NISTIR 8219

Securing Manufacturing Industrial Control Systems: Behavioral Anomaly Detection

James McCarthy
Michael Powell
Keith Stouffer
CheeYee Tang
Timothy Zimmerman
William Barker
Titilayo Ogunyale
Devin Wynne
Johnathan Wiltberger

National Institute of Standards and Technology
U.S. Department of Commerce

NISTIR 8219

Securing Manufacturing Industrial Control Systems: Behavioral Anomaly Detection

James McCarthy
Michael Powell
National Cybersecurity Center of Excellence
Information Technology Laboratory

Keith Stouffer
CheeYee Tang
Timothy Zimmerman
Intelligent Systems Division
Engineering Laboratory

William Barker
Dakota Consulting
Silver Spring, MD

Titilayo Ogunyale
Devin Wynne
Johnathan Wiltberger
The MITRE Corporation
McLean, VA

November 2018

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

Abstract

Industrial control systems (ICS) are used in many industries to monitor and control physical processes. As ICS continue to adopt commercially available information technology (IT) to promote corporate business systems' connectivity and remote access capabilities, ICS become more vulnerable to cybersecurity threats. The National Institute of Standards and Technology's (NIST's) National Cybersecurity Center of Excellence (NCCoE), in conjunction with NIST's Engineering Laboratory (EL), has demonstrated a set of behavioral anomaly detection (BAD) capabilities to support cybersecurity in manufacturing organizations. The use of these capabilities enables manufacturers to detect anomalous conditions in their operating environments to mitigate malware attacks and other threats to the integrity of critical operational data. NIST's NCCoE and EL have mapped these demonstrated capabilities to the Cybersecurity Framework and have documented how this set of standards-based controls can support many of the security requirements of manufacturers. This report documents the use of BAD capabilities in two distinct, but related, demonstration environments: a robotics-based manufacturing system and a process control system that resembles what is being used by chemical manufacturing industries.

Audience

This report is intended for individuals or entities that are interested in understanding BAD technologies and their application to ICS environments. Additionally, this report is intended for those who are interested in understanding how to implement BAD tools in ICS and other operational technology environments.

Keywords

BAD; behavioral anomaly detection; cybersecurity; Cybersecurity Framework; ICS; industrial control systems; manufacturing; process control

Acknowledgments

NIST and the NCCoE wish to thank Omer Schneider, Phil Neray, and Joe DiPietro of CyberX; Paul J. Geraci, Andrew Duke, and Mark McCoy of OSIsoft; Dennis Murphy, Jason Sharp, and Daniel Trivellato of SecurityMatters; and Erlend A. Engum, Jishnu Nair, Nina Hesby Tvedt, and Siv Hilde Houmb of Secure-NOK for their contributions to this document.

Executive Summary

NIST's NCCoE, with NIST's EL and NCCoE collaborators, offers information regarding the use of BAD capabilities to support cybersecurity in ICS for manufacturing. This National Institute of Standards and Technology Interagency Report (NISTIR) was developed in response to feedback from members of the manufacturing sector concerning the need for cybersecurity guidance.

Cybersecurity attacks directed at manufacturing infrastructure can be detrimental to both human life and property. BAD mechanisms support a multifaceted approach to detecting cybersecurity attacks against ICS devices on which manufacturing processes depend, in order to permit the mitigation of those attacks.

The NCCoE and EL deployed commercially available hardware and software provided by industry, in response to a NIST notice in the Federal Register, in order to demonstrate BAD capabilities in an established laboratory infrastructure. We mapped security characteristics of the demonstrated capabilities to the *Framework for Improving Critical Infrastructure Cybersecurity* [1] based on NISTIR 8183, the *Cybersecurity Framework Manufacturing Profile* [2]. The mapping can be used as a reference in applying specific security controls found in prominent industry standards and guidance.

Introducing anomalous data into a manufacturing process can disrupt operations, whether deliberately or inadvertently. The goal of this NISTIR is to provide practical approaches for manufacturers to use in their efforts to strengthen the cybersecurity of their manufacturing processes. This NISTIR demonstrates how BAD tools can be used as a key security component in sustaining business operations, particularly those based on ICS. The examples provided in this NISTIR illustrate how detecting anomalous conditions can improve the reliability of ICS, in addition to providing specific cybersecurity benefits.

Table of Contents

List of Tables

List of Figures

1. Introduction

The goal of this National Institute of Standards and Technology Interagency Report (NISTIR) is to show practical approaches that manufacturers can use to strengthen cybersecurity in their manufacturing processes. Behavioral anomaly detection (BAD) tools can provide a key security component for sustaining business operations, particularly those based on industrial control systems (ICS). Because introducing anomalous data into a manufacturing process can disrupt operations, whether deliberately or inadvertently, the examples provided in this NISTIR demonstrate how detecting anomalous conditions can improve the reliability of manufacturing and other ICS, in addition to providing the demonstrated cybersecurity benefits.

1.1. Background

As stated in the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-82 [3], ICS are vital to the operation of the United States' critical infrastructures, which are often highly interconnected and mutually dependent systems. While federal agencies also operate many ICS, approximately 90 percent of the nation's critical infrastructures are privately owned and operated. As ICS increasingly adopt information technology (IT) to promote corporate business systems' connectivity and remote access capabilities by using industry-standard computers, operating systems (OSs), and network protocols, the accompanying integration provides significantly less isolation for ICS from the outside world. While security controls have been designed to deal with security issues in typical IT systems, special precautions must be taken when introducing these same approaches in ICS environments. In some cases, new security techniques tailored to the specific ICS environment are needed. NIST recognizes this concern and is working with industry to solve these challenges through the development of reference designs and the practical application of cybersecurity technologies. BAD is one tool for improving ICS security.

NIST's National Cybersecurity Center of Excellence (NCCoE), in conjunction with NIST's Engineering Lab (EL) and NCCoE industry collaborators, has demonstrated a set of BAD capabilities to support cybersecurity in manufacturing organizations. The use of these capabilities enables manufacturers to detect anomalous conditions in their operating environments to mitigate malware attacks and other threats to the integrity of critical operational data. NIST's NCCoE and EL have mapped these demonstrated capabilities to the NIST Cybersecurity Framework [1] and have documented how this set of standards-based controls can support many of the security requirements of manufacturers. This NISTIR documents the use of BAD capabilities in two distinct, but related, demonstration environments: a collaborative robotics-based manufacturing system and a process control system (PCS) that resembles what is being used by chemical manufacturing industries.

1.2. Purpose and Scope

The scope of this NISTIR is a single cybersecurity capability. The security characteristics of different BAD approaches were mapped to the Cybersecurity Framework. The mapping points manufacturers to specific security controls found in prominent cybersecurity standards.

1.3. Challenges

Cybersecurity is essential to the safe and reliable operation of modern industrial processes. Threats to ICS can come from numerous sources, including hostile governments, criminal groups, disgruntled employees, other malicious individuals, unanticipated consequences of component interactions, accidents, and natural disasters. The Cybersecurity Framework [1] addresses identifying threats and potential vulnerabilities; preventing and detecting events; and responding to, and recovering from, incidents. It is not possible to prevent all cyber events. It may not even be possible to identify all threats for which ICS need to be prepared. It is certainly necessary to detect incidents before the response to, or recovery from, the incidents can be undertaken. Therefore, the detection of cyber incidents is an essential element for cybersecurity.

Many incident-detection tools involve monitoring system behaviors for out-of-specification settings or readings or for predefined threat signatures (information elements previously identified as being associated with threats or vulnerability characteristics). However, as previously mentioned, not all threats and vulnerabilities are known beforehand (e.g., zero-day attacks); therefore, not all threats and vulnerabilities can be included among signatures for which monitoring is undertaken. BAD involves the continuous monitoring of systems for unusual events or trends. The monitor looks for evidence of compromise, rather than for the attack itself.

The challenge addressed by this project is to demonstrate example implementations of BAD capabilities that manufacturers can adopt to achieve their cybersecurity goals. Specifically, this project responds to a need within the manufacturing community to improve the ability to detect anomalous behavior in real or near-real time. Early detection of potential cybersecurity incidents is key to helping reduce the impact of these incidents for ICS.

1.4. Approach to Addressing Challenges

The NCCoE developed and demonstrated a set of example approaches for detecting anomalous conditions within manufacturers' ICS environments. These examples include recommendations that are practical for businesses to implement to strengthen cybersecurity in their manufacturing processes, with an additional potential for detecting anomalous conditions not related to security, such as equipment malfunctioning.

The NCCoE examples provide the following capabilities:

- models of BAD capabilities that manufacturers can adopt to achieve their security goals for mitigating the risks posed by threats to cybersecurity
- nonintrusive techniques to analyze industrial network communications, allowing the existing ICS infrastructure to flow through the network without interruption or a performance impact
- establishment of one or more baselines, and notification when specific changes or anomalies occur in the environment over time
- identification of new devices on the ICS network and of assets that have disappeared from the network

- detection of unauthorized configuration changes and of the transfer of files in the network
- increased visibility into network operation and real-time alerting

The NCCoE used commercially available products provided by industry collaborators to address this cybersecurity challenge. These products were provided under Cooperative Research and Development Agreements. This NISTIR does not endorse any products and does not guarantee compliance with any regulatory initiatives. An organization's information security experts should identify the products that will best integrate with their existing tools, processes, and system infrastructure. Organizations can adopt one of the demonstrated approaches or another one that adheres to the suggested guidelines. This NISTIR can also be used as a starting point for implementing BAD.

1.5. Benefits

This NISTIR is intended to help organizations accomplish their goals by using anomaly detection tools for the following purposes:

- detect cyber incidents in time to permit effective response and recovery
- expand visibility and monitoring capabilities within manufacturing control systems, networks, and devices
- reduce opportunities for disruptive cyber incidents by providing real-time monitoring and anomaly-detection alerts
- support the oversight of resources (e.g., IT, personnel, data)
- enable faster incident-response times, fewer incidents, and shorter downtimes

2. Cybersecurity Framework and NIST Manufacturing Profile

The *Framework for Improving Critical Infrastructure Cybersecurity* [1] is a voluntary risk-based assemblage of industry standards and best practices designed to help organizations manage cybersecurity risks. The Cybersecurity Framework, created through collaboration between government and the private sector, uses a common language to address and manage cybersecurity risk in a cost-effective way, based on business needs, without imposing additional regulatory requirements. The *Cybersecurity Framework Manufacturing Profile* [2] defines specific cybersecurity activities and outcomes for the protection of the manufacturing system and its components, facility, and environment. By using the profile, the manufacturer can align cybersecurity activities with business requirements, risk tolerances, and resources. The profile provides a manufacturing sector-specific approach to cybersecurity from standards, guidelines, and industry best practices.

Table 2-1 maps functions addressed by BAD capabilities to NIST Cybersecurity Framework functions as presented in the profile. In Table 2-1, the references to the requirements are American National Standards Institute / International Society of Automation Standard 62443-2-1 (*Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program*) [4], American National Standards Institute / International Society of Automation Standard 62443-2-3 (*Security for Industrial Automation and Control Systems – Part 2-3: Patch Management in the IACS Environment*)

[5], and NIST SP 800-53 (*Security and Privacy Controls for Federal Information Systems and Organizations*) [6].

Table 2-1 Mapping of Cybersecurity Framework Functions Addressed by BAD Capabilities to the Manufacturing Profile

Function	Category	Subcategory	Manufacturing Profile	Reference
Detect	Anomalies and Events (DE.AE)	DE.AE-2	**Low**	62443-2-1:2009 4.3.4.5.6, 62443-2-3:2015 SR 2.8, 2.9
			Review and analyze detected events within the manufacturing system to understand attack targets and methods	AU-6 IR-4
			Moderate and High	
			Employ automated mechanisms, where feasible, to review and analyze detected events within the manufacturing system	AU-6(1) IR-4(1)
		DE.AE-3	**Low and Moderate**	62443-3-3:2013 SR 6.1
			Ensure that event data is compiled and correlated across the manufacturing system by using various sources, such as event reports, audit monitoring, network monitoring, physical access monitoring, and user/administrator reports	IR-5
			High	
			Integrate the analysis of events, where feasible, with the analysis of vulnerability scanning information, performance data, manufacturing system monitoring, and facility monitoring to further enhance the ability to identify inappropriate or unusual activity	AU-6(5)(6) AU-12(1)
		DE.AE-4	**Low**	
			Determine the negative impacts, resulting from detected events, to manufacturing operations, assets, and individuals, and correlate the impacts with the risk assessment outcomes	RA-3
			Moderate	
			Employ automated mechanisms to support impact analysis	IR-4(1) SI-4(2)
			High	
			Correlate detected event information and responses to achieve perspective on the event's impact across the organization	IR-4(4)

3. Demonstration Environment Architecture

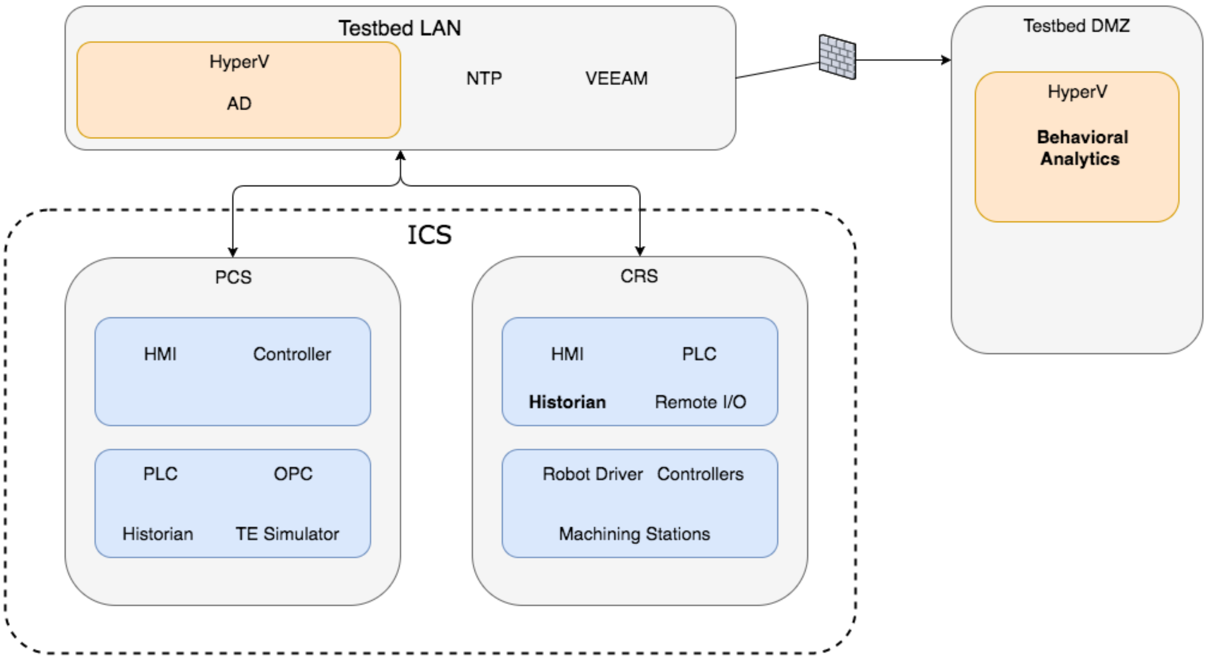
The Cybersecurity for Smart Manufacturing Systems (CSMS) demonstration environment emulates real-world manufacturing processes and their ICS by using software simulators and commercial off-the-shelf hardware in a laboratory environment [7]. The CSMS environment was designed to measure the performance impact on ICS that is induced by cybersecurity technologies. The PCS and the collaborative robotic system (CRS) are the two systems used for the demonstration of BAD capabilities. The PCS and CRS demonstration enclaves are described in Sections 3.1 and 3.2.

Figure 3-1 depicts a high-level architecture for the BAD demonstration environment. The capabilities that are introduced in the demonstration environment are bolded in Figure 3-1 and address the Cybersecurity Framework functions and subcategories listed in Table 2-1.

The local area network (LAN), a firewalled-off cybersecurity tool environment (demilitarized zone [DMZ]), and two ICS environments make up the existing architecture of the CSMS demonstration environment. The LAN consists of a hypervisor for virtualization, a network time protocol (NTP) server for time synchronization, a server for backup and storage, and a virtualized Active Directory server for domain services. Within the demonstration environment's DMZ, there is a hypervisor that allows cybersecurity tools to be deployed within an isolated environment.

Within this architecture, the BAD capability is introduced in two areas that use four collaborator products. Two BAD systems are installed within the demonstration environment's DMZ. One of these BAD systems is agent-based and is installed at multiple endpoints within the CRS and the PCS, while data is aggregated at the demonstration environment's DMZ. The other BAD system is implemented as an additional capability to the historian within the CRS only. This build consisted of performing and introducing the BAD capability into the CRS and PCS environments, one product at a time. In other words, only one product was installed and performing BAD at any given time. Each collaborator's product installation was scheduled to run in sequence to ensure complete autonomy from each product in the build.

Figure 3-1 BAD High-Level Architecture



3.1. Collaborative Robotic System

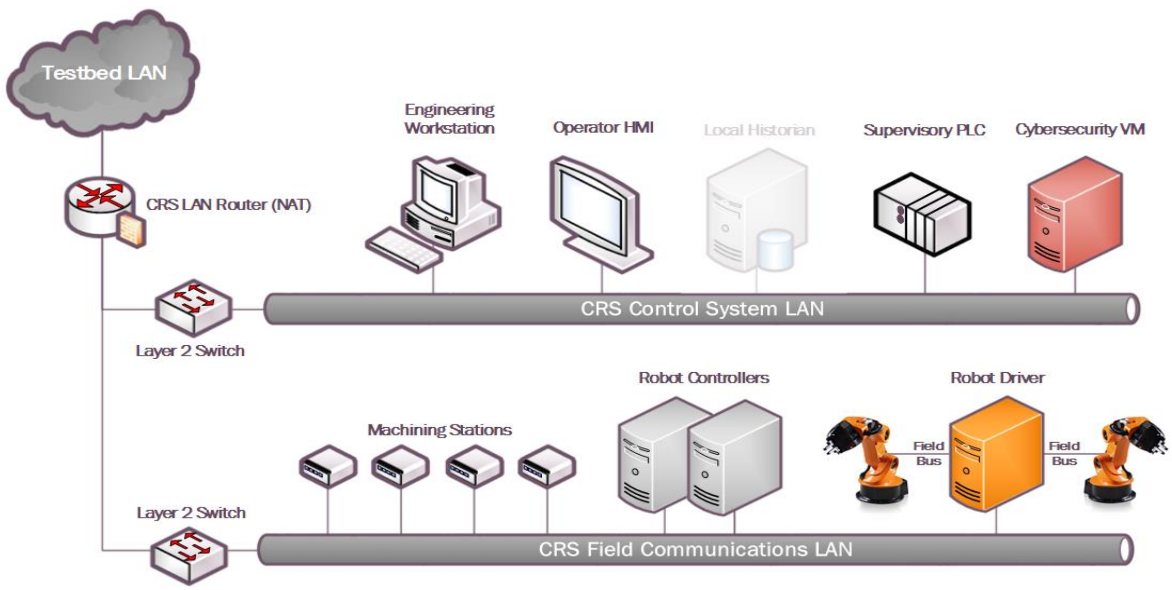
The CRS enclave of the environment is composed of two robotic arms that emulate a material-handling application known as “machine tending” [8]. Robotic machine tending uses robots to interact with the machinery, performing operations that a human operator would normally perform (e.g., loading and unloading parts, opening and closing machine doors, activating operator control-panel buttons). The robots operate in concert according to a material-handling procedure that changes dynamically based on feedback from the simulated machining operations. An architecture of the robotic enclave network is shown in Figure 3-2.

The robot controllers can operate in one of two modes: deployed or virtualized. In the deployed mode, each robot is controlled on a dedicated Dell PowerEdge R420 server running the robot operating system (ROS) on top of Ubuntu Linux. In the virtualized mode, each robot is controlled by virtualized servers within a hypervisor running on a Dell PowerEdge 620 server. The deployed mode supports experiments with a pseudo-ideal configuration. The virtualized mode supports experiments with a resource-restricted configuration and can maintain independent demonstration environments.

The pseudo-ideal configuration provides the robot controller software with computational resources that are well beyond the minimum requirements for unimpeded operations. Operating in this manner is reserved for experiments that do not require server performance impacts to be measured (e.g., network-specific experiments). The resource-restricted configuration allows the researchers to restrict the available resources to the robot controller software and underlying OS (e.g., memory allocation, available hard-disk space, hard-disk access rates, number of central processing unit [CPU] cores).

The hypervisor also allows software-based cybersecurity tools to be deployed within an isolated environment, and allows for the ability to restore the enclave environment to a known-good state, reducing the chances of cross-contamination by residual software modules or services remaining within a virtual machine (VM) post-experiment. Software-based cybersecurity tools are installed on VMs dedicated to specific experiments within the hypervisor and are archived. This allows any tool to be recalled for any experiment that requires its execution.

Figure 3-2 Robotic Assembly Enclave Network



3.1.1. CRS Network Architecture

In addition to the two industrial robots, the enclave includes a supervisory programmable logic controller (PLC), a human-machine interface (HMI), several servers for executing required computational resources and applications, a cybersecurity virtual machine (CybersecVM), and an engineering workstation.

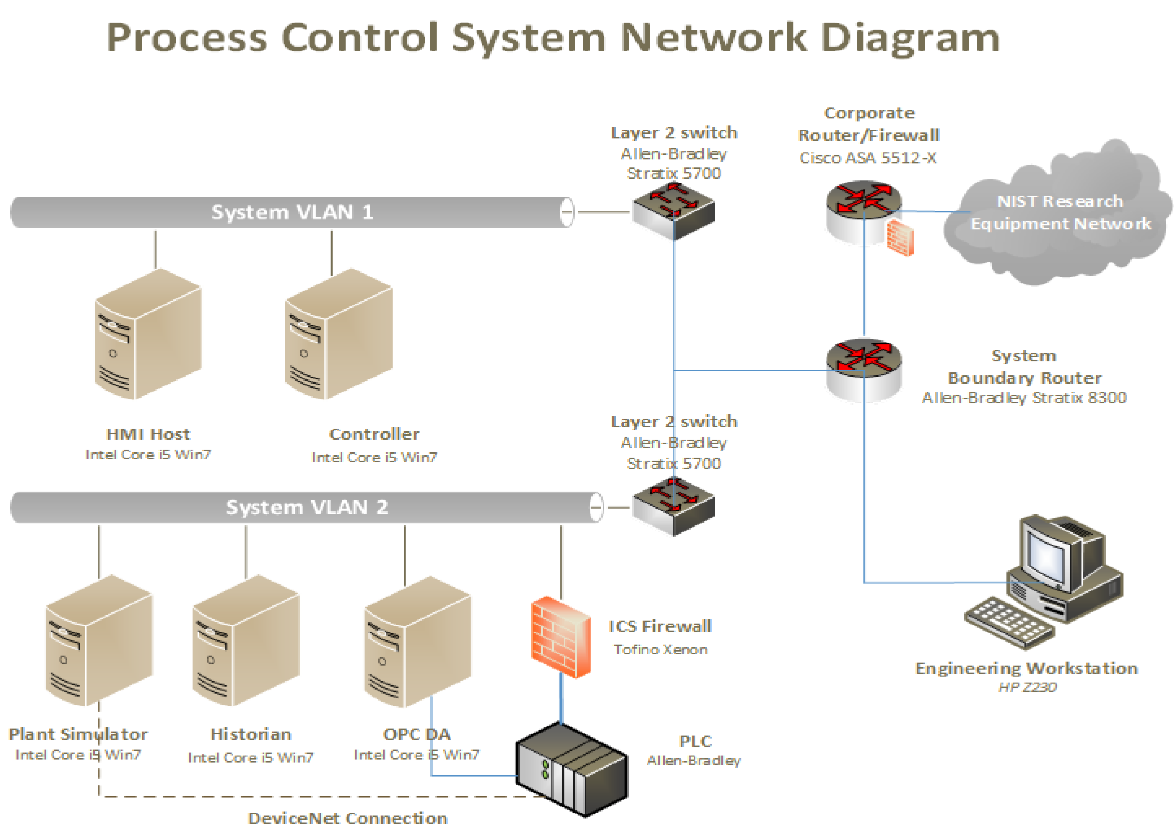
The CRS enclave LAN is constructed as a hierarchal architecture. For the BAD implementation, the reconfigurable design of the enclave enabled the implementation of network segmentation and security perimeters. The local network traffic (CRS LAN) is managed by a Siemens RUGGEDCOM RX1510, and the high-level environment traffic (environment LAN) and its connection to the "corporate network" are managed by a Cisco ASA 5512-X.

The CRS LAN has numerous machines that directly operate and support the operation of the enclave. The robot controllers or driver servers execute the operational code and communicate directly with the robots to direct their actions. The supervisory PLC communicates the status of the machining stations and operator controls to the robot controllers, and of part tracking for manufacturing performance measurements. The operator HMI also communicates with the PLC to display manufacturing process information and performance measurements to the operator. The engineering workstation hosts the programming environment and debugging tools that are used to modify the robot code and to give terminal-level access to other machines within the enclave. The HyperV server provides server virtualization to the enclave, allowing researchers to create servers on demand, as required by specific software tools or packages.

3.2. Process Control System

The PCS enclave emulates an industrial continuous manufacturing system, a manufacturing process to produce or process materials continuously, where the materials are continuously moving, going through chemical reactions, or undergoing mechanical or thermal treatment. Continuous manufacturing usually implies a 24/7 (24 hours a day, seven days a week) operation with infrequent maintenance shutdowns and is contrasted with batch manufacturing. Examples of continuous manufacturing systems are chemical production, oil refining, natural-gas processing, and wastewater treatment [9]. An architecture of the PCS network is depicted in Figure 3-3.

Figure 3-3 PCS Network Architecture

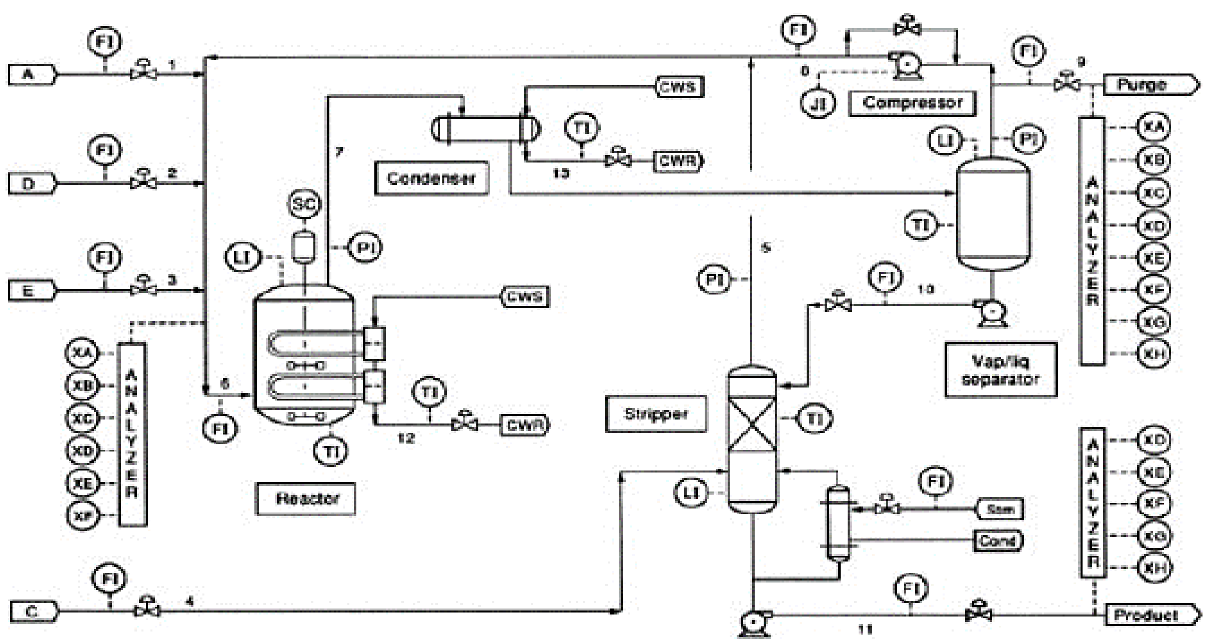


The PCS includes a software simulator to emulate the Tennessee Eastman (TE) chemical reaction process. The TE problem, presented by Downs and Vogel [10], is a well-known process-control problem in continuous chemical manufacturing. The TE control problem was chosen as the continuous process model for several reasons. First, the TE model is a well-known plant model that is used in control-systems research, and the dynamics of the plant process are well understood. Second, the process must be controlled; otherwise, perturbations will drive the system into an unstable state. The inherent unstable open-loop operation of the TE process model presents a real-world scenario in which a cybersecurity attack could represent a real risk to human safety, environmental safety, and economic viability. Third, the process is complex, nonlinear, and has many degrees of freedom by which to control and perturb the dynamics of the process. Finally, numerous simulations of the TE process have been developed with readily available reusable code. We chose the University of Washington Simulink controller design by Ricker [11]. The Ricker Simulink model was chosen for its multiloop control architecture, making distributed control architectures viable. It accurately matches the Downs and Vogel model, and the control code is easily separable from the plant code.

The TE process model is illustrated in Figure 3-4. Downs and Vogel did not reveal the actual substances used in the process; instead, they used generic identifiers for each substance. The process produces two products (G and H) from four reactants (A, C, D, and E). The process is defined as irreversible and exothermic, and the reaction rates of the four reactants are a function of the reactor temperature. The process is broken down into five major operations: a reactor, a product condenser, a vapor-liquid separator, a product stripper, and a recycle compressor. The PCS is housed in a 19-inch rack system. The model has 12 actuators for control and 41 sensors for monitoring. The process description is summarized below.

As previously mentioned, the reaction rates of the reactants are a function of the reactor temperature. The gaseous reactants are combined in the reactor to form liquid products. The reactor temperature is then cooled by using an internal cooling bundle. The reactor product passes through the condenser to the separator. The vapor-liquid separator then separates unreacted gases from the liquid products. The unreacted gases are sent back to the reactor by the recycle compressor. The remaining reactants are removed in a stripping column. Finally, the two end products are sent downstream for further refining and separation.

