Using Application Layer Metrics to Detect Advanced SCADA Attacks

Peter Maynard, Kieran McLaughlin and Sakir Sezer
Centre for Secure Information Technologies, Queen's University Belfast, BT3 9DT, Belfast, U.K.

Keywords: ICS, IDS, Network, SCADA, Security, SIEM.

Abstract: Current state-of-the-art intrusion detection and network monitoring systems have a tendency to focus on the 'Five-Tuple' features (protocol, IP src/dst and port src/dest). As a result there is a gap in visibility of security at an application level. We propose a collection of network application layer metrics to provide a greater insight into SCADA communications. These metrics are devised from an analysis of the industrial control system (ICS) threat landscape and the current state-of-the-art detection systems. Our metrics are able to detect a range of adversary capabilities which goes beyond previous literature in the SCADA domain.

1 INTRODUCTION

The problem with state-of-the-art network intrusion detection systems (IDS) is the lack of ICS-specific insight. Many solutions focus on extracting statistics based on the traditional 5-tuple of flow features. However, this misses potentially valuable information in the SCADA application layer, which may be used to develop more detailed metrics, regarding the security status and performance of the underlying physical system.

The trend in the current state-of-the-art research is to use machine learning methods to discover malicious actions on the network. A common method is to take low level packet captures and analyse them for anomalous activity. e.g. (Terai et al., 2017) who use packet inter-arrival time. We propose the use of application layer packet analysis within the Process Control enclave, and use that data to provide an analyst with deeper insight into the security status of the ICS facility. These metrics can be used for different applications ranging from intrusion detection, forensics, or verifying compliance with security standards and legislation. Legislation, such as the GDPR and NIS Directive require active monitoring and historic measurable indicators of progress. We have derived a set of metrics loosely based on IEC 60870-5-104, which can also be used on other field bus protocols.

Contributions: *(i)* An analysis of industrial threat actors and their capabilities. *(ii)* Review the current state-of-the-art metrics for ICS. *(iii)* Proposed novel metrics that enable deeper insight into SCADA networks, and allow for detecting anomalies within an ICS system. *(iv)* Finally, we compare the current state-of-the-art ICS intrusion detection systems with the proposed metrics to show a much improved coverage of attack activities in reconnaissance, interference, DoS and covert communications. The paper is structured as follows. Section 2 and 3, detail the ICS threat landscape. Section 4 highlights related work and ICS metrics. Section 5 details the proposed metrics drawing comparison between state-of-the-art IDS proposals and our contributions.

2 ICS ENVIRONMENT

A high-level overview of an ICS network typically can be broken down to three enclaves. 1. Business 2. SCADA and 3. Process Control. Historically, SCADA enclaves were air-gapped, however, in recent years it is common to find one-way traffic from SCADA to the business enclave. A SCADA network may contain a data historian, human machine interface (HMI), and remote devices such as programmable logic controllers (PLC). The business enclave facilitates traditional systems used for accounting, human resources and analysing productivity of the plant. The business network is the main entry point for an adversary, typically communicating with the SCADA network, requesting data used within performance reports. The SCADA enclave contains HMI software and monitoring equipment. Similar to the business enclave, as vendors move towards open standards, the hardware requirements are becoming less proprietary, allowing for the use of common off the shelf software (Adobe Flash, Microsoft Windows and

Maynard, P., McLaughlin, K. and Sezer, S.
Using Application Layer Metrics to Detect Advanced SCADA Attacks.
DOI: 10.5220/0006656204180425
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 418-425
ISBN: 978-989-758-282-0

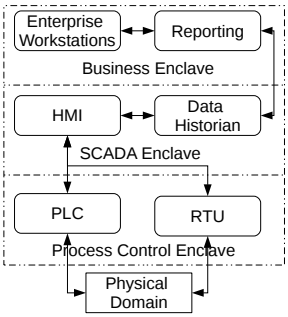

Figure 1: High-level view of an ICS network.

*Unix). The SCADA enclave communicates with the process control enclave, issuing commands and receiving state information from the physical systems.

