and inherits all its rights and permissions.

DD and DTM Handling. None of the IMS/AMS solutions tested performed any publisher or cryptographic verification of 3rd party DTM and DD plug-ins. The attacker is able to modify DTM and DD device plug-ins, load any of the IMS/AMS solutions, and run them in the IMS/AMS process space. The IMS/AMS is therefore vulnerable to DD and DTM trojan attacks.

4.1.3 Instrument DTMs and DDs

DDs and DTMs are used by IMS/AMS solutions to control 3rd party instruments. DDs contain ASCII text-based structures that are interpreted by the IMS/AMS to perform a limited set of device configurations. DTMs provide specialized plug-in user interfaces that enable operators to configure device-unique features. As such, DTMs contain both text-based configuration files and executable code in the form of DLL files that are loaded into the running process space of IMS/AMS applications.

Because they include executable code, DTMs, in particular, pose a significant risk to the IMS/AMS platform. The use of DDs poses less risk to the IMS/AMS platform. Project 12 did not test for this kind of error due to time limitations.

An attacker can create and use a trojan install package that includes modified DD or DTM files and possibly additional malware. Administrative privilege is required to install a DTM, so anything included in the installation package, including malware, automatically inherits administrative privilege on the platform.

Verification of the authenticity and integrity of DDs and DTMs is needed to reduce the risk of accidentally introducing a trojan into the environment. 78% of tested DTMs and DLLs were directly downloadable from the Internet, but none had downloadable cryptographic hashes that could be used to verify their integrity. 22% were downloadable using unencrypted HTTP connections, which risks modification in transit. None of the installers had a verified publisher with a valid certificate. Only 22% had signed DLLs to help prevent modification. Another 22% included debug symbols and/or were written in a language that facilitated easy source-code extraction, which makes it easier to create a trojan that looks and acts like the real plug-in. In all, the test team was able to introduce trojan DDs and DTMs that successfully altered device configurations for 78% of tested devices.

4.1.4 Safety Instrumented Systems

SIS solutions were not a central focus of the Project 12 attack assessment as SISs were tested rigorously under LOGIIC Project 11. Each SIS solution included in Project 12 was found to provide a unique set of protective features that, if implemented correctly, could help mitigate some portion of the risk of unauthorized device configuration changes. Discussion of SIS-specific features are not included in this report to maintain vendor participant confidentiality. Safety system owners should contact their SIS vendors to learn more about product-specific protection mechanisms that may be available for use.

All SIS solutions provided a mechanism to block HART commands with varying degrees of granularity. Commonly, asset owners can choose to block HART common and universal write commands and/or HART device-specific commands. Blocking device-specific commands blocks both read and write operations and thus, prevents the IMS/AMS from reading and displaying the status of any device-specific commands. Blocking device-specific commands is not typically done in practice for this reason.

4.1.5 HART Protocol

The HART 5 and 7 protocol and packet structure (shown in Figure 6) has no built-in security features such as authentication and encryption. It uses a 1-byte XOR checksum for packet integrity, which requires a low level of effort to recompute after packet modification.

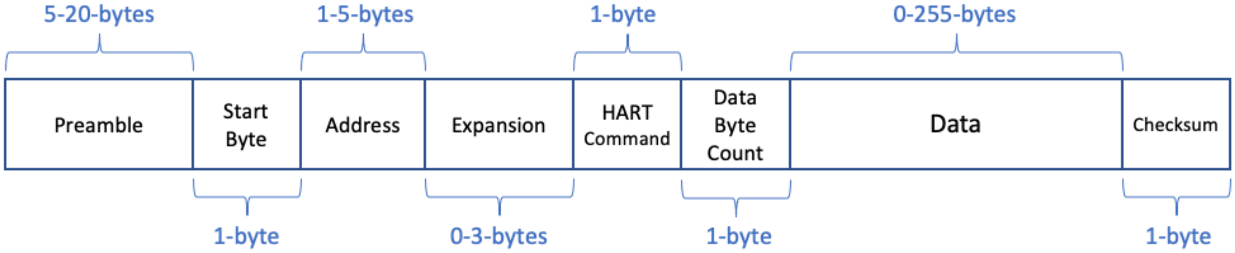


Figure 6. The request HART packet structure has a preamble field (for synchronization and carrier detect), a start delimiter (that designates the start of the communications packet), the device address, an expansion field, a numerical value designating a HART command, a size field for the data, a data field, and a 1-byte checksum for packet integrity.

Using standard, open-source penetration testing tools, an attacker is able to understand, craft, and inject HART commands at will to alter device configurations through IP packet insertion or manipulation when using an SIS without effective protections. HART commands can also be injected through serial connection to a MUX. Because HART is an open protocol and devices have little built-in protection, asset owners must implement external protections.

HART supports common, universal, and device-specific commands. Security-relevant commands are not standardized. Vendors implement write protection, logging, and alerting in a wide variety of non-standard and insecure ways. The lack of a common standard for security-relevant commands complicates efforts to monitor for and detect rogue configuration changes made over the network using the HART-IP protocol.

The HART protocol can be implemented by using a serial communication protocol such as RS485. Limiting the use of HART to serial communications would require an attacker to gain access to the connected computer's serial port (or USB to serial device) to launch an attack. This would reduce the attack surface over using HART-IP, which only requires network access for attack.

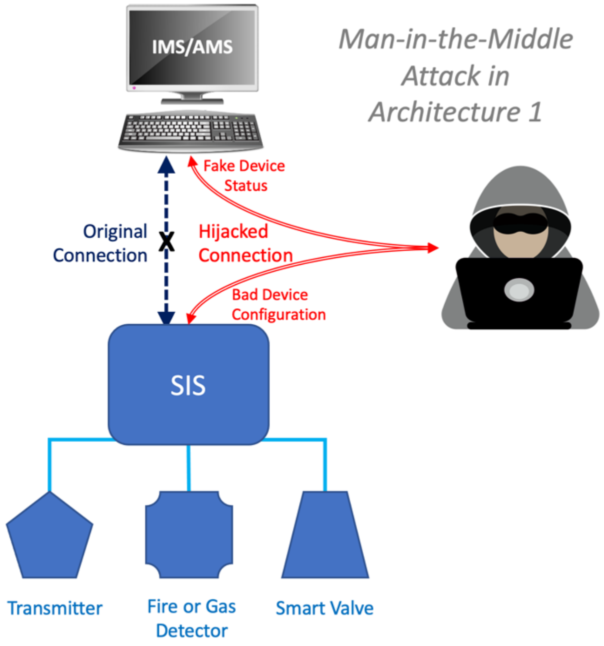


Figure 7. Under certain conditions, unencrypted communications can be hijacked by attackers and used to block or change operator-initiated device commands and send false device status to the IMS/AMS operator console. Using alternate techniques, a similar capability could be achieved through co-resident malware running directly on the IMS/AMS.