Figure 3-4 TE Process Control Model



3.2.1. PCS Network Architecture

The PCS includes a software simulator to emulate the TE chemical reaction process. The simulator is written in C code and is executed on a computer running Windows 7. In addition, the system includes a PLC, a software controller implemented in MATLAB, an HMI, an object linking and embedding for process control (OPC) data access (DA) server, a data historian, an engineering workstation, and several virtual LAN switches and network routers.

The PCS network is segmented from the demonstration network via a boundary router. The router is using a dynamic routing protocol, Open Shortest Path First, to communicate with the main demonstration environment router. All network traffic needs to go through the boundary router to access the main demonstration network. There are two virtual network segments in the system. Each network is managed by an Ethernet switch. The HMI and the controller are in Virtual Local Area Network (VLAN)-1, while the plant simulator, data historian, OPC DA server, and PLC are in VLAN-2. VLAN-1 simulates a central control-room environment in which the HMI and the controllers are virtually located in the same network segment. VLAN-2 simulates the process operation environment, which typically consists of the operating plant, PLCs, OPC DA server, and data historian. These network switches and routers are highly reconfigurable and therefore allow the system to implement various network topologies for demonstration.

A Tofino Xenon security appliance, a firewall specially designed for ICS application, is installed to protect the PLC. The firewall rules are configured to allow only certain network nodes and specific protocols to access the PLC, and to deny all other traffic. All of the computer nodes in the system have the Windows firewall enabled. Rules are configured to allow computer access to only traffic specific to their applications. For example, the firewall of the OPC DA server computer allows only a restricted range of remote procedure call and Distributed Component Object Model (DCOM) ports for the OPC clients to access, and it restricts the source Internet Protocol (IP) address of the OPC clients.

The plant simulator is implemented in C code, which was based on the Fortran code originally developed by Downs and Vogel. The plant simulator requires a controller to provide a control loop in order to operate continuously. A decentralized controller implemented in Simulink, developed by Ricker, is used as the process controller. The Ricker implementation accurately matches the plant simulator, and the controller is a separate software process that runs on a separate computer from the plant simulator. To provide communication between the plant simulator and the controller, a hardware PLC with an industrial network protocol capability is used. The industrial network protocol is used to communicate between the plant simulator and the PLC. The plant simulator sends its sensor information to the controller, and the controller algorithm uses the sensor inputs to compute the desired values of the actuators and then sends those values back to the plant simulator.

In the plant simulator computer, a multinode DeviceNet card was installed. DeviceNet is a common industrial protocol that is used in the automation industry to exchange data between control devices. The multinode card allows a single hardware device to emulate multiple virtual DeviceNet nodes. In this case, each sensor and actuator point are dedicated nodes. Therefore, 53 virtual nodes (41 for sensors and 12 for actuators) were configured in the system. A software interface was developed to send and receive sensor and actuator values between the plant simulator and the PLC, through DeviceNet. An OPC DA server is running on a Windows 7 computer, acting as the main data gateway for the PLC. The PLC communicates to the OPC DA server to update and retrieve all of the sensor and actuator information, respectively. This sensor and actuator information is also known as a "tag" in PLC terminology. The controller has a MATLAB Simulink interface that directly communicates with the OPC DA server.

An HMI and a data historian are implemented in the system. The HMI provides a graphical user interface (GUI) to present information to an operator or user about the state of the process. The data historian serves as the main database to record all of the process sensor and actuator information. Both the HMI and the data historian have built-in interfaces to establish connections to the OPC DA server to access all of the process information. An engineering workstation is used in the system for engineering support, such as PLC development and control, HMI development and deployment, and data-historian data retrieval.

All systems in the PCS are synchronized with the NTP server environment. A network packet analyzer tool is installed in all of the computers in the system to capture and analyze network packets. Other specialized software tools are also used to monitor the system. For example, an OPC data analyzer is used to monitor OPC data exchange, and DeviceNet logging is used to log DeviceNet-level traffic.

3.3. Behavioral Anomaly Detection Capabilities Demonstrated

The BAD capability was demonstrated by installing single products into each environment. Only one product was installed and performing BAD at any given time. The BAD capability is achieved by three different detection methods: network-based, agent-based, and historian/sensor-based. CyberX and SecurityMatters SilentDefense demonstrated network-based detection. Secure-NOK's SNOK Detector demonstrates agent-based detection. The OSIsoft Process Information (PI) System's PI Data Archive (historian) demonstrates sensor-based detection from historian data.

3.3.1. SecurityMatters SilentDefense

SecurityMatters SilentDefense utilizes sensors to passively sniff traffic at the Layer 3 peer-to-peer switches to monitor critical networks for anomalies. The SilentDefense product also uses a command center to manage and collect data from all sensors at an enterprise site. The installation and configuration procedures undertaken for the SecurityMatters SilentDefense product are provided in Appendix A.

3.3.2. Secure-NOK SNOK

Secure-NOK's SNOK is a cybersecurity monitoring and detection system tailored for industrial networks and control systems. SNOK utilizes nonintrusive endpoint monitoring agents and passive network monitoring from Layer 2 and Layer 3 switches. The SNOK network intrusion detection system (IDS) comes preinstalled on an appliance, and endpoint monitoring agents are integrated into the asset owner's environment. The installation and configuration procedures undertaken for the Secure-NOK SNOK appliance are provided in Appendix B.

3.3.3. CyberX

The CyberX platform delivers continuous operational technology (OT) threat monitoring and asset discovery, combining a deep understanding of industrial protocols, devices, and applications with OT-specific behavioral analytics, threat intelligence, risk and vulnerability management, and automated threat modeling. The platform is delivered as a preconfigured appliance, including the IP address, subnet mask, default gateway, and Domain Name System (DNS) servers utilized in the build environment. The installation and configuration procedures undertaken for the CyberX appliance are provided in Appendix C.

3.3.4. OSIsoft PI Data Archive

The OSIsoft PI System's PI Data Archive is a component of the PI System that retrieves, archives, and enables high-performance data storage and rapid retrieval by using minimal disk space. The installation and configuration procedures undertaken for OSIsoft's PI System software are provided in Appendix D.

3.4. Behavioral Anomaly Detection Methods and Security Functions

Table 3-1 identifies methods used in this project and provides a mapping between the method type, the function performed, and the security control(s) provided. Refer to Table 2-1 for an explanation of the Cybersecurity Framework subcategory codes.

Table 3-1 BAD Methods and Security Functions

Type	Function	CSF Subcategories
Network-based	Identifies, monitors, and reports anomalous ICS traffic that might indicate a potential intrusion. Collects ICS network traffic via passive (agentless) monitoring. The system uses deep packet inspection to dissect traffic from both serial and Ethernet control network equipment.	DE.AE-1, DE.AE-2, DE.AE-5, DE.CM-1, DE.CM-4, DE.CM-7, DE.DP-4
Historian/sensor-based	Gathers raw data, records process data, and creates calculations. Provides monitoring and performance alerts of the process historian. The historian accesses historical data and consolidates it with current, real-time data. It allows for investigating intermittent issues, troubleshooting equipment failures, comparing current versus past production performance, and measuring new-plant startups against existing facilities.	Does not support a NIST Cybersecurity Framework subcategory in and of itself. It provides the data to be monitored by the ICS behavior monitor (next item). Related subcategories: DE.AE-5, DE.CM-1

Type	Function	CSF Subcategories
Agent-based	Identifies, monitors, and reports anomalous ICS traffic that might indicate a potential intrusion. Uses nonintrusive software agents to monitor the ICS network that requires no updating. The network IDS passively collects data from the ICS / Supervisory Control and Data Acquisition (SCADA) network via Switch Port Analyzer (SPAN)/mirroring ports. The nonintrusive host-monitoring agents collect data from within endpoints. The agents send event information to the detector, which looks for early warnings of cybersecurity attacks, and alerts on the anomalies detected by using a web interface.	DE.AE-1, DE.AE-2, DE.AE-5, DE.CM-1, DE.CM-4, DE.CM-7, DE.DP-4

3.5. Typographic Conventions

Table 3-2 presents the typographic conventions used in this NISTIR's descriptions of scenarios and demonstration findings.

Table 3-2 Typographic Conventions

Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For detailed definitions of terms, see the *CSRC Glossary*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit.**
`Monospace`	command-line input, on-screen computer output, sample code examples, and status codes	`Mkdir`
`Monospace Bold`	command-line user input contrasted with computer output	**`service sshd start`**
blue text	link to other parts of the document, a web Uniform Resource Locator (URL), or an email address	All publications from NIST's NCCoE are available at https://www.nccoe.nist.gov.

4. Demonstration Scenarios and Findings

With both the robotic and process-control infrastructures available for immediate use, the implementation of the BAD capabilities consisted of installing and integrating a single tool with the existing infrastructures. The BAD products are installed within the demonstration environment's DMZ of the existing infrastructure.

4.1. Network-Based Behavioral Anomaly Detection

Network-based anomaly detection requires the aggregation of all network traffic into a single collection point. Multiple appliances can also be used with centralized management to collect network traffic data from different zones and sites. Network traffic is examined and compared with a preexisting baseline, which is assumed to be normal at the time that it is captured. Should the network traffic show deviations from this baseline or show any other types of behavior considered suspicious or unauthorized, an alert will be generated based on preconfigured parameters.

During network-based anomaly detection, network traffic from the CRS and PCS LAN networks is aggregated at the demonstration environment’s DMZ via SPAN ports. At the demonstration environment’s DMZ, the traffic is inspected by the CyberX and SilentDefense platforms. Once a baseline of network traffic is established as normal, this aggregation of traffic can show deviations from the baseline, triggering an alert based on preconfigured parameters. Parameters can be configured to trigger alerts relating to network-traffic deviations, user-behavior deviations, volumetric deviations, and protocol deviations.

4.2. Agent-Based Behavioral Anomaly Detection

Agent-based anomaly detection combines some of the features of network-based anomaly detection with the nonintrusive monitoring of endpoints. Agent-based anomaly detection uses distributed software agents installed onto or close to devices, such as servers, HMIs, network switches, and controllers. Agents collect and preprocess device information, such as the use of removable media; logged-in users; ingress/egress traffic; device configurations; process and program details; and device parameters, such as memory, disk, and processor utilization. The collected information is sent securely to a detection engine. The detection engine alerts on deviations from preconfigured security policies and preexisting baselines. The preexisting baselines are reviewed and accepted as normal at the time that they are captured.

During agent-based anomaly detection, the behavior of Windows 7 devices in the PCS network, and of Ubuntu Linux devices in the CRS network, was monitored. The host agent information and network traffic are inspected by the Secure-NOK SNOK Detector. Once a baseline of the device configuration and behaviors is established as normal, deviations will trigger alerts.

4.3. Historian-Based and Sensor-Based Behavioral Anomaly Detection

Operational historian/sensor-based anomaly detection relies on the collection of sensor data into ICS network components, such as operational historians. Because historians are constantly being fed real-time operational data, which has already been configured within operational bounds, or set points, any deviations from these thresholds will produce an alert that can be captured. Typically, this would be considered an operational anomaly. OSIsoft’s PI Data Archive performs historian/sensor-based detection.

4.4. Demonstration Results and Findings

The demonstration effort examined 16 classes of BAD. These 16 classes for which anomalous events were successfully detected include the detection of the following items:

- plaintext passwords
- user authentication failures
- new network devices
- abnormal network traffic between devices
- internet connectivity
- data exfiltration
- unauthorized software installations
- PLC firmware modifications
- unauthorized PLC logic modifications
- file transfers between devices
- abnormal ICS protocol communications
- malware
- denial of service (DoS)
- abnormal manufacturing system operations
- port scans/probes
- environmental changes

Each of the demonstration events addressed threats that would not normally be detected by current security tools that involve monitoring system behaviors for predefined out-of-specification settings or readings or that involve threat signatures (information elements previously identified as being associated with threats or vulnerability characteristics, such as with an IDS or an intrusion protection system). Network-based, agent-based, and historian/sensor-based detection capabilities were examined. Each product that was demonstrated performed as expected.

As indicated in Section 4.1, individual products were examined in different scenarios, and not all types of detection events were examined in each scenario. As a result, no comparison of product detection capabilities can usefully be made or is appropriate to this NISTIR.

The installation, configuration, anomaly scenarios, and results for each tool are described in the appendixes of this document.

5. Conclusion

The goal of this project was to demonstrate BAD techniques that businesses can implement and use to strengthen the cybersecurity of their manufacturing processes. The BAD project demonstrated three different detection methods: network-based, agent-based, and operational historian/sensor-based. We have shown that BAD techniques can serve as a key security component in sustaining ICS operations. This NISTIR illustrates the use of the different BAD capabilities, to provide a better understanding of what each of the techniques offers and how to apply each of these techniques in different ICS network environments.

Appendix A. SecurityMatters SilentDefense Supplemental Information

SecurityMatters SilentDefense utilizes sensors to passively sniff traffic at the Layer 3 peer-to-peer switches to monitor critical networks for anomalies. The SilentDefense product also uses a command center to manage and collect data from all network-based sensors within a manufacturing system.

A.1. Build Architecture

The SilentDefense dedicated appliance was physically installed in the measurement rack of the Cybersecurity for Smart Manufacturing Systems (CSMS) environment. Three existing Switch Port Analyzer (SPAN) ports from each system (collaborative robotic system [CRS] and process control system [PCS]) were connected to dedicated network interfaces on the appliance, for a total of six SPAN ports. The SPAN port connections to the appliance, within the PCS and CRS networks, are shown in Figure A-1 and Figure A-2, respectively.

The appliance network connection was connected to the demilitarized zone (DMZ) network located in the test bed's measurement rack, to isolate the appliance's network traffic from the rest of the network. Engineering laptops were used to interface with the SilentDefense graphical user interface (GUI) via network connections to the DMZ. More information regarding the specific configuration of the test-bed network can be found in Section 3.

Figure A-1 SPAN Port Connections to the SilentDefense Appliance in the PCS

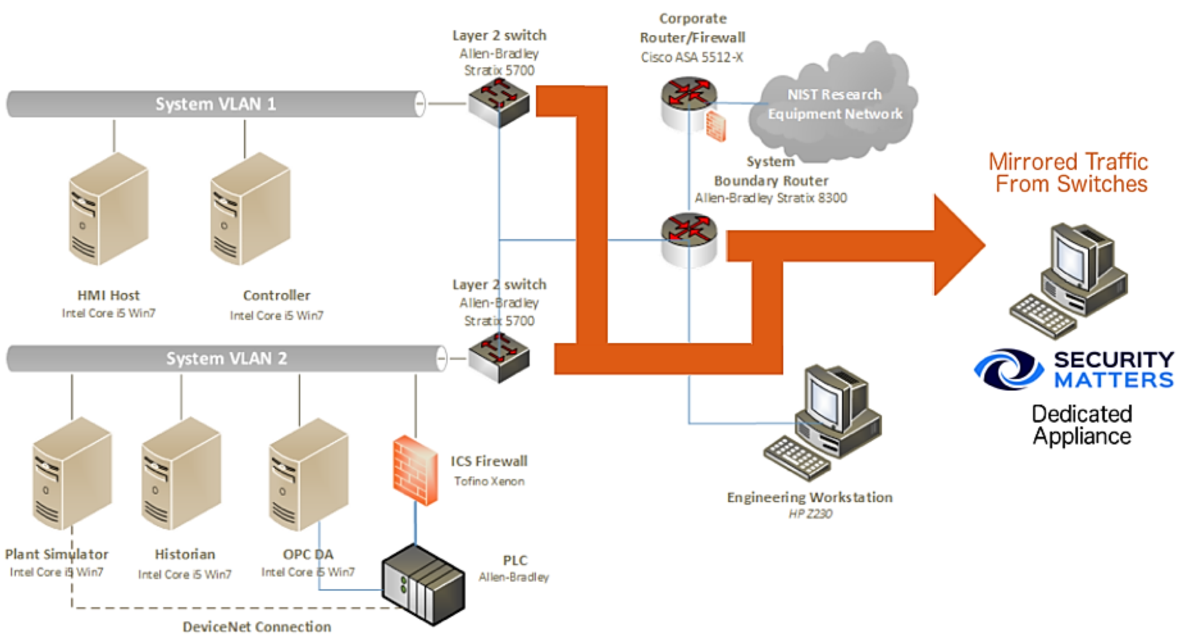
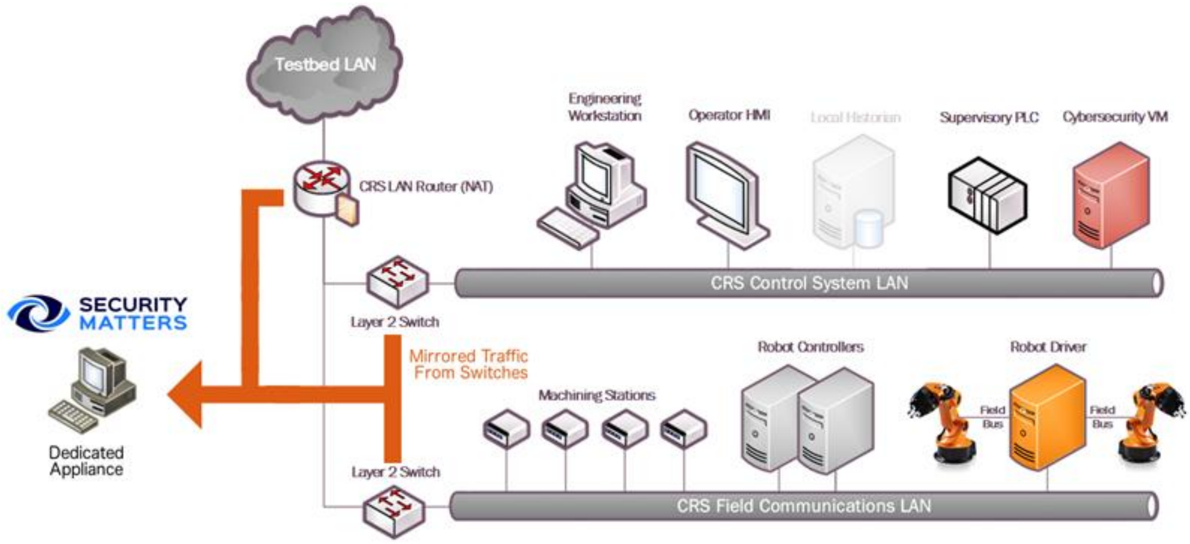


Figure A-2 SPAN Port Connections to the SilentDefense Appliance in the CRS



A.2. Installation and Configuration

Physical hardware and software were provided by SecurityMatters for this demonstration. After the hardware appliance was received, it was installed into the CSMS test bed. Soon after the initial installation, engineers from SecurityMatters arrived on site to complete the installation and configuration of the tool. The following subsections describe the steps taken to install and configure the appliance.

A.2.1. Hardware

The SilentDefense appliance was installed as a bundle (with the sensor and the command center on the same hardware). Typically, these functions are separated in production installations; however, because this was a lab system, the bundle was sufficient for the demonstration environment. The bundled hardware was a Dell R630 1U Rackmount Server with the following specifications:

- central processing unit (CPU): Intel Xeon E5-2620, 2.4 gigahertz, 5-megabyte (M) cache, 6C/12T (6 cores and 12 threads)
- random-access memory: 32 gigabytes (GB), registered dual in-line memory module, 2,400 megatransfers per second
- hard drive: 800 GB, solid-state drive
- redundant array of independent disks controller: PERC H730, 1 GB cache
- sniffing network interface card (NIC): Intel i350 Quad Port Peripheral Component Interconnect Express Card

A.2.2. Operating System

SilentDefense 3.11.1 uses the Ubuntu 16.04.3 Long-Term Support (LTS) Server operating system (OS), which is modified with two scripts. First, there is a SecurityMatters OS update script to update libraries to the latest versions and to install some new libraries necessary for SilentDefense operation. The OS is then modified with a main-configuration script, which hardens the OS by performing operations, such as disabling users, setting iptables, and setting the update repository addresses to local hard-drive folders (so that, automatic updates are not from the internet). The steps for modifying the OS are as follows:

1. Install the Ubuntu 16.04.3 LTS Server OS.
2. Run the SilentDefense OS by using the following command:

```
sudo ./update_os_16.04.3_to_29.11.2017.run
```

3. Reboot the system by using the following command:

```
sudo reboot now
```

4. Run the SilentDefense main-configuration script by using the following command:

```
sudo ./main configuration_29.11.2017.run
```

A.2.3. Configure Sniffing Ports

The Intel i350 card has four sniffing ports to configure. This configuration is done through the SilentDefense `sdconfig` utility:

1. Run the SilentDefense configuration utility by using the following command:

```
sudo sdconfig
```

2. Choose the option **Configure New Monitoring Interface**.
3. Select the four Intel i350 NIC interfaces by using the space bar on your keyboard.
4. Click **OK**.
5. Choose the option **Exit this configuration Utility**.

A.2.4. Configure the Management Port Internet Protocol Address

The SilentDefense system has a management port that is used to connect to the sensors and for the SilentDefense administrators and analysts to access the system GUI. This configuration is done through the SilentDefense `sdconfig` utility:

1. Run the SilentDefense configuration utility by using the following command:

```
sudo sdconfig
```

2. Choose the option **Remove management interface configuration**.
3. Choose the option **Configure management interface**.
4. Type in the following information:
    a. **IP address** (Internet Protocol address)

    b. **subnet mask**
    c. **gateway**
    d. **Domain Name System server(s)**
5. Press **OK**.

A.2.5. Configure the SPAN Ports on Layer 3 Network Switches

The SilentDefense passive monitoring system uses SPAN ports to intercept and analyze network packets. The process to configure a SPAN port varies among different makes and models of networking hardware. For SPAN port configuration information, consult the current configuration manual or user guide for the specific networking hardware.

A.2.6. Log into SilentDefense

The SilentDefense GUI has a default username and password of `admin`. Upon the first login, you are required to change the password to something more secure. The SilentDefense software will not allow the new username and password to be the same.

1. Browse to the SilentDefense GUI from a web browser, using the following Uniform Resource Locator (URL):

```
https://<mgmt_ip_address>
```

2. Type the username `admin` and the password `admin` in the login fields, and then click **Sign in**.
3. A new window pops up, requiring you to change the password. Type in a new password that meets the following requirements:
    a. Contains eight characters minimum
    b. Does not contain the account name
    c. Contains at least three character groups (e.g., uppercase, lowercase, number, special)
4. Click **Apply**.
5. The dashboard now appears, and you can begin to use SilentDefense.

A.3. Anomaly Scenarios

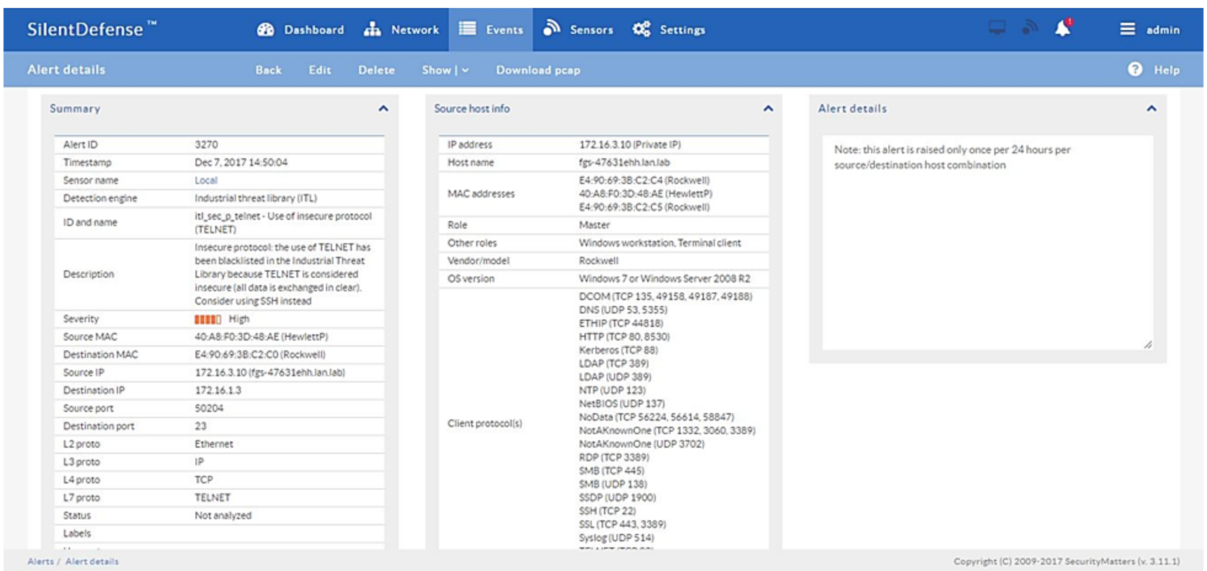
The network-based anomaly detection method was demonstrated for the scenarios detailed in the following subsections. Each scenario includes a description of the anomaly, a detailed description of how each demonstration event was conducted in the CSMS environment, and the observed results.

For the sake of brevity, only a subset of the alerts observed during each anomaly scenario is shown. However, each anomaly scenario includes a screenshot of the alerts summary (or aggregated summary) observed after the anomaly scenario had completed.

A.3.1. Unencrypted Passwords Are Used to Access a Networking Device

Unencrypted or plaintext passwords transmitted over a network are a vulnerability for industrial control system (ICS) networks. If packets containing these credentials are intercepted, then the passwords can be easily unmasked and can be used to obtain unauthorized access to devices or services that use those credentials. This vulnerability can be amplified if multiple devices utilize the same credentials.

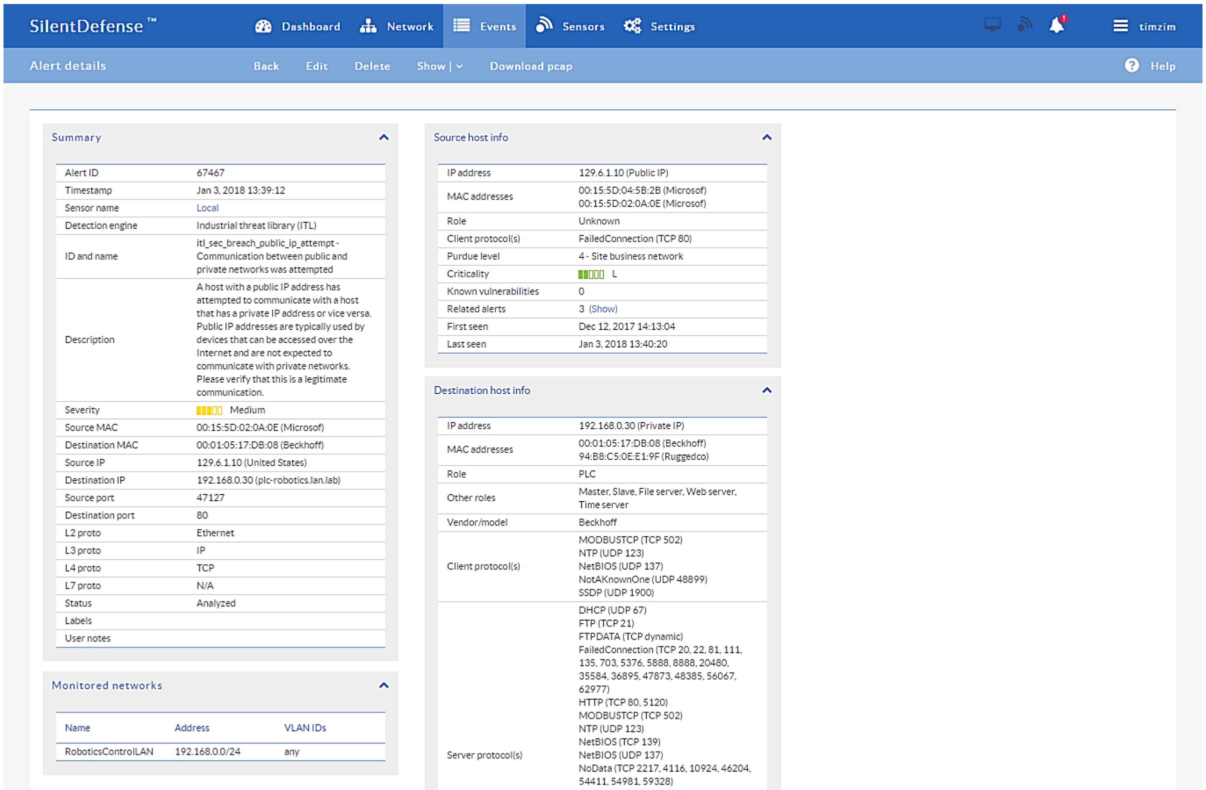
This anomaly was executed on the PCS. The network switches and router provide a Telnet service for remote management. This protocol transmits user credentials as plaintext. A Telnet connection was opened between the engineering workstation and Virtual Local Area Network (VLAN)-1 by using the open-source `PuTTY` [12] client.

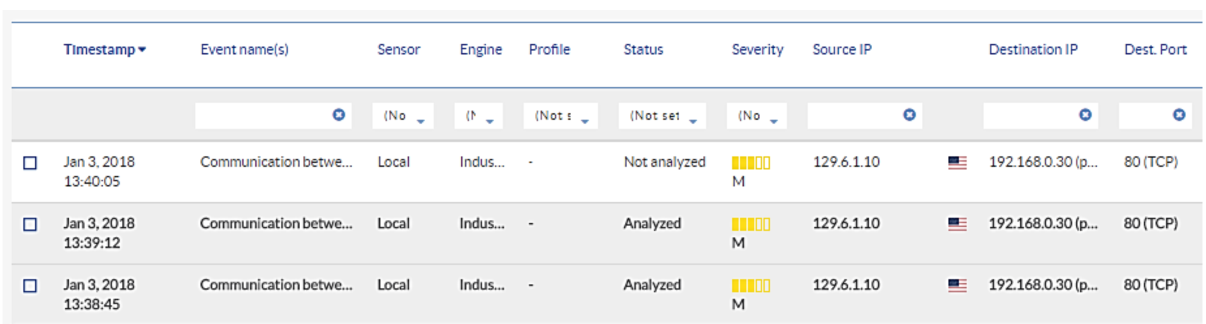


A.3.2. Transmission Control Protocol Connection Requests Are Received from the Internet

When attempting to form a connection by using the transmission control protocol (TCP), a connection request first must be sent to the server. If a TCP connection request is received from the internet (i.e., it has a public Internet Protocol [IP] address), then this can indicate a network misconfiguration, a device misconfiguration, or an unidentified internet connection within the lower levels of the ICS network.