The process control enclave contains devices that communicate using specialised field bus protocols i.e. IEC104, Modbus-TCP, DNP-3 etc. The process control enclave can potentially span a large physical area. On the other hand, this physically large enclave is often not segregated, and is configured as a single flat LAN. Such networks range from on-premises, to remote locations such as autonomous substations. This is a mission critical network which is sensitive to latency. For example, if this were an electrical substation there may be various meters providing power readings that are collected by an RTU or IED and transmitted up to the SCADA network. Alternatively, an operator might need to issue a command to shut down a system, which needs to be completed in a timely and reliable manor.

Figure 1 shows the high-level of separation and relationships between the four enclaves. This segmentation works well with (Shostack, 2014) suggestion of using 'Trust Boundaries' to identify where a risk may initiate. Once the SCADA encalve has been breached, any data received from the process control enclave can not be trusted. This paper focuses on the communications between the SCADA and Process control enclave because of the restricted types of protocols which should be seen between these networks. The aim is to enable detection of malicious events affecting the most critical systems as a priority to counter system risks.

3 INDUSTRIAL THREAT MODEL

Typical threat modelling processes are asset/software-centric. While this provides valuable insight into attack paths and vulnerabilities within software, it fails to highlight possible methods to detect network intrusions. ICS networks are well understood, but the lack of information about the threat actors limits development of defence systems. As such, the authors propose to perform an adversary/attack-centric approach to modelling potential threats. The aim is to better align detection metrics at the application layer with the characteristics of realistic threats to the process network.

Without understanding adversaries' capabilities, it will be difficult to know where to place active defences such as network security monitoring. This section will deconstruct and model known adversaries of industrial systems in order to highlight possible types of attacks. NIST SP-800-82 (Stouffer et al., 2015) defined four primary adversary actors as Individual, Group, Organisation, Nation-State. Each of these have their own characteristics and capabilities. Stouffer's work aligns well with (Robinson, 2013), which performed a comprehensive analysis of the SCADA threat landscape. The proposed approach therefore begins by aligning adversaries with attacks identified from the state-of-the-art detection literature. Table 1 shows different attack capabilities which can be performed by each threat actor. Depending on a perceived threat model (e.g., a high risk of being attacked by a group or organisation), defences can be prepared against certain types of attacks.

Different classes of adversaries are now briefly defined. An individual is a single person working alone. Insiders which can range from on-site employees, remote contractors or partner companies. The scope of damage which these individuals could cause depends on their level of privilege within the system. The likely person could be a disgruntled employee or script kiddie. Types of attacks from individuals can range from port and device scanning to insider compromise resulting in significant damage.

A group is one or more persons working together. The amount of technical and domain knowledge is potentially moderate to high. Groups can fall into two categories, ad-hoc and established. An ad-hoc group is likely to be performing basic incoherent attacks. An established group would have more resources to attract individuals with domain expertise. Their motivation would likely be political and fall into the 'Hacktivist' definition. Types of attacks which might be performed are spear-phising, custom malware and noisy reconnaissance.

Organisations, not unlike groups, consist of more than one person, but have more domain knowledge and technical abilities. They may be industrial competitors, suppliers, partners, or customers. Their intention may be corporate espionage or financial gain, an organisation would be able to perform advanced reconnaissance as well as targeted malware for well known devices. Nation-states have similar features

Table 1: Comparison of actors and capabilities.

Attack Stages	Individual	Group	Organisation	Nation-State
Reconnaissance				
Network Scan	•	•	•	•
Device/Protocol Scan	-	•	•	•
Report Server Information	-	-	•	•
Read Device Identification	-	-	•	•
Interference				
Command Replay	-	•	•	•
Command Injection	-	•	•	•
Unauthorised Write	-	-	•	•
Unauthorised Read	-	•	•	•
Clear Counter/Diagnostic Registers	-	•	•	•
Rouge Device	-	-	•	•
Firmware Tampering	-	-	•	•
Denial of Service				
TCP/UDP Spam	•	•	•	•
Remote Restart	-	-	•	•
Force PLC into Listen Mode	-	-	•	•
Covert				
Covert Comms.	-	-	-	•
Stealthy Deception Attack	-	-	-	•
Malicious Firmware	-	-	•	•

to the organisation threat actor, but have considerably more capabilities than the others. A nation-state's intentions may be espionage and warfare. Typically, they would use advanced methods such as the attacks detailed by (Kleinmann et al., 2017), in which they propose a stealthy attack on a SCADA system that is protocol and process aware. A nation-state might use covert channels within the SCADA enclave, such as the one proposed by (Lemay et al., 2016), used for command and control. Examples of such nation-state attacks are the Ukrainian power outages of 2015/16, and the Stuxnet malware.