4.1.6 Safety Systems Communications

The IMS/AMS communicates with devices to receive status and make configuration changes using the HART protocol. This protocol is open and easy to manipulate, as previously discussed. HART-IP and proprietary protocols that envelop HART packets for use in ethernet networks increase the level of attack complexity. However, if these protocols are unencrypted, they can be reverse engineered given sufficient time and access. When coupled with ARP spoofing to intercept network packets, this enables an attacker to create a man-in-the-middle connection to change commands going to devices and send false information to the IMS/AMS that is then displayed to the operator (see Figure 7.) Therefore, this communications path is a rich attack vector that should be protected and monitored as much as possible. This points to the need for encrypted communications using a secure protocol. This need must be balanced with monitoring requirements.

Some SIS solutions offer encrypted communications between the IMS/AMS and the SIS; this feature is typically disabled by default. Enabling optional encryption mechanisms significantly improved network security and stopped password sniffing and man-in-the-middle (among other) attacks initiated from non-IMS/AMS platforms. Enabling encrypted communications was not generally straightforward.

Two encrypted communications approaches were considered: host-to-host encryption and application-to-application encryption. Project 12 demonstrated that when using host-level encryption, IMS/AMS co-resident malware can craft and insert HART (or HART-wrapped) packets directly into the network stack (see Figure 9, page 31.) These packets are transmitted through the host-level encrypted tunnel to the SIS. The SIS then decrypts and passes the enclosed HART command packets to devices for execution.

Project 12 further demonstrated that, when using application-layer encryption, a trojan DTM DLL can be loaded into the IMS/AMS process space and used to invoke unauthorized device commands that are transmitted through the application-level encrypted tunnel to the SIS. Similarly, the SIS decrypts the commands and passes them on to the device for execution.

In both host- and application-layer encryption, the unauthorized changes are made at will and cannot be seen by any network monitoring system unless that system can decrypt the network packets for inspection. This weakness points to the need for a multi-layered security design solution.

4.2 Hypothesis Revisited

Project 12's primary hypothesis was that a safety system architecture in which an SIS mediates communications between an IMS/AMS and the instruments it manages (Architecture 1) poses less risk than an architecture in which the IMS/AMS connects to instruments through a MUX (Architecture 2). A number of assessment questions were formulated to provide the information needed to confirm or deny this hypothesis.

4.2.1 IMS/AMS Platform Compromise

Can an attacker compromise IMS/AMS platforms?

> Yes. An attacker can install a trojan IMS/AMS DLL, DD, or DTM on all assessed IMS/AMS platforms. Exploiting IMS/AMS host operating system vulnerabilities was not tested in Project 12. This is an additional path to platform compromise.

Can an attacker gain administrative privilege on the IMS/AMS?

> Yes. Administrative privilege is required to install a DTM in every solution tested, so installing a trojan DTM would give an attacker the ability to run malware with administrative privilege on the host operating system. Exploiting IMS/AMS host operating system vulnerabilities was not tested in Project 12. This is an additional path to gaining administrative privilege.

Can an attacker gain remote control of an IMS?

> Yes. Installation of a malicious trojan DTM was demonstrated on all assessed IMS/AMS platforms. The malware was remotely controlled from another point on the network and was able to change instrument configurations. Due to the lack of signing of DTM DLL files, an attacker can modify a DTM DLL that the IMS/AMS then loads and directly executes within its process space. Project 12 demonstrated this scenario with every

vendor IMS/AMS solution tested. An attacker can embed remote control functionality inside the loaded DTM DLL.

Can an attacker compromise the IMS software/system (i.e., modify or install a trojan version) either from the IMS system's host platform or by remote means?

Yes. Project 12 demonstrated the ability to install a trojan or modified DD or DTM by loading it from removable media (also known as "crossing the air gap"). The project also demonstrated the ability to install trojan components in IMS/AMS solutions. Additional means are possible but were not directly tested due to time constraints.

4.2.2 Instrument Compromise

Project 12 explored a number of ways to compromise instruments. If no available protections were used, all tested attacks that executed unauthorized commands succeeded. Testing found that MUXs do not provide any protections against these attacks. These attacks also worked in Architecture 1 with no built-in SIS protections.

When engaging SIS protections in Architecture 1, most of the attacks using HART common and universal command attacks failed. HART device-specific command attacks succeeded when the SIS was not blocking device-specific commands. No SIS had a means to thwart HART device-specific command attacks without also disabling the update of device-specific values in the IMS/AMS operator view. In practice, blocking device-specific commands is not typically done for this reason.

These findings apply to all instrument compromise questions.

Can an attacker affect smart instruments by remotely controlling the IMS software using stolen or cached credentials with or without IMS administrative privilege?

Project 12 demonstrated the ability to install a remotely controlled trojan DTM on the IMS/AMS. No stolen or cached credentials were required other than installing the trojan. Since DTMs must be installed using administrative privilege, this attack may be achieved through social engineering, the supply chain, or malicious insider attack. Because credentials were not necessary for compromise, tests that attempted to steal and reuse IMS/AMS credentials were not attempted.

The test team was able to use a compromised and remotely controlled IMS/AMS platform to make unauthorized device changes in the absence of a protection that would block the associated commands.

Can an attacker intercept a safety instrument password by using keystroke analysis, memory leakage, or network sniffing?

Yes. In all evaluated instruments that have passwords or passcodes, the passwords are sent in clear text between the IMS/AMS and instruments and can therefore be intercepted through network sniffing. The test team was also able to capture instrument passwords using a keylogger on some IMS/AMS platforms.

Architecture 1: Because the IMS/AMS is on the PCN along with the SIS, network sniffing is feasible. Some SISs have options to encrypt communications between the IMS/AMS and the SIS. When encrypted communications were used, this attack was mitigated.

Architecture 2: P12 considered a MUX solution that required a computer to connect to it using a serial connection (see Figure 1, Architecture 2). Eavesdropping with this type of MUX requires access to the serial communications and cannot be done directly from the network. MUX solutions that are ethernet networked may be subject to eavesdropping; however, no such MUX was evaluated as part of P12. These communications are not encrypted.

Can an attacker bypass any instrument physical lock or password to make any changes on the instrument?

It depends on what is being bypassed. Project 12 assumed the attacker had no physical access to deployed devices. Using only network access from the IMS/AMS, the test team was unable in the allotted time to bypass any physical instrument write-protect locks. The team was able to bypass all tested software-based write-protection mechanisms, including passwords, passcodes, and write-protect toggles implemented in software.

Evaluated devices implemented software write protections using HART device-specific commands. Unless a device hardware switch or SIS protective mechanism blocks these commands, passcodes can be bypassed, and unauthorized device changes can be made.

Can an attacker affect smart instruments using a vulnerability exploit?

Attackers can affect device state by exploiting the lack of authorization checking to execute unauthorized commands[3]. While no instrument vulnerability exploits such as buffer overflows were found given the limited scope and time of the assessment, a small percentage of the instruments were found to have input parsing errors. Parsing errors can cause vulnerabilities. The level of effort required to exploit instrument protection weaknesses is significantly less than that required to find and weaponize a parsing vulnerability.

Architecture 1: Some SIS solutions helped mitigate instrument input-based attacks by limiting the allowed argument size or verifying that the argument size matched the size specified in the command packets.

Architecture 2: Using a MUX to mediate communications mitigated some instrument input-based attacks by limiting the argument size allowed.

4.2.3 Evading Detection

Can an attacker change an instrument parameter to an unsafe setting while evading detection of the parameter change?