This anomaly was executed on the CRS. The packet manipulation tool Scapy [13] was used with Python [14] to create a TCP SYN packet with a public IP as the source address (129.6.1.10) and with the programmable logic controller (PLC) IP as the destination address, and was injected into the CRS local area network (LAN).

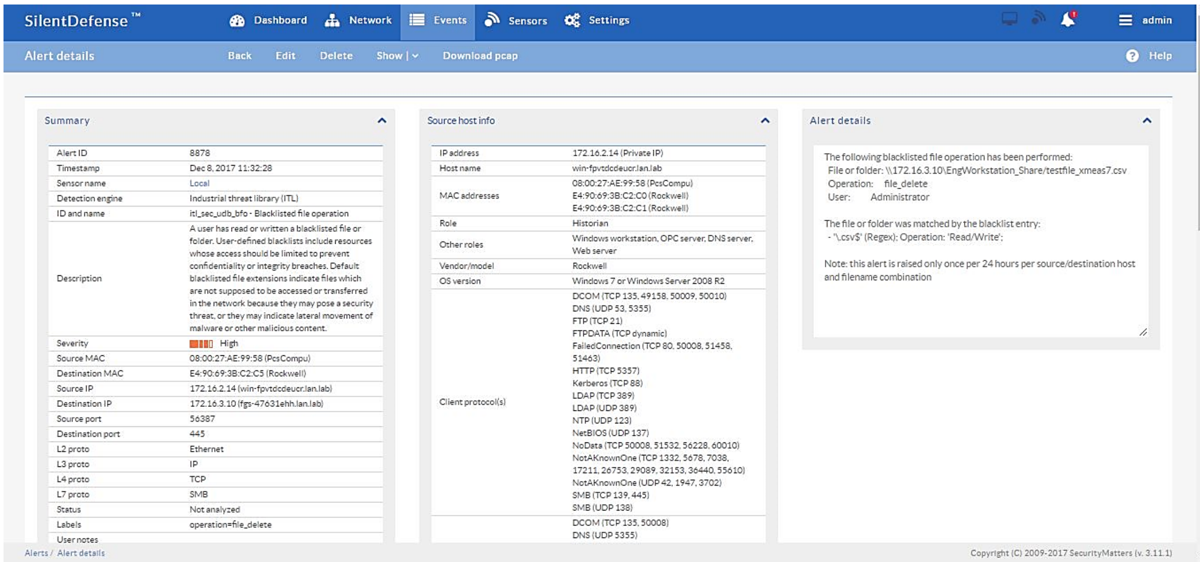


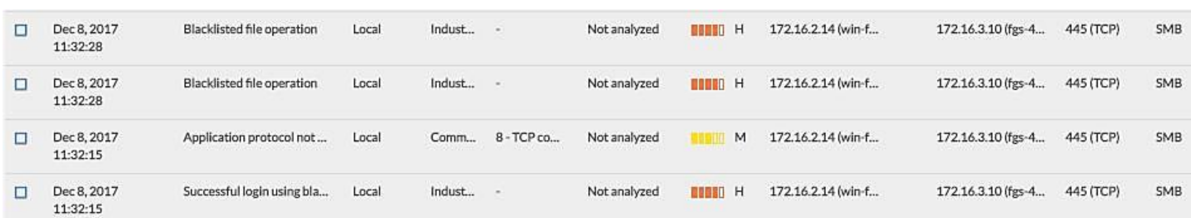


A.3.3. Data Exfiltration Between ICS Devices via Server Message Block

Vulnerable devices within an ICS network can be used as a pivot to bring higher-value targets within reach to exfiltrate data (e.g., using a vulnerable Internet of Things device to pivot and leverage attacks against a PLC on the same network). Monitoring for abnormal communication patterns between ICS devices can help detect these types of attacks, especially if the affected devices do not communicate during normal operations.

This anomaly was executed on the PCS. An unauthorized Windows File Share (using the Server Message Block protocol) was configured between the human-machine interface (HMI) server and the engineering workstation. Three types of files were transferred over the share: a comma-separated values (CSV) file, a Microsoft Excel workbook (XLSX) file, and an Adobe Portable Document File (PDF).

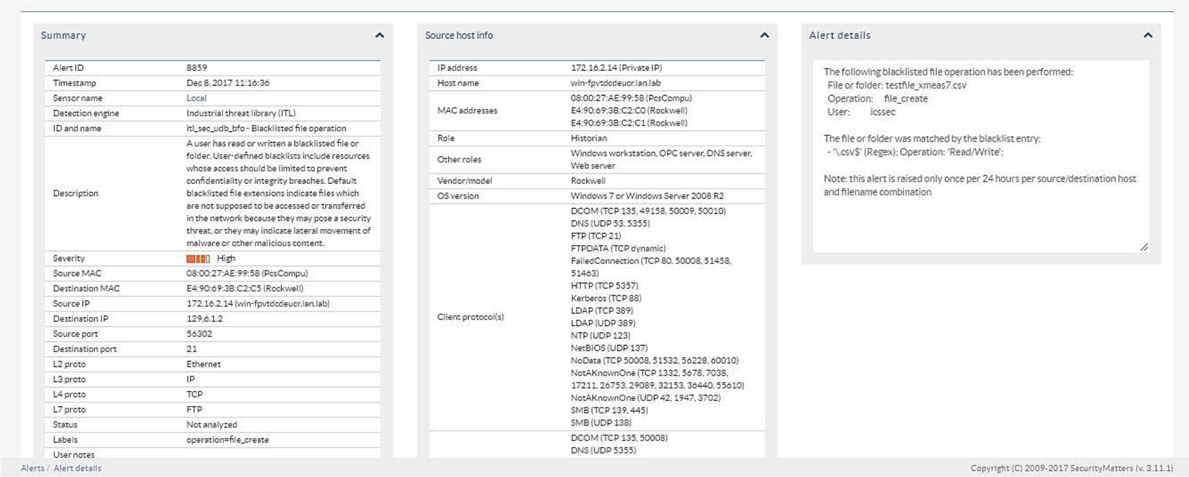


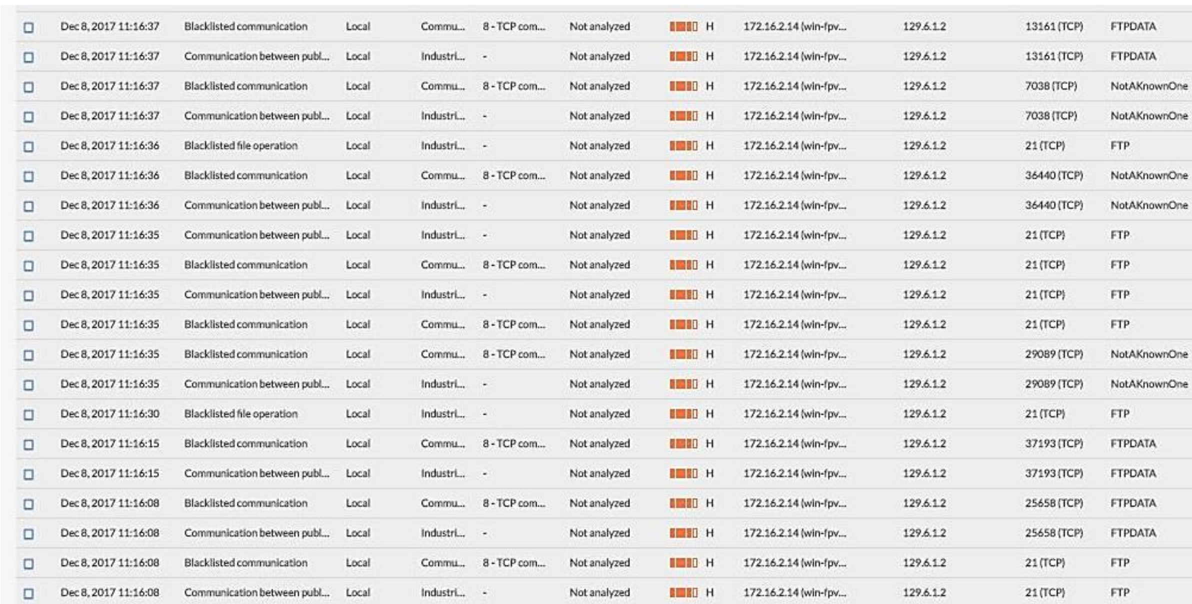


A.3.4. Data Exfiltration to the Internet via File Transfer Protocol

Attacks against ICS, with the goal of information gathering, must (at some point) attempt to exfiltrate the data from the ICS network, likely utilizing the internet as a transport mechanism. Monitoring for ICS devices communicating over the internet can help detect data exfiltration events, especially if the affected device does not normally communicate over the internet. Depending on the protocol used for exfiltration, the file contents and/or data being exfiltrated may be ascertainable (e.g., file names, file types, data transferred using the File Transfer Protocol [FTP]), providing insight into the impact of the anomaly.

This anomaly was executed on the PCS. An FTP server was installed and configured on a server with an internally routed public IP address (129.6.1.2). The FileZilla FTP client [15] was installed on the historian server and was used to transfer three types of files to the simulated "internet-based" FTP server: a CSV file, an XLSX file, and an Adobe PDF.



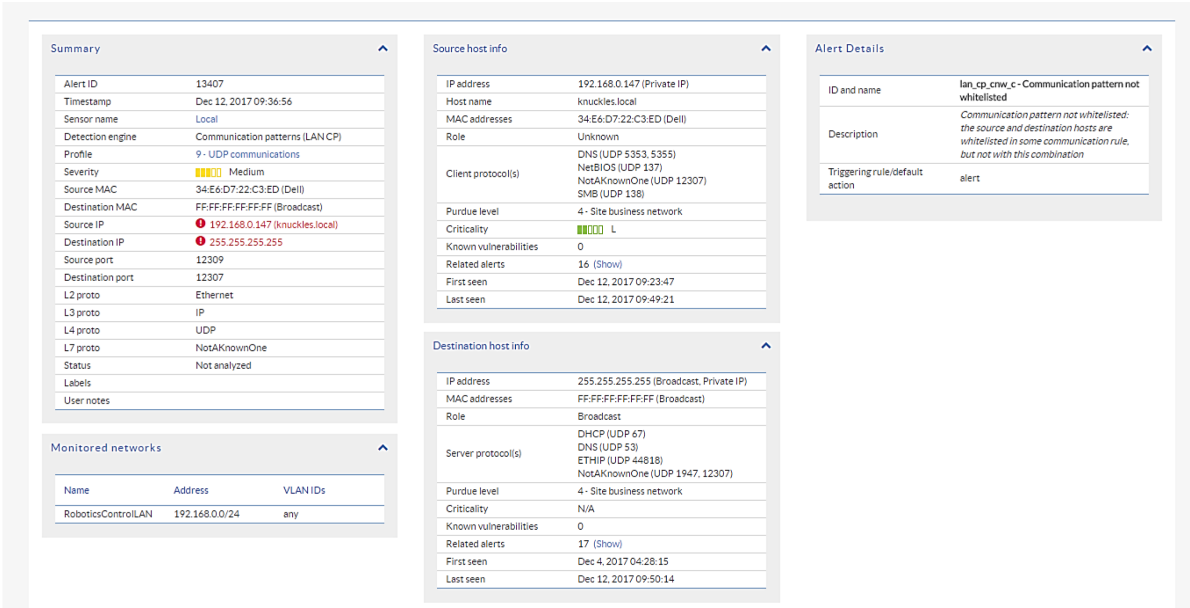


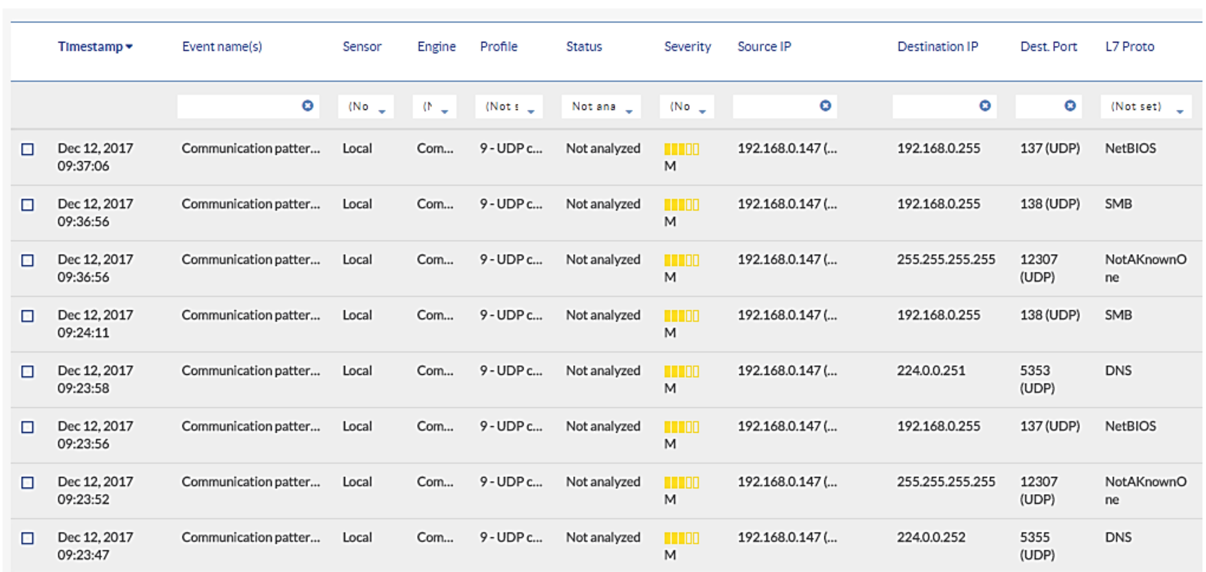
Dec 8, 2017 11:16:37	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	13161 (TCP)	FTPDATA
Dec 8, 2017 11:16:37	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	13161 (TCP)	FTPDATA
Dec 8, 2017 11:16:37	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	7038 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:37	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	7038 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:36	Blacklisted file operation	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:36	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	36440 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:36	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	36440 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:35	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:35	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:35	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:35	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:35	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	29089 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:35	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	29089 (TCP)	NotAKnownOne
Dec 8, 2017 11:16:30	Blacklisted file operation	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:15	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	37193 (TCP)	FTPDATA
Dec 8, 2017 11:16:15	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	37193 (TCP)	FTPDATA
Dec 8, 2017 11:16:08	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	25658 (TCP)	FTPDATA
Dec 8, 2017 11:16:08	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	25658 (TCP)	FTPDATA
Dec 8, 2017 11:16:08	Blacklisted communication	Local	Commu...	8 - TCP com...	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP
Dec 8, 2017 11:16:08	Communication between publ...	Local	Industri...	-	Not analyzed	H	172.16.2.14 (win-fpv...	129.6.1.2	21 (TCP)	FTP

A.3.5. Unauthorized Device Is Connected to the Network

It is important to identify all devices on the ICS network, for a complete risk analysis and for minimizing potential attack vectors. The detection of unauthorized devices attached to the ICS network may indicate anomalous activity. These unauthorized devices are important to find and remove, especially because the purpose of an unauthorized device is unknown and may be malicious.

This anomaly was executed on the CRS. The engineering laptop (Windows 7 OS) was removed from the network during the baseline phase of the tool configuration and was later connected to the CRS LAN to execute the anomaly. After the initial connection, background traffic was automatically generated onto the network by the laptop.

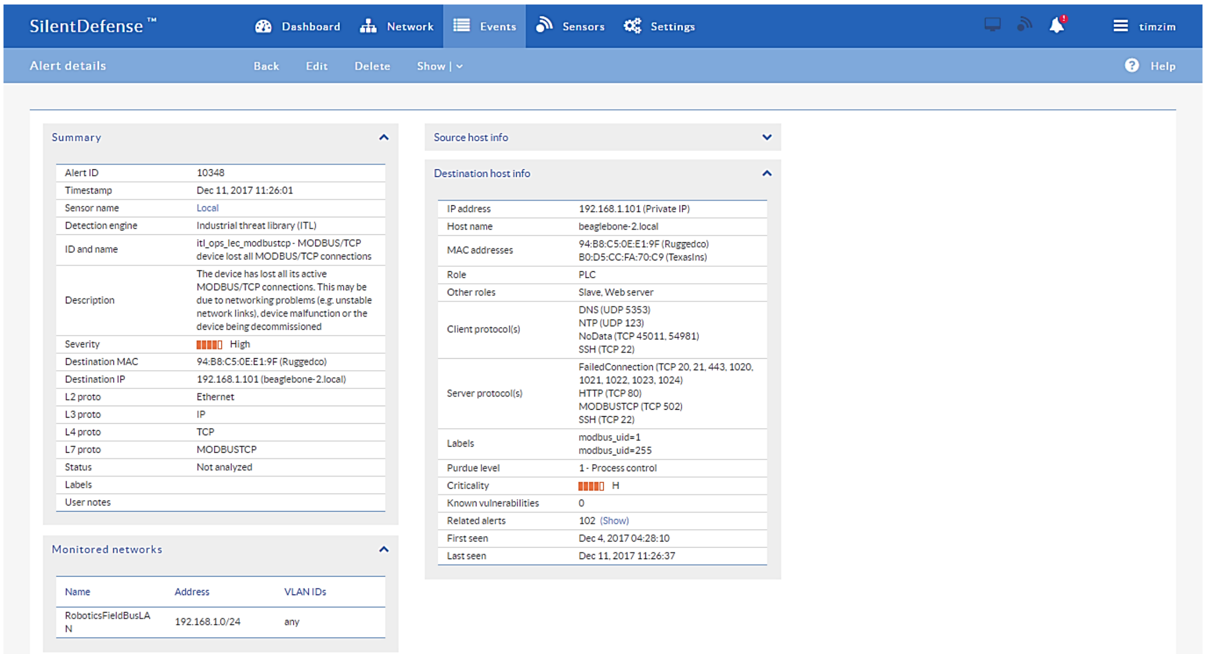


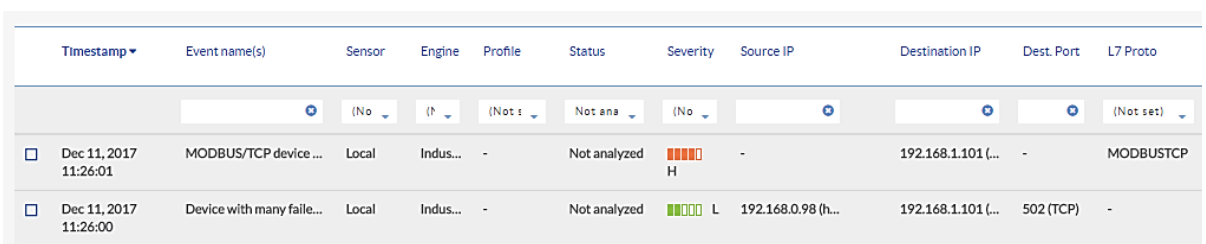


A.3.6. Loss of Communications with Modbus TCP Device

ICS devices must exhibit high availability to support manufacturing operations. This quality becomes more important as the speed of manufacturing operations increases (i.e., short cycle times). If an ICS device hosting a network service becomes unavailable during manufacturing operations, then this may be a sign of anomalous activity and should be investigated. Loss of communications with a device or service may be caused by a multitude of anomalies, including device restarts, software faults, high network utilization, and an increased processing load on the device.

This anomaly was executed on the CRS. A firewall rule was added to the Linux iptables (Linux kernel firewall) on Machining Station 1 to block all incoming packets on Modbus TCP Port 502. The firewall replied with a TCP reset for each incoming packet or connection request, to make it appear as is if the Modbus server had terminated and the TCP socket was closed.



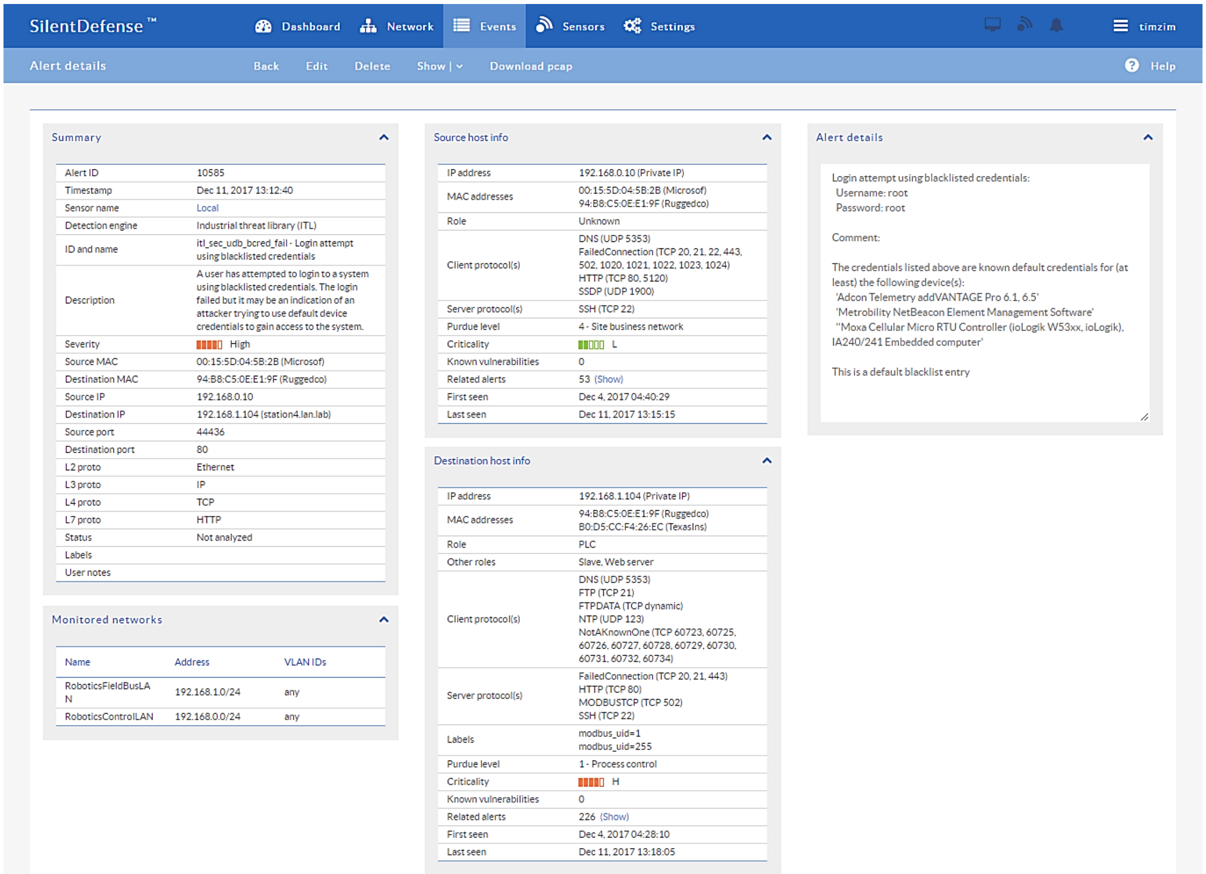


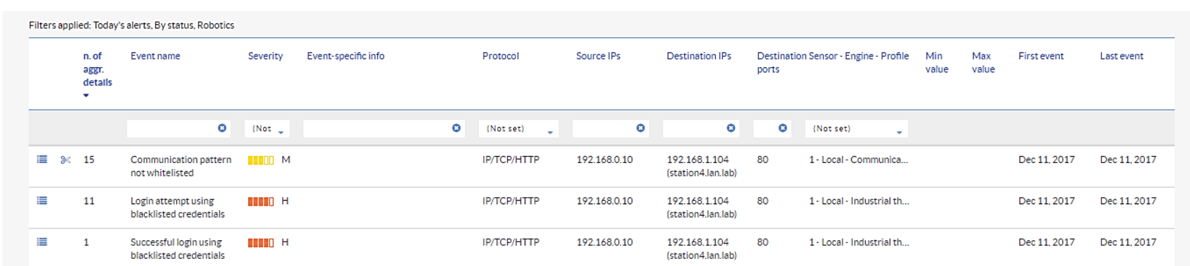
	Timestamp	Event name(s)	Sensor	Engine	Profile	Status	Severity	Source IP	Destination IP	Dest. Port	L7 Proto
			(No	(N	(Not s	Not ana	(No				(Not set)
☐	Dec 11, 2017 11:26:01	MODBUS/TCP device ...	Local	Indus...	-	Not analyzed	H	-	192.168.1.101 (...	-	MODBUSTCP
☐	Dec 11, 2017 11:26:00	Device with many faile...	Local	Indus...	-	Not analyzed	L	192.168.0.98 (h...	192.168.1.101 (...	502 (TCP)	-

A.3.7. Brute-Force Password Attack Against an ICS Device

Authentication systems that are not rate-restricted may be vulnerable to password-guessing attacks, especially if the default credentials of the device have not been changed. Compiled lists containing default user credentials are freely available on the internet, as are lists of commonly used usernames and passwords. Given enough time, an attacker may be able to access vulnerable systems by using a brute-force password attack.

This anomaly was executed on the CRS. The software Nmap [16] was used to generate the brute-force password attack by using the script `http-brute`. The attack was pointed at an Apache [17] Hypertext Transfer Protocol (HTTP) server on Machining Station 4, containing a directory that was protected by HTTP basic authentication. The HTTP server was not configured to limit the number of authentication attempts.

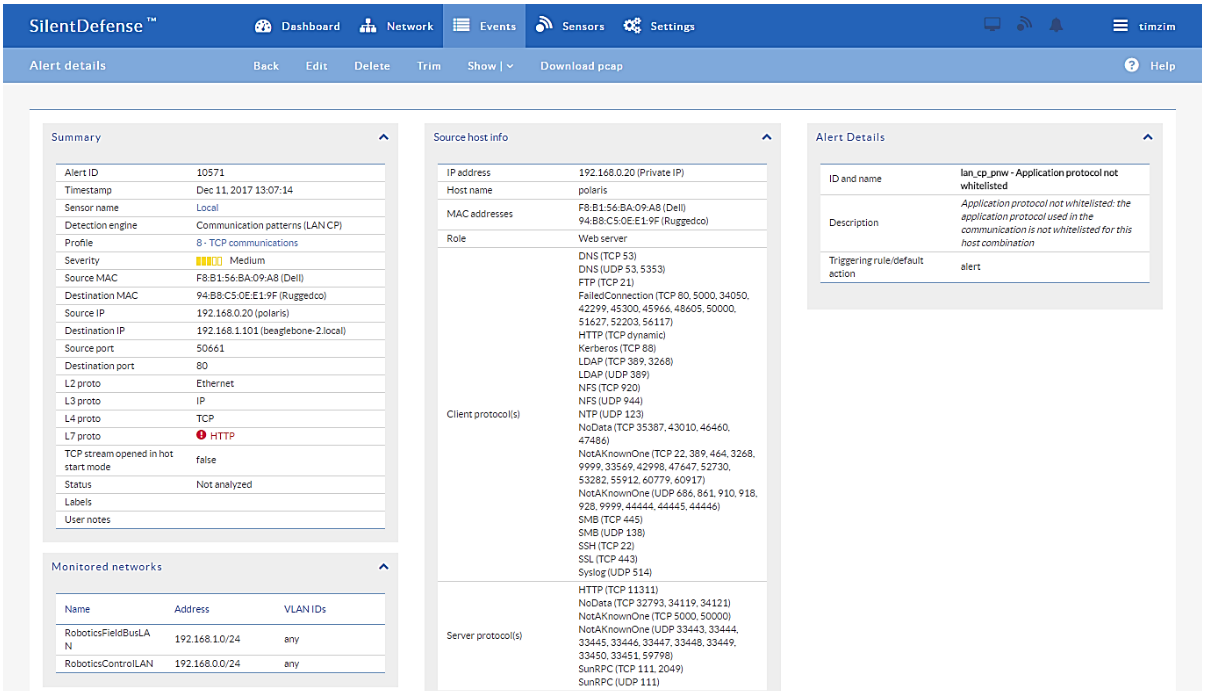


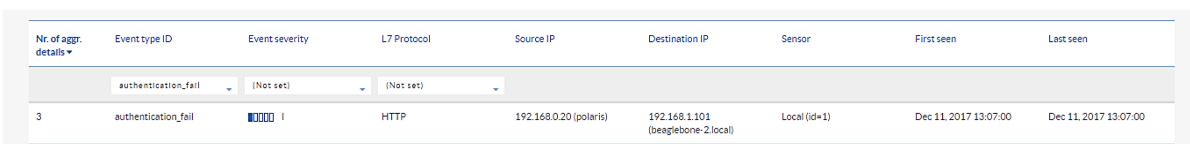


A.3.8. Invalid Credentials for Remote Access

While it can be expected that some users will accidentally enter invalid credentials on a daily basis, it is important to monitor these events for trends of anomalies. Large quantities of invalid-credential usage may indicate a password-guessing attack. These credentials may also be used to authenticate connections between ICS devices. With the increasing use of remote access for ICS devices, it is important to monitor these services for attempts made by attackers to gain unauthorized access.

This anomaly was executed on the PCS. A remote desktop session was initialized from the engineering workstation to the HMI server and required authentication with the Microsoft Active Directory service. Invalid credentials were submitted for authentication.

