# Getting to the HART of the Matter

## An Evaluation of Real-World Safety System OT/IT Interfaces, Attacks, and Countermeasures

Laura S. Tinnel
SRI International
Arlington, VA, USA
laura.tinnel@sri.com

Mike Cochrane
TotalEnergies
London, UK
mike.cochrane@TotalEnergies.com

## ABSTRACT

This paper discusses our experience evaluating attack paths and security controls in commonly used, real-world ICS safety system architectures. Specifically, we sought to determine if an SIS-mediated architecture could provide better protection against unauthorized and malicious safety instrument configuration changes than could a MUX-mediated architecture.

An assessment question-driven approach was layered on top of standard penetration assessment methods. Test cases were planned around the questions and a sample set of vendor products typically used in the oil and gas sector. Four systems were composed from different product subsets and were assessed using the test cases. We analyzed results from the four assessments to illuminate issues that existed regardless of system composition.

Analysis revealed recurring vulnerabilities that exist in all safety systems due to issues in the design of safety instruments and the HART protocol. We found that device-native hardware write-protections provide the best defense, followed by SIS write protections. We concluded that, when using SIS security controls, an SIS-mediated system can protect against unauthorized device reconfigurations better than can a MUX-based system. When SIS security controls are not used, there is no added security benefit.

We present lessons learned for ICS stakeholders and for people who are interested in conducting this kind of evaluation.

## CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; **Systems security**; **Access control**.

## KEYWORDS

Industrial control system, safety instrumented system, safety instruments, asset management, countermeasures, security controls, HART, cyberattack, assessment methodology

**ACM Reference Format:**
Laura S. Tinnel and Mike Cochrane. 2021. Getting to the HART of the Matter: An Evaluation of Real-World Safety System OT/IT Interfaces, Attacks, and Countermeasures. In *CSET '21: Cyber Security Experimentation and Test, August 9, 2021, Virtual, CA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474718.3474726

*CSET '21, August 9, 2021, Virtual, CA, USA*

ACM ISBN 978-1-4503-9065-1/21/08…$15.00
https://doi.org/10.1145/3474718.3474726

## 1 INTRODUCTION

The Linking the Oil and Gas Industry to Improve Cybersecurity (LOGIIC) consortium studies cybersecurity issues in Industrial control systems (ICSs) that could impact operational safety. LOGIIC has conducted three projects focused on various aspects of safety systems. Two earlier projects focused on safety instrumented systems (SIS) controllers [5],[11]. The latest project focused on safety instruments and management [13],[14] and is the subject of this paper.

ICSs control high-risk physical processes in manufacturing and industrial facilities but do not sufficiently manage process risk. SISs independently monitor operations and take corrective actions to bring a system back to a safe state when pre-determined hazardous processing conditions arise [12]. If SISs do not perform their function correctly, the result can be catastrophic. For example, in 2005 a Texas refinery suffered an explosion and fire that killed 15 people and injured 180. The real-world consequences of such failures depend entirely on the system context. SISs are used globally in chemical and petrochemical processing, wastewater treatment, nuclear power production, and more, so consequences can be far-ranging. SISs rely on instruments (or devices) that provide the inputs needed (e.g., pressure, temperature, or valve positions) to make safety decisions about process state. SIS instrument attacks can prevent needed corrective actions from being taken or force a process shutdown unnecessarily, causing a denial of service. An attacker could, for example, change the safe limits on a pressure sensor which could cause the SIS to fail to take appropriate action. This is concerning because nation state actors have already targeted SISs [9], and device attacks are an easy way to cause physical harm.

Modern safety instruments provide smart features, such as valve partial stroke testing or advanced diagnostics. They are typically connected to an SIS using direct cabling and communicate via analog signals. Smart data is superimposed over analog communications using the Highway Addressable Remote Transducer (HART) protocol [6], which is the industry standard for safety instrument communications. HART enables safety systems to monitor and modify device configurations and states.

HART implements three types of commands for reading device state and updating function parameters: universal, common, and device specific. All HART devices are required to implement the universal command set. Common commands are implemented on many, but not all, devices. Device-specific commands are used for device unique features. One can communicate with devices through HART-based handheld devices, HART passthrough SIS I/O cards, or a HART passthrough multiplexor (MUX). In the latter two cases, an information-technology (IT)-based instrument or asset management system (IMS/AMS) can be used to manage devices over an internet protocol (IP)-based network using HART enveloped in HART-IP or other proprietary protocol. Deployments that use IT-based computers with operational-technology (OT) inherit all the typical cybersecurity issues associated with IT and enable attackers to use IT systems to affect OT system functions.

This paper discusses our experience evaluating attack paths and security controls in two commonly used, real-world safety system architectures. While our effort examined numerous issues with safety systems, this paper focuses specifically on vulnerabilities in HART-based devices, the HART protocol, and on security controls to prevent unauthorized device configuration changes. We evaluated four product types (SIS, MUX, IMS/AMS, and instruments) and three classes of instruments in the context of the two architectures to determine which architecture, if either, was able to better protect against attacks.

