MAHIVE: Modular Analysis Hierarchical Intrusion Detection System Visualization Event Cybersecurity Engine for Cyber-Physical Systems and Internet of Things Devices

Stu Steiner
Eastern Washington University
ssteiner@ewu.edu

Ibukun A. Oyewumi
University of Idaho
ibukunoyewumi@outlook.com

Daniel Conte de Leon
University of Idaho
dcontedeleon@ieee.org

Abstract

Cyber-Physical Systems (CPS), including Industrial Control Systems (ICS) and Industrial Internet of Things (IIoT) networks, have become critical to our national infrastructure. The increased occurrence of cyber-attacks on these systems and the potential for catastrophic losses illustrates the critical need to ensure our CPS and ICS are properly monitored and secured with a multi-pronged approach of prevention, detection, deterrence, and recovery. Traditional Intrusion Detection Systems (IDS) and Intrusion Detection and Prevention Systems (IDPS) lack features that would make them well-suited for CPS and ICS environments. We report on the initial results for MAHIVE: Modular Analysis Hierarchical IDS Visualization Event cybersecurity engine. MAHIVE differs from traditional IDS in that it was specifically designed and developed for CPS, ICS, a IIoT systems and networks. We describe the MAHIVE architecture, the design, and the results of our evaluation using two ICS testbed penetration testing experiments.

1. Introduction

Industrial Control Systems (ICS) are real-time infrastructure that operate and automate industrial processes [1]. Internet of things (IoT) devices are interrelated computing devices that collect and exchange data without human interaction. Intrusion Detection Systems (IDS) are devices and/or software applications that actively monitor systems for malicious activity including policy violations. IDS that also attempt to prevent intrusions are usually named Intrusion Detection and Prevention Systems (IDPS).

1.1. The Current Problem

The susceptibility of Cyber-Physical Systems (CPS) to external threats is well documented [2, 3], Recent cybersecurity reports show an increase in cyber-attacks against CPS, for example, the 2019 Symantec Internet Security Report showed an increase of approximately 30% from 2018 to 2020 [4]. The Cisco 2020 Benchmark Cybersecurity Report surveyed 2900 cybersecurity professionals across 13 countries, including United States, Brazil, China, and India. In this report, 69% of the respondents witnessed a cyber-attack or expected an attack at any moment [5].

As previously stated, traditional IDS are not effective at identifying attacks against the heterogeneity of the individual components. What is required for IDS for CPS is a comprehensive and distributed IDS application that features functions that are specific to ICS and IoT devices. This comprehensive IDS must be easily configurable to the specific ICS and/or IoT devices, with the ability to enable selective, low processing penalty, sensing and detection at different points in the control system.

1.2. Contribution

This paper presents MAHIVE: Modular Analysis Hierarchical IDS Visualization Event cybersecurity engine. MAHIVE's functionality includes: (1) to inspect selected logs from CPS-related protocols, including attached IoT devices; (2) to integrate detection alerts with big-data frameworks for stream processing; (3) to visualize high level, correlated log alerts on dashboards; and (4) to integrate intelligence feeds to help identify Indicators of Compromise (IOCs).

1.3. Overview of this Article

The rest of this paper is organized as follows: Section 2 provides background for IDS and CPS. Section 3 describes the MAHIVE architecture. Section 4 describes the testing and validation environment. Section 5 describes MAHIVE's simulated CPS environment. Section 5 details MAHIVE's detection, visualization, and correlation capabilities. Section 6 details simulated experimental validation results. Section 7 presents related work. Section 8 presents

