HAMIDS: Hierarchical Monitoring Intrusion Detection System for Industrial Control Systems

Hamid Reza Ghaeini
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore, 487372
Ghaeini@acm.org

Nils Ole Tippenhauer
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore, 487372
Nils_Tippenhauer@sutd.edu.sg

ABSTRACT

In this paper, we propose a hierarchical monitoring intrusion detection system (HAMIDS) for industrial control systems (ICS). The HAMIDS framework detects the anomalies in both level 0 and level 1 of an industrial control plant. In addition, the framework aggregates the cyber-physical process data in one point for further analysis as part of the intrusion detection process. The novelty of this framework is its ability to detect anomalies that have a distributed impact on the cyber-physical process. The performance of the proposed framework evaluated as part of SWaT security showdown (S3) in which six international teams were invited to test the framework in a real industrial control system. The proposed framework outperformed other proposed academic IDS in term of detection of ICS threats during the S3 event, which was held from July 25-29, 2016 at Singapore University of Technology and Design.

Keywords

SCADA, Intrusion detection, EtherNet/IP

1. INTRODUCTION

The term Industrial control systems (ICS) covers three major types of control systems used in industrial systems. These systems are SCADA (Supervisory Control And Data Acquisition), DCS (Distributed Control Systems), and PLC (Programmable Logic Controller). Critical infrastructures are vital systems that are controlled by these industrial control systems.

SCADA technologies are designed for monitoring and controlling large scale distributed industrial systems like water treatment plants, power grids, and other critical infrastructure systems. The upcoming *Industry 4.0* will lead to SCADA (Supervisory Control And Data Acquisition) industrial systems that introduce aspects of the Internet of Things [26]. In those systems, industrial control protocols will be used along with technologies such as VLANs, VPN, and HTTP in the industrial control system.

The SCADA system simplifies management of remote equipment using coded signals in the communication network. For example, the industrial EtherNet/IP protocol allows the use of the CIP (Common Industrial Protocol) over TCP, IP and standard Ethernet cabling. Connecting those controlled systems to the Internet is one of the aspects of IoT (Internet of Things) in the Industry 4.0. But exposing such a systems to the Internet would increase the chance of top security threats from public networks like Internet [18]. Security vulnerability of such industrial systems could cause severe damage, e.g. in the context of public infrastructures.

Intrusion detection systems (IDS) are computing components that monitor the system or network to detect activities that compromise the confidentiality, integrity or availability of a resource. The major types of intrusion detection system based on detection mechanism are:

- Signature-based IDS
- Anomaly-based IDS
- Specification-based IDS

The signature-based IDS look at the specific patterns inside the network traffic to find the intrusion. Anomaly-based IDS mostly use learning algorithms to find a model of trustworthy activity and classify the traffic based on this model. Specification-based IDS base on a formal specification of the intended system operations to detect unwanted traffic and related states. Current network-based intrusion detection systems in ICS are often designed to detect particular system related attacks in the site manufacturing and area supervisory control segments of the networks. As IDS are often adapted from general "office network" security domains, those IDS do not perform deep packet inspection of industrial SCADA protocols [13], and instead rely on application layer data collected from other software like SCADA server or HMI. As a result, the IDS has to trust the input provided by those application. In addition, the IDS detection speed is related to the speed of data collection.

The newer intrusion detection systems need explicit knowledge of the ICS application and the process that is controlled by the network, and need to be able to observe traffic in all distributed segments of the ICS network. In particular, the traffic in lower levels such as the process (or fieldbus) networks have to be monitored as well [29]. Unlike [29], we currently do not focus on stateful physical process simulation to detect attacks, instead HAMIDS focuses on a comprehensive detection of network-based malicious traffic.

Bro [22] is an open source IDS framework that is used widely for intrusion detection inside industrial communication networks. While Bro supports a broad range of protocols, its support for industrial control protocols is still limited.

In this work, we propose the HAMIDS framework (based on Bro), designed to extract the real-time industrial network traffic. In addition, HAMIDS is designed to be able to detect important industrial control treats directly from the network traffic. Another important feature of this framework is the compatibility with other big data frameworks. It uses Hadoop for further industrial control network traffic analysis. Detection of new advanced persistent

CPS-SPC'16, October 28 2016, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2994487.2994492

threats (APT) like Stuxnet need an in-depth analysis of the whole industrial process events and the HAMIDS framework provides detailed data about the industrial process events like detected network threats, detailed connection history, EtherNet/IP commands, etc.

HAMIDS achieves detailed insight into the plant due to its use of a hierarchical monitoring system. By leveraging multiple distributed IDS sensors on different layers in the network, the IDS can fully observe the traffic in disjoint individual networks (e.g., Fieldbus networks, plant networks). To implement HAMIDS for our ICS plant, we were required to implement support for EtherNet/IP by limited reverse engineering. Leveraging that EtherNet/IP support, we are able to demonstrate the efficacy of our solution with several attacks.

We summarize our contributions as follows:

- We propose HAMIDS, an hierarchical monitoring intrusion detection system for ICS.
- We implement HAMIDS based on Bro in a water treatment ICS.
- Our implementation of HAMIDS allows inspection of selected SCADA protocols EtherNet/IP (as Bro extension).
- We demonstrate the effectiveness of our framework by detecting a wide range of attacks in different ICS levels in a real ICS by invited team from 6 academia and industry.

The remainder of this paper organized as follows. Section 2 discusses the background of the work. Section 3 introduces and explains the proposed framework. Intrusion detection and discussion presented in Section 4. We discuss the related work in Section 5. Finally, we conclude the paper in Section 6.

2. BACKGROUND

In this section, different aspects of intrusion detection in industrial control systems are discussed. First, the SWaT plant will be introduced. Then, details on the Ethernet/IP protocol and its features will be presented.