4 RELATED WORK ON METRICS

Measurements provide a single-point-in-time view of specific, discrete factors, while metrics are derived by comparing a baseline to two or more measurements over time. Measurements are made by counting, metrics are made from analysis. Another way of thinking is that measurements are objective raw data and metrics are either objective or subjective human interpretations of those data (Payne, 2006). A metric should communicate useful information and be consistent and cheap to produce. For a metric to be used in further analysis it should be quantifiable, repeatable and specific to the context where it is applied. (Rathbun and Homsher, 2009) (Pendleton et al., 2016) reviewed the current state-of-the-art security metrics, which can be broken down into four categories: *1)* Attack; 2) System Vulnerabilities; 3) Situational; and, 4) Defences. They discuss the use of intrusion detection metrics with a focus on the effectiveness of an IDS. This can be used to compare detection systems. For actual attack measurements, they focus on zero-day vulnerabilities which provides a measurement of the attack consequences in retrospect. They propose measurements which allow for monitoring a botnet's applicability and how resourced it is, which can be used in developing and deploying mitigation strategies e.g. sink holes. (Zhang et al., 2016) have modelled network diversity as a security metric to evaluate network robustness against zero-day vulnerabilities. They measured the number of distinct resources within the network such as hosts, and host connectivity, resources and any similarities. They designed a probabilistic model which identifies a path of least resistance to compromise an asset.

Building on these concepts, the following sections present and analyse existing work on the development of metrics specific to ICS, focusing on three main areas: 1) generic network metrics i.e. packet timing/length; 2) Physical metrics using data from the physical domain; and 3) the SCADA application layer as a source for deriving indicators of compromise.

4.1 Generic Network Metrics

(Terai et al., 2017), propose the use of Support Vector Machines (SVM) to identify attacks on an ICS network. However the SVM was trained on generic network traffic and would thus be unable to detect specific intrusions like SCADA based covert communication channels, which disguise command and control (C2) within genuine ICS traffic. (Rudman and Irwin, 2016) developed a framework to automatically identify network indicators of compromise (IOC) from packet captures. By running the Dridex malware in a dynamic sandbox they identified basic high-level IOCs such as IP address ranges of control server and frequently used protocols and ports. This is valuable data which can be used to analyse the behaviour of specific malware variants. However, it fails to generate any ICS specific IOCs.

(Wang et al., 2014), propose that by modelling network connectivity along with policies and configurations of a host, they can predict the number of zero-day compromises needed to disrupt a critical asset. The higher the number, the less likely the asset is to be compromised. The model depends on the existence of network connectivity which is not always available.

(Lemaire et al., 2017) have used machine reasoning to identify vulnerabilities in an industrial system based on NIST and ICS-CERT. They use the Systems Modelling Language (SysML) to model the system and perform an analysis. With this design, they are limited by the vulnerability database which is a) based on known exploits and b) manually updated, which can cause delays in detection.

(Zhang et al., 2015) have expanded on the model of 'mean-time-to-compromise' (MTTC), used to estimate the time it takes for an adversary to compromise an asset. Using Bayesian network attack graphs, Zhang et.al are able to identify potential attack steps and quantify various attack scenarios. The first attack graph is used to determine the probabilities of successful exploitation of a vulnerability in the SCADA enclave to gain root access. This is calculated by considering the ratio of exploits via CVE/CVSS scores. The second is used to evaluate the probability of successful compromise of the communication links between the SCADA and process enclaves. They decompose a man-in-the-middle (MITM) attack and counter-measures are included within the model.

(Vasilomanolakis et al., 2016) developed a honeypot that emulates an ICS system, whereby they are able to generate IDS rules automatically from the results. They have emulated a range of ICS processes, and able to detect different classes of attacks: Single-Protocol Level Detection (SPLD); Multi-Stage Level Detection (MSLD); and Payload Level Detection (PLD). However, they are restricted by collection biases, because the types of attacks that honeypots typically detect are automated scanning of well known attacks.