URI: https://hdl.handle.net/10125/71475
978-0-9981331-4-0

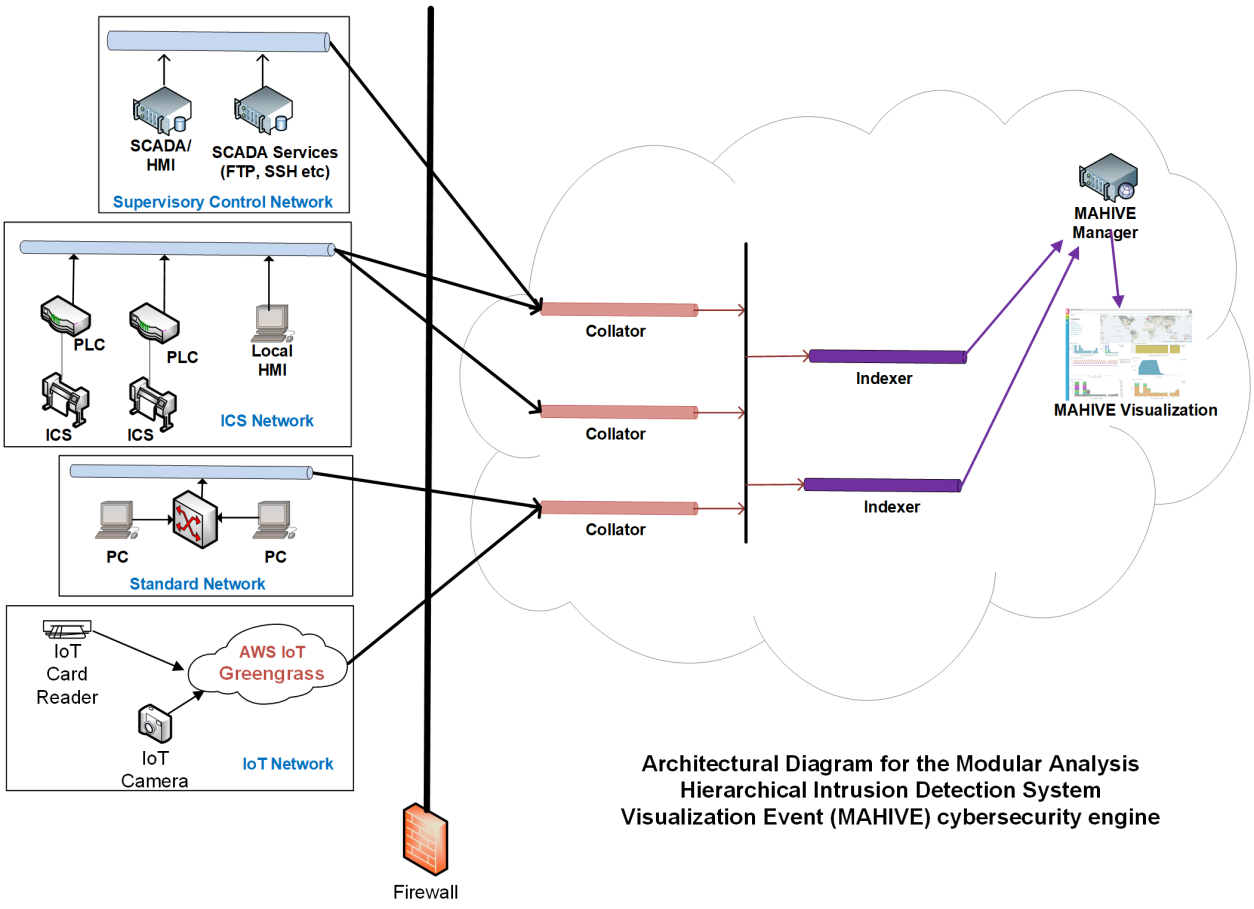


Figure 1. Architectural Overview for the MAHIVE cybersecurity engine

the conclusion and Section 9 presents MAHIVE future work.

2. Background

Current IDS are defined under one of four categories.

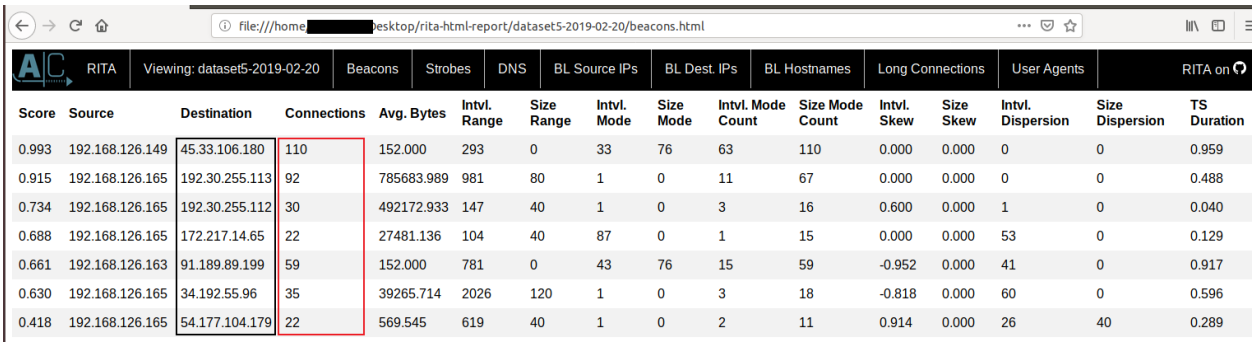
- Network Intrusion Detection Systems (NIDS) consists of an independent platform that identifies intrusions by examining network traffic including monitoring multiple hosts.
- Host Intrusion Detection Systems (HIDS) consists of an agent on a host that identifies intrusions by analyzing system calls.
- Perimeter Intrusion Detection Systems (PIDS) detects and pinpoints the location of intrusion attempts on critical infrastructure perimeter fences.
- Virtual Machine (VM) based Intrusion Detection System (VMIDS) detects intrusions using virtual machine monitoring.