2.1 SWaT ICS

The Secure Water Treatment (SWaT) industrial control system (see [16]) is a facility for research in the area of cyber security at SUTD in Singapore (see Figure 1 and Figure 2). The SWaT plant is built with the goal of advancing the security of critical infrastructure and particularly Cyber-Physical Systems (CPS) such as water treatment systems, power grids, and oil and natural gas refinery. The SWaT plant contains a modern six-stage process. The water treatment system receives raw water in the first process, and it adds necessary chemicals to it. Then, the ultrafiltration system will filter the water and pass it to dechlorination using UV lamps. Afterward, a reverse osmosis system will clean the water. A backwash process is able to clean the UF membranes by using the water produced by reverse osmosis.

The SWaT plant consists of major SCADA components such as communication infrastructure, both wired and wireless telemetry system, Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) workstation, Human Machine Interfaces (HMIs), Historian, and Remote Terminal Unit (RTU). The historian system records Boolean events and sensor data from different parts of the plant for subsequent analysis. Most of the SWaT components equipped with both wired and wireless communication telemetry system and the wired links are redundant to increase the robustness of the system against link failures.

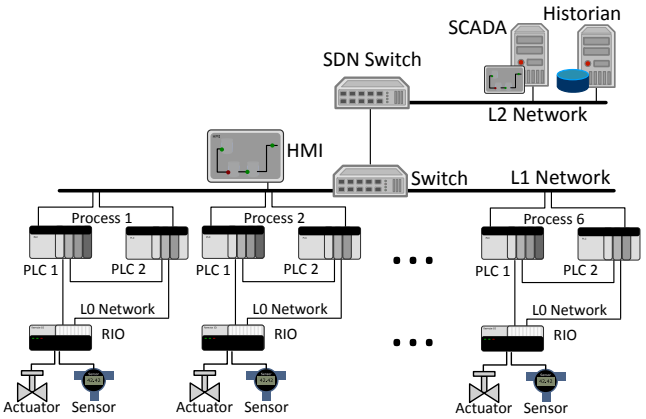

Figure 1: Abstract structure of the SWaT industrial control system network

Figure 2: SWaT testbed used for experimental validation

The sensors and actuators are connected to PLCs for aggregating sensor values and producing control commands. The IoT leads us to modern computer network technologies like SDN, OS and network virtualization, and advanced connectivity technologies. Nevertheless, there are many concerns about security, reliability, and interoperability of such industrial computer networks [14]. In contrast to home area networks (HAN), such computer networks need additional devices to provide some degree of security, reliability, and interoperability. Examples of such additional devices are Device Level Ring (DLR) networks, and industrial switches. In addition, industrial control system networks require custom protocols such as Ethernet/IP [19].

As in many industrial control systems, the SWaT industrial network consists of four compartments (also called levels) [20]. The main aim of this architecture is to clearly separate the industrial control network from other parts of the industrial business to reduce the risk of attacks reaching critical processes through vulnerabilities in other business networks. Following the Purdue model for control hierarchy [7] the SWaT plant consist of the following levels:

1. Site Manufacturing Operations and Control (Level 3/L3): This level designed for the management of the processes in the industrial control systems. The most important parts of this level are Historian, SCADA workstation, network management devices, and engineering workstation.
2. Area Supervisory Control (Level 2/L2): The area supervisory control contains manufacturing operations equipment. This level typically have HMI devices, control workstation, and alarm system.
3. Basic Control (Level 1/L1): This level contains process control equipment that read the sensor values, compute the desired information and send the data to a destination. This

level typically have DCS, PLC, and RTU. These devices have their own vendor provided operating system and software. Also, these devices are vulnerable to industrial control specific vulnerabilities.

4. Process (Level 0/L0): This level contains sensors and instrumentation elements which are controlled by level 1 devices. This level typically has sensors and actuators.

2.2 EtherNet/IP

EtherNet/IP is an industrial network protocol that allows using the Common Industrial Protocol (CIP) over standard Ethernet and IP. According to the global ICS scanner *Shodan ICS radar* (https://ics-radar.shodan.io), EtherNet/IP is one of the most used ICS protocols in the world. In many cases, industrial control system networks are directly connected to the Internet. EtherNet/IP standardized by ODVA [21], Inc., to provide better connectivity of ICS and modern Ethernet technologies. Also, the newer version of Ethernet/IP provides the option of featuring both CIP and Ethernet/IP in ICS.

Ethernet/IP is an application-layer protocol that allows using the Common Industrial Protocol services with the principal elements of IEEE 802.3 protocol. EtherNet/IP used in different Ethernet industrial modules from vendors like Allen Bradley, Schneider Electric, and Omron. Some of the CIP services that could integrate into EtherNet/IP are request individual services from the target devices like PLC or HMI, set configuration options, and downloading to/from a PLC. The Rugged Media Converter (RMC) is a passive EtherNet/IP device that supports some standard CIP objects. These CIP objects can be accessed through CIP service requests and these objects include various services and attributes.

CIP is a communications protocol uses a request-response model. This protocol is for transferring automation data between two devices where a device could send a request to the other device (for reading a value read from an industrial component) and the other device then reply the request (by providing the requested value or an error code). In the CIP Protocol, every network device contains a series of objects. Ethernet/IP used for handling communication sessions with or without handshaking. There are two types of network connections in Ethernet/IP: Explicit and Implicit. Ethernet/IP features TCP/IP client/server transactions to send explicit messages. But in implicit messaging the data field only contains real-time I/O data without any information about the protocol. Also multi-casting EtherNet/IP features the TCP/UDP model with message multicasting ability.