4.2 Physical Metrics

(Urbina et al., 2016), model the physical state of the system and compare the predicted values to actual readings. The attacks they can detect are MITM-type scenarios where the adversary is sending false sensor data to the PLC. As they noted this is a noisy approach and easily detectable. They propose a sophisticated stealthy attack, in which they monitor a specific value within the system and track it for changes. They proposed a stateful and stateless method to detect changes of the physical system to determine if an attack had occurred. (Luchs and Doerr, 2017) proposed IDS monitoring in remote field devices, with the intention to gain more visibility into the process control enclave. They proposed additional monitoring vectors such as acoustic levels, to identify complex attacks such as Stuxnet. Luchs and Doerr monitored the status of field devices independently from the HMI/PLC readings allowing for additional validation. This was used to confirm whether the device is operating within the manufacture's specification, which is another vector for detecting anomalous activity. While the proposed IDS does provide superior insights into the process control enclave over traditional vectors, it would be unlikely to be adopted by ICS operators in its current state, due to the complexity of deployment. This is unlike traditional network IDS, which can be quickly deployed and integrated with existing systems with out bespoke configuration.

4.3 SCADA Metrics

(Almalawi et al., 2016), have proposed a novel data-driven clustering approach, based on the contents of SCADA protocols. They are able to identify a set of working states which represent the SCADA system. From there, it is possible to cluster the states into normal and abnormal. The states are derived from data extracted from the Modbus-TCP protocol, i.e: a) water flow pressure, demand and level; b) Command value status and setting; and c) Pump status and speed. A MITM attack was simulated and false water readings were reported, which would cause the tanks to empty. Traffic is legitimate Modbus-TCP so no alerts are generated. A possible solution to detect this would be to monitor all the tanks and their respective water levels, instead of individually.

(Gonzalez and Papa, 2007), developed a system to passively monitor Modbus-TCP communications and extract data relating to RTUs, e.g. device status and network topology. They developed three components: 1) Network Scanner; 2) Transaction Checker consisting of: pairs that match sets of messages which are passed on to the 3) Incremental network mapper, which maintains a dynamic data structure storing the network topology and device status information. (Nivethan and Papa, 2016), reviewed open source firewalls with regards to industrial control usage. They choose to use iptables due to its ability to perform application layer inspection. They identified some Modbus-TCP fields for use in their experiments, primarily focusing on the function code and the data field. They were able to prevent the following attacks and enumeration known to work on the ModBus-TCP protocol: a) Unauthorised Write; b) Unauthorised Read; c) Clear counters and diagnostic registers; d) Remote restart; e) Force PLC into listen-only mode; f) Report Server information; and g) Read device identification. This is a typical prevention mechanism which needs to be placed between the PLC and the HMI, so it would only be put into production if it is absolutely certain the blocked ac-

tions will not be needed in an emergency. Rather than filtering traffic which meets the criteria, real system operators are more likely to log the data for future analysis.

(Jardine et al., 2016), built a non-invasive IDS, which focuses on the legacy Siemens S7 protocol. They extract data from packet captures, such as read, write and logic code downloads. Additional traffic such as TCP/ARP/UDP are classed as 'Other'. They are able to identify intrusions using the following heuristics: a) Quantity; b) Temporal; c) IP (Communication between the PLC and others); and d) PLC Logic code download. A baseline was built on a network capture of a few hours, features are then verified by a domain expert. Their system is limited to one PLC per instance of the IDS, which needs to be configured to that specific PLC. (Hadziosmanovic et al., 2012), performed an analysis of network and host-based data to determine what is viable for use in detecting network anomalies. They focused on the results of the network analysis and analysed the application payload along with basic TCP/IP features. They mapped common communication patterns of an ICS network by extracting TCP flows and performing a coarse payload analysis. This was followed by extracting activity patterns for each device and protocol. As with other related work they used 5-tuple and Modbus function code. We have chosen six proposed systems discussed above to perform an analysis of our proposed metrics. They were chosen to specifically cover the three categories, generic network, physical, and SCADA specific. They are all able to detect their targeted threat, yet not one is able to detect the full range of threats.