This paper presents some of the unique considerations and challenges we faced in conducting this assessment. We discuss our overall approach and methodology, threat model, product selection, measurements, test cases and test harness, and results. Finally, we present lessons learned, limitations of our work, and related efforts. We then draw some final conclusions. Full project results are included in the project's final report [13].

## 2 ASSESSMENT APPROACH

This project sought to understand 1) how attackers can compromise IT-based IMS/AMS solutions and use them to alter the configuration of OT-based smart instruments to create unsafe operating conditions, render instruments inoperable, and/or take control away from asset owners and 2) how to prevent such attacks using available security controls. These goals required developing a methodology that examined possible attacks against the IMS/AMS and instruments and the attack prevention efficacy of security controls available in safety system architectures typically used in the oil and gas (O&G) industry.

IMS/AMS compromise is possible; we demonstrated trojan vendor software during the assessments. This paper focuses on the portion of the evaluation that used a compromised IMS/AMS to make unauthorized device configuration changes.

We used a combination of penetration (pen) testing, hypothesis-based question-driven testing, and cross-cutting analysis for this assessment. A number of externally imposed constraints affected the planning and conduct of this evaluation. We will discuss these as needed to help the reader understand the evaluation design decisions.

### 2.1 Hypothesis and Questions

LOGIIC's hypothesis was that a safety system architecture where an SIS mediates communications between an IMS/AMS and the instruments it manages (Figure 1) can counter attacks better than can an architecture where a MUX mediates these communications (Figure 2). We sought to test this hypothesis and extract lessons that could help LOGIIC members and other safety system operators make good design choices to protect their operational processes.

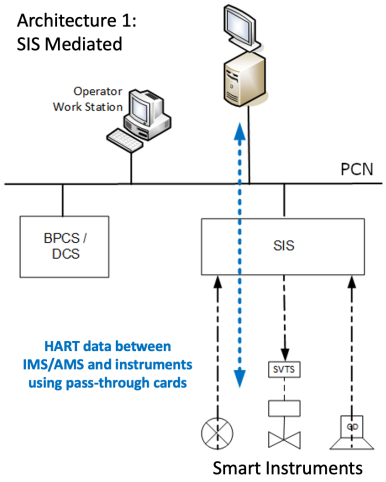


**Figure 1: Reference Architecture 1: The IMS/AMS and SIS communicate over the process control network (PCN). The SIS mediates communications between the IMS/AMS and devices.**

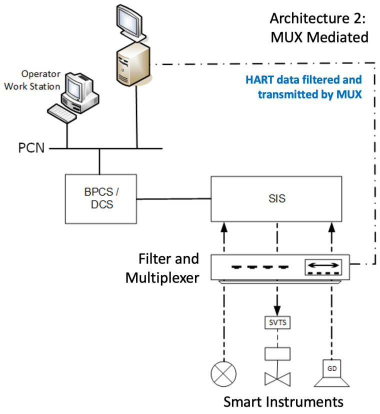


**Figure 2: Reference Architecture 2: The IMS/AMS is connected via serial cable to a MUX, which mediates communications between the IMS/AMS and devices. This may be required due to the network architecture or because the SIS does not support HART passthrough.**

To test the hypothesis, we needed to define the term "better" in a meaningful and objectively measurable way. "Better" could be defined simply as "one architecture blocks more attacks than does the other". This can be measured by creating a corpus of diverse, working instrument attacks and then running those attacks in the two architectures and comparing the number of successful and failed attacks. The result, however, may not be operationally useful, particularly in the case where one architecture blocks only one more attack than does the other. In such a case, the investment required to change architectures may not be warranted.

Another possible definition would be that one architecture blocks more critical attacks than does the other. However, "critical" depends entirely on the system deployment context, and we had no context for comparison.

There could be varying degrees of "better" (e.g., how much "better" as defined by some delta percentage of blocked attacks), but this is not so straight forward when considering combinations of security controls that may or may not be enabled. Ultimately, we measured the effectiveness of specific security controls in blocking attacks and then compared whether the controls were present or absent in a given architecture (e.g., SISs have the ability to block write commands to devices and hence, attacks, but the MUX-mediated architecture does not.)

Attack effects are important to provide meaning to results. We worked with LOGIIC to craft a series of binary, effects-based assessment questions of interest that, if answered affirmatively for the MUX and negatively for the SIS (when using security controls), would provide the evidence needed to support the hypothesis. Questions included the following:

(1) Can an attacker capture an instrument password?
(2) Can an attacker affect smart instruments by remotely controlling the IMS software?
(3) Can an attacker affect smart instruments using a vulnerability exploit?
(4) Can an attacker change an instrument parameter to an unsafe setting while evading detection?
(5) Can an attacker bypass instrument write-protection to:
    (a) cause the instrument to give a false reading?
    (b) force the instrument into commissioning mode so it will send any attacker-specified value to the SIS?
    (c) cause a device to fail to execute an authorized parameter or state-update commands?
    (d) force a device offline or to become unresponsive?
    (e) change an instrument password?
    (f) lock authorized users out of instrument control?