2.3 Bro

The Bro IDS [23] is developed by the National Center for Supercomputing Applications and the International Computer Science Institute. Bro is an open source scripting platform which designed for security monitoring and network traffic analysis. The powerful event handling scripting language of Bro made the strong incentive for researchers from academia and industry to adopt this platform in many security monitoring and analysis applications [13]. By default, Bro has support for a range of network protocols, and can provide detailed log files based on online analysis of the traffic. These features make Bro a popular choice among the tools for network traffic monitoring and analysis.

In addition, Bro features useful functionality such as deep protocol decoding, traffic logging, event handling and notification which are paramount in security monitoring and analysis. Bro provides deep packet inspection with a detailed support of most well-known and some not well-known Internet protocols, and it can be used to generate event logs from protocols of the Internet protocol stack.

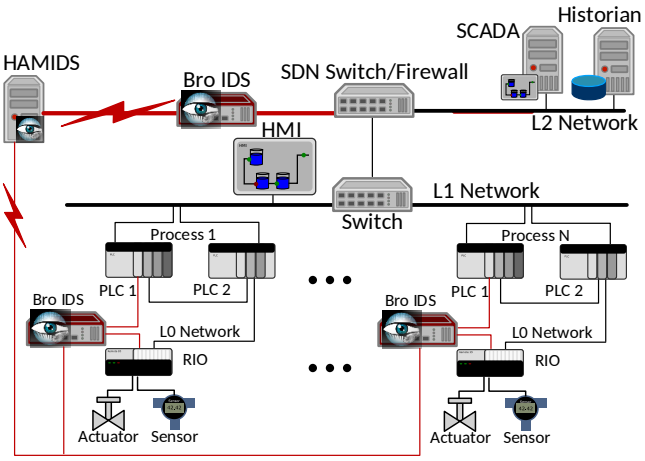

Figure 3: HAMIDS architecture in the SWaT network

One of the interesting features of Bro is its dynamic protocol detection, that is able to detect protocols that are running on non-standard ports.

Unfortunately, a range of newer ICS protocols are not yet well-supported by Bro, in particular, the EtherNet/IP and CIP protocol. In addition, no general protocol parser for the Ethernet/IP protocol is available as open source. Nevertheless, Bro provides a scripting platform to define custom protocols, and it allows to parse and decode that protocol by custom scripts. Using such script, it is possible to define and register specific event handlers to detect malicious actions inside the ICS. In this framework, we will use these Bro capabilities to implement the distributed IDS in the HAMIDS framework.

3. HIERARCHICAL MONITORING INTRUSION DETECTION

We now propose HAMIDS, our HierArchical Monitoring Intrusion Detection System design, based on extensions for Bro to allow distributed detection in a hierarchical SCADA system with several disconnected network segments. Our scheme uses a cluster of SCADA intrusion detection systems that support CIP and EtherNet/IP traffic parsing. These specialized traffic handlers will trigger on traffic using ports relating to both CIP and EtherNet/IP, and then pass the results to the Bro cluster manager. As such, the hierarchical aspect of our system refers to the detection in several layers and segments of the ICS network, aggregated by a cluster of Bro instances connected to a cluster manager.

3.1 Problem Statement

The problems that intrusion detection systems for industrial control systems face are:

- The attacker is assumed to be familiar with the industrial protocols used, and can leverage vulnerabilities in those protocols to obtain confidential data, manipulate devices, or even crash devices connected to the network.
- The attacker could interact with sensors and actuators through local process/fieldbus communications, that cannot be observed on higher network levels.
- Current implementation of open source IDS such as Bro do not support SCADA protocols such as EtherNet/IP.

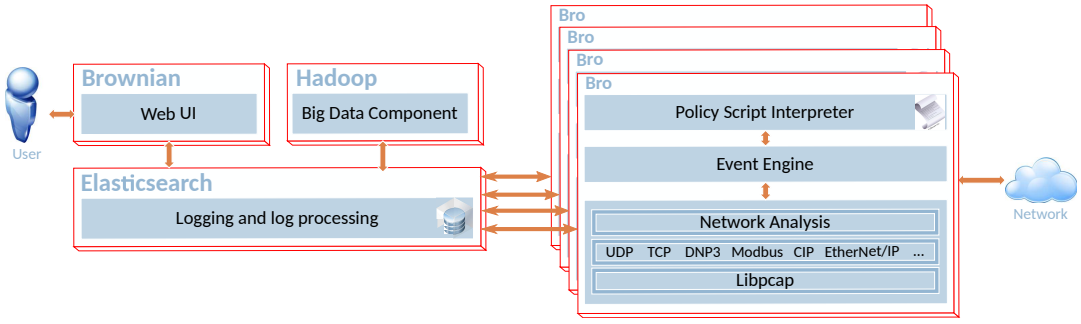

Figure 4: HAMIDS Architecture

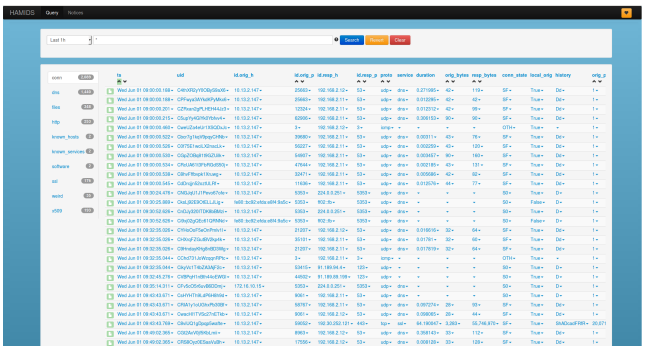

Figure 5: HAMIDS web interface (based on [11]). In this example, the web interface shows a view of currently ongoing communication in the network.

- Current implementation of IDS are not able to detect insider attacks that have impact PLC and fieldbus systems.