In many cases, yes. The test team was able to make device changes and, on all devices that supported it, reset the change bit to immediately acknowledge the change. This effectively kept IMS/AMS solutions from giving any visual indication of change. Detection was mostly limited to IMS/AMS logging that could be used after the fact in forensic analysis. The degree to which changes were logged and where the logging occurred varied significantly depending on the specific products used. In some cases,

[3] https://cwe.mitre.org/data/definitions/862.html

log entries inappropriately attributed changes to legitimate system components rather than to some unknown software. Alarming on changes was less common than logging.

Architecture 1: Some SIS solutions provided additional logging capability.

Architecture 2: The MUX provides no additional logs or alerts.

4.2.4 Potential Attack Effects

Project 12 explored the effects of attacks and whether any available protections would prevent attack success. Vendor solutions have a variety of available protections that can be used to prevent unauthorized changes to instruments. These include instrument write-protection features (jumpers, passwords, toggle modes), IMS password protections, and a range of common and unique SIS protective features.

Discussion of unique SIS protective features and device attacks are not included in this report to maintain vendor participant confidentiality. Safety system owners should contact their SIS vendors to learn more about available product-specific protection mechanisms.

For all questions in this section, the use of hardware-based device write-protection features blocked all attacks. Software-based device write protection and IMS/AMS password protections were bypassed and were ineffective in blocking attacks. In Architecture 1, attack blockage depended on the available and enabled SIS protective features. In Architecture 2, all attacks were possible.

Can an attacker cause the instrument to give a false reading (e.g., change the range on the instruments to send the wrong analog signal to the SIS)?

Project 12 demonstrated the ability to put devices into fixed current mode and send a false value to the SIS when not using hardware-based write-protections and when using Architecture 2 or Architecture 1 with no protections. Other attack avenues were also found. Some attacks leveraged HART common or universal commands and worked across multiple vendor devices.

Can an attacker force an instrument into commissioning mode so it will send the attacker-specified value to the SIS?

Project 12 demonstrated the ability to put devices into fixed current mode and send any specified value to the SIS when not using hardware-based write-protections and when using Architecture 2 or Architecture 1 with no protections. Fixed current mode is used during commissioning and plant maintenance.

Can an attacker cause an instrument to fail to execute authorized parameter and/or state update commands?

Project 12 demonstrated the ability to cause devices to be unreachable and therefore fail to execute parameter update commands when not using hardware-based write-protections and when using Architecture 2 or Architecture 1 with no protections. Some attacks leveraged HART common or universal commands and worked across multiple vendor devices.

Can an attacker cause an instrument to go offline or otherwise become unresponsive?

Project 12 demonstrated the ability to force some devices offline and cause them to be unreachable or become completely unresponsive when not using hardware-based write-protections and when using Architecture 2 or Architecture 1 with no protections. Some attacks leveraged HART common or universal commands and worked across multiple vendor devices.

Can an attacker change a device password?

Project 12 demonstrated the ability to change passcodes on all devices when using Architecture 2 or Architecture 1 when not blocking device-specific commands. If a passcode was already set, the test team was able to first guess the passcode and then change it.

Can an attacker lock the administrator out of controlling the instrument?

Project 12 demonstrated the ability to change passcodes on all devices when using architecture 2 or architecture 1 when not blocking device-specific commands. In some cases, passcodes can be set to strings that cannot be typed on a keyboard, making it even more difficult for an operator to regain control.

4.3 Safety Systems Architectures

Project 12 found little to prevent an attacker from making harmful changes to safety instruments when using Architecture 2 (MUX-mediated communications). The MUX used in Project 12 provided no protection against rogue device command execution. It provided some protection against long command strings intended to overload device input parsers. The only fully effective means of preventing unauthorized changes was using hardware switch device write protections.

Architecture 1 (SIS mediated communications) results varied depending on the SIS solution and whether and how SIS protective measures were used. No SIS provided full protection against the attacks. In general, using an SIS with one or more protective features reduced more risk than did a MUX-based solution. When SIS protective measures were not used, results were similar to those of Architecture 2. All of the SIS solutions filtered long command strings.

Table 3 on page 25 shows the effects of individual protective mechanisms and approaches in preventing unauthorized device configuration changes. These effects are derived from a meta-analysis of Project 12 findings and should be considered in the design of a secure safety system, as discussed in section 5.1.

5. Discussion and Recommendations

Project 12 uncovered numerous security-relevant issues across multiple parts of the industry. Specific recommendations are offered to different stakeholders with a goal of helping safety system owners improve the overall security of operational safety systems and manage risk.

5.1 Asset Owners

Safety systems must be protected from attack-induced dangerous and potentially catastrophic conditions. While a fully secure, zero-risk state can never be achieved, risk can be mitigated by applying multiple overlapping protections to reduce the overall attack surface, identifying gaps where residual risk exists, and monitoring and alerting for evidence of attacks that are trying to or have taken advantage of those gaps. This requires a disciplined, holistic approach to security design.

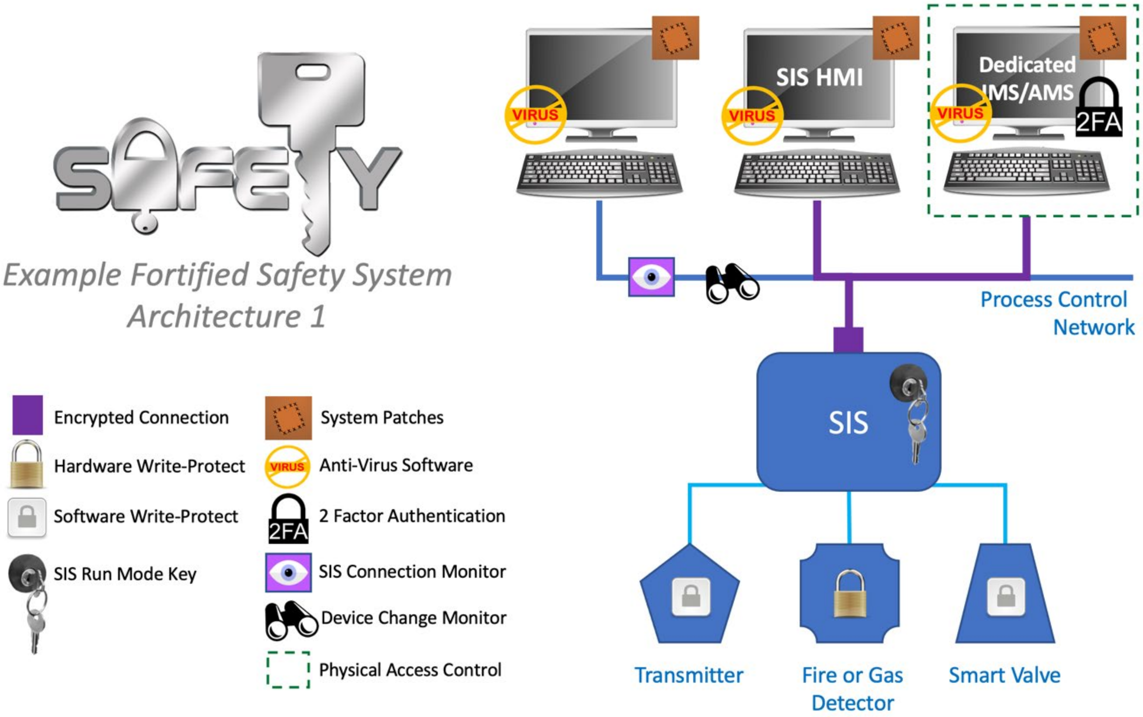


Figure 8. Manage risk by implementing a layered, defense-in-depth approach that fortifies safety systems and their operating environments against network-based, insider, and supply-chain attacks.