CPS is the combination of physical systems and cyber systems through the integration of ICS, IoT devices and IDS. There has been a significant growth with the integration of CPS into our everyday life. For example, CPS permeate the electrical power grid, transportation systems, household appliances, and personal and industrial healthcare systems. One fundamental issue is CPS security. With CPS being a combination of many different systems including ICS and IoT devices the opportunity for new attack vectors also significantly increases.

The current state of CPS is most IDS don't detect or prevent intrusions based on the heterogeneity of the individual components.

3. MAHIVE Architecture

MAHIVE is divided into three modules: 1) hierarchical components for monitoring network security; 2) analytics and visualization engine; 3) threat intelligence sources. Figure 1 illustrates the MAHIVE architecture.



Score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Size Range	Intvl. Mode	Size Mode	Intvl. Mode Count	Size Mode Count	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion	TS Duration
0.993	192.168.126.149	45.33.106.180	110	152.000	293	0	33	76	63	110	0.000	0.000	0	0	0.959
0.915	192.168.126.165	192.30.255.113	92	785683.989	981	80	1	0	11	67	0.000	0.000	0	0	0.488
0.734	192.168.126.165	192.30.255.112	30	492172.933	147	40	1	0	3	16	0.600	0.000	1	0	0.040
0.688	192.168.126.165	172.217.14.65	22	27481.136	104	40	87	0	1	15	0.000	0.000	53	0	0.129
0.661	192.168.126.163	91.189.89.199	59	152.000	781	0	43	76	15	59	-0.952	0.000	41	0	0.917
0.630	192.168.126.165	34.192.55.96	35	39265.714	2026	120	1	0	3	18	-0.818	0.000	60	0	0.596
0.418	192.168.126.165	54.177.104.179	22	569.545	619	40	1	0	2	11	0.914	0.000	26	40	0.289

Figure 2. Beaconing Behavior Detection Using RITA

3.1. Network Security Monitoring

MAHIVE is built using the Zeek Network Security Monitor (ZNSM) [6, 7]. ZNSM is an open source tool for traffic analysis and network monitoring, capable of detecting real-time network intrusions through passive monitoring. ZNSM uses a policy script interpreter event engine. The script interpreter is capable of performing tasks over a range of network protocols. BNSM's event engine reduces kernel-filtered network traffic into a series of higher-level events. It is capable of capturing live data on multiple interfaces, including log files that can be generated to create an archive for the captured data. By default, ZNSM includes a protocol parser for Modbus and DNP3 protocols with decoding and event handling functionalities. It also supports a distributed and clustered architecture deployment with a policy script interpreter, that interprets event handlers written in a high-level policy language.

3.2. Visualization and Log Analytics

MAHIVE's visualization analytics component use the industry standard ELK Stack, Elasticsearch, Logstash, and Kibana [8]. Elasticsearch (ELCS) is a REpresentational State Transfer (RESTful) system that features a distributed search and analytics engine built on Apache Lucene [9]. ELCS provides a distributed, multi-tenant-capable, full-text search engine with an HTTP web interface and schema-free JSON documents. Logstash is a service used for log collection, processing, and data ingestion into ELCS. Kibana provides data visualization capabilities for content obtained from the ELCS cluster. Kibana's visualization capabilities facilitates the creation of customized charts providing easier comprehension of log and feed data. The ELK Stack provides an end-to-end log analysis solution, which aids in analytics, visualization, and geospatial support of logs.

MAHIVE uses Apache Kafka, a multi-node big data cluster, to manage incoming log streams and message queuing [10]. Kafka is a distributed streaming platform for creating real-time data pipelines. Kafka was chosen because of its strong fault tolerance, automated re-balancing algorithms, and ability to support large scale real-time data streams. Combining Kafka with Apache Spark allowed advanced data streaming capabilities [11] and the use of external libraries integrating the Spark SQL library, for extended query processing and the MLib library, for machine learning.

3.3. Threat Intelligence

MAHIVE's threat intelligence uses the Intel Critical Stack (ILCS) [12] coupled with RITA (Real Intelligence Threat Analytics) [13]. ILCS provides threat intelligence feeds and signature analysis. RITA provides network traffic analysis using beaconing and DNS tunneling detection. RITA searches for signs of beaconing behavior in and out of a CPS network. Figure 2 illustrates the beaconing behavior using RITA.

4. Simulated CPS Environment

In order to test and evaluate MAHIVE, select components of a micro-grid power utility was created [14]. The micro-grid power utility consists of two hydro generators and transmission buses, and the simulation uses the open source ModbusPal Java simulator. ModbusPal reproduces real and complex CPS environments with native support for TCP/ IP and scripting. ModbusPal, coupled with its automation tool SineGenerator, was used to generate customized and dynamic generator measurement values.