In HAMIDS, our proposed intrusion detection framework for industrial control systems, different aspects of a successful network attack based on SCADA protocols are detected and logged using an efficient and effective distributed system. The main features of the proposed intrusion detection framework are:

- Our IDS implementation has industrial specific intrusion detection and could be deployed in both level 1 and level 0.
- In contrast to other proposed IDS for ICS, we offer a hierarchical online monitoring IDS.
- Our IDS applies leverage deep packet inspection of the industrial application layer protocol, in particular, the CIP and EtherNet/IP protocol.
- Our IDS implementation is scalable and could distribute in large scale industries with logging and log processing features.
- The logging system can be connected to other big data analysis tools for real-time intrusion detection in a large scale network with learning techniques.

3.2 Hierarchical monitoring

Figure 3 shows the proposed HAMIDS framework in the context of our SWaT plant network. The network architecture of the distributed industrial IDS is constructed to comply to ISA-99, a security standard designed for industrial automation and control systems. This standard suggests three concepts of Zones, Conduits, and Layers. This standard offers traffic control and level segmenting in the control and communication network. A conduit controls the flow of different zones which provides secure communication between the zones.

In HAMIDS, several Bro IDS are located at different network levels (see Figure 3). Each Bro instance can obtain a copy of the traffic by a) a mirroring port on a switch, b) a passive tap (e.g. the network throwing star [24]), or c) by bridging the the network connection actively. In our specific case, we chose the following configuration: The Bro instance on the Layer 2 network obtains a copy of the traffic from a mirroring port on the central Layer 3 switch. The other Bro instances have access to traffic at the Level 0 network (several disconnected device level rings).

In [20], the authors used a firewall between the Level 3 network and lower levels. Such an architecture does not allow to detect or prevent attacks that target devices in the Level 0 network, as their traffic never passes the central firewall [29]. Our proposed architecture allows detecting such attack by extending the analysis to the broadcast area of all ICS levels (including fieldbus networks between sensors and PLCs).

3.3 Implementation

Figure 4 shows the HAMIDS architecture. As described before, the main component of this framework is Bro. The Bro IDS sensors are in the device level ring (DLR) between the PLCs and RIOs, and bridge the ring. As such, the IDS is able to capture all traffic sent and received from the PLC in the L0 ring. In addition, one Bro IDS sensor records the traffic from a mirroring port of the L1 network switch. Combining detection on several network segments in the ICS is an important feature of the HAMIDS.

The Bro IDS sensors will parse received packets and generate the logs of different network protocols like TCP, UDP, ARP, CIP or EtherNet/IP. These logs will aggregate at the central data processing unit of the HAMIDS framework by Elasticsearch [8]. Elasticsearch is a distributed full-text search engine that could work with JSON with RESTful APIs. The Elasticsearch store the logs from the Bro IDS sensors. There is two way of access to the stored logs. The first method is the web user interface that presents the raw logs without data processing (leveraging Brownian [11]). The second way of access is designed for big data processing that using machine learning techniques for classification of the network traffic. In this paper, our focus is on the IDS components and the results are based on detection of treats from the raw logs. Figure 5 shows the HAMIDS web interface.

In addition the individual analysis of the traffic within a network segment, our system also collects data aggregated by all IDS instances in the central logging and log processing unit by Elasticsearch. The connections between the logging unit and Bro IDS

sensors are secured by secure shell (SSH) tunnels. In order to capture the Device Level Ring (DLR) traffic, we leverage a Raspberry PI 3 with two NICs to create a network bridge with the capability of DLR packet capturing. Figure 6) presents the placement of the Raspberry PI device near the PLC.

We run the HAMIDS framework and Bro IDS sensor as virtual machines on a central computing host. By using the high-speed packet capture, filtering and analysis tool (PF_RING) [4] we are capable of increasing the number of Bro IDS sensors in the virtual machine to provide packet inspection of network traffic depending on the target ICS traffic requirements. The HAMIDS framework collects the logs from the distributed IDS sensors and provides a comprehensive logging of network traffic events and alerts from both Level 1 to Level 0.

3.4 Intrusion Detection

EtherNet/IP and CIP traffic use port 44818 (both TCP and UDP on the transport layer), and port 2222 (UDP). To capture that traffic, our IDS instances analyze all traffic originating or targeting those ports. Whenever an IDS receives a packet on the specified ports, it starts handling and extracting the payload data fields from the EtherNet/IP or CIP packets. Then, it initiates the event handler engine to do further analysis. For example, the packet payload types are extracted from the EtherNet/IP or CIP traffic as one feature for the intrusion detection.

When a new network device connects to the ICS network, it starts to exchange ARP traffic to map other IP addresses to MAC addresses, and announce its own MAC address through ARP replies when requested. For our distributed IDS, we propose to use an ARP database that includes information about all legitimate devices on the network, and details the respectively used switch port, Medium Access Control (MAC) addresses, and IP address corresponding to the MAC. To achieve that, the proposed IDS captures both the ARP requests and ARP replies and verifies their consistency based on the ARP database. If the ARP requests or ARP replies violate the consistency, our IDS generates a corresponding log entry, and then it passes the possible attack information for further security analysis investigation, particularly ICS related security analysis at the application layer (EtherNet/IP and CIP) payloads of the suspicious device in the ICS.

In addition, there are several industrial devices specific vulnerabilities that the implemented IDS discovers by inspection of CIP and EtherNet/IP payload. When the IDS detects suspected malicious data in the payload of a packet, it creates a notification to the logging system of the Bro cluster manager. The total size of the line of codes for our IDS implementation is about 2210 lines of codes.