Our test case corpus was designed to achieve these effects.

## 2.2 Methodology

Industry standard security assessments are aimed at specific target network and product instances (e.g., an organization's internal network). In contrast, we needed to examine systems more generally and draw higher-level conclusions about reference architectures (or architecture templates) that could be instantiated using any combination of available products and configurations.

Standard assessments are typically conducted using a battery of pass/fail test cases that are based on common weaknesses in computer systems [1] and known vulnerabilities [2]. This approach works well when assessing specific instances of products or networks. Since our assessment focused on system architecture designs rather than on real systems, our effort required a different or augmented assessment approach.

We evaluated available attack paths and security controls in the two reference architectures by assessing a sampling of real products configured in different instances of safety systems. Evaluating a single pair of MUX- and SIS-mediated systems using one set of products would not suffice to draw any firm conclusions. Instead, we assessed multiple pairs and conducted cross-pair analysis to allow recurrent issues to surface (Figure 3.)

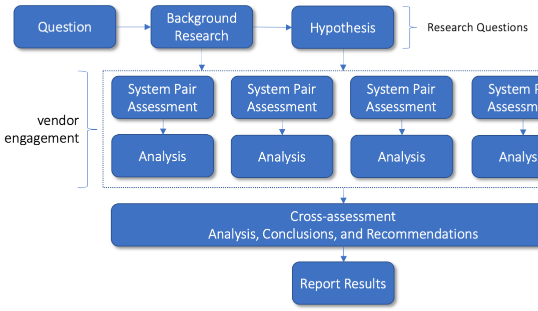


**Figure 3: The overall evaluation methodology assessed four pairs of instantiated safety systems and cross-analyzed results to identify issues common across all system pairs.**

The system pair assessment methodology is shown in Figure 4. We picked a set of products for the system instances; assessed relevant protocols; assessed individual devices using standard pen testing; identified working attacks of interest (as defined by the assessment questions); determined available device and SIS security controls; formulated attacks and security controls into test cases; instantiated the two reference architectures using the selected products; ran test cases in both instantiated systems with and without the available security controls and recorded which attacks worked and which were blocked; and analyzed results.

Upon completion of each system pair assessment activity, we compared the number of successful and failed attacks for each security control and the nature of the attacks.

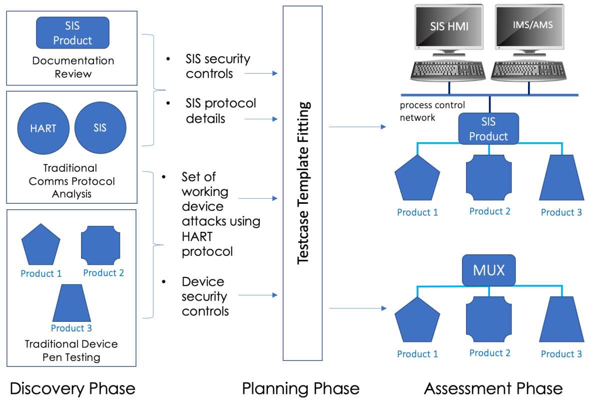


**Figure 4: The System Pair Assessment methodology used standard industry assessment methods to flesh out test cases specific to the instantiated SIS- and MUX-mediated safety system pair. The set of test cases was then executed in the two systems, and results were collected.**

**Special Considerations.** Our approach was sensitive to inconsistencies across the four system pair assessments, as each product had unique vulnerabilities and security controls. Standard vulnerability analysis to find product-specific vulnerabilities alone would not yield the consistency needed to support cross-assessment analysis, but it could be used in a discovery phase to determine available instrument attack surfaces. Those attack surfaces could then be fitted to pre-planned test case templates designed around the assessment questions. This allowed us to focus on consistency across assessment question effects and fit individual product vulnerabilities and security controls to those effects.

SIS vendors allowed LOGIIC to use test systems in their product labs for our individual assessments. The project could not have been done without this level of access but use of vendor labs meant that each safety system instance was tested in a different environment. We therefore sought logical consistency in the safety system instantiations in each test environment and required a week of test environment validation and sample data collection before starting the tests.

Finally, a high degree of planning and controlled test execution was needed to ensure consistency. We used a templated test plan as the basis for all individual assessment activities. The plan included rules of engagement (RoE) designed to ensure that testing was consistent and rigorous across all individual assessments and that results were fully repeatable.

## 2.3 Threat Model

The project used a threat model to guide and limit the scope of the pen testing activities and test case design. Our attacker had insider-sourced knowledge of the operational safety system from an O&G company (e.g., specific vendor products and versions). This insider also provided physical access to the IMS/AMS platform and the network switch but had no other direct access.

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 product development lifecycle attacks by injecting malware into vendor firmware as the attacker did not have access within the vendor to perform such attacks. 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.