This approach to security orchestrates protective and monitoring mechanisms to fortify the overall system against concerted attack, as shown in Figure 8. Specific recommendations for asset owners to prevent attacks and monitor for attack attempts and effects are discussed in the following sub-sections.

5.1.1 Attack Prevention

Project 12 results clearly show that when SIS protective features are enabled, Architecture 1 poses less risk for unauthorized device modification than does Architecture 2. SIS features vary widely depending on the SIS and alone, are not enough to protect vulnerable devices.

Table 3. Effects of Different Device Protective Measures and Residual Risks

Security Mechanism or Approach	How It Works	Demonstrated to be Bypassable?	Attacks Stopped	Attack Launch Point	Devices Protected	Residual Gaps
Device write protection: hardware-based	Device will not process device update commands while switch is in protect position	No	All unauthorized updates sent to the target device	Anywhere, including when attached to MUX	Single	None for that device
Device write protection: software-based	Device will not process device update commands until device is unlocked by entering a passcode	Yes, by sniffing or guessing the passcode	Unauthorized updates sent to the target device	Anywhere on PCN	Single	None for that device
Device common and universal write protection: SIS enforced	SIS blocks HART common and universal write commands; device-specific commands are not blocked	Depends on the implementation; generally, no	Unauthorized updates sent to any device that use HART common and universal write commands	Anywhere on PCN	All	Unauthorized updates using HART device-specific commands
Device device-specific write protection: SIS enforced	SIS blocks all HART device-specific commands, including read commands, which negatively impacts operator view	Depends on implementation; generally, no	Unauthorized updates sent to any device that uses HART device-specific commands	Anywhere on PCN	All	Unauthorized updates using HART common and universal commands
IMS/AMS user authentication	IMS/AMS will not send device update commands	Yes, by running co-resident malware on the IMS/AMS host platform	Unauthorized updates through hands-on access to the IMS/AMS	IMS/AMS platform	All	All
Limit device connections to only authorized hosts	Whitelisting or required authentication mechanism blocks connections from unauthorized hosts	For whitelisting, possibly via spoofing	All updates sent to any device from unauthorized hosts	Any unauthorized host on PCN	All	Unauthorized updates from authorized hosts (e.g., IMS/AMS)
IMS/AMS to SIS or SIS proxy communications encryption	Uses public/private key exchange to authenticate senders and receivers; encrypts network-based communications	Not from network Host-level encryption: co-resident malware on the IMS/AMS platform can use encrypted tunnel Application-level encryption: trojan DLLs loaded in the IMS/AMS address space can use encrypted tunnel	Modification of device updates in transit Connecting directly to SIS from unauthorized host and/or application and sending unauthorized device updates	Any unauthorized host (or application, if using application layer encryption)	All	Host-level encryption: co-resident malware on IMS/AMS platform is able to make unauthorized changes Application-level encryption: trojan DLLs that are loaded by the IMS/AMS are able to make unauthorized changes

Table 3 shows common protections examined during Project 12 and their effects on attacks. It does not include additional SIS product unique protections, which may provide additional attack coverage. No protective measure was found that could comprehensively protect the whole safety system. For example, the strongest protection was the hardware device write-protection, but only 33% of sampled devices had hardware-implemented switches. Other write-protection mechanisms can be bypassed with a low level of effort.

The combined use of several security mechanisms is therefore necessary to provide comprehensive protection.

The most effective combined set of mechanisms to prevent unauthorized safety system device changes are hardware-based write-protection mechanisms, limiting change requests to only authorized applications on authorized hosts, and encrypting communications from the IMS/AMS to the SIS. None provides 100% protection across the whole system, but together, the three can greatly reduce the attack surface.

Project 12 found that many security mechanisms can be bypassed with little effort. Asset owners should contact vendors about addressing weaknesses in security mechanisms and use strong procedures to mitigate risk in the interim. For example, reducing the possibility of DTM trojans being introduced into the environment would shore up IMS/AMS application-level encryption and increase effectiveness in thwarting attacks.

Recommendations for protective measures within specific areas of safety system architectures follow.

Instrument and Asset Management Solutions. The IMS/AMS solution is a trusted platform in safety systems architectures. Because of this, the platform can be used to launch any of a number of attacks against the safety system and anything connected to the PCN. It is therefore of vital importance to protect this platform using security best practices and limit exposures to potential attacks.

Recommendations

1. Use a dedicated IMS/AMS solution. Do not install any software on the system other than that needed to manage the devices and provide security to the system (e.g., antivirus).
2. Keep the IMS/AMS system patched with the latest vendor patches and use updated antivirus software.
3. Scan all files and software, including vendor-provided software and updates, with the latest virus signatures before loading onto the IMS/AMS.
4. Device DTMs and DDs
    a. Do not use device DTMs in safety-critical applications unless they are absolutely required. Require vendors to securely provide cryptographic hashes to verify the integrity of DTM installers and to sign all individual DTM files. Verify the authenticity and integrity of all currently used and new DTMs before installing a DTM in the safety-system environment.
    b. Use vendor-provided DD files for device management where possible. Like DTMs, check the integrity of the DD installer. Input parsing bugs are a major cause of software vulnerabilities and can be exploited by using input files with corrupt contents. Project 12 did not test for input parsing bugs in IMS/AMS solutions.
5. Limit system console access and authorizations to reduce opportunities for insider attack. Apply access control best practices.
    a. Use strong authentication for system logins. Use two-factor authentication where possible.

   b. Practice “least privilege access.” Users should have unique, non-shared non-administrator accounts for normal operations. Administrator access should only be used during IMS/AMS system maintenance performed by authorized personnel.
   c. Audit system logins for accountability.
   d. Limit physical access to the system to prevent malicious insiders from installing malware on it. Maintenance should only be performed by vetted individuals.
6. Consider turning off the IMS/AMS when not in use to limit network-based attack exposure time. If following this recommendation, implement and audit a manual procedure to keep the system patches and virus signatures up to date.
7. Create a safe computing environment for the IMS/AMS by applying the latest system and software patches and antivirus signatures to all other network connected systems. Prevent remote access by using network segregation or a host-based firewall.

Device Write-Protection Enforcement. Generally speaking, the farther from the device that write-protection is placed, the easier it is to bypass. The use of built-in hardware-based device write protection is the only fully effective means of preventing unauthorized changes; however, not all devices have this feature. Software-based device protections require little effort to bypass and make unauthorized changes. Most SIS solutions can block HART common and universal write commands, but device-specific write commands cannot be blocked without also blocking device-specific read commands. IMS/AMS enforcement is useful for keeping unauthorized persons from using the system console to make device changes but, it is not useful in preventing co-resident malware from making these changes. Bypassable device write-protections pose significant risk. It is therefore important to enforce device write-protections as close to the device as possible.