Figure 6: Raspberry PI placement on top of a PLC

4. EXPERIMENTAL VALIDATION

In the experimental validation, our assumption is that an attacker can compromise network devices, but it can not change the physical architecture of the industrial network. We validated our framework in the SWaT plant shown in Figure 2. For all experiments, the Bro IDS sensors are running on a Raspberry PI 3B device with a quad-core CPU with 1 GB RAM. We defined three scenarios of major CPS attack types to evaluate the distributed intrusion detection system. The first scenario is the SCADA-specific attacks. In such attacks, the attacker tries to prevent routine use of industrial control system. The second scenario is Man-in-the-Middle (MitM) attack. In such attacks, the attacker sits between sender and receiver and could actively monitor, capture or control the communication between sender and receiver in the industrial control system. The third scenario is the results of SWaT security showdown events that six international hacking teams invited to test the system under a realistic attack situation. Table 1 shows the traffic characteristics in SWaT plant. The rest of this part describes those three scenarios. For each of the individual attacks in the first and second scenarios, we test the framework five times in our plant to determine the false and true positive rate of the system. In the third scenario, we will present the average false and true positive rates of the framework.

4.1 SCADA-Specific Attacks

The PLC devices used in SWaT plant are the Allen-Bradley ControlLogix family, and feature both CPU and Ethernet modules. Although there are some efforts to secure such modules by vendors such as Rockwell Automation and Schneider Electrics, new vulnerabilities in modules and protocols are constantly discovered.

For example, there are some well-known remote command vulnerabilities for such devices. In particular, one of the most significant remote command vulnerability is the multi-CIP command attack introduced in the Metasploit framework [17]. That attack tries to stop or crash the CPU and Ethernet controller, by targeting the CPU and Ethernet modules with malicious payloads. Two of these SCADA-specific attacks highlight the CPU vulnerability issues.

- Stop CPU—this attack will stop the ControlLogix
- Crash CPU—this attack will crash the PLC CPU

Some commands in the EtherNet/IP protocol might cause security issues with the ControlLogix PLC. In particular, we observed that none of these commands are authenticated in our testbed. That problem could be harmful to most of EtherNet/IP devices including Rockwell Automation, Schneider, Wago, and others manufacturing systems. The two other SCADA-specific attacks are due to Ethernet controller errors in the ControlLogix.

- Crash Ethernet—this attack will Crash the Ethernet Controller
- Reboot Ethernet—this attack will cause a temporary outage of the ControlLogix Ethernet interface

Table 1: Traffic characteristics of SWaT plant

	Traffic feature	Value
1	Throughput	7650 Kbps
2	Average packet size	106.5 Bytes
3	Overall Packet per second	8990.7
4	UDP packet percentage	25.4
5	TCP packet percentage	74.5
6	EtherNet/IP packet percentage	36.7

Table 2 shows the attacks used in our experiment. Figure 7 shows the attacker placement in our experimental validation with SCADA-specific attack. Table 3 shows the results of our experimental validation. The results indicate that Bro consumes about 25 percent of CPU resources and about 8.8 percent of memory resources which is a small fraction of computation resources. The proposed framework could detect 100 percent of the SCADA-specific attacks and it has zero percent false positive rate.

4.2 General Network Attacks

The communication between SCADA systems and PLCs is not encrypted or authenticated. Hence, an adversary might try to listen or even change the communication between sender and receiver. In our implemented general network attacks, an adversary uses ARP poisoning to take control of the communication between two or more SCADA devices. In the local cache of IDS, the ip_to_mac and mac_to_port dictionaries are stored. Once the adversary took the communication control, she will change values inside the EtherNet/IP packets to take the physical process under her control.

To detect such attacks, the implemented IDS will analyze the ARP traffic and compare the traffic with its own ARP cache. The learner mechanism of ARP cache consist of two lists of hosts with corresponding MAC addresses. These two lists are the whitelist and blacklist.

When the adversary attacks the local ARP cache, our IDS will generate an alarm in the log files and change the status of the host from white to black. Based on the implemented EtherNet/IP and ARP handling scripts, it will pass that logs for further security analysis. In particular, our system currently would raise an alarm if components are changed for legitimate reasons (e.g. upgrades), in that case it is up to the operator to verify and dismiss the alert message.

We evaluate our distributed IDS against both ARP poising and MitM attacks. These attacks are implemented by some Python scripts featuring Scapy module. Another general types of attacks used in our experimental evaluation are IP scanning and Port scanning. The main aim of these network attacks are to discover active devices inside the network and possible vulnerabilities based on open ports. So the attacker spreads various discovery requests to the network.

We summarize the attacks used in our experiment in Table 4. Figure 8 shows the attacker placement in our experimental validation with active attacks. The placement for the passive attacker are summarized by Figure 7.

Table 5 shows the results of our experimental validation. Similar to the first experiment, the results indicate that Bro consumes about 25 percent of CPU resources and about 8.8 percent of memory resources which is a small fraction of computation resources. The proposed framework was able to detect the general attacks in all test cases with 100 percent accuracy. In general, we did not see any false positives. The only exception is occurring after the successful detection of an ARP poisoning attack.