## 2.4 Product Selection

One of LOGIIC's goals is to help vendors improve the security of their products, which in turn, helps with the security of systems deployed by member companies; therefore, fostering good vendor relationships is essential. Surreptitiously acquiring and using products without vendor consent would hurt relationships and not result in the product improvements that LOGIIC seeks. Thus, this project was conducted with the full support and cooperation of safety system product vendors. This impacted the design of the evaluation in that the product sample set could not be random. Instead, it was representative of the products LOGIIC members use in their operations and was based on existing vendor relationships.

LOGIIC identified six product types for the evaluation and proposed candidates for each type. The final sample set of products was dictated by vendor decisions to participate (or not). Our goal was for each assessment to have a unique product in each category, with the exception of MUX; however, we were limited by vendor participation and some instruments were used in two assessment activities (Table 2).

Based on LOGIIC experience and a review of product literature, we concluded that MUX products are passthrough devices that do not provide protection, so using one MUX for all assessments would sufficiently represent MUX protective capabilities (or the lack thereof).

While our instrument sample size was small (nine out of hundreds of available products), all products use the HART protocol, and the same HART-sourced issues were found across the whole sample set. Based on a review of IEC device standards and identified

**Table 1: Threat Model Attacker Assets and Accesses**

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

**Table 2: Product Classes and Sample Sizes**

| Product Class | Sample Size |
|---|---|
| Instruments | 9 |
| Pressure and temperature transmitters | 3 |
| Fire and gas detectors | 3 |
| Smart valve positioners and solenoids | 3 |
| IMS/AMS solutions | 4 |
| SIS solutions | 4 |
| MUX solutions | 1 |

issues with the HART protocol, we are confident that additional instruments would suffer from the same problems.

## 2.5 Measurements

We measured the effectiveness of security controls in blocking attacks and then compared whether the controls were present or absent in a given architecture This allowed us to answer the assessment questions in the context of the available security controls in each architecture. We required at least one successful attack to answer a question in the affirmative, which eliminated the need to demonstrate all the possible ways that the question could be answered affirmatively. We then determined which, if any, security controls blocked each attack. We used a question-to-test case map to aid with this process. Test results were plugged into the matrix and a simple OR function applied (i.e., if any one test attack succeeded, the assessment question answer was “yes”.)

Write protection effectiveness was measured by the non-existence of unblocked write commands (from our command sample set), the inability to bypass the protection, and the lack of collateral damage. We used unblocked write commands because any such command could be abused by an attacker. Collateral damage was defined as any potentially adverse side effect from using a security control. For example, SIS blocking of device-specific commands caused a portion of the operator console to stop displaying device status information.

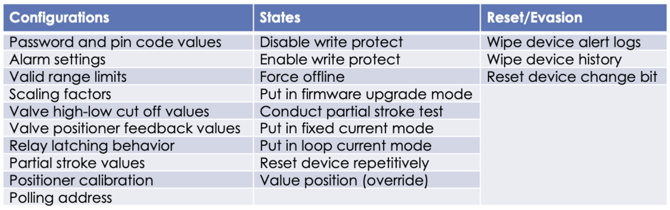
Communication encryption effectiveness was measured by its ability to cause an attack to fail and requirements for bypassing it. For the purposes of hypothesis evaluation, we did not consider specific encryption implementation vulnerabilities as we were focused on what encryption could do in the SIS architecture (if implemented correctly) rather than on specific product solutions. Use of binary assessment questions mirrors standard product security assessment measures, which are typically pass/fail. These are simple and relatively easy to measure. Often in security assessments, binary measures are rolled into a risk score or metric. We did not attempt to do this because 1) our goal was to determine which of two designs provided better protection rather than “how secure” each architecture is, and 2) risk is entirely dependent on the operational context impacted by attacks. Because safety systems are used in a broad range of applications, the context (and therefore risk) can vary greatly. Furthermore, we were not assessing real deployed systems, so there was no operational context to consider.

## 2.6 Test Cases

Test cases were to be 1) useful in achieving one or more assessment question effects, 2) based on the threat model, and 3) traceable and reproducible. Test cases were documented step-by-step and automated to the extent possible so that vendors could reproduce the test results in their own labs. We planned test cases for instruments, communications, and systems. Instrument and communications tests were used in system test cases.

**Instrument test cases** focused on command abuse. Each instrument was examined during the discovery phase to identify a sample set of commands in the HART common, universal, and device-specific sets that could be used by an attacker to achieve one or more of the assessment questions effects. Example command functions used are shown in Figure 5. We used the same set of device common and universal commands where possible to provide more consistency.

**Communications test cases** examined the use of encrypted and unencrypted network traffic. Tests focused on which attacks were prevented and how to bypass the encryption. For example, when considering application-layer encryption between the IMS/AMS and SIS, attacks used ran as part of the IMS/AMS through a trojan dynamic-link library (DLL) to send unauthorized commands. We conducted some encryption implementation tests to help vendors better secure their products. For example, we looked at the use of self-signed certificates, uni- versus bi-directional authentication, and whether cryptographic components had known vulnerabilities (e.g., older version of TLS). Specific product implementation issues were not included in the general architecture measurements.



| 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 | | |

**Figure 5: Device commands were used individually in test cases or combined with other commands and attacks to achieve a greater effect.**

**System test cases** applied each of the available security controls to instrument test cases to determine the control's effect on the attack. Attacks were first run with no enabled security controls and then with each control that had the potential to affect attack success. In essence, the security controls were used as “test control knobs” to determine which controls, if any, would cause attacks to fail. Security controls examined included various write-protection methods, IMS/AMS authentication, limiting allowed connections to authorized hosts, and various encryption schemes. Device-native write-protections were the only protection common to both architectures and (since MUXes have no security controls) were the only protection mechanism tested in the MUX-mediated system. Test cases were also planned to determine if security controls could be bypassed.