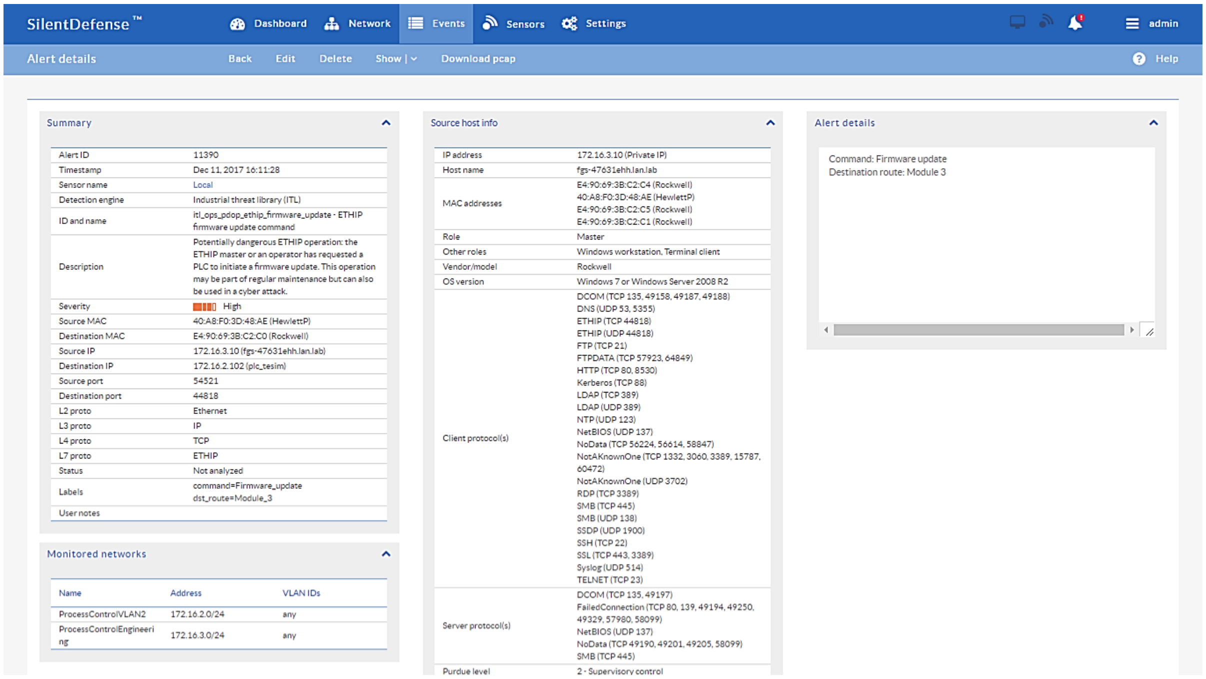




A.3.9. Unauthorized ICS Device Firmware Update

Many ICS devices provide services to remotely update firmware over the network. These network services can also provide a mechanism for attackers to replace valid firmware with malicious firmware if the device is not protected.

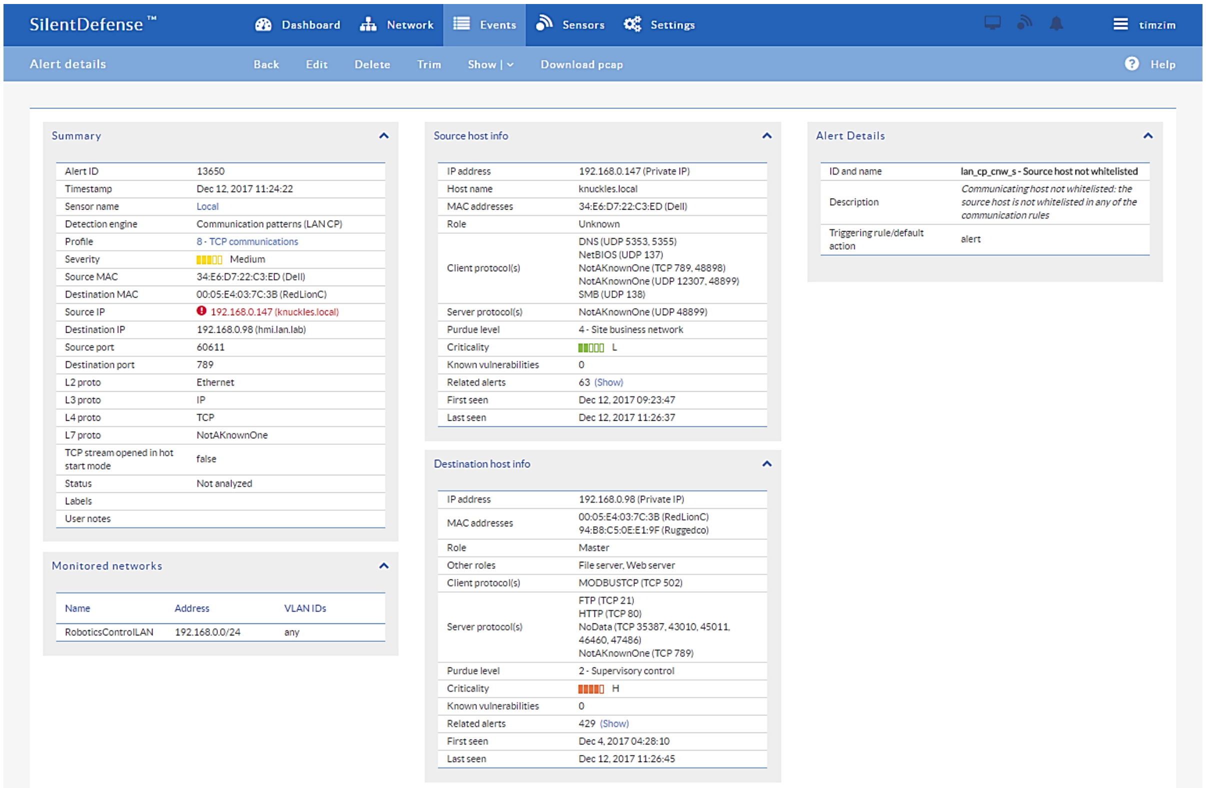
This anomaly was executed on the PCS. The Allen-Bradley PLC implemented in the PCS contains an Ethernet module (1756-EN2T) that allows its firmware to be upgraded and downgraded over Ethernet/IP. The firmware was upgraded or downgraded using the ControlFLASH firmware upgrade tool.

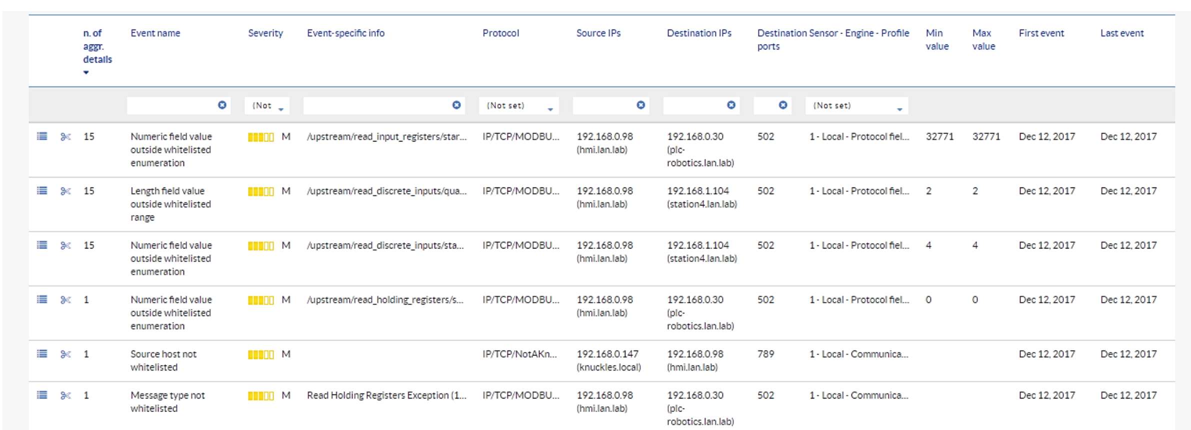


A.3.10. Unauthorized HMI Logic Modification

Many ICS devices provide services to remotely update control logic over the network. These network services can also provide a mechanism for attackers to replace valid control logic with malicious logic if the device is not protected. This is especially important for HMIs, as they are typically used by operators to monitor and manipulate the manufacturing process in a safe and controlled manner.

This anomaly was executed on the CRS. The database implemented on the CRS Red Lion HMI (Model G310) was modified and uploaded to the HMI by using the Red Lion Crimson 3.0 software. The Modbus TCP registers in the modified database differed slightly from those in the original database.

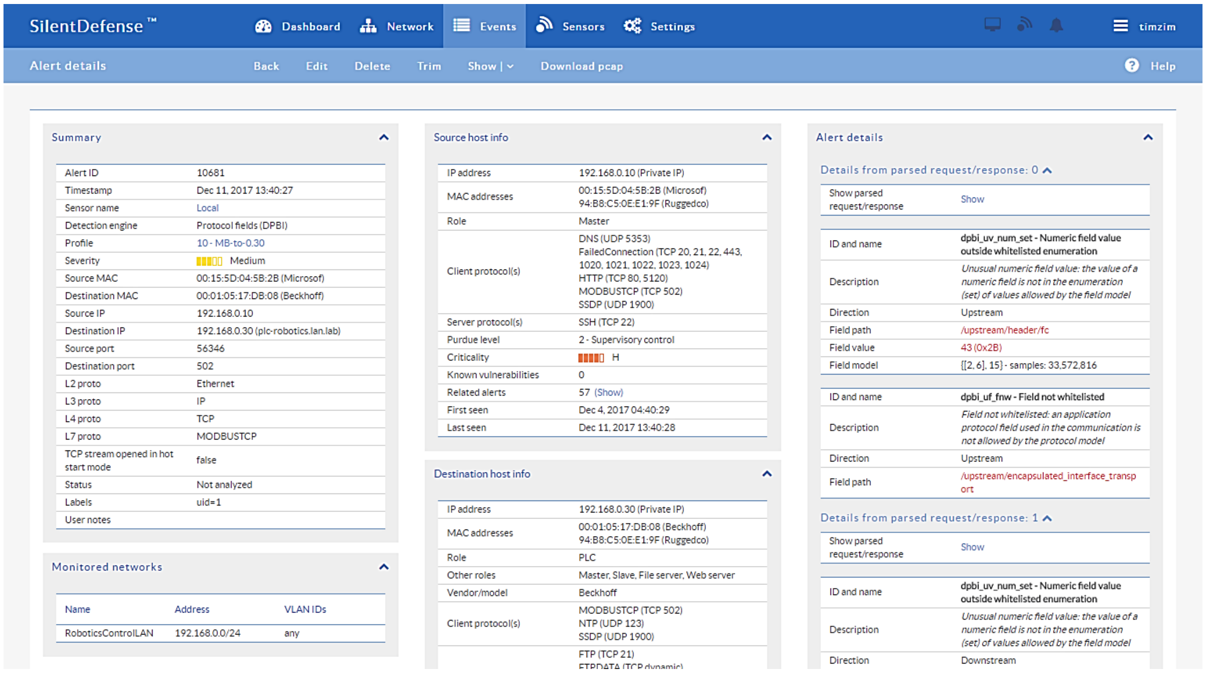


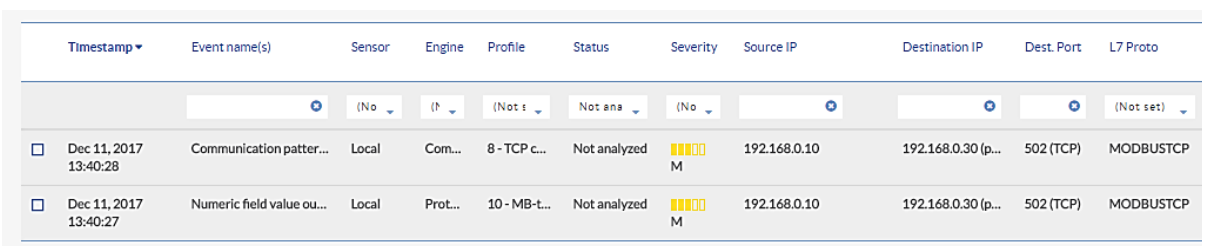


A.3.11. ICS Device Receives Diagnostic Modbus TCP Function Codes

Certain ICS network protocols enable diagnostic access to ICS devices. While this type of functionality enables remote maintenance and diagnostics to authorized personnel, it may also be leveraged by aggressors to compromise ICS devices.

This anomaly was executed on the CRS. Python [14] was used to create a Modbus TCP message with the diagnostic function code value of 43 (`0x2B`), known as encapsulated interface transfer. The message was generated by the cybersecurity virtual machine (CybersecVM) and was transmitted to the PLC Modbus server.

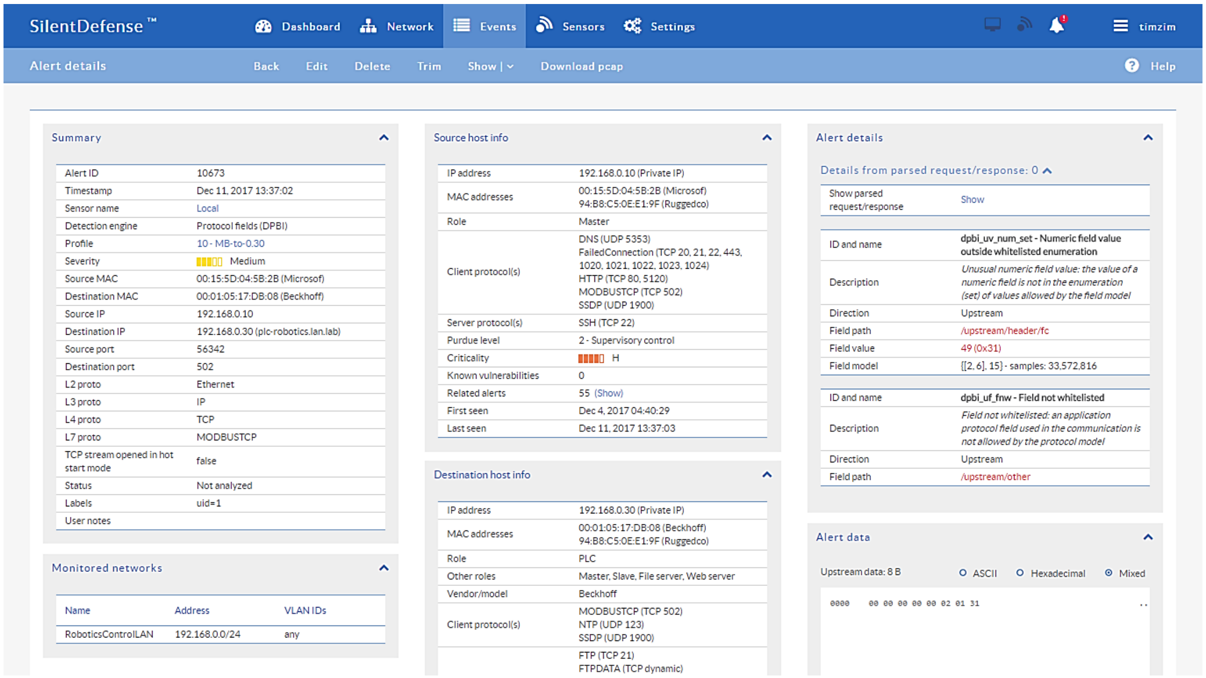


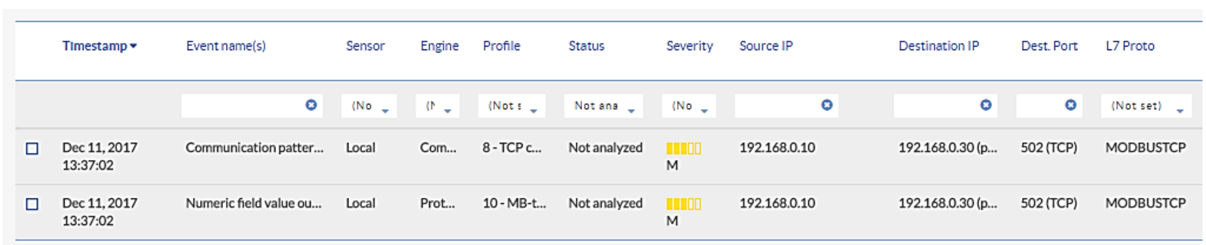


A.3.12. ICS Device Receives Undefined Modbus TCP Function Codes

Communications that do not conform to the defined specifications of the industrial protocol may cause an ICS device to act in an undefined or unsafe manner. Depending on the manufacturing process and the ICS device, the nonconforming communications may or may not be impactful, but investigation into the cause is warranted.

This anomaly was executed on the CRS. Python [14] was used to create a Modbus TCP message with the undefined function code value of 49 (`0x31`). The message was generated by the CybersecVM and was transmitted to the PLC Modbus server.

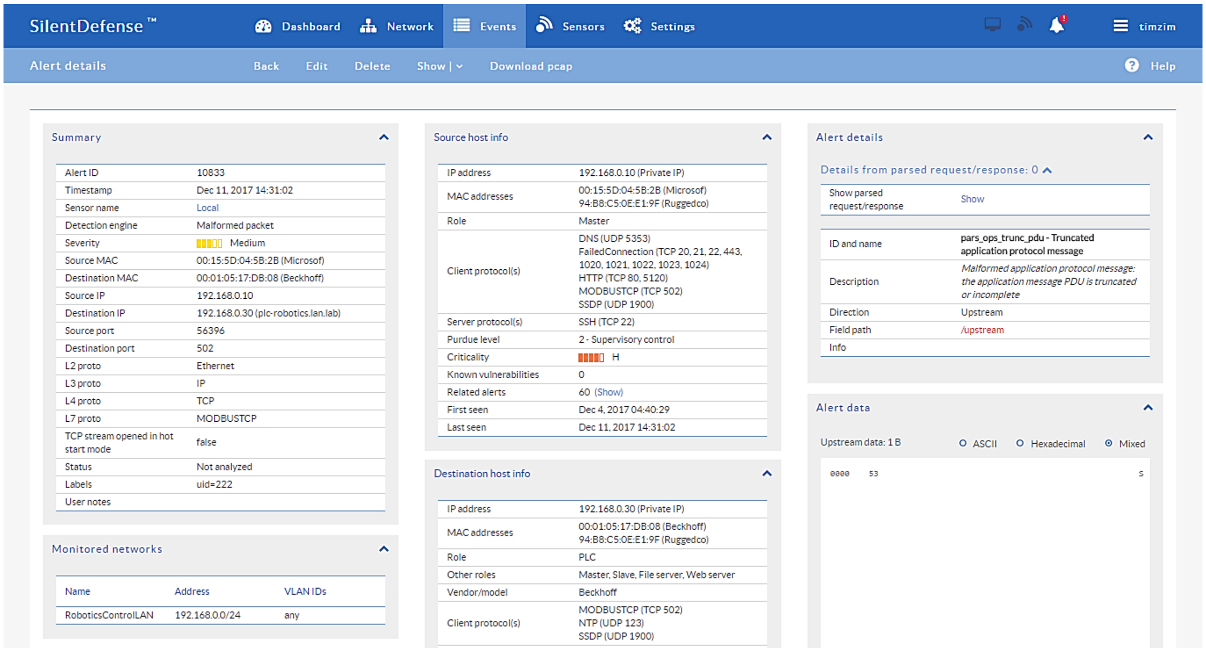


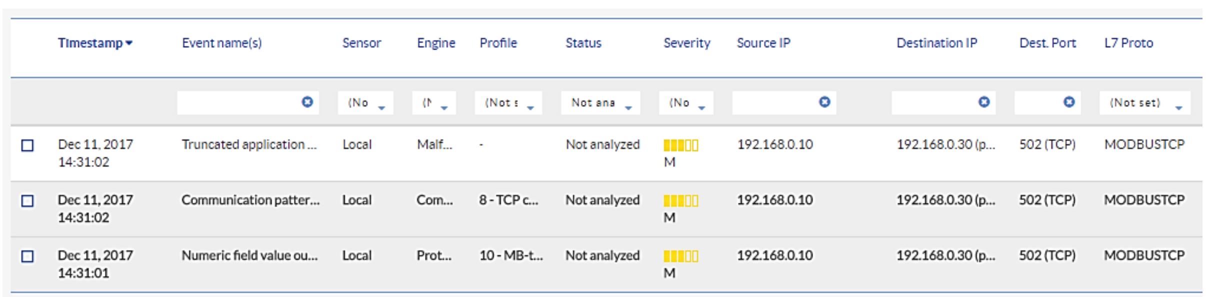


A.3.13. ICS Device Receives Malformed Modbus TCP Traffic

Communications that do not conform to the defined specifications of the industrial protocol may cause an ICS device to act in an undefined or unsafe manner. Depending on the manufacturing process and the ICS device, the nonconforming communications may or may not be impactful, but investigation into the cause is warranted.

This anomaly was executed on the CRS. Python [14] was used to create a malformed Modbus TCP message. The message was generated by the CybersecVM and was transmitted to the PLC Modbus server.

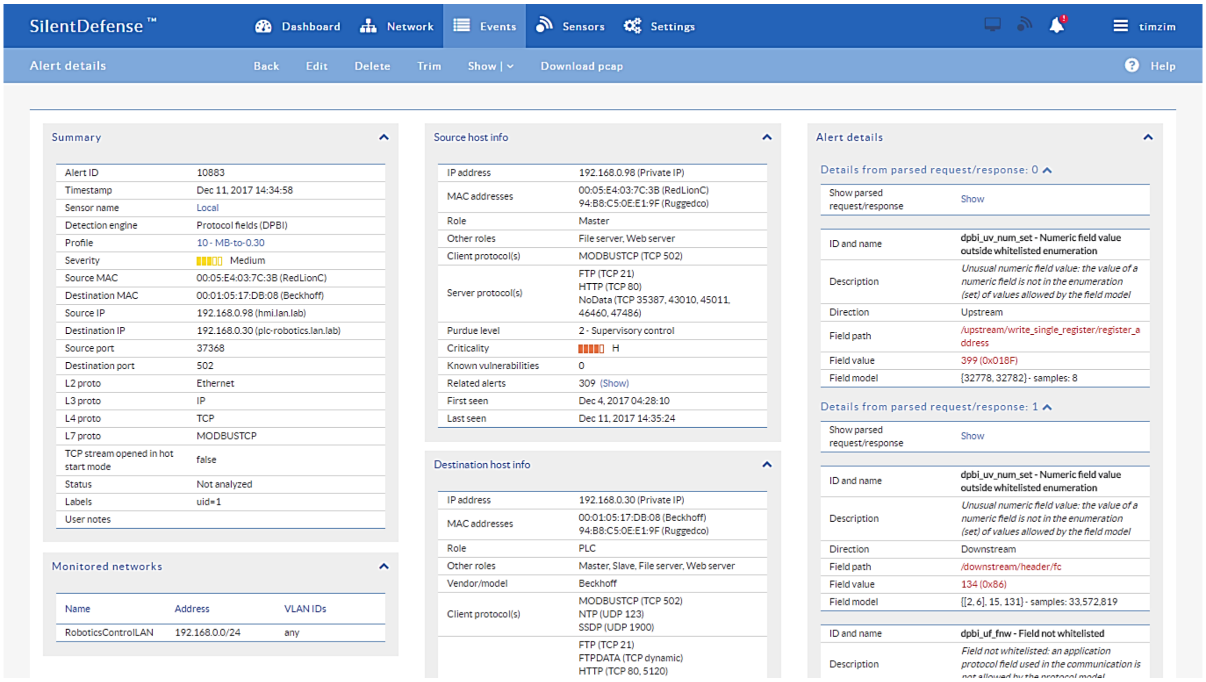


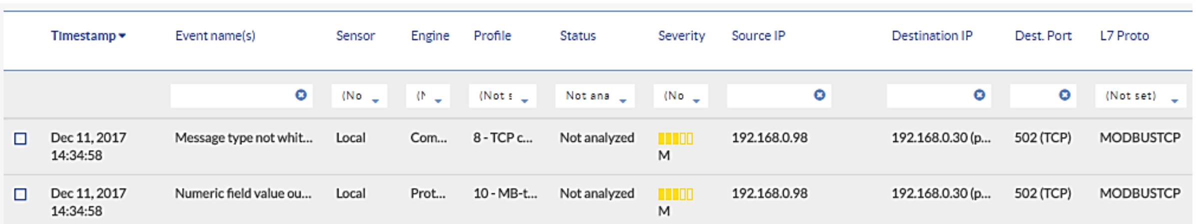


A.3.14. Illegal Memory Addresses of ICS Device Are Accessed

Some industrial protocols (like Modbus) require relative addressing to access ICS device registers. Attackers may attempt to modify illegal memory locations of ICS devices by using these types of industrial protocols or may attempt to cause the ICS device to act in an undefined or unsafe manner by modifying data located in a protected memory location.

This anomaly was executed on the CRS. The HMI database was modified to access an illegal register on the PLC Modbus TCP server when the anomaly was activated. The valid Modbus address range for the PLC registers is `0x8000` to `0x80FF`.

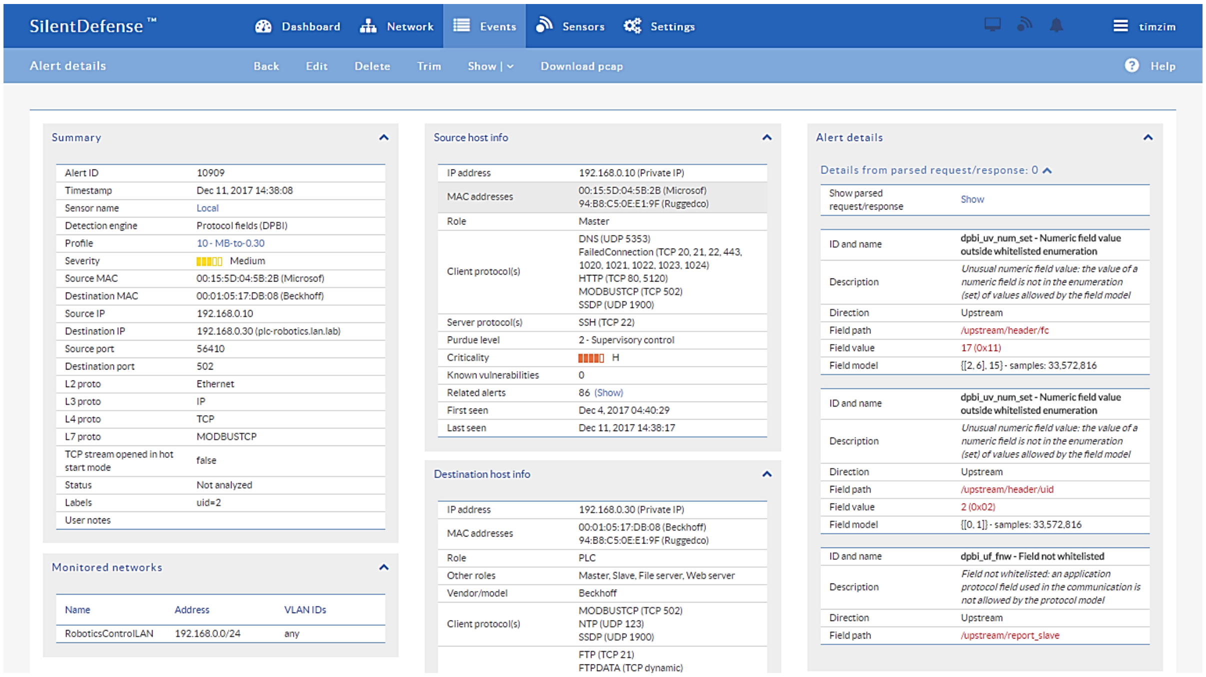


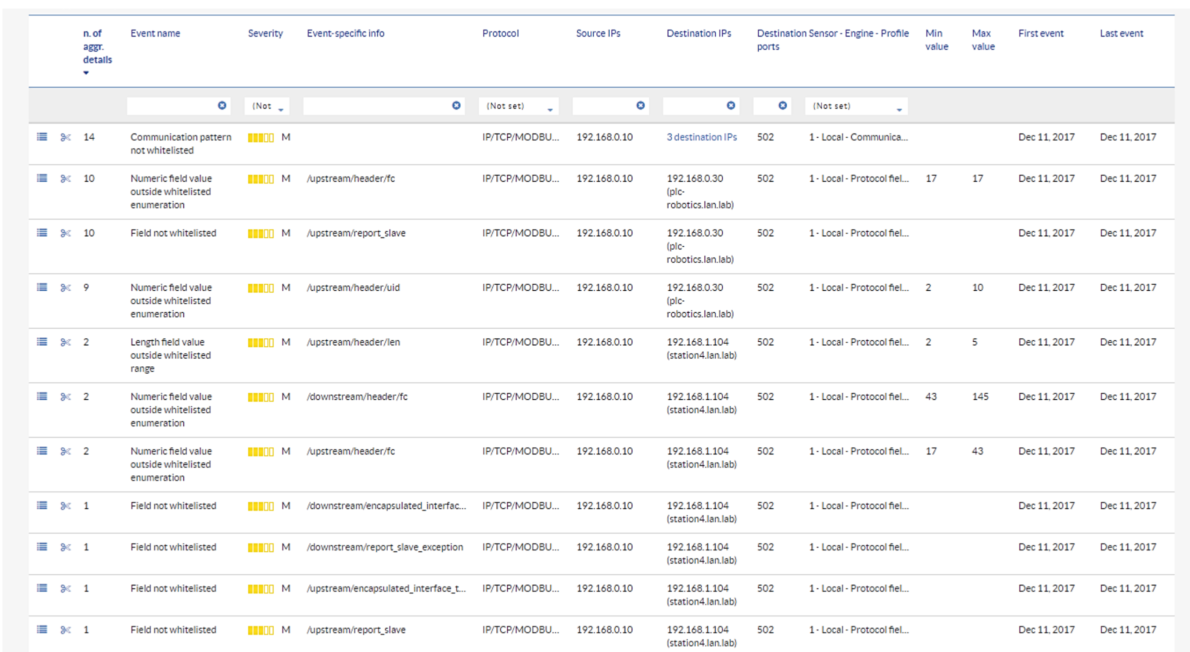


A.3.15. ICS Device Scanning Is Performed on the Network

During the reconnaissance phase, an attacker may attempt to locate vulnerable devices in an ICS network and will likely probe for ICS-specific services (e.g., Modbus TCP). Once a vulnerable service is discovered, an attacker may attempt to exploit that service.

This anomaly was executed on the CRS. The software Nmap [16] was used to generate the Modbus device scan by using the script `modbus-discover` [18]. The attack was directed at two ICS devices: the PLC and Machining Station 4.