Table 2: SCADA-specific Attacks considered in this work

	Attack Name	Detail	Type
1	Multi CIP	CPU Stop	Active
2	Multi CIP	CPU Crash	Active
3	Multi CIP	Reboot Ethernet	Active
4	Multi CIP	Crash Ethernet	Active

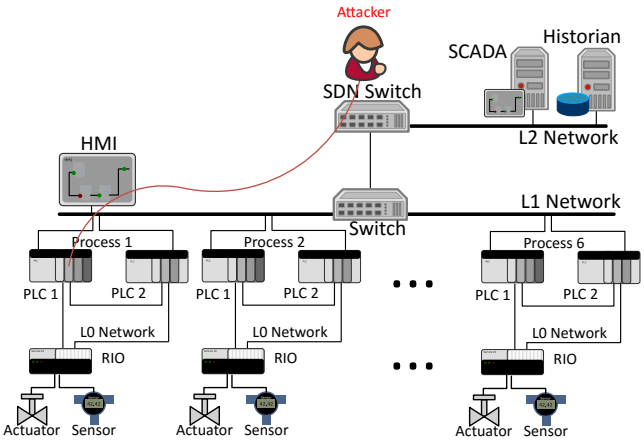

Figure 7: Attacker placement for our experimental validation with port scanning and SCADA specific attacks

Table 3: SCADA-specific Attacks detection properties

	Attack Name	CPU (%)	Mem. (%)
1	CPU Stop	25.85	8.8
2	CPU Crash	26.36	8.83
3	Reboot Ethernet	26.04	8.8
4	Crash Ethernet	25.03	8.855

Table 4: General Attacks considered in this work

	Attack Name	Detail	Type
1	PLC 1 MitM	R/W Attr.	Active
2	PLC 2 MitM	R/W Attr.	Active
3	HMI MitM	R/W Attr.	Active
4	SCADA MitM	R/W Attr.	Active
5	ARP poisoning	-	Active
6	SYN flooding	-	Active
7	DHCP attack	-	Active
8	IP scanning	-	Passive
9	Port scanning	-	Passive

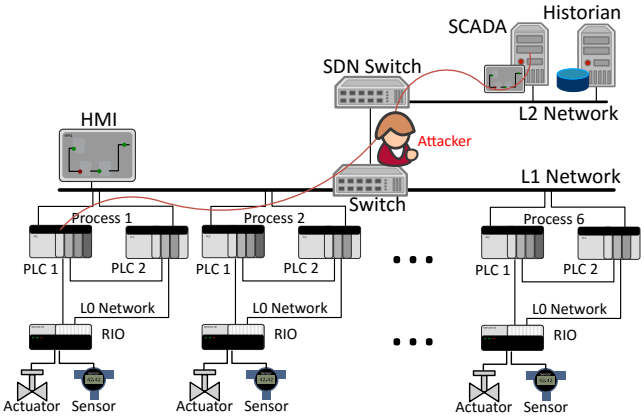

Figure 8: Attacker placement for our experimental validation with general network attack and ARP poising

Table 5: General Network Attacks Detection Properties

	Attack Name	CPU (%)	Mem. (%)
1	PLC 1 MitM	24.77	8.8
2	PLC 2 MitM	25.45	8.8
3	HMI MitM	25.2	8.8
4	SCADA MitM	24.7	8.89
5	ARP poisoning	25.1	8.88
6	SYN flooding	25.6	8.8
7	DHCP attack	25.6	8.8
8	IP scanning	25.1	8.89
9	Port scanning	24.85	8.8

After an ARP poisoning attack has been discovered, any ARP packet sent by the attacker is classified as attack packet. That results in potentially harmless traffic of the attacker being classified as attack traffic, which we count as false positive. Overall, less than 1 percent of all test-cases showed that artifact.

4.3 Validation by Invited Hackers

We invited six international teams to attack the SWaT plant as part of the SWaT security showdown (S3) event, which was held from July 25-29, 2016 at Singapore University of Technology and Design.

The attacker teams were asked to choose an attacker profile from one of three options (cyber criminal, insider, strong). The cyber criminal and strong attacker were not limited to the software they were using, while the insider and strong attacker was are allowed to access the control software used in the SWaT plant, and use the HMI to manipulate the process. The intended target for the attack were as following:

- Valves (open or close valve)
- Pumps (start or stop pump, change pump speed)
- Pressure (raise or lower pressure in a pipe segment)
- Tank fill level (change physical level)
- Chemical dosing (change amount of dosing)
- Historian (manipulate/influence sensor data received)
- HMI/SCADA (manipulate/influence sensor data received)
- PLC (manipulate/influence sensor data received)
- Display (manipulate/influence sensor data received)

The HAMIDS framework was deployed in the L1 and L0 networks of the SWaT plant, and was able to reliably detect network-based attacks targeting to HMI/SCADA, PLC, and Historian. The detected attacks included DoS attacks using Ettercap targeting HMI, SCADA, and PLC (at level 0); ARP Man-in-the-Middle attacks on HMI and PLC components, and DoS attacks using SYN-flooding EtherNet/IP port of a PLC. Our implementation of the proposed HAMIDS framework detected all of those attacks while having zero percent false positive rate.

As expected, the HAMIDS framework was not able to detect insider attacks in which the attacker was using the HMI or SCADA to directly send malicious control messages, or change setpoints of the control logic (related to the physics of the system, e.g., chemical dosing attacks) without having an impact on the network traffic behavior at both level 1 and level 0. In our future work we plan to investigate promising venues such as machine learning techniques to detect attacks targeting the physical process by changing the values in the ICS software, leveraging the big data processing component of HAMIDS framework.

5. RELATED WORK

In the context of CPS and ICS, a number of recent works were focused on anomaly-based intrusion detection methods. While signature based intrusion detection methods have been shown to perform well for a particular set of common attacks, their use in detecting novel attacks is limited. SNORT [25] is a signature-based intrusion detection system for real-time network traffic monitoring and analysis. Machine learning techniques became popular with industrial control security researchers and works such as [30] and [15] extract machine learning features from the network traffic to detect abnormal traffic in SCADA systems. Nevertheless, there are two major issues in these techniques: first the detection ratio is related to the learning data set and false positive rate is high, second there are some cases that these techniques could not efficiently detect the malicious traffic [13, 19].