Recommendations

1. Follow the IEC 61511-1 standard, which requires that all SIS devices have write-protection.
2. For devices that have hardware-based write protections, write protect the devices using the hardware switch, even if the device also has a software-based password. Only disable the hardware protection when maintenance requires changing device settings.
3. For devices that only have software-based write protections, use SIS-enforced write protection, if available. If the SIS supports blocking device-specific commands, consider procedurally blocking those commands when not using the IMS/AMS to reduce the opportunities for unauthorized device-specific changes.
4. Use software-based passwords as a last resort.

SIS Connections and Network Communications. Because devices assume all received HART commands are legitimate, it is important to restrict which hosts and applications are able to send commands to the devices. While it is relatively easy in some safety solutions to modify HART commands in transit, it is far simpler to send unauthenticated HART packets to devices. Use of correctly implemented encrypted communications has two important properties: it authenticates the sender and receiver, and it protects the integrity of the HART data being sent between the IMS/AMS and the SIS.

Recommendations

1. Use an SIS to mediate all communications between the IMS/AMS and the devices it manages. If a MUX solution must be used, do not connect the IMS/AMS to the PCN.
2. Implement encrypted communications between the IMS/AMS and devices to prevent attacks on communication integrity and confidentiality.
    a. Use application-level encryption, if supported by the safety system solution, to prevent IMS/AMS co-resident malware from making unauthorized changes.
        a. If the safety system only supports host-level encryption, use that to prevent attacks on network-based communication integrity and confidentiality.
        b. Where possible, configure the system to always require encrypted sessions and use bi-directional certificate-based authentication to validate the SIS and the IMS/AMS.
        c. If the safety system does not support encrypted communications, consider using a proxy solution that establishes a VPN between authorized hosts and a point in front of the safety instrumentation (and the SIS, if using one).
        d. Encrypting communications complicates content-based network monitoring. If communication content monitoring is required, use null-cipher encryption to enforce authentication between the SIS and the IMS/AMS and protect network packet integrity while not encrypting packet content. Communication confidentiality attacks, including password interception, are possible with this configuration, so alternative mitigations must be used.
3. Implement a mechanism to limit connections to the devices from the minimally required set of hosts. If the SIS solution can provide this function, enable it there. Otherwise, consider an SIS proxy that can restrict allowed connections to designated hosts.
4. SIS solutions offer a wide range of features that can help reduce risk. Typically, these features are not enabled by default. Work with your vendor to understand available options and to select and correctly implement the right ones for your environment.

5.1.2 Attack Detection

Layered protections can contain gaps that attackers can get through. It is important to identify if and when this occurs so that appropriate actions can be taken. Attacks generate evidence that can be detected during active attacks or after attacks complete. Actively monitoring for this evidence can result in recognizing an attack more quickly and, potentially, before serious negative consequences occur.

Evidence and other information (e.g., maintenance logs) that can be important to forensic analysis exist in many locations within the safety system, including on the IMS/AMS, the SIS, and on the network.

Instrument and Asset Management Systems. IMS/AMS solutions are the main network interface for configuring and managing safety system instrumentation. Many changes are made from this platform, although changes can also be made from handheld devices using wireless HART. The auditing and logging features of IMS/AMS solutions varies widely across all tested solutions. Some solutions create detailed logs of device changes while others create no logs. Some also provide visual indicators when a device's configuration has changed.

Recommendations

1. Work with vendors to understand available IMS/AMS logging and alerting capabilities and how the captured information can be used to determine if unanticipated device states are the result of changes made using the IMS/AMS.
2. Enable audit logs on the host operating system to capture account login, file write, and process execution events.
3. Periodically save IMS/AMS and host operating system logs offline for post-event forensic analysis, if needed.
4. Consider running a file system and registry integrity checker on the IMS/AMS platform to raise awareness when changes are made. Investigate unexpected changes to determine if they are the result of malicious or benign actions. This will help catch the installation of malware earlier, perhaps before serious damage can occur.

SIS Connections and Network Communications. Devices assume all HART commands are valid and will execute any commands received. Because of this, unexpected connection attempts to the SIS should be monitored and investigated.

Recommendations

1. Implement a PCN network monitor and log all connection attempts to the SIS. Alert on attempts from unauthorized sources. This can be done whether or not communications are encrypted.
2. Implement PCN network packet inspection and log all HART commands sent to devices for execution. Alert on any commands that affect security features of devices or make changes that could be unsafe. This monitoring approach relies on packet content analysis and can only be done on unencrypted communications. If communications between the IMS/AMS and the SIS are encrypted, this monitoring is not possible; however, it can be done for unencrypted communications from other sources.

Device State. Project 12 centered around an attacker's ability to affect device state by making unauthorized configuration changes. Because this is possible and can be done surreptitiously, it is important to constantly monitor devices for unexpected configuration changes or state. Some IMS/AMS solutions have a partial ability to do this through log files and other means. However, because the IMS/AMS platform can be compromised, LOGIIC recommends using an additional, independent device state monitoring mechanism.

Recommendations

1. Work with the SIS vendor to determine what, if any, capabilities the SIS may have that could help with detecting unexpected device state changes.
2. Consider implementing a solution that snapshots device configurations after maintenance updates, periodically compares the current device state to the snapshot, and alerts on changes. Automation of this analysis task can help operators discover harmful changes before damage occurs.

5.1.3 Safety System Operators

Humans are often the weakest link in the security of systems. They can be fooled by social engineering and accidentally introduce malware into systems. This risk is typically managed by training, limiting accesses, and implementing security controls.

Recommendations

1. Keep a written maintenance log of all changes to the IMS/AMS including vendor software installation and patches. This information can be useful for determining if and when malware was introduced into the environment.
2. Train operators on the unique security-relevant features of all components of safety systems. This includes how to properly write-protect devices and how to review audit logs for potentially unauthorized device changes.
3. Develop a robust safety system security policy and procedures guide. The guide should address relevant security issues such as not sharing user accounts, acceptable and unacceptable uses of the IMS/AMS system, acceptable ways to install software on the system, and how and when to conduct system security maintenance and auditing. The policy should clearly specify how to handle DTMs and verify their integrity before installation on the IMS/AMS.
4. Develop safety system security training materials. Train all operators on the security policies and procedures. Provide refresher training annually and when security-relevant changes are made to the system. Training materials should address relevant security issues such as phishing and social engineering and should include consequence-based examples of how attackers can use insiders to install malware. Examples include using the IMS/AMS to directly connect to the internet, downloading software from untrusted sites, and using USB thumb drives from untrusted sources.

5.2 Safety System Project Vendors

Safety system standards and designs should be evaluated and refined to address modern realistic attack paths and motivated attackers.

5.2.1 Instrument and Asset Management Solution Vendors

The IMS/AMS solution is a trusted platform in safety systems architectures. If compromised, the platform can be used to launch any of a number of attacks against the safety system and anything connected to the PCN. The introduction of 3rd party DTM or DD files poses significant risk to the platform. In addition, solutions that rely on backend databases are especially at risk if the databases can be modified without authentication. IMS/AMS solutions can be improved to reduce the risks of compromise and attacks on the PCN.