No test cases were planned to look for vulnerabilities in SIS and MUX components as these were out of project scope.

**Test Case Example 1** (Questions 5.c, 5.d). Cause a device to become unresponsive: send a rapid succession of reset commands to a device.

**Test Case Example 2** (Questions 1, 5.e, 5.f). Bypass a device's software write-protection and lock the administrator out of the device: intercept device software write-protect passcode to gain access, then disable write protection, change the passcode, and re-enable write protection.

## 2.7 Testing and Test Harness

We engaged two assessors who worked in parallel: one focused on the instruments and the MUX architecture and the other focused on the instrument manager and on the instantiated SIS system. This section describes the instrument and system assessments.

**Instrument Testing.** The four goals of instrument testing were to 1) identify available protective features, determine how they worked, and determine how they could be bypassed; 2) identify device-supported HART commands that could be used to make unauthorized instrument changes; 3) identify any undocumented device features and commands; and 4) identify any potential input parsing errors that might provide opportunity for attack.

Devices were assessed by reviewing product documentation, directly probing devices, observing HART communications between the IMS/AMS and instruments while configuring and pulling status using the operator interface, and conducting limited fuzz testing.

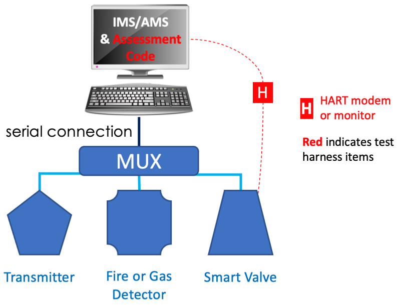


**Figure 6: The MUX-mediated test environment consisted of multiple instruments, an IMS/AMS, a MUX, and test harness components (shown in red).**

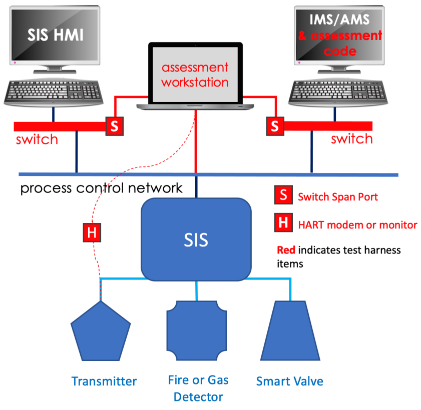


**Figure 7: The SIS-mediated test environment consisted of multiple instruments, an IMS/AMS, an SIS, an SIS HMI, and test harness components (shown in red).**

**System Testing.** The MUX-mediated environment (Figure 6) was used to validate that the instrument test cases worked. Because MUXes are passthrough devices and do not offer any protective features, the project treated the MUX architecture as the baseline for what an attacker could do to an unprotected instrument. Network-based attacks (e.g., man-in-the-middle or packet sniffing) were not applicable to this environment because the IMS/AMS was serial connected to the MUX.

**Table 3: Device-Native Protection Efficacy**

| Write-Protection | Hardware | Software | None |
|---|---|---|---|
| Type | 3 (33%) | 6 (66%) | 1 (11%) |
| Effectiveness w/o Bypass | 3 (100%) | 5 (83%) | 0 (0%) |
| Bypassed in Testing? | 0 (0%) | 6 (100%) | n/a |
| Overall Effectiveness | 3 (100%) | 0 (0%) | 0 (0%) |

Assessment attacks against the SIS-mediated systems were mainly launched using co-resident malware (attack scripts) on the IMS/AMS platform. Man-in-the-middle and password sniffing attacks were conducted from a pen tester assessment workstation (see Figure 7.)

Each SIS-mediated assessment was conducted in a different lab to obtain access to SIS test systems. The assessment team ensured logical consistency and adherence to planned test structure shown in Figure 7. Individual test cases performed in these labs were repeated by the assessors to ensure the reliability of the results and were then demonstrated to the test director to confirm the results. Tests that affected the SIS were demonstrated a fourth time to the SIS vendor's staff.

## 3 RESULTS AND ARTIFACTS

Participating vendors did not want discovered product issues to be shared either publicly or with other vendors. LOGIIC pledges confidentiality to participating vendors in all projects and uses non-disclosure agreements (NDAs) to enforce the agreements. This impacted the extent to which results could be shared. Issues unique to specific products could only be reported to the LOGIIC members and the product's vendor. Recurring issues that existed across all or most assessments, regardless the vendor products in use, could be reported publicly.