SilentDefense™
Dashboard
Network
Events
Sensors
Settings
timzim
Alert details
Back
Edit
Delete
Trim
Show |
Download pcap
Help
Summary
Alert ID 10909
Timestamp Dec 11, 2017 14:38:08
Sensor name Local
Detection engine Protocol fields (DPBI)
Profile 10 - MB-to-0.30
Severity Medium
Source MAC 00:15:5D:04:5B:2B (Microsof)
Destination MAC 00:01:05:17:DB:08 (Beckhoff)
Source IP 192.168.0.10
Destination IP 192.168.0.30 (plc-robotics.lan.lab)
Source port 56410
Destination port 502
L2 proto Ethernet
L3 proto IP
L4 proto TCP
L7 proto MODBUSTCP
TCP stream opened in hot start mode false
Status Not analyzed
Labels uid=2
User notes
Monitored networks
Name Address VLAN IDs
RoboticsControlLAN 192.168.0.0/24 any
Source host info
IP address 192.168.0.10 (Private IP)
MAC addresses 00:15:5D:04:5B:2B (Microsof) 94:B8:C5:0E:E1:9F (Ruggedco)
Role Master
Client protocol(s) DNS (UDP 5353) FailedConnection (TCP 20, 21, 22, 443, 1020, 1021, 1022, 1023, 1024) HTTP (TCP 80, 5120) MODBUSTCP (TCP 502) SSDP (UDP 1900)
Server protocol(s) SSH (TCP 22)
Purdue level 2 - Supervisory control
Criticality H
Known vulnerabilities 0
Related alerts 86 (Show)
First seen Dec 4, 2017 04:40:29
Last seen Dec 11, 2017 14:38:17
Destination host info
IP address 192.168.0.30 (Private IP)
MAC addresses 00:01:05:17:DB:08 (Beckhoff) 94:B8:C5:0E:E1:9F (Ruggedco)
Role PLC
Other roles Master, Slave, File server, Web server
Vendor/model Beckhoff
Client protocol(s) MODBUSTCP (TCP 502) NTP (UDP 123) SSDP (UDP 1900)
FTP (TCP 21) FTPDATA (TCP dynamic)
Alert details
Details from parsed request/response: 0
Show parsed request/response Show
ID and name dpbi_uv_num_set - Numeric field value outside whitelisted enumeration
Description Unusual numeric field value: the value of a numeric field is not in the enumeration (set) of values allowed by the field model
Direction Upstream
Field path /upstream/header/fc
Field value 17 (0x11)
Field model [[2, 6], 15] - samples: 33,572,816
ID and name dpbi_uv_num_set - Numeric field value outside whitelisted enumeration
Description Unusual numeric field value: the value of a numeric field is not in the enumeration (set) of values allowed by the field model
Direction Upstream
Field path /upstream/header/uid
Field value 2 (0x02)
Field model [[0, 1]] - samples: 33,572,816
ID and name dpbi_uf_fnw - Field not whitelisted
Description Field not whitelisted: an application protocol field used in the communication is not allowed by the protocol model
Direction Upstream
Field path /upstream/report_slave


n. of aggr. details
Event name
Severity
Event-specific info
Protocol
Source IPs
Destination IPs
Destination ports
Sensor - Engine - Profile
Min value
Max value
First event
Last event
14 Communication pattern not whitelisted M IP/TCP/MODBU... 192.168.0.10 3 destination IPs 502 1 - Local - Communica... Dec 11, 2017 Dec 11, 2017
10 Numeric field value outside whitelisted enumeration M /upstream/header/fc IP/TCP/MODBU... 192.168.0.10 192.168.0.30 (plc-robotics.lan.lab) 502 1 - Local - Protocol fiel... 17 17 Dec 11, 2017 Dec 11, 2017
10 Field not whitelisted M /upstream/report_slave IP/TCP/MODBU... 192.168.0.10 192.168.0.30 (plc-robotics.lan.lab) 502 1 - Local - Protocol fiel... Dec 11, 2017 Dec 11, 2017
9 Numeric field value outside whitelisted enumeration M /upstream/header/uid IP/TCP/MODBU... 192.168.0.10 192.168.0.30 (plc-robotics.lan.lab) 502 1 - Local - Protocol fiel... 2 10 Dec 11, 2017 Dec 11, 2017
2 Length field value outside whitelisted range M /upstream/header/len IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... 2 5 Dec 11, 2017 Dec 11, 2017
2 Numeric field value outside whitelisted enumeration M /downstream/header/fc IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... 43 145 Dec 11, 2017 Dec 11, 2017
2 Numeric field value outside whitelisted enumeration M /upstream/header/fc IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... 17 43 Dec 11, 2017 Dec 11, 2017
1 Field not whitelisted M /downstream/encapsulated_interfac... IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... Dec 11, 2017 Dec 11, 2017
1 Field not whitelisted M /downstream/report_slave_exception IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... Dec 11, 2017 Dec 11, 2017
1 Field not whitelisted M /upstream/encapsulated_interface_t... IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... Dec 11, 2017 Dec 11, 2017
1 Field not whitelisted M /upstream/report_slave IP/TCP/MODBU... 192.168.0.10 192.168.1.104 (station4.lan.lab) 502 1 - Local - Protocol fiel... Dec 11, 2017 Dec 11, 2017

Appendix B. Secure-NOK SNOK Supplemental Information

Secure-NOK SNOK is a cybersecurity monitoring and detection system tailored for industrial networks and control systems. In the installation, the SNOK network intrusion detection system (IDS) comes preinstalled on an appliance that is integrated into the asset owner's environment.

B.1. Build Architecture

Two SNOK dedicated appliances were physically installed in the measurement rack of the Cybersecurity for Smart Manufacturing Systems (CSMS) environment. One appliance was dedicated to the process control system (PCS), and the other appliance was dedicated to the collaborative robotic system (CRS). Three existing Switch Port Analyzer (SPAN) ports from each system (PCS and CRS) were connected to a VERSAstream packet broker (VS-1208BT-S) to aggregate the mirrored traffic from the PCS and the CRS into two respective streams, for a total of six SPAN ports. The appliance connections within the PCS and CRS networks are shown in Figure B-1 and Figure B-2, respectively.

The PCS appliance network was connected to the demilitarized zone (DMZ) network located in the test bed's measurement rack, to isolate the appliance's network traffic from the rest of the network traffic. The engineering laptop was used to interface with the SNOK graphical user interface (GUI) via physical connections to the DMZ. The CRS appliance network was connected to the industrial control system (ICS) local area network (LAN), and the SNOK GUI was accessed via the engineering workstation. More information regarding the specific configuration of the test-bed network can be found in Section 3.

Figure B-1 SPAN Port Connections to the SNOK Appliance in the PCS (Including the Hosts with SNOK Agents)

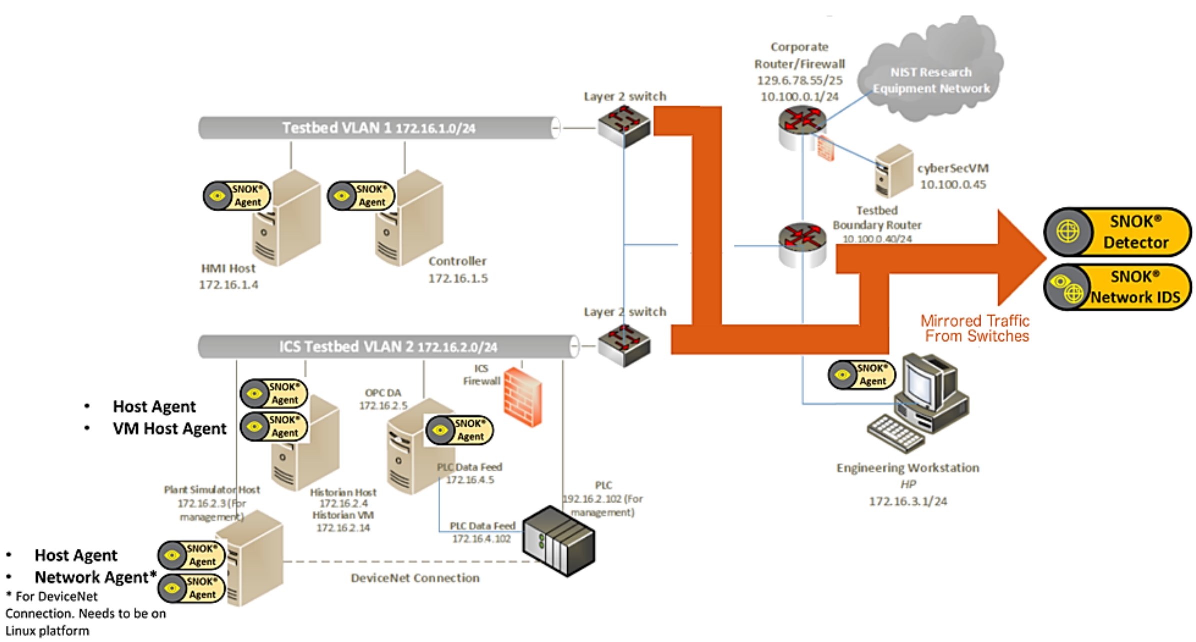
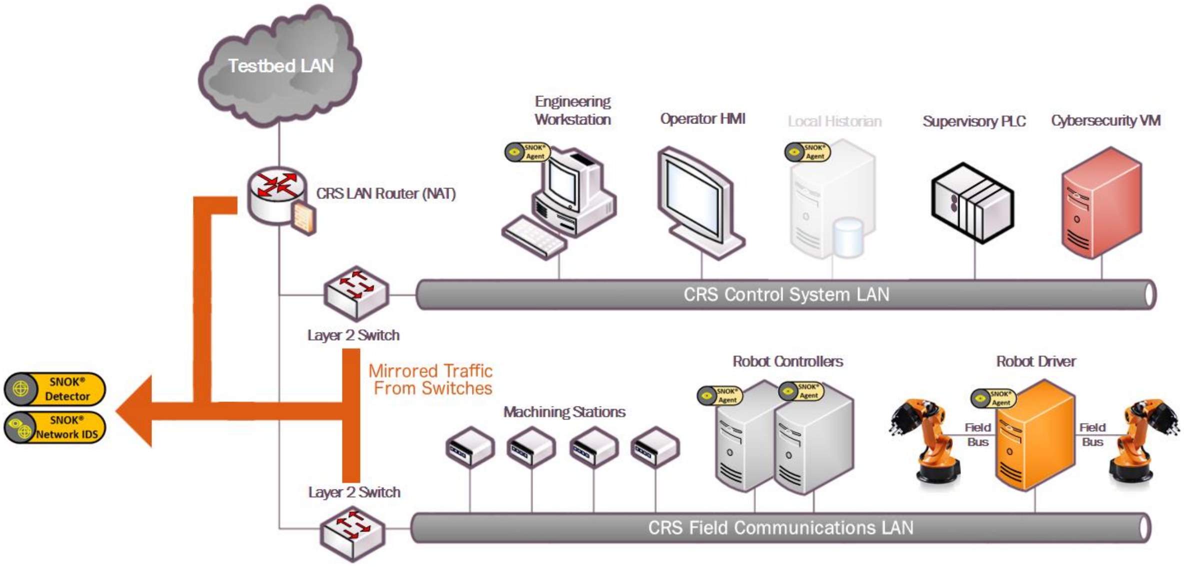


Figure B-2 SPAN Port Connections to the SNOK Appliance in the CRS (Including the Hosts with SNOK Agents)



B.2. Installation and Configuration

Physical hardware appliances and software were provided by Secure-NOK for this demonstration. After the hardware appliances were received, they were installed into the CSMS test bed. Soon after the initial installation, engineers from Secure-NOK arrived on site to complete the installation and configuration of the tool. The following subsections describe the steps taken to install and configure the appliances.

B.2.1. Hardware

The hardware used included two Siemens SIMATIC industrial personal computers (IPCs) executing the SNOK services: a SIMATIC IPC227E for the PCS and a SIMATIC IPC427E for the CRS. A VERSAstream packet broker (VS-1208BT-S) was used to aggregate the mirrored traffic from the PCS and the CRS into two respective streams, one for each IPC.

B.2.2. Windows XP / Windows 7 / Windows Server 2012 Installation

The steps in this section describe the installation of SNOK Agents on endpoints with Microsoft Windows operating systems (OSs).

1. Launch *SNOKAgentSetup.exe* from the Windows Agent folder in the installation pack.
2. Click **Next>**.
3. Select both components, and then click **Next>**.
4. Input the username and password for administrative privileges, and then click **Install**.

5. Modify the configuration file located at *<installation directory>\SNOK-agent\bin\snokagentconfig.txt* to include the following information:
    a. **idAgent**: a unique identifier (ID) that will not be used by any other agent that reports to the same SNOK Detector
    b. **detectorIP**: the Internet Protocol (IP) address of the SNOK Detector to which the agent will report
    c. **licenseKey**: the license key provided for the SNOK Detector

B.2.2.1. Start SNOK Agent Manually

If the installation did not include selecting **Automatically start agent**, then follow the steps below to manually start the agent:

1. Open the command prompt.
2. Change the directory to <installation directory>\bin\ by using the following command:

```
> cd C:\SNOK\bin\
```

3. Run the agent by using the following command and then pressing the **Enter** key:

```
> SNOKAgent.exe
```

B.2.2.2. Stop SNOK Agent Manually

1. Open the **Task Manager**.
2. Open the **Processes** tab.
3. Select the process name **SNOKAgent.exe**.
4. Click the **End Task** button.

B.2.3. Ubuntu 12 / Ubuntu 14 Installation

1. Copy the file *snoknetmonagent_<version>.deb* into the */home* directory of the IPC.
2. Add the Debian Wheezy universe to the apt sources file by using the following command:

```
> sudo echo “deb http://httpredir.debian.org/debian wheezy
main” >> /etc/apt/sources.list
```

3. Install the libpcap-dev package by using the following command:

```
> sudo apt-get install libpcap-dev
```

4. Install the SNOK Agent from the Debian software package file by using the following command:

```
> sudo dpkg -I ~/snoknetmonagent_<version>.deb
```

5. Modify the configuration file *snok-netmonconfig.txt* located in the directory */etc/default/* to include the following information:

    a. **idAgent**: a unique ID that will not be used by any other agent that reports to the same SNOK Detector

    b. **detectorIP**: the IP address of the SNOK Detector to which the agent will report

    c. **licenseKey**: the license key provided for the SNOK Detector

B.2.4. SNOK Detector Configuration

The SNOK Detector comes installed as part of a preconfigured appliance, requiring final configuration before integration into the asset owner's environment. The following configuration must be completed on the appliance before installation:

1. Obtain a license key from Secure-NOK. The media access control (MAC) address of the network interface is needed to generate the license. On the appliance, execute the following command to obtain the address, which will be the hexadecimal number after `HWaddr`, in the format of `xx:xx:xx:xx:xx:xx`:

```
sudo ifconfig eth0
```

2. Copy the license key file *snoklicense.key* to the directory */home/snok/*.

3. Start the configuration software by using the following command:

```
> sudo /usr/share/snok/snok-config.sh
```

4. Ensure that *1 (VMs Installation) SNOK Detector with local Visualizer* is highlighted, and then press the **Enter** key.

5. On the Database VM IP page, enter the IP address of the preconfigured appliance, and then press the **Enter** key.

6. On the Detector mode page, ensure the messages received are <u>not</u> forwarded (isolated) and is selected, and then press the **Enter** key.

7. On the Date-Time Synchronization page, select **1 Enter IP for Simple Network Time Protocol Server** (Network Time Protocol [NTP] / Simple Network Time Protocol [SNTP] server available), and then press the **Enter** key.

8. On the NTP/SNTP IP page, type the IP address of the NTP server, and then press the **Enter** key. The lab NTP server (10.100.0.15) was used for the build environment.

9. On the External Event Reporting page, select any reporting methods, and then press the **Enter** key. The build configuration did not require any external reporting (e.g., syslog, email), so the *option 3 Go to next step* was selected.

10. When prompted, enter the database password to enable the automated configuration.

11. Start the snok-box service by using the following command:

```
> sudo service snok-box start
```

12. Start the snok-dumper service by using the following command:

```
> sudo service snok-dumper start
```

B.3. Anomaly Scenarios

The agent-based anomaly detection method was demonstrated for the scenarios detailed in the following subsections. Each scenario includes a description of the anomaly, a detailed description of how each demonstration event was conducted in the CSMS environment, and the observed results.

For the sake of brevity, only a subset of the alerts observed during the demonstration is shown. However, each anomaly scenario includes a screenshot of the alerts summary observed after the anomaly scenario had completed.

B.3.1. Web Browser Is Used to Access the Internet

The detection of unauthorized internet traffic on ICS networks is important for mitigating risk to the manufacturing system. Internet-accessible network connections introduce a gateway for malware into the ICS network, as well as a gateway for sensitive manufacturing system data to be exfiltrated out of the ICS network.

This anomaly was executed on the CRS. A Hypertext Transfer Protocol (HTTP) server was installed and configured on a server with an internally routed public IP address (129.6.1.2). The Firefox web browser was used to connect to a web page, from the engineering workstation to the internet-based HTTP server.

Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: firefox-esr

NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/16/2018 13:12:02	Unexpected new connection	A new IP address has been detected in the network	Source IP: 129.6.1.2 Destination IP: 192.168.0.20 Source MAC Address: 94:b8:c5:0e:e1:9f Destination MAC Address: f8:b1:56:ba:09:a8

129.6.1.2 94:b8:c5:0e:e1:9f	192.168.0.20 f8:b1:56:ba:09:a8	HTTP	02/16/2018 13:11:59	02/16/2018 13:13:01	0.15	0.98

B.3.2. Data Exfiltration to the Internet via HTTP

Attacks against ICS, with the goal of information gathering, must (at some point) attempt to exfiltrate sensitive or proprietary data from the ICS network, potentially utilizing the internet as a transport mechanism. Monitoring for ICS devices communicating to other devices over the internet can help detect data exfiltration events, especially if the affected device does not normally communicate over the internet.

This anomaly was executed on the CRS. An HTTP server and the PHP (Hypertext Preprocessor) server-side scripting language [19] were installed and configured on a server with an internally routed public IP address (129.6.1.2). A PHP web page was created to enable file uploads over HTTP. The web page was accessed by the Firefox web browser on the engineering workstation, and the sensitive file *ControlsSchematic.dwg*, an AutoCAD drawing file, was selected and uploaded to the server.

NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/16/2018 13:07:52	Unexpected new connection	A new IP address has been detected in the network	Source IP: 129.6.1.2 Destination IP: 192.168.0.20 Source MAC Address: 94:b8:c5:0e:e1:9f Destination MAC Address: f8:b1:56:ba:09:a8	Authorized by: **admin** Timestamp: 02/16/2018 13:12:00 Message:

NIST EL Collaborative Robots Sys All segments Engineering WS	02/16/2018 13:10:23	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: wget	Authorize

Source IP Source MAC	Destination Destination MAC	Protocol Type	Start Timestamp	End Timestamp	Packets/second	kBits/second
192.168.0.20 f8:b1:56:ba:09:a8	129.6.1.2 94:b8:c5:0e:e1:9f	HTTP	02/16/2018 13:23:27	02/16/2018 13:24:30	0.17	0.08

B.3.3. European Institute for Computer Antivirus Research Virus Test File Is Detected on Host

Computer viruses and malware are serious threats to the ICS. They can undermine the ICS security, confidentiality, and stability, and can even sabotage the ICS. Providing the ability to detect viruses and malware in the ICS network is important.

This anomaly was executed on the PCS. Before the CyberX platform tool was installed, a European Institute for Computer Antivirus Research (EICAR) test file was created and stored on the engineering workstation.

Host	Timestamp	Type	Description
NIST EL Process Control All Segments Engineering WS	02/21/2018 12:47:00	Security Policy Violation	Security Policy Violation : Anti-Virus Event : Anti-Virus Disabled Network Segment Security Policy Violation

B.3.4. Host Scanning Is Performed on the Network

During the reconnaissance phase, an attacker may attempt to locate vulnerable devices on an ICS network. A host scan is one method to discover hosts or devices in the network. Once a host or device is discovered and identified, an attacker may attempt to exploit the host or device.

This anomaly was executed on the PCS. The software Nmap [16] was used to perform a host discovery scan of the ICS network on the subnet `172.16.2.0/24`. The scan originated from the cybersecurity virtual machine (CybersecVM), logically located in the test-bed LAN.

NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.80 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorized by: **admin** Timestamp: 02/21/2018 13:06:25 Message:
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.82 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorized by: **admin** Timestamp: 02/21/2018 13:06:15 Message:
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.84 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorized by: **admin** Timestamp: 02/21/2018 13:06:21 Message:
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.97 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.91 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.86 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.93 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorized by: **admin** Timestamp: 02/21/2018 13:06:18 Message:
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.88 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.95 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.9 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.99 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.85 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.8 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.81 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorized by: **admin** Timestamp: 02/21/2018 13:06:12 Message:
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.83 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.90 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.98 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.92 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:05:15	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.2.87 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize

B.3.5. Port Scanning Is Performed on the Network

During the reconnaissance phase, an attacker may attempt to locate vulnerable services in an ICS network, likely probing for any open network ports to determine if a specific network service is available (e.g., Modbus). Once a vulnerable service is discovered, an attacker may attempt to exploit that service.

This anomaly was executed on the CRS. The software Nmap [16] was used to perform a network scan for devices with the Modbus service enabled (Port 502). The scan originated from the CybersecVM, logically hosted on the historian located in the test-bed LAN.

NIST EL Collaborative Robots Sys All segments Historian	02/20/2018 13:43:42	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: nmap	Authorize

NetMonAgent	Timestamp	Type	Description	Connection Details	Authorization
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 13:44:29	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.0.10 Destination IP: 192.168.0.30 Source MAC Address: 00:15:5d:02:0a:0e Destination MAC Address: 00:01:05:17:db:08	Authorize
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 13:44:29	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.0.10 Destination IP: 192.168.0.60 Source MAC Address: 00:15:5d:02:0a:0e Destination MAC Address: 00:30:de:00:c4:3c	Authorize
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 13:44:29	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.0.10 Destination IP: 192.168.1.104 Source MAC Address: 00:15:5d:02:0a:0e Destination MAC Address: 94:b8:c5:0e:e1:9f	Authorize
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 13:44:29	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.0.10 Destination IP: 192.168.1.104 Source MAC Address: 94:b8:c5:0e:e1:9f Destination MAC Address: b0:d5:cc:f4:26:ec	Authorize

NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:33:00	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.1.5 Source MAC Address: e4:90:69:3b:c2:c4 Destination MAC Address: 0c:c4:7a:31:3e:d7	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:33:00	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.1.4 Source MAC Address: e4:90:69:3b:c2:c4 Destination MAC Address: 0c:c4:7a:31:44:47	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:33:00	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.1.4 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize
NIST EL Process Control All Segments PCS NetMon	02/21/2018 13:33:00	Unexpected new connection	A new IP address has been detected in the network	Source IP: 10.100.0.28 Destination IP: 172.16.1.5 Source MAC Address: 00:15:5d:02:0a:08 Destination MAC Address: e4:90:69:3b:c2:c1	Authorize

Source IP Source MAC	Destination Destination MAC	Protocol Type	Start Timestamp	End Timestamp	Packets/second	kBits/second
192.168.0.10 00:15:5d:02:0a:0e	192.168.0.30 00:01:05:17:db:08	Modbus/TCP	02/20/2018 13:43:24	02/20/2018 13:44:27	0.49	0.00
192.168.0.10 00:15:5d:02:0a:0e	192.168.0.60 00:30:de:00:c4:3c	Modbus/TCP	02/20/2018 13:43:24	02/20/2018 13:44:27	0.15	0.00
192.168.0.10 00:15:5d:02:0a:0e	192.168.1.104 94:b8:c5:0e:e1:9f	Modbus/TCP	02/20/2018 13:43:24	02/20/2018 13:44:27	0.76	0.01
192.168.0.10 94:b8:c5:0e:e1:9f	192.168.1.104 b0:d5:cc:f4:26:ec	Modbus/TCP	02/20/2018 13:43:24	02/20/2018 13:44:27	0.38	0.00

B.3.6. Unauthorized Installation of Software

Many Linux distributions provide an automated method to download and install packages. Often, these packages originate from third parties and may not be validated against the ICS environments. Attackers may install unvalidated, or even malicious, packages to the ICS. The ability to detect unauthorized downloads and unauthorized installations of software is important.

This anomaly was executed on the CRS. The Advanced Package Tool (apt-get) was used to install a small package with minimal dependencies (md5deep). The installation was performed on the engineering workstation via the command line.

Host	Timestamp	Type	Description	Authorization
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:35	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: python	Authorize
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:33	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: //bin/dbus-daemon	Authorize
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:05	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: [dpkg]	Authorize

NetMonAgent	Timestamp	Type	Description	Connection Details	Authorization
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 11:13:12	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.1.5 Destination IP: 91.189.94.25 Source MAC Address: a0:ce:c8:1f:bd:99 Destination MAC Address: 94:b8:c5:0e:e1:9f	Authorize
NIST EL Collaborative Robots Sys All segments NetworkAgent 1	02/20/2018 11:12:09	Unexpected new connection	A new IP address has been detected in the network	Source IP: 192.168.1.5 Destination IP: 91.189.91.23 Source MAC Address: a0:ce:c8:1f:bd:99 Destination MAC Address: 94:b8:c5:0e:e1:9f	Authorize

Source IP Source MAC	Destination Destination MAC	Protocol Type	Start Timestamp	End Timestamp	Packets/second	kBits/second
192.168.1.5 a0:ce:c8:1f:bd:99	91.189.94.25 94:b8:c5:0e:e1:9f	HTTP	02/20/2018 11:12:05	02/20/2018 11:13:08	0.03	0.00

B.3.7. Unauthorized Programmable Logic Controller Firmware Update

Many ICS devices provide services to remotely update firmware over the network. These network services can also provide a mechanism for attackers to replace valid firmware with malicious firmware if the device is not protected.

This anomaly was executed on the PCS. The Allen-Bradley programmable logic controller (PLC) implemented in the PCS contains an Ethernet module (1756-EN2T) that allows its firmware to be upgraded and downgraded over Ethernet/IP. The firmware was upgraded or downgraded using the ControlFLASH firmware upgrade tool.

Host	Timestamp	Type	Description	Authorization
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:35	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: python	Authorize
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:33	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: //bin/dbus-daemon	Authorize
NIST EL Collaborative Robots Sys All segments Robotic Driver	02/20/2018 11:12:05	Installed Base Mismatch	Installed Base Mismatch : Unknown Process detected Process Name: [dpkg]	Authorize

Source IP Source MAC	Destination Destination MAC	Protocol Type	Start Timestamp	End Timestamp	Packets/second	kBits/second
192.168.1.5 a0:ce:c8:1f:bd:99	91.189.94.25 94:b8:c5:0e:e1:9f	HTTP	02/20/2018 11:12:05	02/20/2018 11:13:08	0.03	0.00

B.3.8. Unauthorized PLC Logic Download

Many PLCs enable remote access for uploading and downloading control logic to and from the controller. This service provides great convenience, but also provides a mechanism for attackers to remotely access the control logic and proprietary manufacturing information if the PLC is not protected.

This anomaly was executed on the PCS. The Allen-Bradley software Studio 5000 was used to download the logic from the PCS PLC to the engineering workstation. Physical access to the PLC was required in order to change the operation mode from RUN to REMOTE RUN.

NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:24:41	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is high in both number of packets and traffic bandwidth usage	Source IP: 172.16.2.102 Destination IP: 172.16.3.10 Source MAC Address: 00:1d:9c:c9:6d:42 Destination MAC Address: e4:90:69:3b:c2:c5 Protocol: CIP [41(kbps) > 0(kbps)] [157(pps) > 0(pps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:24:41	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is low in traffic bandwidth usage	Source IP: 172.16.2.4 Destination IP: 172.16.3.10 Source MAC Address: 0c:c4:7a:31:44:bd Destination MAC Address: e4:90:69:3b:c2:c5 Protocol: TCP [0(kbps) < 1(kbps)]	Authorize

B.3.9. Unauthorized PLC Logic Modification

As previously mentioned, many PLCs enable remote access for uploading and downloading control logic to and from the controller. This service provides great convenience, but also provides a mechanism for attackers to replace valid control logic with malicious logic if the device is not protected.

This anomaly was executed on the PCS. The Allen-Bradley software Studio 5000 was used to upload new logic from the engineering workstation to the PCS PLC. Physical access to the PLC was required in order to change the operation mode from RUN to REMOTE RUN.

NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:27:49	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is high in both number of packets and traffic bandwidth usage	Source IP: 172.16.3.10 Destination IP: 172.16.2.102 Source MAC Address: 40:a8:f0:3d:48:ae Destination MAC Address: e4:90:69:3b:c2:c0 Protocol: CIP [50(kbps) > 0(kbps)] [19(pps) > 0(pps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:27:49	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is high in both number of packets and traffic bandwidth usage	Source IP: 172.16.3.10 Destination IP: 172.16.2.102 Source MAC Address: e4:90:69:3b:c2:c5 Destination MAC Address: 00:1d:9c:c9:6d:42 Protocol: CIP [99(kbps) > 0(kbps)] [37(pps) > 0(pps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:27:49	Abnormal communication pattern between IP addresses	The communication between two IP addresses is high in terms of both packet and traffic bandwidth usage	Source IP: 172.16.2.102 Destination IP: 172.16.3.10 Source MAC Address: 00:1d:9c:c9:6d:42 Destination MAC Address: e4:90:69:3b:c2:c5 [37(kbps) > 0(kbps)] [125(pps) > 0(pps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:27:49	Abnormal communication pattern between IP addresses	The communication between two IP addresses is high in terms of both packet and traffic bandwidth usage	Source IP: 172.16.3.10 Destination IP: 172.16.2.102 Source MAC Address: 40:a8:f0:3d:48:ae Destination MAC Address: e4:90:69:3b:c2:c0 [50(kbps) > 0(kbps)] [19(pps) > 0(pps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:27:49	Abnormal communication pattern between IP addresses	The communication between two IP addresses is high in terms of both packet and traffic bandwidth usage	Source IP: 172.16.3.10 Destination IP: 172.16.2.102 Source MAC Address: e4:90:69:3b:c2:c5 Destination MAC Address: 00:1d:9c:c9:6d:42 [99(kbps) > 0(kbps)] [37(pps) > 0(pps)]	Authorize

B.3.10. Unauthorized Connection Is Established Between ICS Devices

An unauthorized connection between two ICS devices may indicate anomalous activity and is important to discover, especially when the devices do not normally communicate.

The anomaly was executed on the PCS. An unauthorized remote desktop session was initialized from the human-machine interface (HMI) server to the object linking and embedding for process control (OPC) server. Valid credentials were used to complete the connection.

NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:45:34	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is high in traffic bandwidth usage	Source IP: 172.16.2.5 Destination IP: 172.16.1.4 Source MAC Address: 0c:c4:7a:32:b3:01 Destination MAC Address: e4:90:69:3b:c2:c5 Protocol: TCP [116(kbps) > 109(kbps)]	Authorize
NIST EL Process Control All Segments PCS NetMon	02/22/2018 16:45:34	Abnormal communication pattern on a specific protocol between IP addresses	The communication between two IP addresses on a specific protocol is high in traffic bandwidth usage	Source IP: 172.16.2.5 Destination IP: 172.16.1.4 Source MAC Address: e4:90:69:3b:c2:c4 Destination MAC Address: 0c:c4:7a:31:44:47 Protocol: TCP [77(kbps) > 73(kbps)]	Authorize

B.3.11. Host-Based Firewall Is Disabled

The host-based firewall is an important part of the overall network security strategy. Attackers may attempt to disable the firewall to gain access to the host. Any change in the operating state of the host-based firewall may indicate malicious activity.

This anomaly was executed on the PCS. The engineering workstation utilized the Microsoft Windows 7 OS, which included the Windows Firewall component. The Windows Firewall was manually disabled and enabled to generate the anomaly.

All Segments	Engineering WS	02/23/2018 15:41:47	Windows firewall status	Windows firewall enabled
All Segments	Engineering WS	02/23/2018 15:41:45	Windows firewall status	Windows firewall enabled
All Segments	Engineering WS	02/23/2018 15:41:45	Security Policy Violation	Network Segment Security Policy Violation

B.3.12. Host-Based Anti-Virus Software Is Disabled

The anti-virus software is an important part of the overall ICS security strategy. Attackers may attempt to disable the anti-virus software to download malwares to the host. Any change in the operating state of the anti-virus software may indicate malicious activity.

This anomaly was executed on the PCS. Symantec Endpoint Protection anti-virus software was installed and operational on the engineering workstation. The software was manually disabled and enabled to generate the anomaly.

All Segments	Engineering WS	02/23/2018 15:43:07	Antivirus status	AntiVirus protection disabled
All Segments	Engineering WS	02/23/2018 15:43:07	Security Policy Violation	Network Segment Security Policy Violation

B.3.13. Host Central Processing Unit Load Is Increased

Most hosts in the ICS environment are running a predefined set of tasks or schedules. The system load of each host usually closely follows a routine or pattern. Any change or deviation from the routine could indicate malicious activity or abnormal or fault behavior of the ICS.

This anomaly was executed on the PCS. The software Prime95 [20] was installed on the engineering workstation to generate the anomaly. The Prime95 torture test option “Blend” was used to execute a search for large prime numbers, resulting in a central processing unit utilization increase that was continuously greater than 95 percent.

NetworkSegment	Host	Timestamp (Detector)	Type	Description
Any All Segments	Any All Segments::HMI Host All Segments::Controller All Segments::Historian VM All Segments::OPC DA Server All Segments::Engineering WS	From To	Any Agent started Detector started USB event CPU load	Free text search
All Segments	Engineering WS	02/27/2018 11:41:16	CPU load	CPU usage normal CPU Load: 32%
All Segments	Engineering WS	02/27/2018 11:38:44	CPU load	CPU usage increased CPU Load: 99%
All Segments	Engineering WS	02/27/2018 11:27:39	CPU load	CPU usage normal CPU Load: 31%

B.3.14. Unauthorized Detachment of Keyboard to Host

While access to unused Universal Serial Bus (USB) ports can be denied through numerous physical means, the potential may still exist for an attacker to simply remove an attached USB device to gain access to a USB port. Detection of the disconnection of an input device may indicate malicious activity.

This anomaly was executed on the PCS. A USB keyboard attached to the engineering workstation was temporarily disconnected from the USB port.

NIST EL Process Control All Segments Engineering WS	02/27/2018 11:47:18	Security Policy Violation	Security Policy Violation : USB Event : Device Inserted Site Security Policy Violation (Device class: Device)	Authorize
NIST EL Process Control All Segments Engineering WS	02/27/2018 11:47:10	Security Policy Violation	Security Policy Violation : USB Event : Device Removed Site Security Policy Violation (Device class: Device)	Authorize

B.3.15. Unauthorized Insertion of USB Storage Device

Portable USB storage devices could be a threat to the ICS. An unauthorized USB device may contain malware. Once inserted into a host, the malware can potentially gain control of the host and infect other hosts in the ICS network.

This anomaly was executed on the PCS. A USB storage device (flash drive) was temporarily connected to the engineering workstation.

Host	Timestamp	Type	Description	Authorization
NIST EL Process Control All Segments Engineering WS	02/27/2018 11:19:01	Security Policy Violation	Security Policy Violation : USB Event : Device Inserted Site Security Policy Violation (Device class: Device)	Authorize
NIST EL Process Control All Segments Engineering WS	02/27/2018 11:18:56	Security Policy Violation	Security Policy Violation : USB Event : Device Removed Site Security Policy Violation (Device class: Device)	Authorize

Appendix C. CyberX Supplemental Information

The CyberX platform delivers continuous operational technology (OT) threat monitoring and asset discovery, combining a deep understanding of industrial protocols, devices, and applications with OT-specific behavioral analytics, threat intelligence, risk and vulnerability management, and automated threat modeling. The platform is delivered as a preconfigured appliance, including the Internet Protocol (IP) address, subnet mask, default gateway, and Domain Name System (DNS) servers utilized in the build environment.

C.1. Build Architecture

The CyberX appliance was physically installed in the measurement rack of the Cybersecurity for Smart Manufacturing Systems (CSMS) environment. Three existing Switch Port Analyzer (SPAN) ports from each system (collaborative robotic system [CRS] and process control system [PCS]) were connected to dedicated network interfaces on the appliance, for a total of six SPAN ports. The SPAN port connections to the appliance, within the PCS and CRS networks, are shown in Figure C-1 and Figure C-2, respectively.

Enterprises typically deploy multiple CyberX appliances across various geographically distributed sites, along with a central manager that is used to aggregate asset, vulnerability, and threat information from each CyberX appliance, and to manage software updates and configurations for each individual appliance.

The appliance network was connected to the demilitarized zone (DMZ) network located in the test bed’s measurement rack, to isolate the appliance’s network traffic from the rest of the network traffic. Engineering laptops were used to interface with the CyberX console graphical user interface (GUI) via physical connections to the DMZ. More information regarding the specific configuration of the test-bed network can be found in Section 3.

Figure C-1 SPAN Port Connections to the CyberX Appliance in the PCS

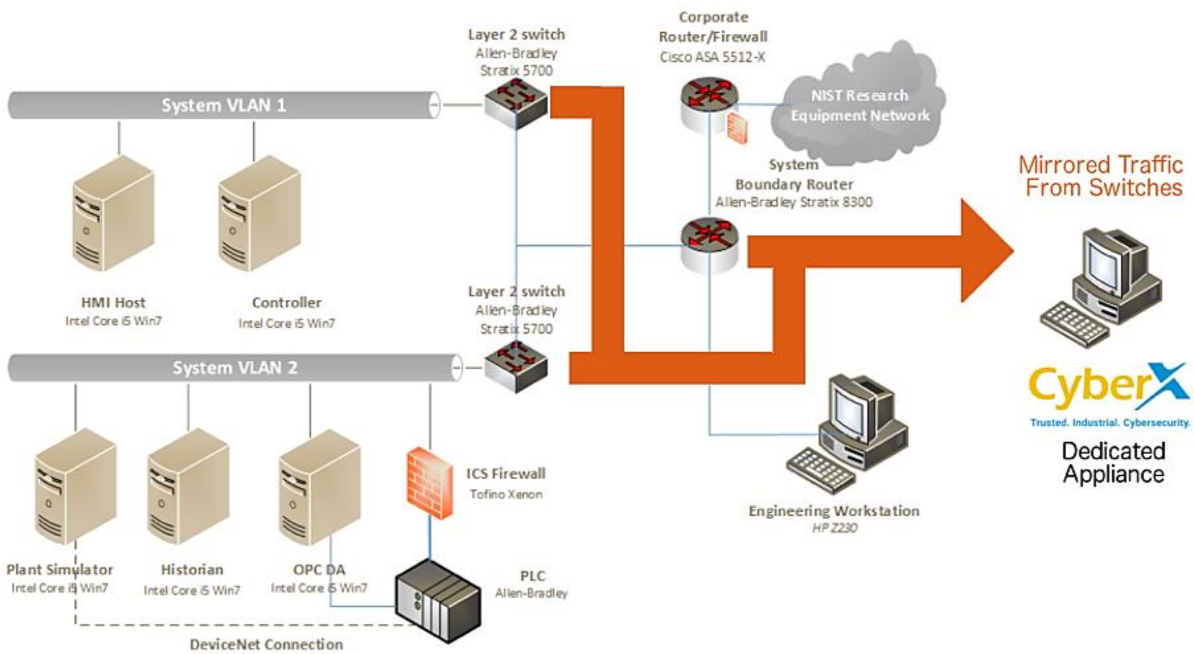
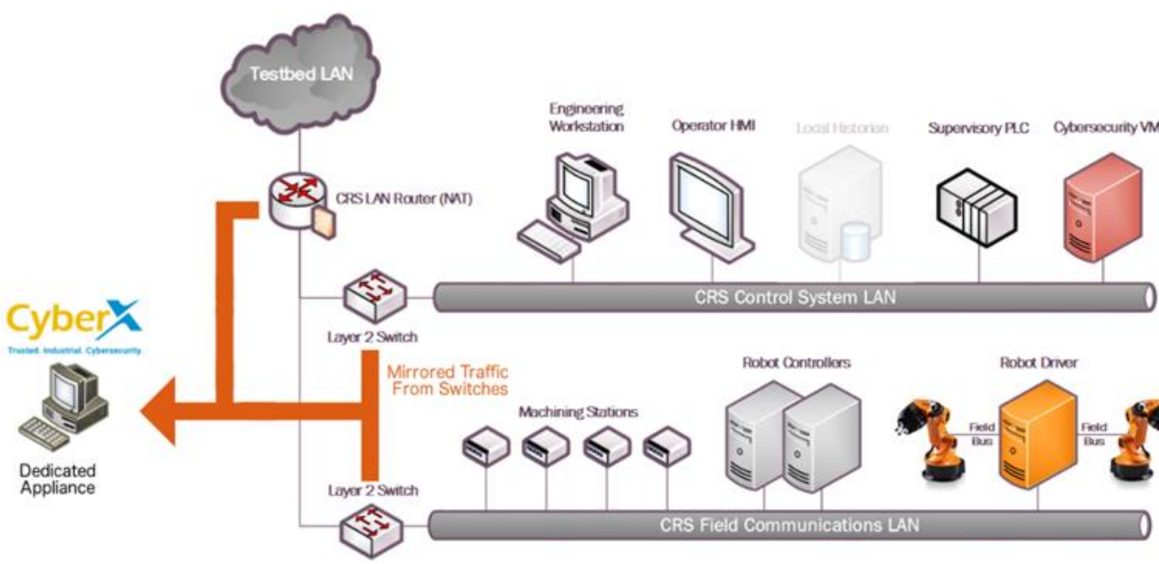


Figure C-2 SPAN Port Connections to the CyberX Appliance in the CRS



C.2. Installation and Configuration

Physical hardware and software were provided by CyberX for this demonstration. After the hardware appliance was received, it was installed into the CSMS test bed. Soon after the initial installation, engineers from CyberX arrived on site to complete the installation and configuration of the product. The following subsections describe the steps taken to install and configure the appliance.

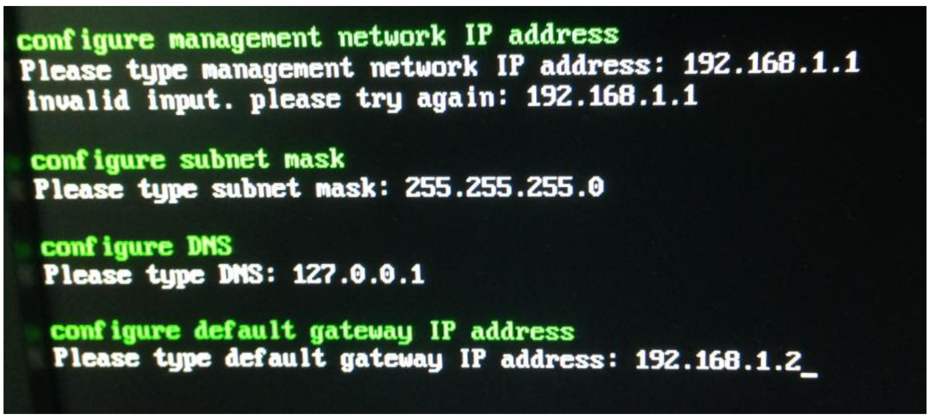
C.2.1. Configuration Guide

The CyberX appliance was received preconfigured for the build environment, with the proper IP address, subnet mask, default gateway, and DNS server. If reconfiguration is needed, then access the server via the command line and type the following command:

```
> cyberx-xsense-network-reconfigure
```

This will open a dialog for the configuration, similar to the dialog shown in Figure C-3.

Figure C-3 CyberX Network Reconfiguration Program on the Appliance



C.2.2. Configuration of Forwarding Rules

The CyberX platform is typically combined with an existing security information and event management (SIEM) system. The following steps describe the process to forward data from CyberX to the SIEM:

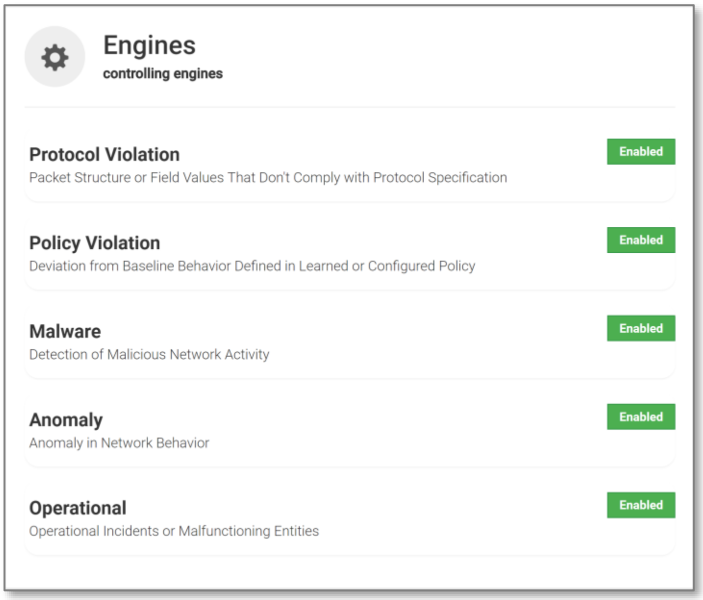
1. Select **Forwarding** from the navigation menu on the CyberX console.
2. Select **Create Forwarding Rule**.
3. Complete the required information for the Forwarding Rule, and then select **Submit**.

C.2.3. Enabling Self-Learning Analytics

The CyberX platform has five different self-learning analytics engines that are used to detect various types of behavioral anomalies within the network. The following steps describe the process to enable individual analytics engines:

1. Select **System Settings** from the navigation menu on the CyberX console.
2. Click the **Enabled/Disabled** button next to each engine to enable or disable the engine. If an engine is enabled, then the button will indicate **Enabled** and will be illuminated with a green background color. An example with all five engines enabled is shown in Figure C-4.

Figure C-4 Example Screenshot with All Five Self-Learning Analytics Enabled



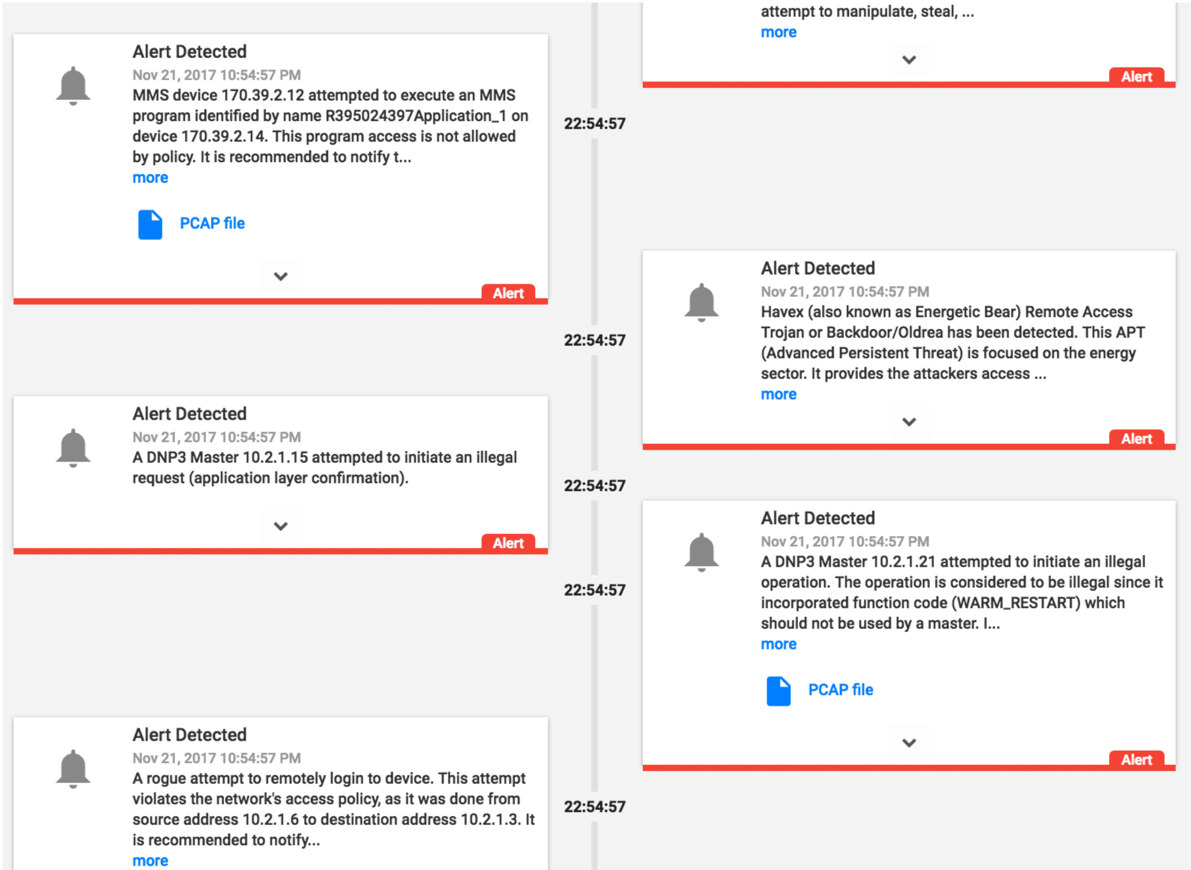
C.3. Anomaly Scenarios

The network-based anomaly detection method was demonstrated for the scenarios detailed in the following subsections. Each scenario includes a description of the anomaly, a detailed description of how each demonstration event was conducted in the CSMS environment, and the observed results.

For the sake of brevity, only a subset of the alerts observed during the demonstration is shown. However, each anomaly scenario includes a screenshot of the alerts summary observed after the anomaly scenario had completed.

Alerts can be observed in the Alerts dashboard, grouped by the severity and type of alert, as well as in the Event Log (timeline view). The Event Log view is shown in the screenshot in Figure C-5.

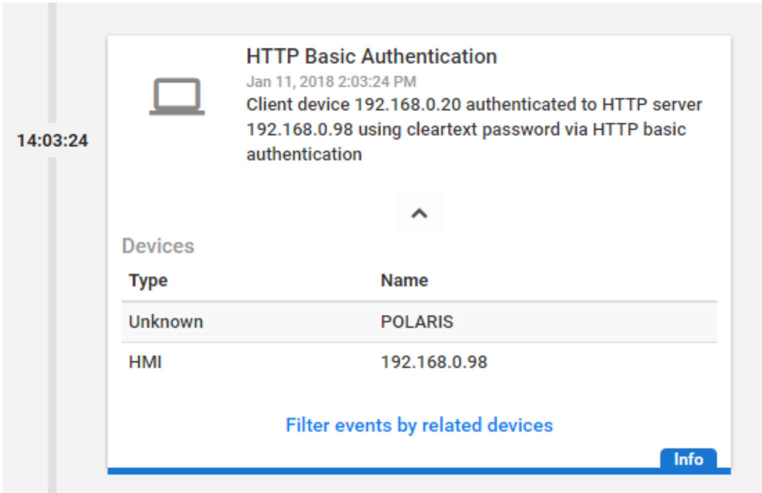
Figure C-5 Event Log (Timeline View) of Real-Time Alerts in the CyberX Console



C.3.1. Unencrypted Hypertext Transfer Protocol Credentials Are Detected on the Network

Unencrypted or plaintext credentials transmitted over a network are a vulnerability for industrial control systems (ICS) networks. If packets containing these credentials are intercepted, then the credentials can be easily unmasked and can be used to obtain unauthorized access to devices or services that use those credentials. This vulnerability can be amplified if multiple devices utilize the same credentials.

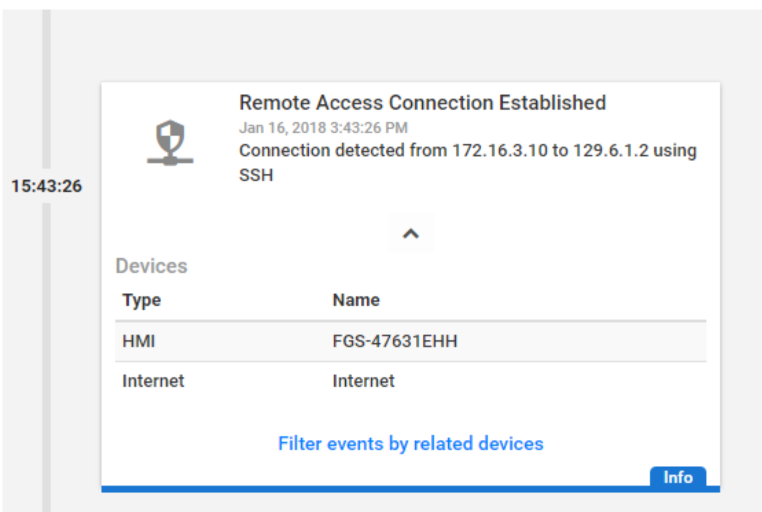
This anomaly was executed on the CRS. An Apache [17] Hypertext Transfer Protocol (HTTP) server was configured on Machining Station 1 and contained a directory that was protected by HTTP basic authentication. The web pages hosted in the protected directory enabled an operator to remotely view machine status information. The connection was initiated from the Firefox browser on the engineering workstation.



C.3.2. Unauthorized Secure Shell Session Is Established with an Internet-Based Server

A Secure Shell (SSH) session is an encrypted and secure connection for remotely sending commands over a network. However, unauthorized SSH sessions with internet-based servers could indicate malicious activity. Attackers can use an SSH session to gain access to the ICS device and network.

This anomaly was executed on the PCS. The OpenSSH [21] suite was installed and configured on a server with an internally routed public IP address (129.6.1.2). The open-source SSH client `PuTTY` [12] was used to establish a connection with the SSH service from the engineering workstation to the internet-based server.

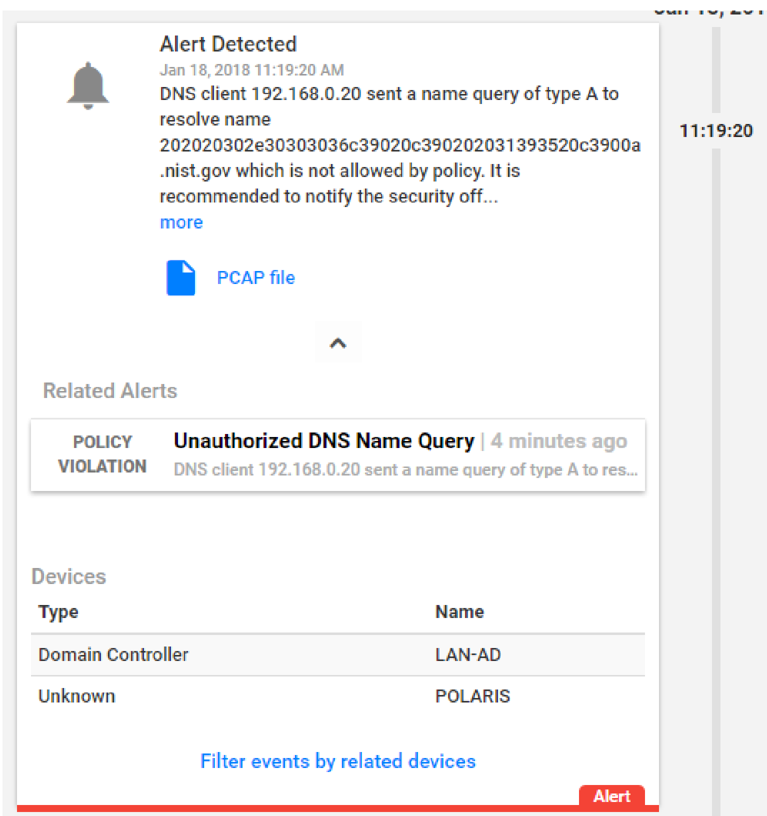


C.3.3. Data Exfiltration to the Internet via DNS Tunneling

Attacks against ICS, with the goal of information gathering, must (at some point) attempt to exfiltrate sensitive or proprietary data from the ICS network, potentially utilizing the internet as a transport mechanism. Monitoring for ICS devices communicating to other devices over the internet can help detect data exfiltration events, especially if the affected device does not normally communicate over the internet.

This anomaly was executed on the CRS. A script was written in Python [14] to exfiltrate the file contents via DNS tunneling. The DNS request functionality was enabled by the Linux command-line tool `nslookup`. A DNS Type A was record was added to the test-bed DNS server, mapping the *.nist.gov domain to our local internet-based server IP address (129.6.1.2).

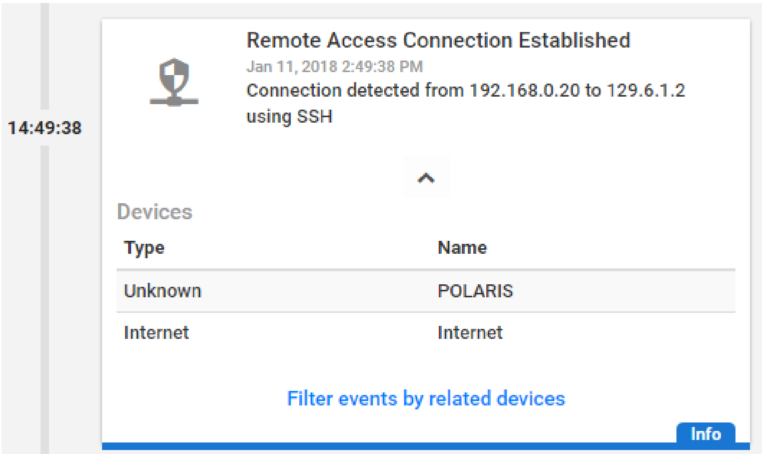
To exfiltrate the file, the Python script would first read 30 bytes from the file measurements.cmm, convert the bytes into a hexadecimal representation encoded as an American Standard Code for Information Interchange string, and concatenate the string as a subdomain with the Uniform Resource Identifier (URI) `.nist.gov`. The resulting URI is sent to the `nslookup` tool, which subsequently transmitted the DNS request. This process repeated until the complete file contents were exfiltrated.



C.3.4. Data Exfiltration to the Internet via Secure Copy Protocol

As previously mentioned, attacks against ICS, with the goal of information gathering, must (at some point) attempt to exfiltrate the data from the ICS network, potentially utilizing the internet as a transport mechanism. Monitoring for ICS devices communicating to other devices over the internet can help detect data exfiltration events, especially if the affected device does not normally communicate over the internet. Depending on the protocol used for exfiltration, the file contents and/or data being exfiltrated may be ascertainable (e.g., specific file types transferred using the File Transfer Protocol [FTP] protocol), providing insight into the impact of the event.

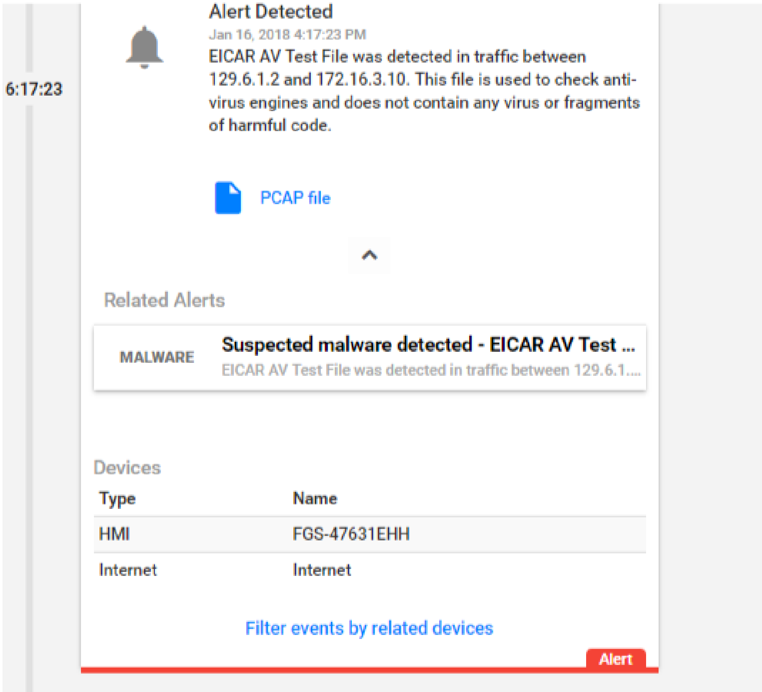
This anomaly was executed on the CRS. The OpenSSH [21] suite was installed and configured on a server with an internally routed public IP address (129.6.1.2). The secure copy protocol was then used to transfer a sensitive file over SSH from the engineering workstation to the internet.



C.3.5. European Institute for Computer Antivirus Research Virus Test File Is Detected on the Network

Malware and computer viruses are serious threats to ICS. Malware can undermine ICS security, confidentiality, and stability, with the potential to sabotage the ICS. Providing the ability to detect the presence of viruses and malware in the ICS network is important for minimizing risk to the manufacturing system.