5.1 ICS-specific Approaches

Example ICS protocols supported by Intrusion detection systems include Modbus, and DNP3. The authors of [6] extract some features from Modbus protocol and use it for intrusion detection. The work in [2] detects intrusions based on the specification of advanced metering infrastructure (AMI). In addition, the work in [13] uses a DNP3 specification-based model for intrusion detection in SCADA system. We note that DNP3 is also supported by Bro, and thus could be supported by HAMIDS. In [29], the authors present a Fieldbus-layer single detection system that monitors the EtherNet/IP traffic, extracts control commands and sensor values, and leverages stateful physical process simulation to detect a mismatch between the predicted and observed reaction of the system. The work is focused on understanding the physical process-related traffic and is not considering more general network based attacks.

The authors of [28] proposed an extension Bro for the IEC 60870-5-104 protocol of SCADA. They evaluated their extension by a set of experiments using a captured traffic and reproducing attack traffics by Scapy module. The results of this paper show that the extended version of Bro is capable of detection of port scanning attacks, ARP spoofing, and Man-in-the-Middle attacks. Unlike to this paper our proposed framework tested in a real network traffic and all of the evaluation process done by an online attacker and some online IDS sensors. In addition, in our proposed framework the sensors are capable of detecting SCADA-specific attacks. Another advantage of our proposed framework is the distributed architecture of the framework that makes it possible to detect attacks at different levels of the industrial control systems, both of level 1 and level 0. Table 6 shows a comparison between our proposed framework, SNORT and [28] in terms of detection of a variant type of attacks in ICS.

As the Table 6 summarizes, the HAMIDS framework could detect several ICS attacks which other IDS are not able to detect. To our best knowledge, the most complete proposed academic IDS for SCADA systems and EtherNet/IP are SNORT [25] and [28]. In [28], the authors are not able to detect EtherNet/IP-specific attacks because their setup does not parse the CIP and EtherNet/IP packets. Although both our proposed framework and SNORT are able to detect EtherNet/IP-based attacks by using signature-based detection mechanism, lower level attacks that target PLCs cannot be detected by SNORT and the system proposed in [28] due to the lack of information on the network in the fieldbus layer. For that

reason, our proposed hierarchical monitoring framework observes the SCADA system both at the L0 and L1 of the ICS network. As result, the proposed framework outperformed other proposed academic IDS in term of detection of ICS threats.

Newer commercial systems such as SilentDefense from Security Matters [27] seem to take a similar approach to HAMIDS in the sense that they provide industrial-protocol aware IDS. In addition to processing industrial protocols, attacks using ARP and general TCP can be detected by such frameworks. To the best of our knowledge, there are no academic works with details on those systems available.

5.2 Smart Grid

In [3], the authors proposed an intrusion detection design based on the concept of Critical State Analysis and State Proximity. In that design, they provide additional detection mechanisms that could be helpful alongside with traditional IDS. The authors of [9] proposed a design approach leveraging filtering systems based on state analysis of the system monitored. In that design, the authors use a state-based mechanism to detect attacks that use a sequence of multiple SCADA commands. The designs of [3] and [9] are designed in the context of smart grid networks. In contrast, our HAMIDS framework was designed to be more generic, and could support a wide range of SCADA protocols (in addition to the currently supported EtherNet/IP protocol). In addition, the HAMIDS framework includes functions of most of traditional IDS (as it is based on Bro).

The authors of [10] proposed a systematic approach to configuring the anomaly detection engines for detection of attacks that violate connection patterns specific to ICS. The authors of [12] proposed a method to detect cyber attacks targeting values sent to PLC. Their approach is based on Gaussian mixture model that build a cluster based on sensor values. In addition, the authors use a cluster assessment technique known as the silhouette to classify various operational states. Both of these methods are sitting at the communication layer between SCADA and PLC and attempt to predict the sensor values based on their internal models.

The authors of [1] presented the functional features of existing anomaly-based methods to improve the context-awareness in control of Smart Grid applications. In addition, the authors of [5] offer a study for determination of the IDS requirements by using the Non-Functional Requirements (NFR) Framework. Both of the works demonstrate the applicability of IDS in the context of the smart grid applications. Our proposed HAMIDS framework is designed as core packet inspection framework for anomaly-based IDS to provide detailed information on traffic in L0 and L1 of ICS systems. HAMIDS structure can be used as a main packet inspection component for data analysis, control theory, and system state researchers in different ICS applications like water treatment and distribution systems, smart grids, and oil and gas industries.

6. CONCLUSIONS

In this work, we proposed HAMIDS, a hierarchical monitoring SCADA intrusion detection system for industrial control systems. We implemented the framework by extending the Bro open source IDS with added support for EtherNet/IP and CIP protocols. By using IDS sensors in different network segments, HAMIDS is able to detect attack events in a hierarchy of network levels (e.g., the fieldbus connecting the sensors, actuators, and PLCs).

We employed HAMIDS in the SWaT plant that emulates a water treatment system. The results of our analysis of the implemented IDS proves the soundness and effectiveness of detection of the SCADA-specific threats in a realistic ICS. In particular, we were able to detect most classes of attacks considered in all of our test cases. The only false positives we observed were caused by the IDS flagging all traffic by an identified ARP spoofer as attack, even if the content was non-malicious. Our framework was also able to detect level 0 attacks such as *CPU crash* and *Reboot Ethernet* attacks that previous work was not able to detect it at level 0.

Table 6: Comparison between SNORT, Udd *et al* [28] and HAMIDS, the ● denotes that the IDS will detect the attack at 100% true positive rate, - denotes insufficient information about the detection, and ○ denotes that the IDS could not detect such attacks

	Udd et al	SNORT	HAMIDS
IP scanning	●	●	●
Port scanning	●	●	●
ARP poisoning	●	●	●
DHCP attack	●	●	●
SYN flooding	●	●	●
CPU Stop	○	●	●
CPU Crash	○	●	●
Reboot Ethernet	○	●	●
Crash Ethernet	○	●	●
PLC 1 - DoS	-	○	●
PLC 2 - DoS	-	○	●
HMI - Crash	-	○	●
SCADA - Crash	-	○	●
HMI/SCADA Insider	-	○	○

The performance of the proposed framework evaluated as part of SWaT security showdown (S3) in which six international teams were invited to test the framework in a real industrial control system. The proposed framework outperformed other proposed academic IDS in term of detection of ICS threats during the S3 event, which was held from July 25-29, 2016 at Singapore University of Technology and Design.