**"Insecure by Design" Findings.** Safety instruments process unchecked commands under the assumption that all received commands are from a legitimate source. This was true for all 9 instruments. The HART and HART-IP protocols evaluated included no security concepts. Vendor proprietary protocols were similarly deficient. These all tie to fundamental issues identified in the MITRE Common Weakness Enumeration (CWE) [1] as exploitable weaknesses. These findings are directly responsible for our ability to bypass security controls.

**Device Write Protection Results.** Security controls to block writing changes to devices were available at three points on the network: on the device, on the communications mediator (SIS only), and on the IMS/AMS. In general, we found that the closer the protection mechanism was placed to the device, the better it worked. Any protection that relied on the IMS/AMS platform at all could be bypassed if that system was compromised.

*Device-native write-protections* worked independently of the three HART command types and architecture; non-bypassed protections generally blocked all tested write operations. We found a few minor exceptions with maintenance-type commands.

In our sample set, three devices had hardware-only write-protections, five devices had software or hybrid software/hardware write-protections,

**Table 4: SIS Command Blocking Efficacy**

| SIS Block | Common/Universal | Device-Specific |
|---|---|---|
| HART Command Blocking | 4 (100%) | 3 (75%) |
| Effectiveness w/o Bypass | 4 (100%) | 3 (100%) |
| Bypassed in Testing? | 1 (25%) | 1 (25%) |
| Collateral Damage | 0 (0%) | 3 (100%) |
| Overall Effectiveness | 3 (75%) | 0 (0%) |

one device had independent hardware and software write-protections, and one device had no native write-protections of any kind (Table 4).

Strictly hardware-based jumpers and switches worked without fail to block all unauthorized modification attempts, with the one exception previously noted. We considered this performance to be “effective.” None of these protections were bypassed during our testing.

Software write-protections were implemented in variety of inconsistent ways across products, even within same-vendor product families. All relied on the IMS/AMS for users to enter passcodes and were bypassed in testing. In every case, the products supported only weak passcodes (e.g., a four- or eight-character code.) Passcodes were easily guessed, as none of the tested devices supported lockouts for failed attempts.

In the SIS systems, protection passcodes were transmitted to devices over the network in cleartext and could be sniffed from network traffic. This was true whether using HART-IP or a vendor-proprietary protocol. In all cases where an SIS had an optional encrypted communications feature that was enabled, network sniffing was unsuccessful. Encryption was disabled by default in all tested products. All tested encrypted communications were implemented using standard, current COTS encryption products.

*SIS write protections* provided the second-best protection against malicious device reconfiguration. Performance was tied to HART command types rather than to individual commands. All SISs were able to block HART common and universal write commands. 75% were able to block device-specific HART commands; however, none could do so without also blocking read commands, which prevented device status updates on operator consoles. This issue is attributed to a deficiency in the HART protocol. We considered this to be “collateral damage.”

We were able to bypass command blocking entirely in one SIS due to the manner in which it was implemented.

**Encrypted Communications Results.** Using encryption generally prevented attacks originating from points other than the IMS/AMS. Host-layer encryption could be bypassed by malware co-located on the IMS/AMS platform. Application-layer encryption required the malware to execute in the application process space.

**Assessment Questions and Hypothesis Results.** All assessment questions were answered in the affirmative in the MUX-based architecture if hardware-based write protections were not engaged. Because these protections were architecture-independent (present in both architectures), we focused on the protections available only in the SIS architecture to answer our hypothesis. SIS security controls reliably thwarted attempts to make malicious changes for common and universal commands with few exceptions. If collateral damage was acceptable, SISs thwarted attacks using device-specific commands. The level of additional protection provided by the SIS-mediated architecture depended on the availability and correct configuration of controls and on the controls not being bypassed. In our testing, if SIS security controls were not enabled, the SIS-mediated architecture did not provide any better protections than did the MUX-mediated architecture.

**Artifacts.** Artifacts from our evaluation effort include test plans and test cases; individual safety system designs with products and product configuration files; attack scripts, and data. Data collected during the evaluation included product configuration files, network traffic, log files, screen shots, and attack script outputs. These artifacts cannot be shared due to confidentiality agreements with participating vendors. All data is held as confidential by LOGIIC. Artifacts specific to any one vendor were provided to that vendor on request to facilitate reproducing results by vendor product teams.

## 4 LESSONS LEARNED

This section briefly discusses safety system design lessons for ICS stakeholders and then presents lessons learned for those interested in conducting evaluations of real-world architectures.

**Safety System Designs.** Numerous lessons were learned from this activity that will help vendors and asset owners in configuring and deploying more secure safety systems. Examples include: 1) Software-based write protections on devices are fully bypassable; therefore, vendors should implement hardware-based write protections instead. 2) The only way to block unauthorized device changes when using a MUX is by using device-native write protections. 3) In the absence of SIS security controls, an SIS provides no better protection than does a MUX. 4) SIS security controls are often difficult to understand and configure. In some cases, misconfigurations caused problems in the system function and security. This may contribute to non-use in practice. A full set of lessons, along with specific recommendations, are documented in the project final report [13].