5 NETWORK BASED METRICS FOR ICS PROTOCOLS

This section will discuss the use of advanced application layer metrics and how they are able to identify anomalies within both SCADA and process control enclaves. Section 4 highlighted common forms of network metrics that are able to detect IT related attacks. Many metrics do not consider the communication context. Focusing only on basic network characteristics, 5-tuple, offers less insight into the process of the underlying system and its security status. An advanced adversary (e.g. with high skill, resources, and domain knowledge) when infiltrating a network will attempt to conceal their actions. A common method is to disguise their communications to appear as if they are normal operation. With generic network metrics, this would not be detected. To address these shortcomings, a range of metrics will now be presented that take the communication context into account.

Table 2 contains the proposed network metrics for use in SCADA and process control enclaves. The metrics are broken down into three columns. *Metric*: this is a short hand for the specific measurement which is used as a reference to the issue. *Metric Computation*: is the formula used to the measure the specific metric. *Purpose*: this provides a concise description of the types of threats the metric could indicate. Not included in the table is direction of the traffic; either from the supervisory enclave or to the process control. It is possible to measure in a single direction or both. Enforcing this can identify breach of the trust boundaries.

5.1 Comparison of Proposed Metrics

Table 3 compares the current state-of-the-art detection methods against the proposed metrics. The comparison is subdivided into four groupings: A) a single metric; B) two metrics; C) three metrics; and D) four or more metrics. Each grouping inherits the previous grouping, thus allowing for a more comprehensive analysis of the traffic. For example, detecting a network scan (group A) can be done using only M0. Where as, detecting an unauthorised read (group B) would need M3 and M4, and since it is in group B, it will inherit group A's M0 which is also needed for the attack detection. We now discuss the groups in more detail.

Group A: Due to the consistently predictable nature of the process control network, it is easy to identify non-targeting scanning and obtrusive communication by using simple metrics such as distinct protocols, ports, IP addresses, and date timestamps.

Group B: By monitoring commands which get/set values of remote devices, we are able to detect abnormal values from legitimate (or illegitimate) devices. This can indicate an adversary attempting to determine the status of the current system to escalate privilege. By analysing the number of commands accepted or rejected along with the values, we are able to detect these actions.

Group C: The advantage of measuring the access of each device at a protocol level allows for greater insight into activities within the system which directly affect the SCADA domain, rather than basic TCP/IP communication patterns. By understanding the everyday operations at this level an analyst (or algorithm) could identify anomalous activities. Monitoring the command type and response at both the source (SCADA) and destination (Process Control) will enable better monitoring for replay and injection

Table 2: Proposed application layer metrics.

Metric	Metric Computation	Purpose
M0: Generic Protocol	Count the number of distinct protocols	Detect any non-SCADA specific communication, this could be a result of network probing.
M1: Firmware update	Count the number of update firmware commands	Detect any unexpected alteration of the behaviour of devices which might produce unwanted and unpredictable results.
M2: Set value	Count the number of set or update value commands	Detect any unexpected alteration of the behaviour of devices which might produce unwanted and unpredictable results.
M3: Get value	Count the number of get value commands	Detect an increase in monitoring commands which could result in a denial of service of the device.
M4: Accepted Command	Count the number of accepted commands	Detect an increase in accepted commands which could identify unexpected behaviour within the network.
M5: Rejected Command	Count the number of rejected commands	Detect an increase in rejected commands which could identify unexpected behaviour within the network.
M6: Command Type	Count the number of distinct command types	Detect any unexpected commands which could alter the behaviour of devices, or indicate unauthorised access on the network.
M7: Response Type	Count the number of distinct response types	Detect any unexpected responses which could indicate strange devices behaviour, or indicate unauthorised access on the network.
M8: Device Address	Count the number of distinct common address (Link Address)	Detect any unexpected devices on the network and identify most communicating devices which could indicate misconfigurations.
M9: Application Address	Count the number of distinct information object addresses (Application Address)	Detect any unexpected application addresses on the network which could identify misconfiguration of a device.
M10: Address	Count the number of distinct addresses (Link and Application)	Detect any unexpected addresses on the network which could identify misconfiguration of a device.
M11: Cause of transmission	Count the distinct CoT	Detect an unexpected CoT which could indicate abnormal behaviour within the network.
M12: Avg. Information Objects	Count the average number of Information Objects	Detect a large amount of data being transmitted to/from a device. Which could result in abnormal behaviour within the network.