MAHIVE tracks the current state of the micro-grid CPS system using Ladder Logic on OpenPLC [15]. OpenPLC is open source Programmable Logic Controller (PLC) that emulates the functions of electromechanical relays based on the configurations of uploaded logic-based programs.

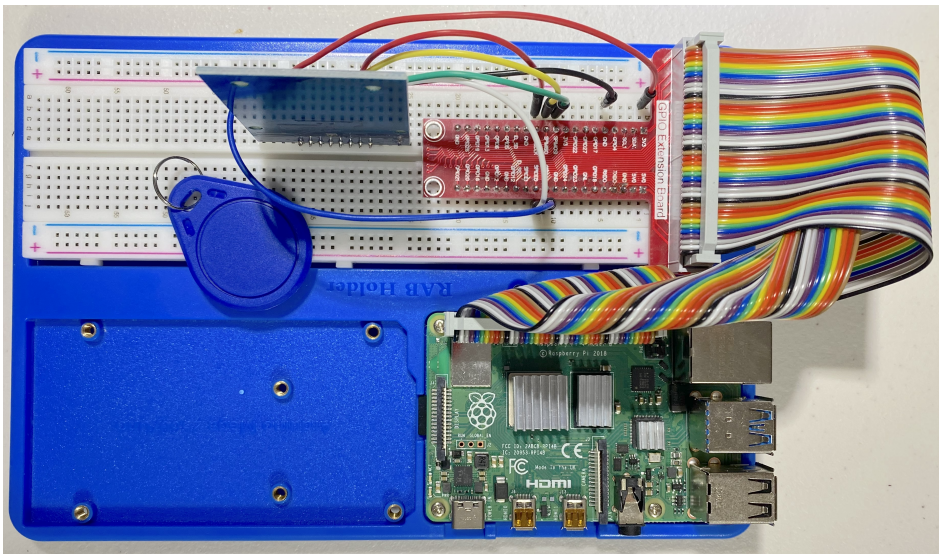


Figure 3. MAHIVE IoT RFID Simulator.

Supervisory Control and Data Acquisition (SCADA) was implemented using ScadaBR [16]. ScadaBR is an open source SCADA system that supports acquisition of data in more than 20 CPS protocols. It also provides supervisory control, an events engine, alarm and report management, and a Human Machine Interface (HMI).

The MAHIVE implementation provides a visualization of OpenPLC server measurement values that are transmitted in real-time as DNP3 tags. The DNP3 data sources were configured with ScadaBR including simple alarms notifications from device state changes and point event detectors.

The MAHIVE simulation environment forms a distributed CPS architecture depicting level 0, level 1, and level 2 of the Purdue Model for Control Hierarchy [17].

The IoT simulation environment was created using a Raspberry Pi 4 Model B. The Pi simulated an RFID card swipe of an employee entering the micro-grid power utility. The data from the card swipe was processed using Amazon Greengrass IoT Core, where lambda functions were applied to record the time and the user identity. Figure 3 displays the Raspberry Pi 4 model B representing the IoT RFID.

5. MAHIVE Implementation

A clustered ZNSM instance provides distributed hierarchical packet capture and anomaly detection across several layers of a CPS organization. The architecture supports packet inspection of several protocols, including DNP3, MODBUS and TCP/IP by using sensors for protocol analysis on traffic streams. Traffic from several layers of the model system is aggregated to a centralized manager (MAHIVE Manager) which is responsible for receiving alerts and notices from the rest of the nodes in the cluster. In Figure 1 the collators act as the logger and proxy nodes for storing log messages from each of the network components. The indexers manage the synchronized state of the variables across network components. The MAHIVE manager handles eliminating multiple similar log events from several devices, ultimately creating a single logged event that is managed by the indexers.

In Figure 1, the sensors in the ICS Network zone capture all traffic sent and received from the PLC. The sensors forward the traffic through the firewall to the collators. Similarly, the Supervisory Control Network and Standard Network zones capture the traffic. The traffic is passed to the collators where the event engine and policy script interpreter, analyze received packets, to generate logs containing several protocols such as TCP, UDP, DNP3, MODBUS, and ARP. Logs generated from matching detection signatures in Intel CriticalStack and RITA are included.

The IoT Network zone is similar to the other zones, except the sensors are replaced with AWS IoT Greengrass [18]. Greengrass seamlessly extends AWS to edge devices acting locally on the generated data, while allowing connected devices to run Lambda functions or Docker containers. Lambda functions allow execution of predictions based on machine learning models. Greengrass forwards its traffic through the firewall to the collators.

The collators groups the data, and then forward that data to the indexers. The aggregated data from

the indexers is ultimately forwarded to the MAHIVE Manager. The MAHIVE Manager forwards the data through the ELK Stack for visualization and analytics.