Recommendations

1. Provide guidance to asset owners on how to install and configure a secure IMS/AMS platform (both base operating system and applications) using security best practices.
2. Provide thorough documentation, training, and hands-on support as needed to help asset owner operators understand logging and other features that may be useful in discovering and understanding unexpected device changes.

3. Implement load-time signature verification of all DD/DTM plug-in DLLs. Do not load DLLs that fail the verification test.
4. Implement a mechanism to detect out-of-band device changes and display those changes in the IMS/AMS so the correct device state is always displayed.
5. Implement a bird's eye device state view that helps the operator to quickly see devices that have changed. Provide a drill down capability for quick access and assessment.
6. Follow software development and cyber security best practices.
   a. Consider reducing the complexity of solutions as system complexities are a major contributor to exploitable software and composition bugs.
   b. Conduct an end-to-end system risk assessment, looking specifically at component interfaces and compositional interactions and implications.
   c. Perform a comprehensive analysis of component authentication across the integrated IMS/AMS solution. Ensure that all components authenticate to each other to prevent the insertion of trojan components into the system or direct interaction with malicious actors.
   d. Use software code analysis tools on all software components to find and remediate software vulnerabilities prior to release.
   e. Perform exhaustive fuzz testing and ensure that input parsers function correctly on all boundary cases. Pay specific attention to DD file parsers as an attacker could use a malicious DD file to exploit any parsing bugs in this part of the IMS/AMS software.
   f. Secure any backend database to prevent unauthorized access and modification by other co-resident or network-based processes. Keep software patched and use appropriate access controls. If the database is co-located with the IMS/AMS software, block external network connections using the Windows host firewall or another similar mechanism.

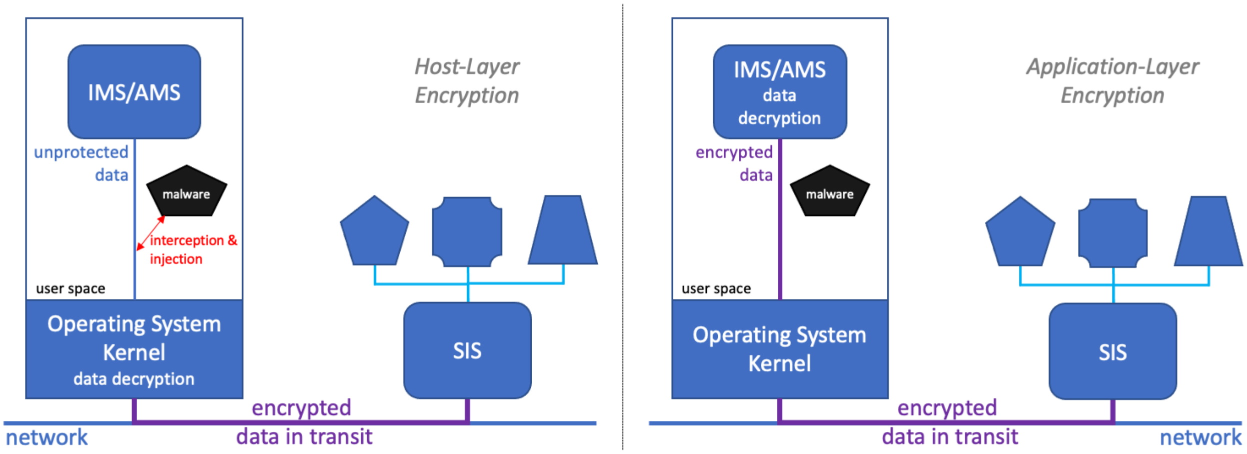


Figure 9. While host-to-host encryption (left) protects against network-based attacks, it does not protect against co-resident malware on an IMS/AMS. Malware can be installed on the platform through trojan DTM installers and other social engineering and lifecycle attacks. Application layer encryption (right) is necessary to protect against malware with this vantage point.

7. Assess your IMS/AMS solution design for common exploitable weaknesses as found in the MITRE CWE[4]. Address weaknesses to shore up your system.
8. Engage a cryptographic application expert to implement secure HART-IP as specified in the HART Network Management Specification Revision 3.0. This standard specifies application-layer authentication and encryption, as shown on the right in Figure 9, and addresses a number of common architectural weaknesses. A cryptographic application expert can help ensure a correct implementation that does not introduce additional architectural weaknesses.

5.2.2 Safety Instrument Product Vendors

Safety instruments assume that all HART commands received are valid. Because of this, attackers can modify device configurations by sending the desired HART packets to the device. Write-protect mechanisms are implemented inconsistently across the industry, with many devices having weak software-based password protections that can be bypassed by attackers using little effort.

Device DTM files are needed to configure device-specific features through the IMS/AMS; however, industry standard practices for distributing and installing these files provides a rich attack path for attackers to install malware on the IMS/AMS platform.

Instrument vendors can take actions that will help reduce these and other related risks.

Recommendations

1. Implement a non-bypassable physical write-protect switch on all new products. If the physical switch does not block all write commands, provide clear guidance to customers on which commands are not blocked so that asset owners can determine how to handle any posed risk.
2. Avoid using commands that require no arguments and change the device state. These device-state changes can be accidentally or purposely triggered, causing the sensor state to change in an unknown way.
3. Document all device-proprietary commands to enable operators to devise ways to detect the execution of potentially dangerous commands.
4. Log device set-point changes to a historian to increase visibility and aid in post-attack forensic analysis.
5. Work with the HART standards body and other vendors to develop standard device security features and standard configuration commands.
6. Establish your organization as a verified Microsoft publisher to enable end customers to verify the publisher of DTM and DD installers.
7. Create cryptographic hashes of all DD/DTM installers and provide them to end customers through secure channels. Provide instructions on how to verify the hash, and thus, the integrity of the installer.

[4] https://cwe.mitre.org/data/definitions/1008.html

8. Cryptographically sign individual DLL files and configuration files. Provide a means for IMS/AMS solutions to verify the signatures.
9. Use software analysis tools on all software components to find and remediate software vulnerabilities prior to customer distribution.
10. Perform exhaustive fuzz testing and ensure that input parsers and logic functions correctly on all boundary cases.
11. Assess device designs for common exploitable weaknesses as found in the MITRE CWE[5]. Address weaknesses to shore up your devices.

5.2.3 Safety Instrumented Systems Vendors

SIS solutions are complex. Each is completely different in design and operations and provides a unique set of features that can be used to prevent a subset of unauthorized device reconfigurations. SIS vendors can help asset owners manage risk in their environments in a number of ways.