**Evaluation Methodology.** Overlaying a hypothesis and assessment questions-based approach on top of exploratory pen testing works well. This method can inject discipline and rigor into what might be an otherwise “messy” exploratory method and can provide the consistency needed to perform cross-system analysis.

Assessing multiple system instances works well in uncovering recurrent problems. By the end of the second system pair assessment, a pattern emerged pointing to systemic issues with safety system industry protocols and product designs. The third and fourth assessments confirmed what we already suspected.

**Collaborative Assessments** involving stakeholders can have significant impact on product security. One vendor fixed found problems before we even completed our assessment, and a LOGIIC member changed its procurement process to require a more secure product configuration as the direct result of interim findings.

**Vendor Confidentiality and Artifacts.** Results involving product-unique features and artifacts such as screen shots and command sets can reveal the specific products used. While product-specific issues cannot be disclosed, issues with industry-shared standards and practices can be revealed through efforts such as this. If evaluations are designed properly, high-impact product-neutral findings can be extracted and shared.

## 5 LIMITATIONS AND FUTURE WORK

**Small Sample Set.** We used a relatively small sample set which makes drawing general conclusions difficult for some metrics. For example, the observed percentage of devices with hardware vs. software write protections may not be representative of all devices. However, we observed that write protection type aligned with classes of instruments rather than with vendors. Additional work is needed to determine if this is an industry-wide phenomenon or was coincident of the sample set. Additional work using a larger sample set could help understand deeper issues that may occur within classes of instruments.

Using the limited sample set, we could draw conclusions about common issues found in all or nearly all test subjects. HART protocol security weaknesses are evident in the documentation and were confirmed through consistent test results across the entire sample set. We focused mainly on these types of measures.

**Only Two Evaluated Architectures.** Due to time and funding limits, we were unable to evaluate alternate architectures that might inherently provide better protection. In particular, we expect that an architecture that does not connect the IMS/AMS to the PCN would have reduced exposure to network-based compromise. This alternative may have other issues that create more attack opportunities (e.g., inability to apply system patches in a timely manner.) Future efforts could explore alternate architectures to understand the pros and cons of each.

**Non-exhaustive Testing.** Exhaustive product testing was not conducted because the project did not require it. Rather than exhaustively test all device HART commands, we tested a sampling from each command type (common, universal, and device-specific). As a result, there could be commands that are not blocked by protective measures. We found evidence of such for some maintenance commands. The test subjects may also contain undiscovered vulnerabilities. We made no attempt made to determine "how secure" a product or architecture is.

**No Risk Score.** We made no attempt to compute a risk score. To understand actual risk, safety system operators must examine the available attacks and countermeasures presented by the products and product configurations used in their own systems and compare with the context-unique impact of successful attacks.

## 6 RELATED WORK

Pen testing simulates cyberattacks against a system to find exploitable vulnerabilities. This form of testing provides knowledge that is specific to the system or product being probed. Our goal was not to break specific devices and report on that breakage. Instead, we wanted to understand more generally if common product-neutral issues existed. We used pen testing to identify a small set of viable attacks against specific instruments that could be used for security control efficacy testing.

Our effort most closely relates to system architecture security assessments. Buckshaw et al. discussed the use of MORDA, a mission impact and adversary-minded risk analysis methodology used to evolve system architectures to be more secure [4]. The methodology is tabletop-based and uses red and blue teams to create attack trees and apply countermeasures to reduce likely attack paths. Because red team opinions are highly diverse, the results may vary and be non-repeatable. The methodology uses sensitivity analysis of the results in attempt to address this issue. Malik et al. applied quantitative modeling and risk analysis to large-scale cyber-physical systems [10]. They focused on assessing risk for implemented large-scale systems rather than understanding architectural and security control impact.

Proof testing is an industry practice used to uncover systematic errors in safety-instrumented systems [8]. Vendors specify what should be tested and when, but cybersecurity concepts are not included. Our work focused solely on cybersecurity.

Bolshev and Malinovsky examined HART, HART-IP, IMSs, and DTMs through exploratory analysis [3]. Our project used exploratory analysis to learn device command sets and how they can be used for malicious purposes. We used this knowledge to refine test cases for architectural testing.

The Department of Energy National SCADA Test Bed staff examined available security defenses for SCADA systems [7]. The architecture and communication portions of their work closely relate to our effort. They found two issues in common with our findings: clear text communications and weak or no authentication. Had their recommended mitigations been applied in this domain, some of the issues we found would not exist.

## 7 CONCLUSIONS

We evaluated attack paths and security controls in commonly used safety system architectures based on a hypothesis that an SIS-mediated architecture could provide better protection against unauthorized and malicious device configuration modifications than could a MUX-mediated architecture. From this, we generated effects-based questions to inform test case design. We ran the tests against a series of four instantiated system pairs, each using different products, and then performed cross-system analysis to illuminate recurring issues that affect the broader industry.