MAHIVE uses a distributed and fault tolerant Apache Kafka cluster to address the four core characteristics of Big Data: [19] 1) Volume 2) Variety 3) Velocity and 4) Value. The Kafka instance is a multi-node cluster comprising of three brokers, that uses Zookeeper to track the status of cluster nodes. MAHIVE correlates logs from the distributed sensors, where intrusion notices are visualized as alerts on the multi-node Apache Kafka.

MAHIVE relies on its network security monitoring system with extended capabilities of logging, anomaly detection, behavioral analysis and signature based intrusion detection. This system also contains an event-driven scripting language. Whenever the event engine receives incoming packet streams, the engine reduces the streams into higher-level events. By reducing the streams into higher-level events the engine can understand the network activity without any policy parsing. Using scripts, packets are analyzed by the sensors and attack patterns matching the site policy's detection scripts are logged in the file *notice.log*. The matching threat intelligence alert is logged in the file *intel.log*. *Intel.log* is then used by the Intel CriticalStack for traffic patterns with known signatures.

6. MAHIVE Validation

The MAHIVE prototype, version 0.1, was validated using two penetration testing scenarios representative of potential cyber-attacks on industrial control systems. The premise for each scenario is that a malicious actor has compromised a server located in any of the network zones. The attack scenarios used are: Network Reconnaissance and Address Resolution Protocol (ARP) Spoofing.

6.1. Network Reconnaissance

Description: During network reconnaissance a malicious actor attempts to determine information about a network, including the network structure, and its applications and services. In this scenario, a malicious actor attempts to map the network using a black box approach including:

- Executing a ping sweep to identify live hosts, devices, or services and obtain information about them and any connected systems.
- Initiating a stealth port scan using Nmap TCP connect and SYN scans. A stealth port scan differentiates between open, closed, and filtered ports.

The Simulated Attack: A malicious actor launched multiple reconnaissance campaigns on the Supervisory Control Network. This simulated attack attempted to sniff network packets and create a list of vulnerable hosts, devices, protocols, and services.

Results: MAHIVE detected all reconnaissance campaigns. A loaded scan policy with custom defined intervals and thresholds for address and port scans was used to detect the reconnaissance campaigns. When the network scans were detected, an alert was logged to `notice.log` on the MAHIVE collator. The collector logged all reconnaissance campaign attempts. However, the collator only forwarded a single log entry to `intel.log` since all scans were considered duplicate scans. The indexer notified the MAHIVE Manager and the scan alert was visualized.

6.2. ARP Spoofing

Description: To build an ARP entries table, outstation devices and other Ethernet based devices broadcast ARP packets. Broadcasting ARP packets initiates a communication channel with the goal of obtaining the Media Access Control (MAC) addresses of other systems. In this scenario, using the ICS network, a simulated malicious actor attempts to spoof a PLC ARP entry. Spoofing the PLC ARP entry allows access to the network where false data can be injected into the ICS.

The Simulated Attack: A malicious actor attempted to disrupt the operational network by launching an ARP spoofing attack. This simulated attack used EtterCap attempting to inject random values into holding registers.

Results: MAHIVE detected all malicious data injections by using a known-masters-slaves script. This script tracks all master and slave devices within the ICS network. Similar to the Network reconnaissance attacks the attempts were logged to `notice.log` and forwarded to indexers. The indexers logged the information to `intel.log` and then forwarded the information to the MAHIVE Manager where the scan alert was visualized.

Figure 4 shows the MAHIVE detection of both the network reconnaissance scans and the simulated ARP spoofing attacks. In both instances MAHIVE alerted the security analyst visually and with log entries to `notice.log` and `intel.log`.