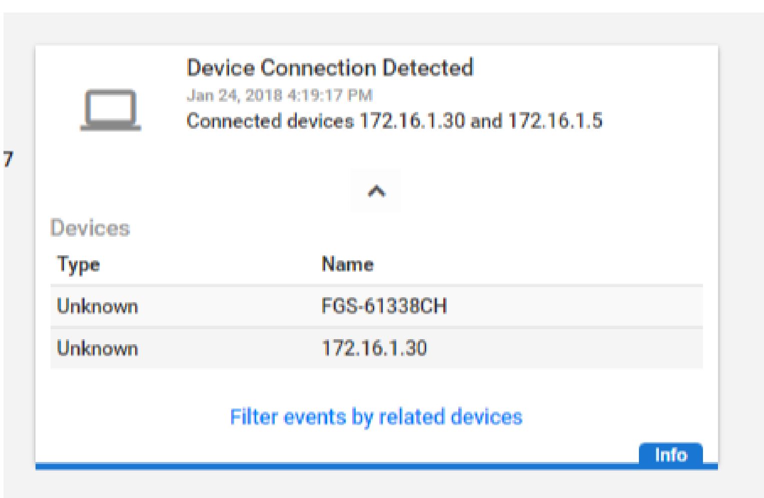
This anomaly was executed on the PCS. The European Institute for Computer Antivirus Research (EICAR) virus test file was transferred from the human-machine interface (HMI) server to the object linking and embedding for process control (OPC) server by using Windows File Sharing (Server Message Block protocol).



C.3.6. Unauthorized Device Is Connected to the Network

It is important to identify all devices on the ICS network, for a complete risk analysis and for minimizing potential attack vectors. The detection of unauthorized devices attached to the ICS network may indicate anomalous activity. These unauthorized devices are important to find and remove, especially because the purpose of an unauthorized device is unknown and may be malicious.

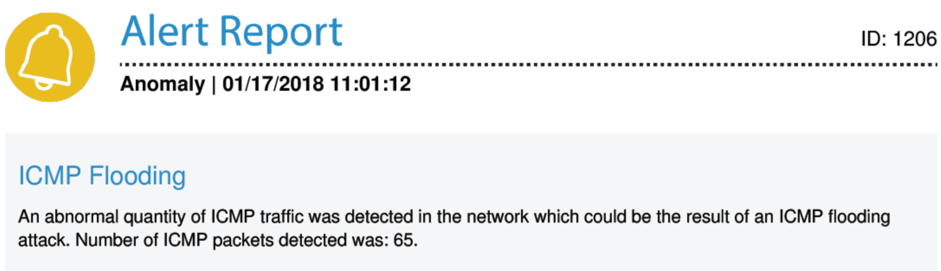
This anomaly was executed on the PCS. The engineering laptop (Windows 7 operating system) was removed from the network during the baseline analysis phase of the product and was later connected to Virtual Local Area Network (VLAN)-2 to execute the anomaly. After the initial connection, background traffic was automatically generated onto the network by the laptop.



C.3.7. Denial-of-Service Attack Is Executed Against the ICS Local Area Network

Disruptive attacks, like a denial of service (DoS), are a serious threat to ICS, especially ICS that rely heavily on networks to communicate. An attacker can launch a DoS attack on ICS and disrupt normal operations, with potentially debilitating effects to the system. The ability to detect such attacks is important to protect the manufacturing system.

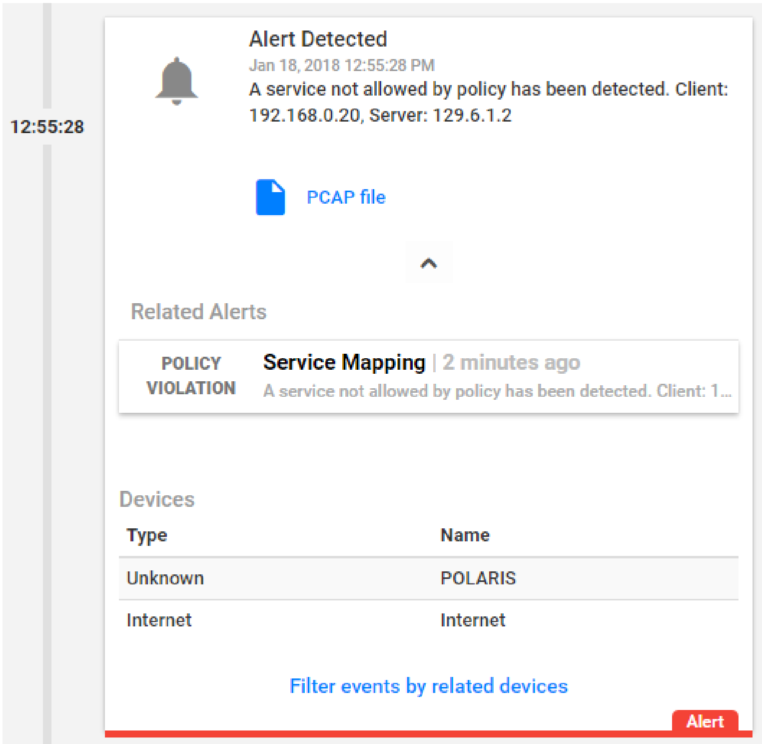
This anomaly was executed on the PCS. The Linux **`ping`** command-line tool was used to transmit a flood of Internet Control Message Protocol (ICMP) packets to the OPC server. The anomaly utilizes **`ping`**'s `flood` flag to inundate the OPC server with ICMP packets. Each ICMP packet requires fragmentation, due to its large size (3,000 bytes), configured using the packet-size flag.



C.3.8. Data Exfiltration Between ICS Devices via User Datagram Protocol

An unauthorized file transfer between two ICS devices could indicate anomalous activity and is important to identify, especially when the devices do not normally communicate or when the exchange of files is unauthorized.

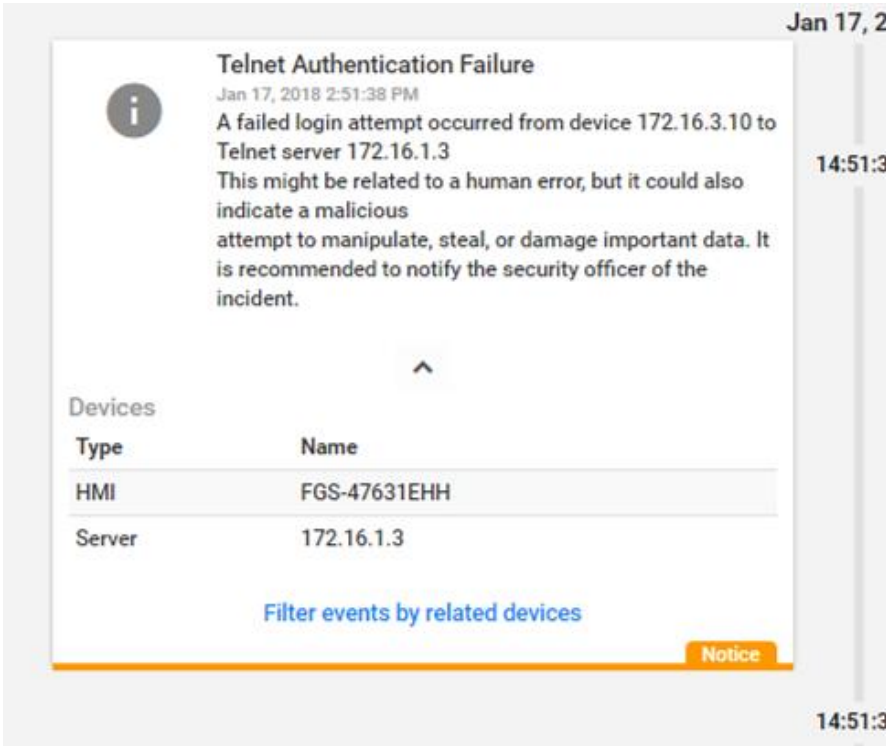
This anomaly was executed on the CRS. A tape archive file was transmitted from the cybersecurity virtual machine (CybersecVM) to the engineering workstation by using the Linux utility netcat and User Datagram Protocol (UDP) sockets. UDP Port 9999 was used for the transfer.



C.3.9. Invalid Credentials Are Used to Access a Networking Device

Authentication systems that are not rate-restricted may be vulnerable to password-guessing attacks, especially if the default credentials of the device have not been changed. Compiled lists containing default user credentials are freely available on the internet, as are lists of commonly used usernames and passwords. Given enough time, an attacker may be able to access vulnerable systems by using a brute-force password attack.

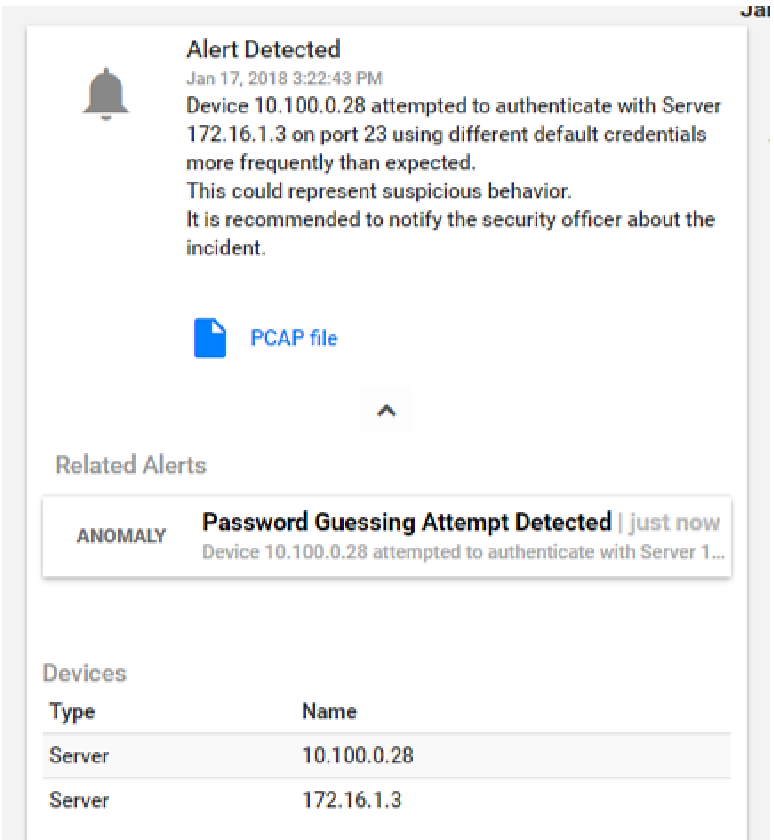
This anomaly was executed on the PCS. The Allen-Bradley software Studio 5000 was used to download the logic from the PCS programmable logic controller (PLC) to the engineering workstation. Physical access to the PLC was required in order to change the operation mode from RUN to REMOTE RUN.



C.3.10. Brute-Force Password Attack Against a Networking Device

As previously mentioned, authentication systems that are not rate-restricted may be vulnerable to password-guessing attacks, especially if the default credentials of the device have not been changed. Compiled lists containing default user credentials are freely available on the internet, as are lists of commonly used usernames and passwords. Given enough time, an attacker may be able to access vulnerable systems by using a brute-force password attack.

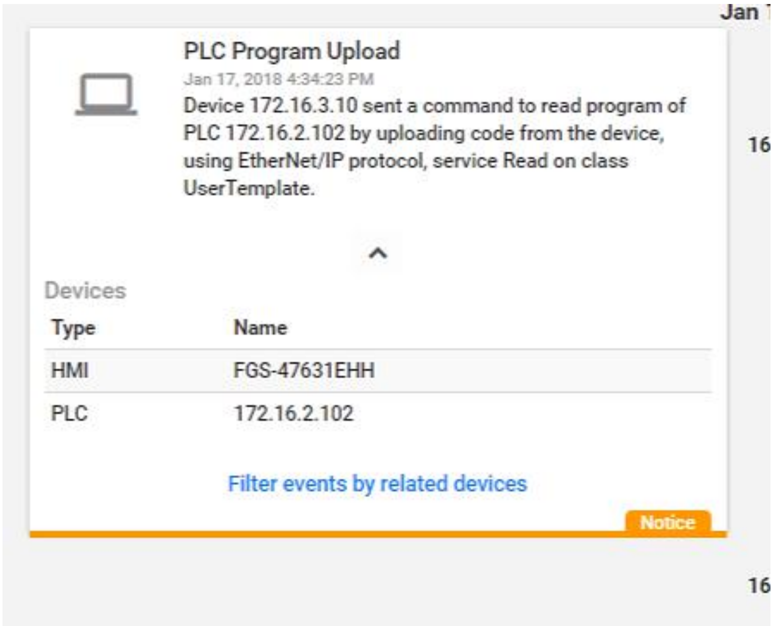
This anomaly was executed on the PCS. The software Nmap [16] was used to generate the brute-force password attack by using the script **`telnet-brute`**. The attack was pointed at the PCS router, which has a Telnet service for remote configuration and is protected by a password. The service was not configured to limit the number of authentication attempts.



C.3.11. Unauthorized PLC Logic Download

Many ICS devices provide services to remotely update control logic over the network. These network services can also provide a mechanism for attackers to replace valid control logic with malicious logic if the device is not protected.

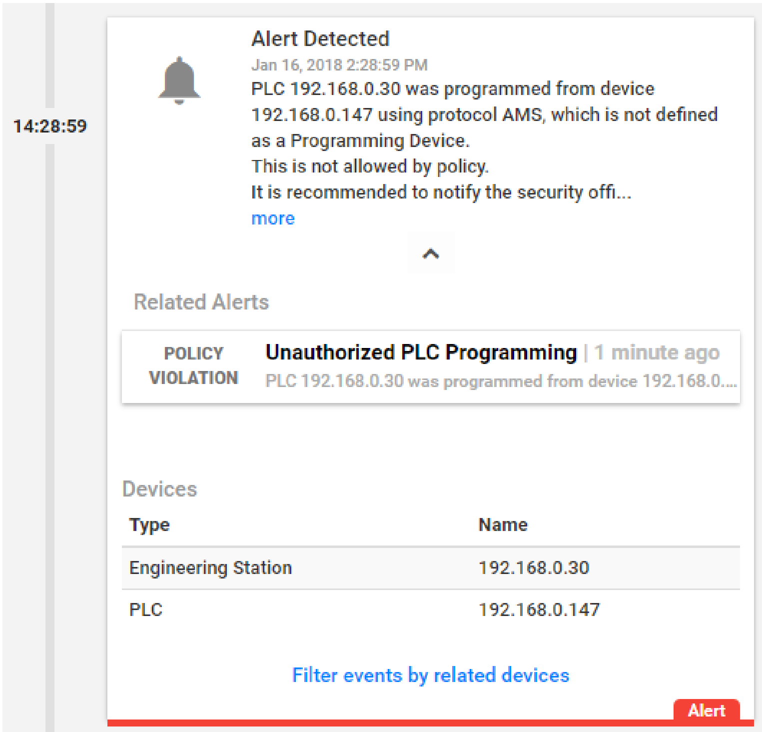
This anomaly was executed on the PCS. The Allen-Bradley software Studio 5000 was used to download the logic from the PCS PLC to the engineering workstation. Physical access to the PLC was required in order to change the operation mode from RUN to REMOTE RUN.



C.3.12. Unauthorized PLC Logic Update – CRS

Many ICS devices provide services to remotely update control logic over the network. These network services can also provide a mechanism for attackers to replace valid control logic with malicious logic if the device is not protected.

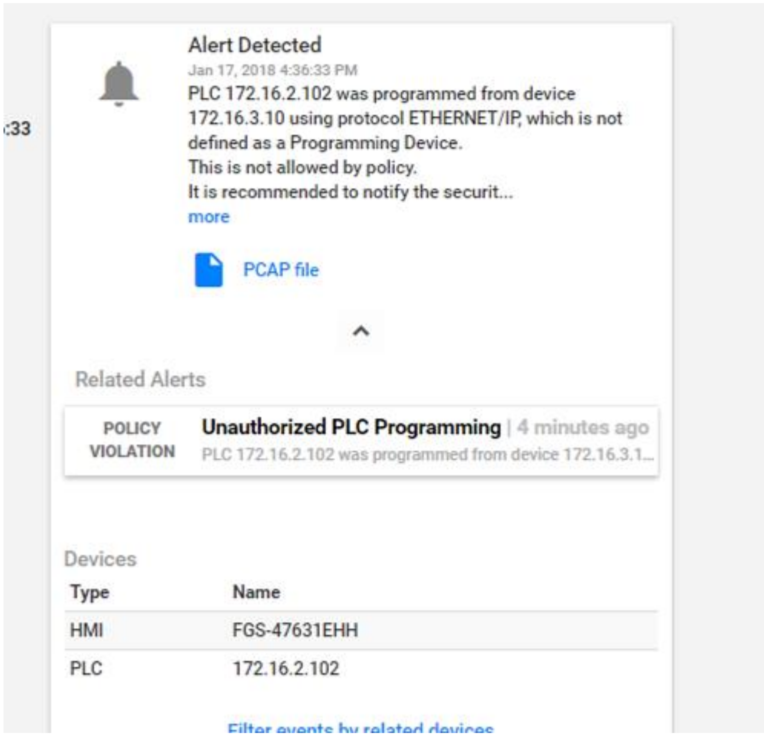
This anomaly was executed on the CRS. The TwinCAT eXtended Automation Engineering (XAE) software from Beckhoff was used to deploy new logic to the CRS PLC. The deployment was performed by using the engineering laptop while the PLC was in the ONLINE mode. The unauthorized logic was functionally compatible with the authorized logic that it replaced, with minor modifications.



C.3.13. Unauthorized PLC Logic Update – PCS

As previously mentioned, many ICS devices provide services to remotely update control logic over the network. These network services can also provide a mechanism for attackers to replace valid control logic with malicious software if the device is not protected.

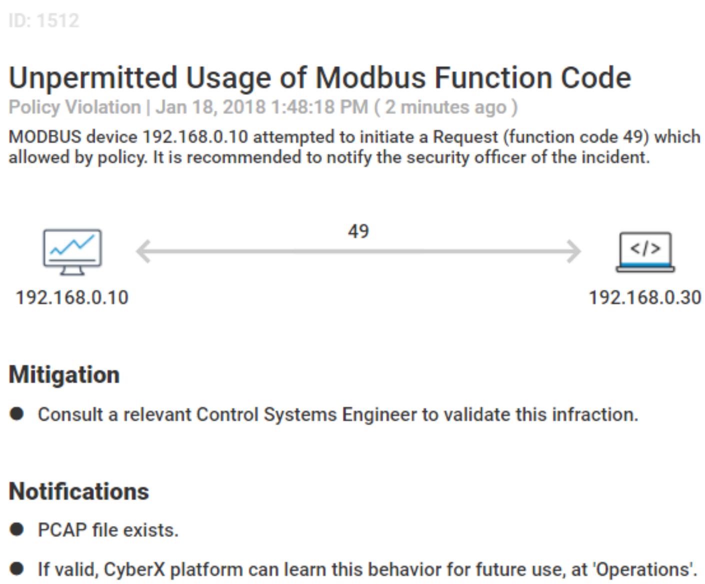
This anomaly was executed on the PCS. The Allen-Bradley software Studio 5000 was used to upload new logic from the engineering workstation to the PCS PLC. Physical access to the PLC was required in order to change the operation mode from RUN to REMOTE RUN.



C.3.14. Undefined Modbus Transmission Control Protocol Function Codes Are Transmitted to the PLC

Communications that do not conform to the defined specifications of the industrial protocol may cause an ICS device to act in an undefined or unsafe manner. Depending on the manufacturing process and the ICS device, the nonconforming communications may or may not be impactful, but investigation into the cause is warranted.

This anomaly was executed on the CRS. Python [14] was used to create a Modbus Transmission Control Protocol (TCP) message with the undefined function code value of 49 (`0x31`). The message was generated by the CybersecVM and was transmitted to the PLC Modbus server.

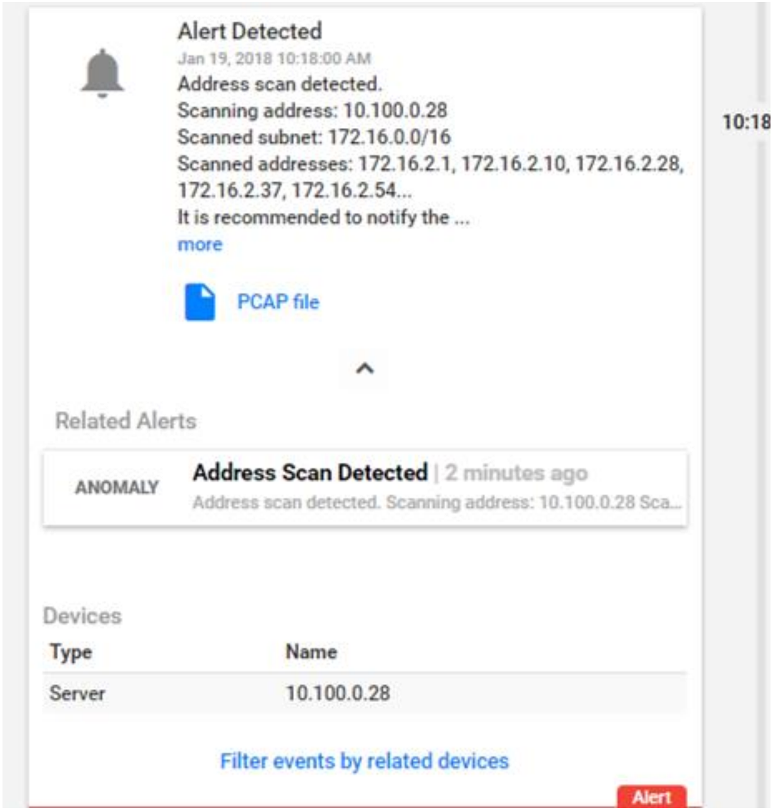


C.3.15. Unauthorized Ethernet/IP Scan of the Network

During the reconnaissance phase, an attacker may attempt to locate vulnerable services in an ICS network and will likely include probing for ICS-specific services (e.g., Ethernet/IP). Once a vulnerable service, host, or device is discovered, an attacker may attempt to exploit that entity.

This anomaly was executed on the PCS. The software Nmap [16] was used to perform a port scan (Ports 1 through 1024) against two hosts: the HMI and the plant controller. The scan originated from the CybersecVM, logically located in the test-bed local area network (LAN).

This anomaly was executed on the PCS. The software Nmap [16] was used to perform an Ethernet/IP device scan by using the script **`enip-info`**. The scan was pointed at the PCS subnet `172.16.2.100/28` and was executed by the CybersecVM in the test-bed LAN.


Alert Detected
Jan 19, 2018 10:18:00 AM
Address scan detected.
Scanning address: 10.100.0.28
Scanned subnet: 172.16.0.0/16
Scanned addresses: 172.16.2.1, 172.16.2.10, 172.16.2.28, 172.16.2.37, 172.16.2.54...
It is recommended to notify the ...
more
PCAP file
10:18
Related Alerts
ANOMALY
Address Scan Detected | 2 minutes ago
Address scan detected. Scanning address: 10.100.0.28 Sca...
Devices
Type
Name
Server
10.100.0.28
Filter events by related devices
Alert

Appendix D. OSIsoft Process Information Supplemental Information

The OSIsoft Process Information (PI) System is a suite of software applications for capturing, analyzing, and storing real-time data for industrial processes. Although the PI System is typically utilized as a process historian, the PI System is also utilized to collect, store, and manage data in real time. Interface nodes retrieve data from disparate sources to the PI Server, where the PI Data Archive resides. Data is stored in the Data Archive and is accessible in the assets defined in the Asset Framework (AF). Data is then typically accessed, either directly from the Data Archive or from the AF Server, by using tools in the PI visualization suite. Typically, most PI System users consume data by accessing the AF Server, rather than directly accessing the Data Archive. This build demonstrates how PI can be leveraged to monitor for specific behavioral anomalies of the process that may be caused by cybersecurity incidents, and to alert operators and cybersecurity personnel of the anomalies.

D.1. Build Architecture

The PI System was installed in a virtual environment (HyperV) that already existed within the collaborative robotic system (CRS). The virtual machine (VM) for the PI System used Windows Server 2008 R2 as the operating system, with four virtual central-processing-unit cores and 16 gigabytes (GB) of random-access memory. The VM was networked directly into the existing network topology of the CRS with a dedicated Internet Protocol (IP) address (192.168.0.21).

D.2. Installation and Configuration

Compared with the other three installations, the PI System was installed locally on existing virtualization hardware. Remote assistance and troubleshooting were provided by OSIsoft for the installation and configuration of the system within the CRS.

Six components were installed in the VM:

- PI AF
- PI Data Archive
- PI Process Explorer
- PI Vision
- PI Modbus Ethernet Interface
- Structured Query Language Server 2012

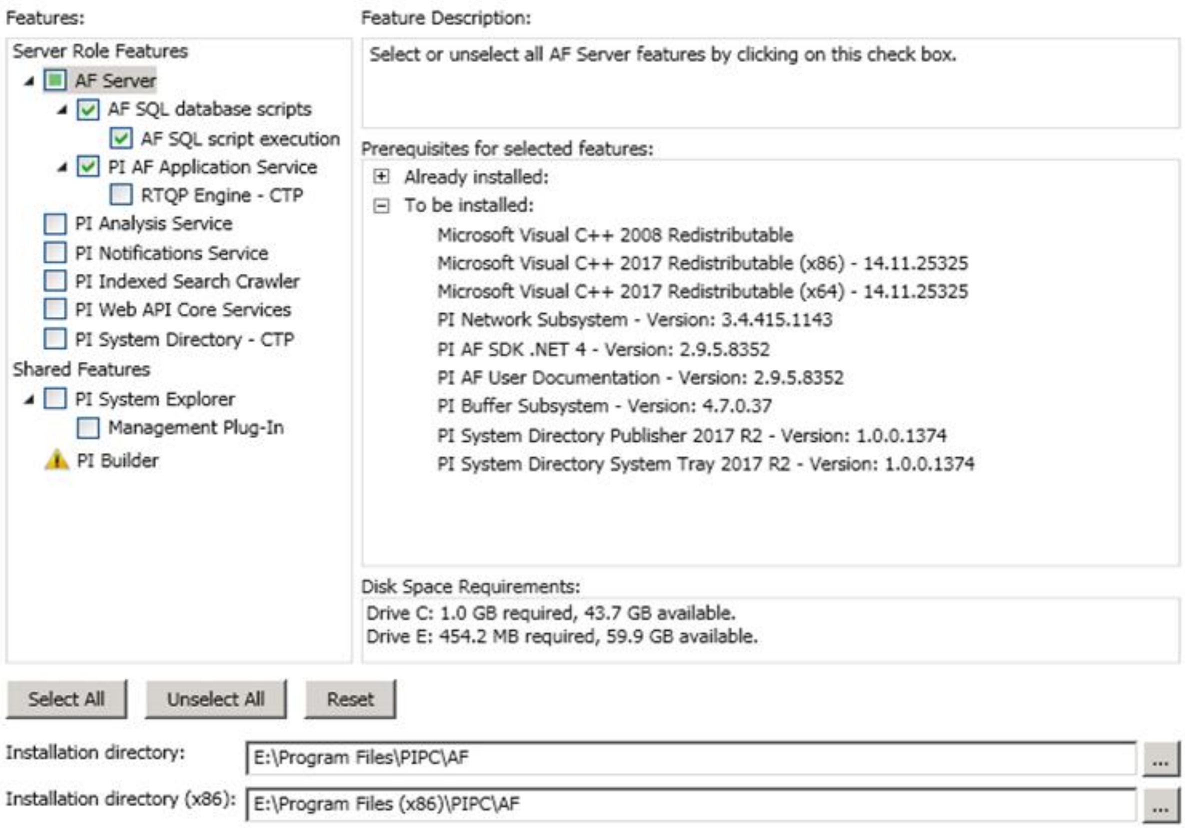
Four additional hard-drive partitions (virtual) were created to support the PI System installation:

- PI Server (E:): 60 GB
- archives (F:): 60 GB
- queues (G:): 30 GB
- backups (H:): 21 GB

D.2.1. PI AF Installation

1. Run *PI-AF-Services_2017-R2-Update-1_Demo.exe* to launch the installer.
2. Select the **Server Role Features** shown in Figure D-1. Ensure that the **Installation Directory** is set to the corresponding drive letter labeled as *PI Server*. Click **Next**.

Figure D-1 Server Role Features to Be Selected During PI AF Installation



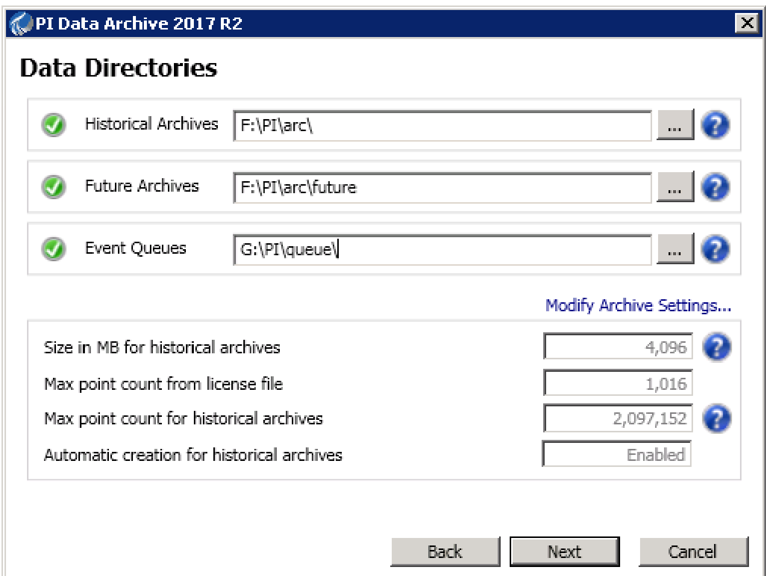
3. Keep the default settings. Click **Next**.
4. Set the **Directory Name** to <*Configure Later*>. Click **Next**.
5. Leave the **Service Account** as default. Click **Next**.
6. Upon completed installation, reboot the server.

D.2.2. PI Data Archive Installation

1. Run the *PI-Data-Archive_2017_R2A_Demo_.exe* file.
2. When prompted for the **License File**, browse to the location of the *pilicense.dat* file from OSIsoft. Click **Next**.
3. Specify a name for the **Default Asset server**, or leave it as the default host name. Click **Next**.