We conducted our evaluation with full cooperation from safety-system and device vendors and safety-systems experts. We found recurring product-independent vulnerabilities that exist in all safety systems due to the insecure design of safety instrument and the HART protocol. These design flaws enable attackers to bypass all software-based device-native write protections. Hardware-based device write protections were the most effective control but were absent in 66% of our sample set.

All SISs offered write protection that could be used in an SIS-mediated system. Because of this, we concluded that the SIS-mediated architecture is able to provide better protection for these devices if security controls are enabled and configured properly. However, if no SIS security controls are used, the SIS acts as a passthrough, just as does the MUX, and provides no added security benefit. SIS security controls are not well known or understood in the operational environment, so while these features are available and can provide protections, they are often not used. SIS vendors and asset owners should work together to identify and implement all applicatble controls to reduce vulnerabilities in process control envrionments.

## ACKNOWLEDGMENTS

The authors wish to thank the numerous participating vendors that cooperated in this project and provided equipment and staff hours to setup for and support testing. The authors also thank the entire project assessment team, who fleshed out the evaluation strategy and performed the system assessments. Finally, the authors wish to thank the LOGIIC project champion and the safety-system expert who provided guidance and subject-matter expertise throughout the evaluation.

LOGIIC is a government-industry partnership between the U.S. Department of Homeland Security's Science and Technology (DHS S&T) Directorate and multi-national member companies from the O&G sector. This project was co-funded with U.S. DHS S&T under Contract No. HSHQDC-16-C-00034. The opinions, findings, conclusions, and recommendations expressed in this material are those of the authors, do not necessarily reflect the views of DHS or the O&G companies, and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DHS, the U.S. Government, or the O&G companies.

## REFERENCES

[1] [n.d.]. MITRE Common Weakness Enumeration (CWE) Database. Online. https://cwe.mitre.org/ Database of known cybersecurity weaknesses in software and systems designs.

[2] [n.d.]. MITRE Cybersecurity Vulnerability Enumeration (CVE) Database. Online. https://cve.mitre.org/ Database of known cybersecurity vulnerabilities in products.

[3] Alexander Bolshev and Alexander Malinovsky. 2013. HART (in)security: How one transmitter can compromise a whole plant. Retrieved Jul 16, 2021 from https://www.slideshare.net/DefconRussia/alexander-bolshev-alexander-malinovsky-hart-insecurity

[4] Donald Buckshaw, Gregory Parnell, Willard Unkenholz, Donald Parks, James Wallner, and O. Saydjari. 2005. Mission Oriented Risk and Design Analysis of Critical Information Systems. Military Operations Research. *Military Operations Research* 10, 2 (2005), 19–38. https://www.academia.edu/2817116/Mission_oriented_risk_and_design_analysis_of_critical_information_systems

[5] The LOGIIC Consortium. 2011. *Cyber Security Implications of SIS Integration with Control Networks*. Technical Report. Automation Federation. https://www.automationfederation.org/Logiic/LogiicProjects Presented at Automation Week.

[6] FieldComm Group 2011-2021. *Highway Addressable Remote Transducer (HART) Protocol Specifications*. FieldComm Group. https://www.fieldcommgroup.org/hart-specifications

[7] Raymond K Fink, David F Spencer, and Rita A Wells. 2006. *Lessons Learned from Cyber Security Assessments of SCADA and Energy Management Systems*. Technical Report. https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/1-NSTB_Control_Systems_Security_Standards_Accomplishments_and_Impacts.pdf

[8] Prasad Gotei. 2018. Proof Testing Safety Instrumented Systems. Retrieved Jul 16, 2021 from https://engineering.purdue.edu/P2SAC/presentations/documents/Proof_Testing_Safety_Instrumented_Systems.pdf

[9] Blake Johnson, Dan Caban, Marina Krotofil, Dan Scali, Nathan Brubaker, and Christopher Glyer. 2017. *Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure*. Retrieved Jul 16, 2021 from https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html Threat research analysis of Triton SIS attack.

[10] Adeel A. Malik and Deepak K. Tosh. 2020. Quantitative Risk Modeling and Analysis for Large-Scale Cyber-Physical Systems. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. 1–6. https://doi.org/10.1109/ICCCN49398.2020.9209654

[11] Annie McIntyre. 2018. *LOGIIC Safety Instrumented Systems Final Report*. Technical Report. Automation Federation. https://www.automationfederation.org/Logiic/LogiicProjects

[12] RealPars. 2018. What is a Safety Instrumented System. Video. Retrieved Jul 16, 2021 from https://www.youtube.com/watch?v=W2YUNnfATBY Short tutorial on what a safety instrumented system is and why it is important.

[13] Laura S Tinnel and Ulf Lindqvist. 2021. *LOGIIC Safety Instrumentation and Management Final Report*. Technical Report. International Association of Automation (ISA). https://www.logiic.org

[14] Laura S Tinnel and Ulf Lindqvist. 2021. When Safety Instrument Control Goes Rogue. Retrieved Jul 16, 2021 from https://gateway.on24.com/wcc/eh/3049745/lp/3076561/logiic-project-12-when-safety-instrument-control-goes-rogue