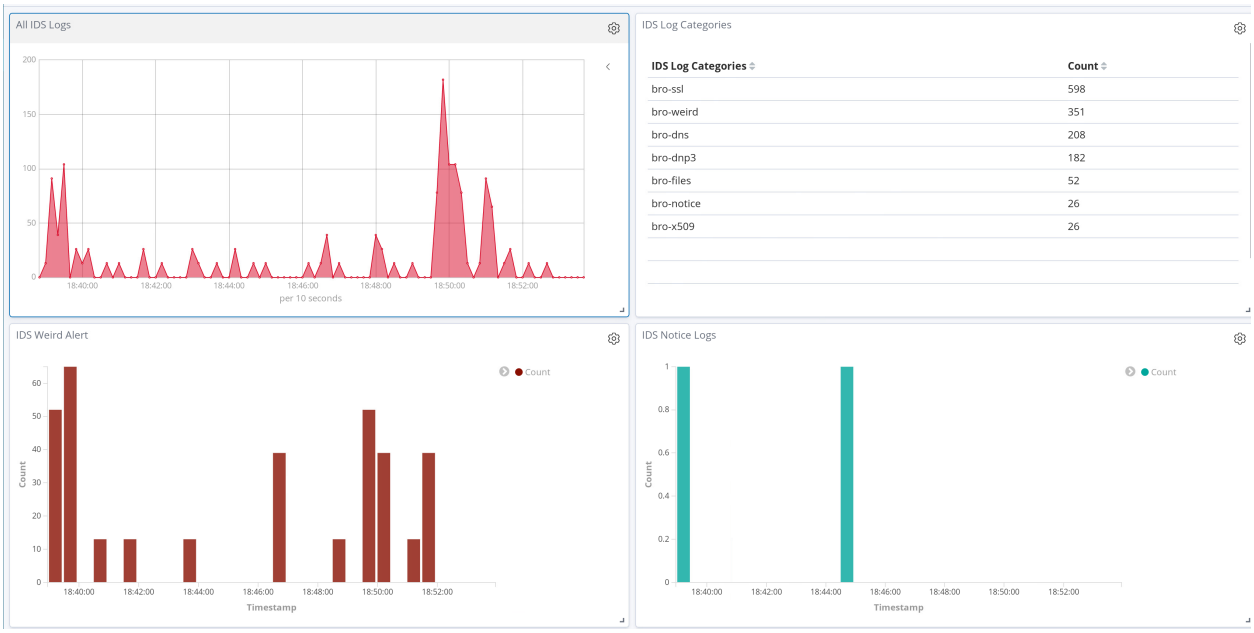


Figure 4. MAHIVE Alerts on Visualization Dashboard

7. Related Work

The research concerning IDS and IDPS for CPS has gained momentum in the past five years, and based on this research a classification system for ICS testbeds has been developed. Geng et al. [20] classified the ICS testbeds into four different categories.

- Physical Simulation Testbed - The testbed uses actual hardware and software to configure both the network and physical layers.
- Software Simulation Testbed - The testbed uses software to simulate the hardware.
- Semi-physical Simulation Testbed - The testbed is mostly comprised of software simulations; however, some actual hardware is used in the test bed.
- Virtualized Simulation Testbed - The testbed is completely virtualized using open source software and controllers such as OpenPLC [21].

MAHIVE was developed based on Ghaeini et al. HierArchical Monitoring Intrusion Detection System (HAMIDS) and Mantere et al. Self-Organizing Maps (SOM).

Rakas et al. further classified IDS testbeds by evaluating the research work of 26 IDS and IDPS for CPS [22]. In Rakas et al. classification system, SOM is not addressed and HAMIDS is classified as a virtualized simulation test bed that is specification-based. Also classified with HAMIDS as a specification-based virtualized simulation testbed is the work of Yang et al..

Ghaeini et al. described HAMIDS as a hierarchical monitoring IDS that detects anomalies at level 0 and level 1 of an industrial control system in a water treatment facility [23]. The HAMIDS framework is built on Zeek IDS, using Hadoop as a storage component.

Mantere et al. introduced the SOM algorithm. This algorithm promotes use of network specific state attributes to provide network anomaly detection [24]. SOM uses the Zeek IDS.

Yang et al. introduced an IDS tailored for IEC 61850 based substations. This work encompassed access control detection, protocol whitelisting, model-based detection, and multi-parameter based detection [25]. The work was completed with data from a real 500 kV smart substation.

The remainder of this related works section discusses the differences of MAHIVE compared to HAMIDS, SOM and IEC 61850, concerning modularity, security monitoring, visualization, intelligence sources and log analytics.

Modularity: HAMIDS allows inspection of selected SCADA protocols as Zeek extensions. SOM allows for inspection of normal logs and alarm logs. IEC 61850 allows for substation configuration monitoring. MAHIVE allows for monitoring as well as the integration of ICS and IIoT devices using a modular approach.

Security Monitoring: HAMIDS allows for IP scanning, Port scanning, ARP poisoning, Dynamic Host Configuration Protocol (DHCP) attack, synchronize (SYN) flooding, and HMI crash. SOM is a complementary mechanism and not a security monitoring tool. IEC 61850 allows for access-control detection, model-based detection and multi-parameter detection. MAHIVE is a comprehensive distributed IDPS suite that allows for the same security monitoring as others; however, MAHIVE has the added capabilities of IIoT monitoring and detection.

Visualization: Based on the published related works, there was difficulty in determining the visualization capabilities of HAMIDS or SOMS. IEC 61850 illustrated basic visualization capabilities of its log files. MAHIVE differs from the others based on its visualization capabilities. MAHIVE's visualization analytics component is built using the ELK Stack: Elasticsearch, Logstash, and Kibana.

Intelligence Sources: HAMIDS, SOM and IEC 61850 are all built from traditional IDS, without the capabilities for IIoT. MAHIVE is also built from a traditional IDS; however, MAHIVE allows for IIoT low power devices attached to the perimeter.