4. Select the **Installation Directory** for the Data Archive. Click **Next**.
5. Set the remaining directories as shown in Figure D-2, corresponding to the correct drive letters. Click Next.
6. Click **Next**, and verify that the service status shows as **Running**. Click **Next** to finish the installation and to reboot the server.

Figure D-2 Data Directories to Be Selected During PI Data Archive Installation



D.2.3. PI System Process Explorer Installation

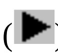
1. Run the *PIProcessBook_2015_R2_SP1_06-Jun-2018.exe* file to start the installation.
2. A screen titled OSIsoft Setup Progress will begin, installing the different required components.
3. A dialog box will appear once the installation is complete.

D.2.4. PI Vision Installation

1. Run the *PI-Vision_2017-R2-Update-1-90-Day-Trial_.exe* file to start the installation.
2. Select the **Operating Configuration Store**. In this build, the Asset Server was called PI-ROBOTICS. Click **Connect**, and then click **Next**.
3. Verify that the **PI Web API port is 443**. Click **Next**.
4. On the Submit URL page, do not change the automatically generated **Indexed Search Crawler Submit URL**. In this build, the automatically generated Uniform Resource Locator (URL) was *https://pi-robotics.lan.lab/piwebapi/*. Click **Next**.
5. Review the changes. Click **Next**.

6. When the installation has completed, review the Confirmation page for errors. If no errors are found, then click **Finish**.
7. The installer will continue installing additional components. Click **Continue** when prompted to install Windows features.
8. If prompted, leave the default installation directories. Click **Next**.
9. Once the installation finishes, click **Finish**.

D.2.5. PI System Modbus Ethernet Interface Installation

1. Run the *ModbusE_ReadWrite_4.2.2.31_DEMO.exe* file to start the installation.
2. Keep all default settings, and complete the installation.
3. Open PI Interface Configuration Utility, and select the interface **PIModbusE1**.
4. Configure the **Display Name**. In this build, the default name was kept.
5. Select the option **Service** in the left navigation panel.
6. Select the **Startup Type** option **Auto**, click **Create**, and then click **Apply**.
7. Click the Start Service () button on the top navigation bar.
8. If the service is running properly, then the label **Running** will appear on the status bar at the bottom of the dialog.

D.2.6. PI System Points and Assets Configuration

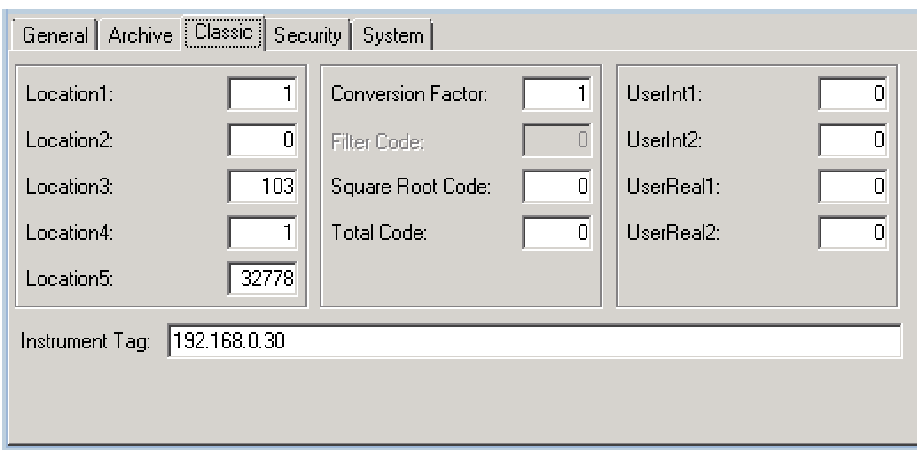
PI System points utilizing the ModbusE interface were manually created using the PI System Management Tools (SMT) software. Modbus device addresses, register names, and register addresses were known prior to configuring the points.

1. Launch the PI SMT by navigating to **Start** > **All Programs** > **PI System** > **PI System Management Tools**.
2. Select **Points** > **Points Builder** from the left navigation pane.
3. Create a new tag, and enter the required attributes (shown in Figure D-3). An example of the configuration for the Point PLC-ExperimentMode is shown in Figure D-4.
4. Click **Save**.

Figure D-3 Configuration Options in the PI Point Builder for Tags Utilizing the ModbusE Interface

Point Builder Tab	Field	Setting
General	Name	ModbusETest
	Point source	MODBUSE
	Point type	Int32
Classic	Location 1	1 (or whatever was used in the Interface ID field in PI ICU)
	Location 2	(Node ID). Example: 1
	Location 3	(Data Type * 100 + Function Code). Example: 103 (which is 1 (for Int16) * 100 + 3 (for holding registry)). Refer the interface manual for a full list of data types and function codes
	Location 4	1 (Scan class Frequency)
	Location 5	(offset from 40000 for holding registry). Example: 52 Represents 40052 register
	Instrument tag	IP address or hostname of the Ethernet communications node. Must match with the IP Addr./Hostname entered

Figure D-4 Example Configuration Settings for the Tag PLC-ExperimentMode

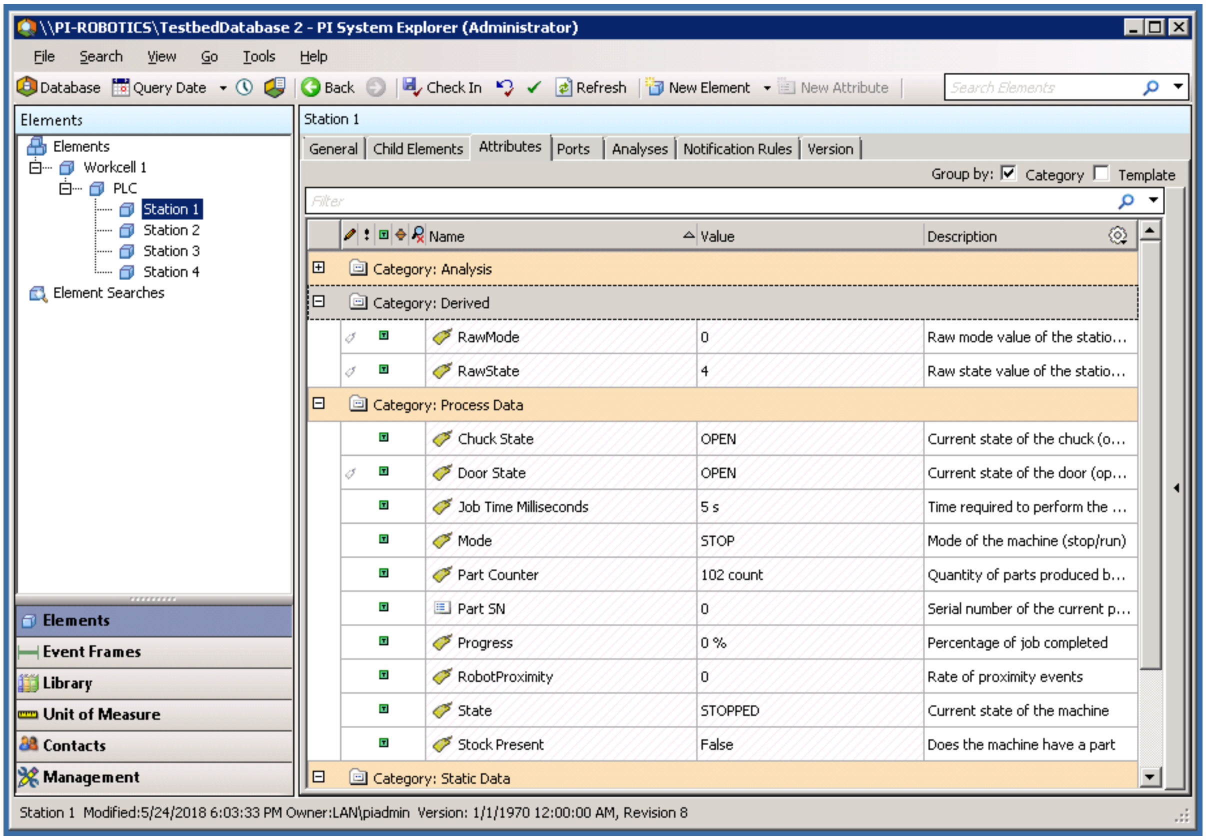


In Figure D-4, the fields **Location1** through **Location5** have different uses, depending on the interface used, and are described in detail in Figure D-3. The **Instrument Tag** field describes the IP address of the Modbus Transmission Control Protocol (TCP) server that the ModbusE interface needs to poll.

The PI System AF and System Explorer were used to define a hierarchical structure for the PI System points, to display tag values for each asset, and to provide an interface for viewing and acknowledging alerts. Because of the relatively simple interactions among elements of the CRS, the structure created in the AF contained the supervisory programmable logic controller (PLC) as the top-level element, and Station 1 through Station 4 as child elements.

Asset templates were created for the PLC and four machining stations to automatically link to the proper PI System points based on the asset. The final configuration of assets is shown in Figure D-5, showing the hierarchical structure of **Workcell 1** > **PLC** > **Station 1**. Also shown in this figure are the **Attributes** for Station 1, as received from the PI System points.

Figure D-5 PI System Explorer View Showing the Configured Assets (Elements), the Resulting Hierarchical Structure of Assets, and Live Attributes Received from Station 1



For both the PLC and machining-station asset templates, analysis functions were created to generate alerts for the operator when identified anomalous events are detected. The anomalous events to be detected are the anomalies described in Section D.3. The analysis functions are described in Sections D.2.7 and D.2.8. Respective event-frame generators for each analysis function were created to generate the actual alerts.

D.2.7. PLC Asset Template Analysis Functions

The analysis functions provided in the following subsections were created to generate alerts in the PLC asset template when their respective anomalous events are detected. For the sake of brevity, the event-frame generation code is not shown. In general, the typical event-frame generator contains logic to activate the event frame when the analysis function result is `TRUE`, and to stop the event frame after the analysis function result is `FALSE` or after a related element variable changes to a value indicating that the failure or fault has been resolved.

D.2.7.1. High Workcell Temperature

If the simulated workcell temperature increases above the value of 29.0 degrees Celsius, then generate an alert by using the following command:

```
R261 := if ('WorkcellTemperature'>= 29.0) then 1 else 0;
```

D.2.7.2. Inspection Failure

If the inspection station reports a failed inspection count greater than or equal to three, then generate an alert by using the following command:

```
Alarm := If('FailedInspectionCounter' >= 3) Then 1 Else 0;
```

D.2.7.3. Station Out-of-Sync

If any of the machining stations is not in the RUN mode while the workcell is in the RUN state, then generate an alert by using the following commands:

```
S1State := '.\Elements[@Name=Station 1]|State';

S2State := '.\Elements[@Name=Station 2]|State';

S3State := '.\Elements[@Name=Station 3]|State';

S4State := '.\Elements[@Name=Station 4]|State';


WCState := If(TimeEq('WorkcellState','*-5s','*',"RUN")>=5)
Then "RUN" Else "Starting";

StationModes := if (S1State = "STOPPED" Or S2State = "STOPPED"
OR
 S3State = "STOPPED" Or S4State = "STOPPED")
 Then 1 Else 0;

Alarm := if (StationModes = 1 And WCState = "RUN") Then 1 Else
0;
```

D.2.8. Machining Station Asset Template Analysis Functions

The analysis functions provided in the following subsections were created to generate alerts in the machining station asset template when their respective anomalous events are detected. For the sake of brevity, the event-frame generation code is not shown. As previously mentioned, in general, the typical event-frame generator contains logic to activate the event frame when the analysis function result is `TRUE`, and to stop the event frame after the analysis function result is `FALSE` or after a related element variable changes to a value indicating that the failure or fault has been resolved.

D.2.8.1. High Trouble Call Count

Two analysis functions were created for this alert. First, determine if the machining station is in the TROUBLE state by using the following command:

```
Trouble := if ('State' = "TROUBLE" AND ((PrevVal('State','*-
1s') = "TROUBLE") = False)) THEN "TROUBLE" ELSE NoOutput();
```

If the machining station has entered the TROUBLE state, then count this event. If the number of times that the machining station has entered the TROUBLE state in the previous 10 minutes is greater than or equal to five, then generate an alert by using the following command:

```
TroubleCount := If (EventCount('Alarm-TroubleCounterEvent','*-
10m','*') >= 5) Then 1 Else 0;

Variable1 := 'State';
```

D.2.8.2. Robot Proximity Fault

If the machining station is in the RUN mode and a robot proximity message has not been received within the previous two minutes, then generate an alert by using the following command:

```
Alarm := If (('Mode' = "RUN") And (PrevVal('Mode','*-2m') =
"RUN") And (TagMax(';RobotProximity','*-2m','*') = 0)) then 1
else 0;
```

D.2.8.3. Station Door Fault

If the machining station is in the ACTIVE state and the door is not closed, then generate an alert by using the following command:

```
Door_Open_Alarm := if (TimeEq('State','*-2s','*',"ACTIVE")>=2
And TimeEq('Door State','*-2s','*',"CLOSED")<1) Then 1 ELSE 0;

Variable1 := TimeEq('Door State','*-2s','*',"CLOSED");
```

D.2.8.4. Station Mode Error

If the register value for the machining station mode (as written by the PLC) is not within the valid range of values (0 to 1), then generate an alert by using the following command:

```
Alarm := If('RawMode' < 0 OR 'RawMode' > 1) Then 1 Else 0;
```

D.2.8.5. Station State Error

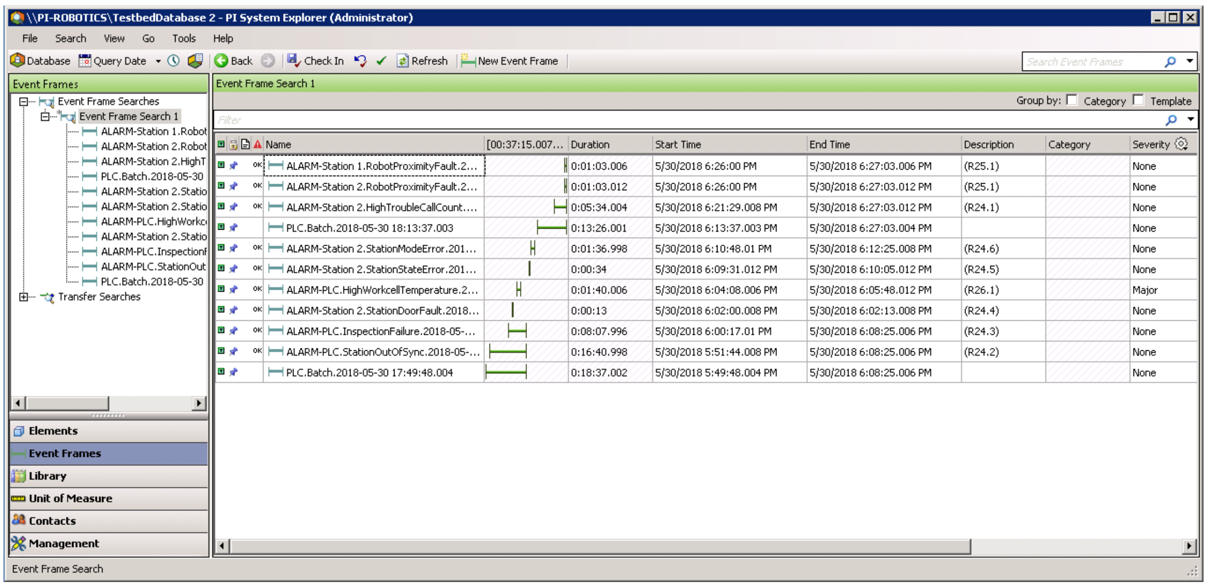
If the register value for the machining station state (as reported by the machining station) is not within the valid range of values (0 to 5), then generate an alert by using the following command:

```
Alarm := If('RawState' < 0 OR 'RawState' > 5) Then 1 Else 0;
```

D.2.9. Viewing and Acknowledging Alerts

The PI System Explorer was used to view and acknowledge alerts (event frames) generated by the analyses templates. An example of the alerts is shown in Figure D-6, showing all of the alerts generated by the anomalies during the execution of the anomaly scenarios.

Figure D-6 PI System Explorer Interface Showing an Example of Alerts Displayed to the Operator for Acknowledgment, as Used During Anomaly Scenario Execution



D.3. Anomaly Scenarios

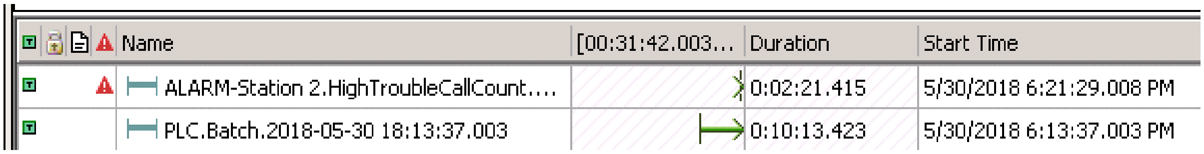
The historian/sensor-based anomaly detection method was demonstrated for the scenarios detailed in the following subsections. Each scenario includes a description of the anomaly, a detailed description of how each demonstration event was conducted in the Cybersecurity for Smart Manufacturing Systems environment, and the observed results.

The anomalies listed below demonstrate the fusion of cybersecurity and manufacturing activities into a cohesive operation for detecting operational/maintenance issues and for potentially identifying issues caused by cybersecurity incidents. In-depth knowledge of the manufacturing system enables engineers to design PI System analysis functions to monitor and alert when anomalous events occur, and to track trends of anomalies over extended periods of time. With proper communication between operators and cybersecurity personnel, anomalous manufacturing process events can be analyzed to determine if they could have been caused by a cybersecurity incident and could have been mitigated.

D.3.1. Frequency Increase of Trouble Calls from a Machining Station

Trouble calls are automatically generated by a machining station when it detects an anomaly during manufacturing operations (e.g., broken tooling, coolant failure).

This anomaly was executed on the CRS. The machining station logic for Station 2 contained a register that enabled trouble calls to be initiated on demand, generating the anomaly. This register was set using a menu option on the human-machine interface (HMI). When enabled, the machining station would enter the TROUBLE state after each part was placed in the machine, and would be automatically cleared after eight seconds had elapsed.

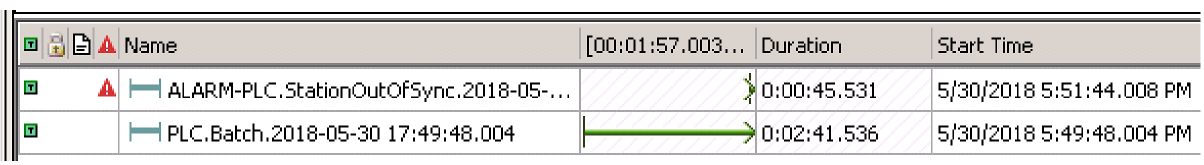


Name	[00:31:42.003...	Duration	Start Time
ALARM-Station 2.HighTroubleCallCount....		0:02:21.415	5/30/2018 6:21:29.008 PM
PLC.Batch.2018-05-30 18:13:37.003		0:10:13.423	5/30/2018 6:13:37.003 PM

D.3.2. Machining Station Shuts Down During Normal Workcell Operations

The workcell requires that all four machining stations are operational and in the RUN mode while the workcell is in the RUN state.

This anomaly was executed on the CRS. The machining station logic for Station 2 contained a register that enabled a “forced shutdown” to be initiated, generating the anomaly. This register was set using a menu option on the HMI. When enabled, the machining station would enter the STOP mode while the rest of the workcell machines were operational.

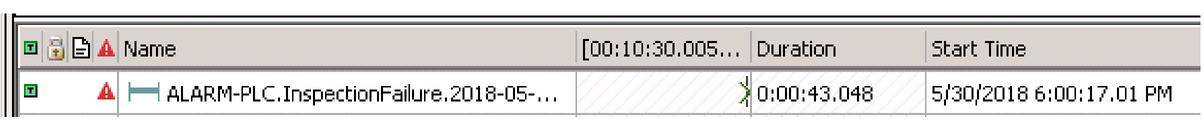


Name	[00:01:57.003...	Duration	Start Time
ALARM-PLC.StationOutOfSync.2018-05-...		0:00:45.531	5/30/2018 5:51:44.008 PM
PLC.Batch.2018-05-30 17:49:48.004		0:02:41.536	5/30/2018 5:49:48.004 PM

D.3.3. Inspection Station Rejects All Parts Leaving the Workcell

The quantity of good and bad parts exiting the inspection station is counted by the supervisory PLC. An increase in the number of rejected parts indicates that the workcell should be inspected by an operator to determine the cause.

This anomaly was executed on the CRS. The station logic for Station 4 contained a register that enabled the “inspection failure of all parts” anomaly. This register was set using a menu option on the HMI. When enabled, the inspection station would report a failed result for every inspection performed until the anomaly was disabled.



Name	[00:10:30.005...	Duration	Start Time
ALARM-PLC.InspectionFailure.2018-05-...		0:00:43.048	5/30/2018 6:00:17.01 PM

D.3.4. Machining Station Door Sensor Fails

The unsafe condition that this sensor failure can cause warrants investigation by an operator. Substantial damage can occur to both the machining station and robots if this sensor failure is not detected. This anomaly could be a goal for an attacker who has the intent to cause production disruption or financial loss through equipment damage.

This anomaly was executed on the CRS. The machining station has a simulated door that must open and close to allow the robot to have access into the machine for placing raw material and removing finished parts. The machining station logic for Station 2 contained a register that enabled the door-sensor failure anomaly. This register was set using a menu option on the HMI. When enabled, the failure of this sensor caused the machining station to report that the door was always "OPEN."

Name	[00:12:13.003...	Duration	Start Time
ALARM-Station 2.StationDoorFault.2018...		0:00:10.492	5/30/2018 6:02:00.008 PM

D.3.5. Abnormal Process Variable Data Is Transmitted to the PLC

Two-way communication occurs between the supervisory PLC and the machining station during normal operations. If a process variable trends outside the known operational range, then this anomaly should be reported.

This anomaly was executed on the CRS. Each machining station contains a Modbus TCP server for communicating operational information to, and receiving commands from, the supervisory PLC. The machining station logic for Station 2 contained a register that enabled specific operational information to be corrupted before it was transmitted to the PLC. This register was set using a menu option on the HMI.

Name	[00:19:44.007...	Duration	Start Time
ALARM-Station 2.StationStateError.201...		0:00:36.826	5/30/2018 6:09:31.012 PM

D.3.6. Abnormal Process Variable Data Is Transmitted to a Machining Station

As previously mentioned, two-way communication occurs between the supervisory PLC and the machining station during normal operations. If a process variable trends outside the known operational range, then this anomaly should be reported.

This anomaly was executed on the CRS. The supervisory PLC contains a Modbus TCP client for communicating commands to, and receiving operational information from, the machining stations. The supervisory PLC contained a register that enabled specific commands to be corrupted before they were transmitted to the machining stations. This register was set using a menu option on the HMI.

Name	[00:21:01.005...	Duration	Start Time
ALARM-Station 2.StationModeError.201...		0:00:42.732	5/30/2018 6:10:48.01 PM

D.3.7. Robots Fail to Send Required Sensor Data to a Machining Station

As previously mentioned, the unsafe condition that this sensor failure can cause warrants investigation by an operator. Substantial damage can occur to both the machining station and robots if this sensor failure is not detected. This anomaly could be a goal for an attacker who intends to cause production disruption or financial loss through equipment damage.

This anomaly was executed on the CRS. The machining station has a simulated door that must open and close to allow the robot access into the machine for placing raw material and removing finished parts. The two robots report their locations (via Modbus TCP) to the machining stations so that they do not attempt to close the door while the robot is still operating within the machine. Robot Controller 1 contains a configuration option to disable this reporting, resulting in Stations 1 and 2 not receiving robot location information. This configuration option was used to generate the anomaly.

Name	[00:36:12.995...	Duration	Start Time
ALARM-Station 1.RobotProximityFault.2...		0:00:31.083	5/30/2018 6:26:00 PM
ALARM-Station 2.RobotProximityFault.2...		0:00:31.086	5/30/2018 6:26:00 PM

D.3.8. Workcell Temperature Increases Above a Specified Threshold

Process variables that impact the output quality of the workcell must be monitored for deviation from expected values. The temperature of the workcell increases during normal operations and must be properly cooled to maintain quality; therefore, the workcell temperature is monitored.

This anomaly was executed on the CRS. The workcell contained a simulated temperature sensor, which was used to "monitor" the temperature within the workcell. The temperature was then displayed to the operator, on the HMI. The workcell temperature would increase to an expected value while the workcell was operational and would decrease to room temperature when the system was shut down. During anomalous conditions, the temperature would increase beyond a threshold, causing all parts produced during that period to be scrapped.

The temperature sensor was simulated by the PLC. The anomalous temperature increase was enabled by a register within the PLC and was set using a menu option on the HMI.

Name	[00:14:21.001...	Duration	Start Time
ALARM-PLC.HighWorkcellTemperature.2...		0:00:33.602	5/30/2018 6:04:08.006 PM

1753 **Appendix E. Acronyms and Abbreviations**

24/7	24 Hours a Day, Seven Days a Week
AF	Asset Framework
BAD	Behavioral Anomaly Detection
CPU	Central Processing Unit
CRS	Collaborative Robotic System
CSMS	Cybersecurity for Smart Manufacturing Systems
CSV	Comma-Separated Values
CybersecVM	Cybersecurity Virtual Machine
DA	Data Access
DCOM	Distributed Component Object Model
DMZ	Demilitarized Zone
DNS	Domain Name System
DoS	Denial of Service
EICAR	European Institute for Computer Antivirus Research
EL	Engineering Laboratory
FTP	File Transfer Protocol
GB	Gigabyte(s)
GUI	Graphical User Interface
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ICS	Industrial Control System
ID	Identifier
IDS	Intrusion Detection System
IP	Internet Protocol
IPC	Industrial Personal Computer
IT	Information Technology
LAN	Local Area Network

LTS	Long-Term Support
M	Megabyte(s)
MAC	Media Access Control
NCCoE	National Cybersecurity Center of Excellence
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NISTIR	National Institute of Standards and Technology Interagency Report
NTP	Network Time Protocol
OPC	Object Linking and Embedding for Process Control
OS	Operating System
OT	Operational Technology
PCS	Process Control System
PDF	Portable Document File
PHP	Hypertext Preprocessor
PI	Process Information
PLC	Programmable Logic Controller
ROS	Robot Operating System
SCADA	Supervisory Control and Data Acquisition
SIEM	Security Information and Event Management
SMT	System Management Tools
SNTP	Simple Network Time Protocol
SP	Special Publication
SPAN	Switch Port Analyzer
SSH	Secure Shell
TCP	Transmission Control Protocol
TE	Tennessee Eastman
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

USB	Universal Serial Bus
VLAN	Virtual Local Area Network
VM	Virtual Machine
XAE	eXtended Automation Engineering
XLSX	Microsoft Excel Workbook File

Appendix F. References

[1] "Framework for improving critical infrastructure cybersecurity," NIST, Gaithersburg, MD, Apr. 16, 2018 [Online]. Available: https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11.

[2] K. Stouffer et al., "Cybersecurity framework manufacturing profile," NIST, Gaithersburg, MD, NISTIR 8183, 2017 [Online]. Available: https://www.nist.gov/publications/cybersecurity-framework-manufacturing-profile.

[3] K. Stouffer et al., "Guide to industrial control systems (ICS) security," NIST, Gaithersburg, MD, SP 800-82 Revision 2, May 2015 [Online]. Available: https://www.nist.gov/publications/guide-industrial-control-systems-ics-security.

[4] *Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program*, ANSI/ISA Standard 62443-2-1, 2009.

[5] *Security for Industrial Automation and Control Systems – Part 2-3: Patch Management in the IACS Environment*, ANSI/ISA Standard 62443-2-3, 2015.

[6] Joint Task Force Transformation Initiative, "Security and privacy controls for federal information systems and organizations," NIST, Gaithersburg, MD, SP 800-53 Revision 4, Apr. 2013.

[7] R. Candell, T. Zimmerman, and K. Stouffer, "An industrial control system cybersecurity performance testbed," NIST, Gaithersburg, MD, NISTIR 8089, Nov. 2015 [Online]. Available: https://www.nist.gov/publications/industrial-control-system-cybersecurity-performance-testbed.

[8] T. Zimmerman, "Metrics and key performance indicators for robotic cybersecurity performance analysis," NIST, Gaithersburg, MD, NISTIR 8177, Apr. 2017 [Online]. Available: https://www.nist.gov/publications/metrics-and-key-performance-indicators-robotic-cybersecurity-performance-analysis.

[9] C. Tang, "Key performance indicators for process control system cybersecurity performance analysis," NIST, Gaithersburg, MD, NISTIR 8188, Aug. 2017 [Online]. Available: https://www.nist.gov/publications/key-performance-indicators-process-control-system-cybersecurity-performance-analysis.

[10] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245-255, 1993.

[11] N. L. Ricker. (2015, Jan. 23). *Tennessee Eastman challenge archive* [Online]. Available: https://depts.washington.edu/control/LARRY/TE/download.html.

[12] S. Tatham. (1999). *Download PuTTY* [Online]. Available: https://www.putty.org/.

[13] P. Biondi. (2008). *Scapy* [Online]. Available: https://scapy.net/.

[14] G. van Rossum. (1990). *Python* [Online]. Available: https://www.python.org/.

[15] T. Kosse. (2001). *FileZilla* [Online]. Available: https://filezilla-project.org/.

[16] G. Lyon. (1997). *Nmap.org* [Online]. Available: https://nmap.org/.

[17] R. McCool. (1995). *Apache HTTP server project* [Online]. Available: https://httpd.apache.org/.

[18] A. Rudakov. (n.d.). *File modbus-discover* [Online]. Available: https://nmap.org/nsedoc/scripts/modbus-discover.html.

[19] PHP Group. (2001). *PHP: Hypertext preprocessor* [Online]. Available: http://www.php.net/.

[20] G. Woltman. (1996). *Free Mersenne prime search software: Prime95 version 29.4 build 7* [Online]. Available: https://www.mersenne.org/download/.

[21] Wikimedia Foundation, Inc. (1999). *OpenSSH* [Online]. Available: https://en.wikipedia.org/wiki/OpenSSH.