Recommendations

1. Provide thorough documentation, training, and hands-on support in securely configuring safety systems using your products.
2. Provide clear information to customers regarding any device write commands are that are not blocked when SIS protection is engaged.
3. Implement connection and communications security features that allow asset owners to designate systems as device managers. Device manager systems should be allowed to configure devices through the SIS while other systems are allowed only to read device configurations and status.
4. Implement an SIS capability to detect and alert on device conditions of interest. Provide clear guidance and training on how to configure this feature.
5. Use software analysis tools on all software components to find and remediate software vulnerabilities prior to customer distribution.
6. Perform exhaustive fuzz testing and ensure that input parsers and logic functions correctly on all boundary cases.
7. Assess your SIS solution design for common exploitable weaknesses as found in the MITRE CWE. Address weaknesses to shore up your system.
8. Engage a cryptographic application expert to implement secure HART-IP as specified in the HART Network Management Specification Revision 3.0. This standard specifies application-layer authentication and encryption, as shown on the right in Figure 8, and addresses a number of common architectural weaknesses. A cryptographic application expert can help ensure a correct implementation that does not introduce additional architectural weaknesses.

[5] https://cwe.mitre.org/data/definitions/1008.html

5.3 Standards Bodies

Cyberattack risks can directly and negatively affect safety. Safety system standards bodies can play an important role in cyberattack risk management by evolving standards to move the industry in a direction that reduces cyberattack risk and improves the overall security and safety of operational safety systems.

5.3.1 International Electrotechnical Commission (IEC)

The IEC 61511 standard requires manufacturers and suppliers of devices for SISs to conform to the IEC 61508 standard. The IEC 61508 Functional Safety for Safety Related Systems standard requires that instruments have a write protection mechanism, key lock, or dedicated tool with password, to be rated for use in safety systems. The evaluated password-based systems were bypassed with little effort. Hardware write-protect switches were not bypassed, but only 33% of sampled devices had hardware switches.

Recommendations

1. Evolve the IEC 61508 standard to require write-protection mechanisms that cannot be bypassed.
2. Examine the IEC 61511 standard and reinforce the requirement for non-bypassable write protection.

5.3.2 DTM Standards Body

DTMs are necessary for configuring device-specific features. These plug-ins are DLLs that are loaded and executed in the process space of trusted IMS/AMS software. Insertion of a trojan DTM is easy and was demonstrated numerous times during Project 12. DDs pose less risk in that they do not contain executable code; however, both DTM and DD installers are executable code that can install malware along with the DTM and DD files. The DTM and DD standards should be evolved to reduce the risk of trojan and other malware insertion in safety system environments.

Recommendations

1. Require all DTM and DD installers to use a verified Microsoft publisher.
2. Encourage all vendors to provide DTM and DD installer cryptographic hashes to asset owners.
3. Require all DTM and DD DLLs and configuration files to be cryptographically signed.

5.3.3 HART Standards Body

Wired HART and HART-IP are open protocols that are easy to manipulate using off-the-shelf tools. Protocol design deficiencies complicate monitoring for and preventing attempts to make unauthorized device changes.

Secure HART-IP[6] was introduced mid-way through P12 and was not included in testing. After P12 completion, the test team conducted a paper-based analysis that compared the revised standard with HART and HART-IP recommendations. The team found that the secure HART-IP standard addresses LOGIIC recommendations of using application-layer

[6] https://library.fieldcommgroup.org/20085/TS20085/3.0/#page=1

encryption between the IMS/AMS and the SIS and would, therefore, have a significant, positive impact on safety system security. The wired HART protocol needs changes to address other issues found in P12.

Recommendations

1. Augment the HART protocol command specifications to include a means to differentiate device-specific read and write commands. This will enable external protection mechanisms to block write commands while not blocking read commands.
2. Work with the vendors to develop standard HART commands to configure security relevant mechanisms (e.g., software passcodes, logging or histories, and configuration-changed bit.) Consider adding these commands as an explicit “security” type command. Include a command that returns the list of security features provided by the device.
3. Engage cybersecurity experts in protocol analysis to thoroughly review the wired HART protocol design and evolve the protocol to a more secure design.

5.3.4 Vendor Opportunities

Project 12 revealed gaps in key areas that present opportunities for vendors. New or repurposed technology can fill some of these gaps. Technologies that could be of benefit include

1. A multi-device state-change detector.
2. A ruggedized safety system security gateway to be placed in front of any SIS or MUX. Desired features include:
    a. A way to designate authorized sources for device management and device monitoring.
    b. A passthrough for SIS HMI for non-HART commands.
    c. Methods to enforce authentication of sources and encrypt communications.
    d. The ability to inspect received HART commands and log all device changes and the source of the changes.

6. Summary

This report captures the scope, approach, method, and findings of LOGIIC Project 12, along with test team findings and recommendations.

Project 12 testing was limited in time and scope and conducted using partial knowledge. Concerted adversaries have ample time and resources to plan attacks, sometimes years in advance. Undocumented commands and firmware vulnerabilities are often discovered by attackers over the course of many months. The SMEs in this assessment devoted the majority of their efforts to examining documented system features and how to abuse those features to achieve attack goals. Even with these limitations, Project 12 revealed numerous consequential and recurring exploitable weaknesses across individual assessments that indicate a systemic and pervasive industry-wide problem. This issue is mainly a consequence of four critical findings:

1. Some safety system designs allow unchecked HART passthrough
2. The current HART and HART-IP protocols have no built-in security
3. Devices do not authenticate the sources of received HART commands and many have bypassable write-protections
4. The industry uses unverified 3rd party software downloaded from the Internet

These combined findings create a situation where little prevents an attacker from making harmful changes to safety instruments when using a passthrough MUX. SIS product protections vary widely, with each providing a unique set of features that can help mitigate some risks of unauthorized device configuration changes.

Successfully demonstrated attacks used a number of commonly available attacker tools and exploited common-knowledge architectural weaknesses [7] that were present in all four assessments. These attacks required a low-to-moderate skill level and included effects that can negatively and significantly impact device safety functions.

Critically, Project 12 found that third-party DTMs used by IMS/AMS solutions to control instruments pose a significant risk for IMS/AMS platform compromise. Loading these software packages on the IMS/AMS platform bypasses any air gap, potentially placing malware directly into the process-control environment. This malware can then take advantage of the IMS platform's trust relationship with the SIS or MUX and the critical findings (above) to launch attacks against the safety system and other PCN-connected systems. ***We cannot sufficiently emphasize the severity of this vulnerability.***

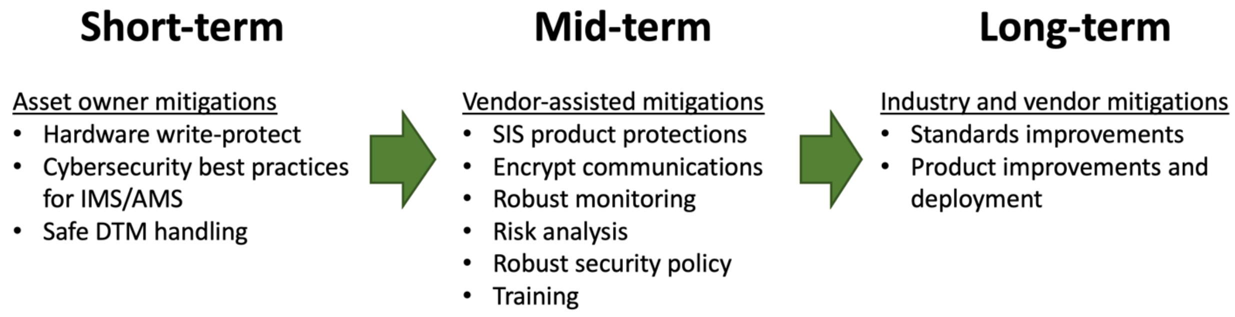
Additional high-level findings include

- Attackers can make unauthorized device changes at will while evading detection. Some changes can result in unsafe operating conditions; therefore, the risk of cyberattack directly impacts safety and must be considered along with hardware faults and other safety considerations.