Log Analytics: HAMIDS and SOM both use Zeek scripting for log analysis. IEC 61850 uses Generic Object Oriented Substation Events GOOSE for its events and log analysis. MAHIVE uses Apache Kafka as a multi-node big data cluster to manage incoming log streams and message queuing. The use of Apache Kafka is unique to MAHIVE as compared to other IDS.

In summary MAHIVE is similar to other IDS systems in its use of traditional IDS intelligence sources. MAHIVE is radically different from the other systems with its visualization analytics component and its capability of adding, monitoring, and detecting attached IIoT devices.

8. Conclusion

Comprehensive and accurate intrusion detection in Cyber-Physical Systems requires practitioners to have an exceptional understanding of the threat landscape and also the structure and operation of the CPS or ICS being protected.

Traditional IDS are not well-suited for adequately monitoring CPS and ICS systems. As more organizations add digitally controlled and network connected CPS, including integrated ICS and IoT devices to the critical national infrastructure, there is a growing demand to ensure CPS are adequately secured. Successful cyber-attacks against critical infrastructures could have catastrophic consequences.

This article introduced MAHIVE: Modular Analysis Hierarchical IDS Visualization Event cybersecurity engine. MAHIVE shows that a distributed hierarchically integrated IDS may help address current drawbacks in IDPS. A detailed analysis concerning the architecture and implementation of MAHIVE was described as well as the results of two penetration testing scenarios used as preliminary evaluation.

In summary, MAHIVE should be considered a next-generation IDS that is capable of monitoring CPS with ICS and IIoT and detecting real-world intrusions.

9. Future Work

Similar to all other signature-based Intrusion Detection Systems, MAHIVE requires expert knowledge related to network protocols, systems and logs, and CPS scripting languages. Future work includes the following additional enhancements.

The current logging system of `notice.log` and `intel.log` will be converted into a high-level policy language, such as the High Level Easy to Use Reconfigurable Machine Environment Specification (HERMES) language [26, 27]. HERMES allows for a block-like structure, domain specific language that contains singular identifiers to differentiate specifications, including a wide range of attributes and fields for relationship representation.

The current MAHIVE anomaly and behavior detection system will be enhanced with an implementation of Machine Learning-based detectors. This will allow a more robust set of anomaly and behavior detection scripts, replacing the simulation loaded scripts.

Additional and richer penetration testing scenarios will be developed to further evaluate MAHIVE's performance.

By implementing these future work items MAHIVE with its monitoring of CPS including integrated IDS and IIoT will further its effectiveness at identifying attacks against the heterogeneity of the individual components.

List of Abbreviations

ARP	Address Resolution Protocol
CPS	Cyber-Physical Systems
DHCP	Dynamic Host Configuration Protocol
DNP3	Distributed Network Protocol 3
ELCS	Elasticsearch
ELK	Elasticsearch, Logstash and Kibana

GOOSE	Generic Object Oriented Substation Event
HERMES	High Level Easy to Use Reconfigurable Machine Environment Specification
HIDS	Host Intrusion Detection Systems
HMI	Human-Machine Interface
ICS	Industrial Control Systems
IDPS	Intrusion Detection and Prevention Systems
IDS	Intrusion Detection Systems
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
ILCS	Intel Critical Stack
IOC	Indicators of Compromise
IoT	Internet of Things
ISAAC	The Idaho Cybersecurity Testbed
IT	Information Technology
LAN	Local Area Network
MAC	Media Access Control
MAHIVE	Modular Analysis Hierarchical IDS Visualization Event cybersecurity engine
NIDS	Network Intrusion Detection Systems
NMap	Network Mapper
OT	Operational Technology
PIDS	Perimeter Intrusion Detection Systems
PLC	Programmable Logic Controller
RESTful	REpresentational State Transfer
RITA	Real Intelligence Threat Analytics
SCADA	Supervisory Control and Data Acquisition
SOM	Self-Organizing Maps
SYN	Synchronize
VM	Virtual Machine
VMIDS	Virtual Machine Intrusion Detection Systems
ZNSM	Zeek Network Security Monitor

Acknowledgments

We would like to thank the University of Idaho, College of Engineering, Center for Secure and Dependable Systems, and Computer Science and Electrical and Computer Engineering Department's technical and administrative staff for their help designing, implementing, and maintaining our research infrastructure and services. We would also like to thank the program committee, conference chairs, and reviewers for their help improving this paper.

This research and the computing and laboratory testing infrastructure used to perform it were partially funded by the Idaho Global Entrepreneurial Mission (IGEM) Grant for Security Management of Cyber-Physical Control Systems, 2016 (Grant Number IGEM17-001), the U.S. National Science Foundation (NSF) CyberCorps® award 1565572, and the M.J. Murdock Foundation. The opinions expressed in this paper are not those of the NSF, the M.J. Murdock Foundation or the State of Idaho.