Our current HAMIDS implementation is already able to extract sensor values from EtherNet/IP traffic in the fieldbus, but we are currently not raising alarms based on those changes. In the future, sensor and control values could be monitored for changes to detect manipulations. In addition, we plan to investigate machine learning approaches to expand our IDS detection space for sophisticated ICS threats.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions, and Pierre Gaulon for his contributions to the EtherNet/IP support in Bro as the intern in 2015. This work supported by SUTD's startup grant SRIS14081.

7. REFERENCES

[1] C. Alcaraz, L. Cazorla, and G. Fernandez. Context-awareness using anomaly-based detectors for smart grid domains. In *International Conference on Risks and Security of Internet and Systems*, pages 17–34. Springer, 2014.

[2] R. Berthier and W. H. Sanders. Specification-based intrusion detection for advanced metering infrastructures. In *Proceedings of Symposium on Pacific Rim Dependable Computing (PRDC)*, pages 184–193. IEEE, 2011.

[3] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta. A multidimensional critical state analysis for detecting intrusions in scada systems. *IEEE Transactions on Industrial Informatics*, 7(2):179–186, 2011.

[4] A. Cardigliano, L. Deri, J. Gasparakis, and F. Fusco. vpf_ring: towards wire-speed network monitoring using virtual machines. In *Proceedings of conference on Internet measurement conference*, pages 533–548. ACM, 2011.

[5] L. Cazorla, C. Alcaraz, and J. Lopez. A three-stage analysis of ids for critical infrastructures. *Computers & Security*, 55:235–250, 2015.

[6] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA security scientific symposium*, volume 46, pages 1–12, 2007.

[7] I. Committee. Manufacturing and control systems security part 1: Models and terminology. Technical report, ISA, 2004.

[8] Elastic Co. Elasticsearch. www.elastic.co/products/elasticsearch.

[9] I. N. Fovino, A. Coletta, A. Carcano, and M. Masera. Critical state-based filtering system for securing scada network protocols. *IEEE Transactions on industrial electronics*, 59(10):3943–3950, 2012.

[10] B. Genge, D. A. Rusu, and P. Haller. A connection pattern-based approach to detect network traffic anomalies in critical infrastructures. In *Proceedings of the Seventh European Workshop on System Security*, page 1. ACM, 2014.

[11] V. Grigorescu. Brownian web interface for bro logs. https://github.com/grigorescu/Brownian.

[12] I. Kiss, B. Genge, and P. Haller. A clustering-based approach to detect cyber attacks in process control systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 142–148. IEEE, 2015.

[13] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer. Adapting Bro into SCADA: building a specification-based intrusion detection system for the DNP3 protocol. In *Proceedings of Cyber Security and Information Intelligence Research Workshop*, page 5. ACM, 2013.

[14] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer. Semantic security analysis of scada networks to detect malicious control commands in power grids. In *Proceedings of the workshop on Smart energy grid security*, pages 29–34. ACM, 2013.

[15] O. Linda, T. Vollmer, and M. Manic. Neural network based intrusion detection system for critical infrastructures. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 1827–1834. IEEE, 2009.

[16] A. Mathur and N. O. Tippenhauer. A water treatment testbed for research and training on ICS security. In *Proceedings of Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*, 2016.

[17] D. Maynor. *Metasploit toolkit for penetration testing, exploit development, and vulnerability research*. Elsevier, 2011.

[18] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri. The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE*, 104(5):1039–1057, 2016.

[19] R. Mitchell and I.-R. Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, 46(4):55, 2014.

[20] L. Obregon. Secure architecture for industrial control systems. SANS Institute InfoSec Reading Room, 2015.

[21] ODVA. EtherNet/IP. Available at: https://www.odva.org/, 2016.

[22] V. Paxson. Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[23] V. Paxson. Bro: A system for detecting network intruders in real-time. *Comput. Netw.*, 31(23-24):2435–2463, Dec. 1999.

[24] F. Risso and L. Degioanni. An architecture for high performance network analysis. In *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, pages 686–693. IEEE, 2001.

[25] M. Roesch et al. Snort: Lightweight intrusion detection for networks. In *LISA*, volume 99, pages 229–238, 1999.

[26] A.-R. Sadeghi, C. Wachsmann, and M. Waidner. Security and privacy challenges in industrial internet of things. In *Proceedings of the Annual Design Automation Conference (DAC)*, pages 54:1–54:6, New York, NY, USA, 2015. ACM.

[27] Security Matters. Silentdefense ics. www.secmatters.com/products-ics.

[28] R. Udd, M. Asplund, S. Nadjm-Tehrani, M. Kazemtabrizi, and M. Ekstedt. Exploiting bro for intrusion detection in a SCADA system. In *Proceedings of Cyber-Physical System Security Workshop (CPSS)*, 2016.

[29] D. Urbina, J. Giraldo, N. O. Tippenhauer, and A. Cárdenas. Attacking fieldbus communications in ICS: Applications to the SWaT testbed. In *Proceedings of Singapore Cyber Security Conference (SG-CRC)*, Jan. 2016.

[30] S. Yasakethu and J. Jiang. Intrusion detection via machine learning for scada system protection. In *Proceedings of Symposium on ICS & SCADA Cyber Security Research*, pages 101–105. BCS, 2013.