[7] MITRE Common Weakness Enumeration (CWE) database. https://cwe.mitre.org/data/definitions/1008.html

- No simple and immediate remedy is available to secure safety systems. A combination of protection and detection measures is required.
- Device hardware-based write protections are the only fully protective means to prevent unauthorized device configuration changes from the network, but only 33% of sampled devices had hardware switches. Because software-based write protections can be bypassed, they do not provide protection against these changes. SIS write protections effectively prevent some, but not all, changes.
- Device write-protection implementations are inconsistent, even across the same vendor's products. This can lead to confusion and devices left accidentally unprotected.
- Safety systems architectures that use a MUX to mediate communications between the IMS/AMS and safety instruments are inherently insecure. Use of an SIS to mediate these communications can provide more protection if SIS protective features are used. If SIS protections are not used, the attack risk is equivalent to using a MUX.
- HART 5 and 7 protocol design deficiencies complicate monitoring for and preventing attacks from making unauthorized changes.
    - The protocol lacks basic security concepts (e.g., authentication, packet integrity).
    - The protocol's common and universal command sets do not include security relevant commands, which leads to inconsistent implementation across devices using device-specific commands. This complicates monitoring for attempts to circumvent device security features.
    - The protocol provides no means to differentiate device-specific read and write commands, making it impossible for any SIS to block device-specific write commands without also blocking read commands. Blocking device-specific commands prevents the IMS/AMS from displaying device-specific status, so it is rarely done in practice.

Project 12 concludes that the safety environment is vulnerable to undetectable malicious attacks and that extreme caution should be taken before installing any software, including DTMs, that could introduce malware into the PCN.



Short-term

Asset owner mitigations

- Hardware write-protect
- Cybersecurity best practices for IMS/AMS
- Safe DTM handling

Mid-term

Vendor-assisted mitigations

- SIS product protections
- Encrypt communications
- Robust monitoring
- Risk analysis
- Robust security policy
- Training

Long-term

Industry and vendor mitigations

- Standards improvements
- Product improvements and deployment

Figure 10. LOGIIC Project 12 Recommended Risk Mitigation Roadmap

The risk of cyberattack against safety systems can be addressed through a series of short-, mid- and long-term actions as shown in Figure 10. The industry has the opportunity now to plan for and address these issues in all stages of the safety system lifecycle. In the short-term, safety system owners should

- Follow the IEC 61511-1 standard, which requires that all SIS devices to have write-protection. Use hardware write-protect switches on all devices that have them. Disable switches only when conducting maintenance.
- Apply security best practices to the IMS/AMS platform to prevent attackers from exploiting its trust relationship with the SIS to launch attacks. Use network segregation or a host-based firewall (e.g., Windows 10 Security firewall) to prevent remote access.
- Avoid using vendor DTMs in safety-critical applications where possible, opting instead for device description (DD) files. Where DTMs are currently in use, verify the pedigree and integrity of all DTMs files. Obtain DTM and DDs directly from vendors. Request a cryptographic hash to verify the integrity of all DTM and DD installers. Ask that vendors sign all individual files. Verify DTM and DD integrity before installation on IMS/AMS platforms. Insist that any DTMs or DDs downloaded from the Internet use HTTPS.

Based on Project 12 findings, these mitigations will substantially reduce the risk to safety systems. In the midterm, safety system owners should

- Use the SIS to mediate communications between IMS/AMS solutions and safety instruments whenever possible. Work with the SIS vendor to identify and implement SIS-specific protective measures to reduce the available attack surface and, therefore, risk.
- Implement a means to allow only authorized hosts to make SIS network connections to prevent unauthorized hosts from making changes.
- Encrypt communications between the IMS/AMS and SIS where possible to avoid network-based attacks that steal passwords and change device commands in transit.
- Implement a robust monitoring system to detect and alert on device changes and on unexpected device states.
- Conduct a full consequences-based risk analysis of all operational safety systems using Project 12 findings to identify any residual risk not mitigated by applied mitigations. Owners should identify and implement additional mitigations based on risk.
- Create a robust security policy for their systems. Operators should be trained on the policy and how to avoid inadvertently introducing malware into the environment.

Longer term fixes should address the larger problems that require vendor product and industry-level changes. These include implementing the secure HART-IP protocol, published as part of the HART Network Management Specification in July 2020.

6.1.1 Acknowledgements

Project 12 was developed and guided by the members of the international LOGIIC forum, who devote their time and expertise to conduct projects that will lead to improvements to cybersecurity in the O&G industry and in the ICS community in general. The Automation Federation serves as the LOGIIC host organization and provides the governance and legal framework for our efforts.

LOGIIC would like to thank the US Department of Homeland Security Science and Technology Directorate for providing leadership, vision, and commitment to enhancing cybersecurity in ICS. We would like to acknowledge the numerous vendors who fully cooperated in this project and provided equipment and many staff hours. This project could not have been done without the support of these vendors. Finally, we would like to thank the Project 12 test team, who fleshed out the evaluation strategy, performed the system evaluations, and authored technical reports.

The work performed on this project by SRI International and its subcontractors was funded by the U.S. DHS S&T Directorate under Contract No. HSHQDC-16-C-00034.

6.1.2 Disclaimers

Findings must have been observed, be reproducible across multiple vendor products, and be common-knowledge architectural weaknesses as documented in the MITRE CWE[8] to be included in this report. Product-specific vulnerability information was provided directly to the appropriate product vendors and will not be disclosed by LOGIIC.

Project 12 was a time-based effort; the test team used a planned set of tests to identify as many commonly known weaknesses as possible in the allotted time. Because of this, exhaustive testing was not performed; therefore, the systems under consideration may contain additional vulnerabilities that were not discovered during this assessment.

The opinions, findings, conclusions, and recommendations expressed in this material are those of the authors, do not necessarily reflect the views of DHS and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DHS or the U.S. government.

6.1.3 Distribution

This report is approved by U.S. Department of Homeland Security and the LOGIIC Executive Committee for unlimited public distribution.

[8] https://cwe.mitre.org/data/definitions/1008.html

Appendix A. Acronyms

Term	**Definition**
AMS	Asset management system
BPCS	Basic process control system
CRC	Cyclic redundancy check
CWE	Common Weakness Enumeration
DCS	Distributed control system
DD	Device description
DHS S&T	Department of Homeland Security, Science & Technology Directorate
DoS	Denial of service
DTM	Device type manager
EWS	Engineering workstation
HART	Highway Addressable Remote Transducer protocol
HART-IP	HART over Internet protocol
HMI	Human machine interface
ICS	Industrial control system
O&G	Oil and Gas
IEC	International Electrotechnical Commission
IMS	Instrument management system
LOGIIC	Linking the Oil and Gas Industry to Improve Cybersecurity
MUX	Multiplexor
PCN	Process control network
RoE	Rules of engagement