10. References

References

[1] A. A. Jillepalli, D. Conte de Leon, M. Ashrafuzzaman, Y. Chakhchoukh, B. K. Johnson, F. T. Sheldon, J. Alves-Foss, P. Tosic, and M. A. Haney, "HESTIA: Adversarial modeling and risk assessment for CPCS," in *14th International Wireless Communications and Mobile Computing Conference (IWCMC-2018)*, pp. 226–231, June 2018.

[2] Dragos, Inc., "Threat proliferation in ics cybersecurity: Xenotime now targeting electric sector, in addition to oil and gas." Online, June 2019.

[3] R. E. Mahan, J. D. Fluckiger, S. L. Clements, C. W. Tews, J. R. Burnette, C. A. Goranson, and H. Kirkham, "Secure data transfer guidance for industrial control and scada systems," Tech. Rep. PNNL-20776, Pacific Northwest National Laboratory (PNNL), 2011.

[4] Symantec Corporation, "2019 internet security threat report." Online, March 2019.

[5] Cisco Systems, Inc., "Cisco 2020 benchmark cybersecurity report." Online, February 2020.

[6] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[7] Z. Manual, "Zeek user manual v3.2.1." `https://docs.zeek.org/en/current/index.html`, May 2020.

[8] Elasticsearch, B. V., "Elastic Search." `https://www.elastic.co/products/elasticsearch`, May 2019.

[9] Apache, Software Foundation, "Apache Lucene." `https://lucene.apache.org/`, May 2019.

[10] Apache, Software Foundation, "Apache Kafka." `https://kafka.apache.org`, May 2019.

[11] Apache, Software Foundation, "ApacheSpark." `https://spark.apache.org`, May 2019.

[12] CriticalStack, Inc., "Critical Stack Intel." `https://intel.criticalstack.com/`, May 2019.

[13] Active, Countermeasures, "Real Intelligence Threat Analytics." `https://github.com/activecm/rita`, May 2019.

[14] I. A. Oyewumi, H. Challa, A. A. Jillepalli, P. Richardson, Y. Chakhchoukh, B. K. Johnson, D. Conte de Leon, F. T. Sheldon, and M. A. Haney, "Attack scenario-based validation of the idaho CPS smart grid cybersecurity testbed (ISAAC)," in *2019 3rd IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, Feb 2019.

[15] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "Openplc: An open source alternative to automation," in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 585–589, Oct 2014.

[16] Sensorweb, B. R., "SCADABR." `http://www.scadabr.com.br/`, May 2019.

[17] P. Ackerman, *Industrial Cybersecurity*. Packt Publishing, 2017.

[18] Amazon, "AWS IoT Greengrass." `https://aws.amazon.com/greengrass/`, May 2020.

[19] J. P. Dijcks, "Oracle: Big Data for the Enterprise," Tech. Rep. ORCL-1453236, Oracle, Redwood Shores, CA (US), 2014.

[20] Y. Geng, Y. Wang, W. Liu, Q. Wei, K. Liu, and H. Wu, "A survey of industrial control system testbeds," *IOP Conference Series: Materials Science and Engineering*, vol. 569, p. 042030, aug 2019.

[21] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "Openplc: An open source alternative to automation," in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 585–589, 2014.

[22] S. V. B. Rakas, M. D. Stojanovi, and J. D. Markovi-Petrovi, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93083–93108, 2020.

[23] H. Ghaeini and N. Tippenhauer, "Hamids: Hierarchical monitoring intrusion detection system for industrial control systems," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, CPS-SPC-2016, pp. 103–111, ACM, 2016.

[24] M. Mantere, M. Sailio, and S. Noponen, "A module for anomaly detection in ics networks," in *Proceedings of the 3rd International Conference on High Confidence Networked Systems*, HiCoNS '14, (New York, NY, USA), pp. 49–56, ACM, 2014.

[25] Y. Yang, H. Xu, L. Gao, Y. Yuan, K. McLaughlin, and S. Sezer, "Multidimensional intrusion detection system for iec 61850-based scada networks," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 1068–1078, 2017.

[26] A. A. Jillepalli, D. Conte de Leon, S. Steiner, and F. T. Sheldon, "Hermes: A high-level policy language for high-granularity enterprise-wide secure browser configuration management," in *Proc. 2016 IEEE 07th Symposium Series On Computational Intelligence (SSCI-2016)*, December 2016.

[27] D. Conte de Leon, M. G. Brown, A. A. Jillepalli, A. Q. Stalick, and J. Alves-Foss, "High-level and formal router policy verification," *Journal of Computing Sciences in Colleges*, vol. 33, pp. 118–128, October 2017.