Table 3: Comparison of proposed metrics and SoA detection methods.

Attack Stages	(Vasilomanolakis et al., 2016)	(Terai et al., 2017)	(Nivethan and Papa, 2016)	(Gonzalez and Papa, 2007)	(Jardine et al., 2016)	(Luchs and Doerr, 2017)	Proposed Metrics	Grouping
Reconnaissance								
Network Scan	•	•	•	-	•	-	M0	A
Device/Protocol Scan	•	-	•	•	•	-	M3,6,7	C
Report Server Information	•	-	•	•	-	-	M3,6,7	C
Read Device Identification	•	-	•	•	-	•	M7,12	B
Interference								
Command Replay	-	•	-	-	-	•	M4,5,7	C
Command Injection	-	-	-	-	-	•	M4,5,7	C
Unauthorised Write	-	-	•	•	•	-	M2,4	B
Unauthorised Read	-	-	•	•	•	-	M3,4	B
Remotely Clear Registers	-	-	•	-	•	-	M6,7	B
Rouge Device	-	-	-	•	•	-	M8,9,10	C
Firmware Tampering	-	-	-	•	•	•	M0,1	B
Denial of Service								
TCP/UDP Spam	•	•	•	-	•	-	M0	A
Remote Restart	-	-	•	-	•	-	M4,6,7	C
Force PLC into Listen Mode	-	-	•	-	-	-	M4,6,7,11	D
Covert								
Covert Comms.	-	-	-	-	-	•	M6,7,11,12	D
Stealthy Deception Attack	-	-	-	-	•	•	M2,3,4,6,7	D
Malicious Firmware	-	-	-	-	•	•	M0,1,5,6, 7,9,10,11	D

attempts.

Group D: These metrics are dependent on the protocol used. Because of this, we are able to measure the cause of a transmission (whether the packet is routine, manual, or as a reaction to an event). This can provide an indication of the desired effect an attacker is attempting to achieve, e.g. an intruder performing a DoS by initiating packets designed to cause the remote device to operate in diagnostic or other non-standard modes.

In the case of the stealthy deception attack (Kleinmann et al., 2017), it would be a matter of placing measuring sensors at multiple points across the network. Using a combination of all the metrics, it is possible to highlight this attack. Covert command channel proposed by (Lemay et al., 2016) would be detected by monitoring the values within the command's request and response (M6-7), along with the number information objects and cause of transmissions (M11-12). Malicious firmware where a self-propagating worm was developed to run within the CPU of the PLC such as (Spenneberg et al., 2016), would be detected by monitoring the device addresses (M9-10), firmware upload requests (M1) and command types used (M4-7). Unavoidable variations in these metrics caused by this attack will allow this attack pattern to

be discerned from the normal operations of the system.

6 CONCLUSION

This paper highlights the threat actors that exist and their capabilities in regards to an ICS network. This is important when choosing what defences to deploy. This work has expanded on (Gonzalez and Papa, 2007), and proposes novel metrics which are able to detect a wider range of threats. This work also addresses a gap in the existing state-of-the-art and commercial systems. In particular, (Terai et al., 2017; Zhang et al., 2015) and (Vasilomanolakis et al., 2016) are unable to detect adversaries which disguise their activities within the application layer. The proposed metrics can detect such intrusions. The metrics were developed with the intention of providing a classification system with deeper insight into a SCADA network. (Nivethan and Papa, 2016) and (Jardine et al., 2016), consider the application layer, but they are limited to a single host with limited visibility to the application. Without encoding knowledge of the system into a detection system it will require a user with domain knowledge to act on the results. As a next step we intend to apply the metrics to real world experiments to confirm their effectiveness. This can be validated by inputting the data into a SIEM and performing baseline comparison with known normal operations, as well as attack patterns. Finally, we plan to experiment with one class SVMs to discover malicious actions on the network. In conclusion, we show that by creating and analysing metrics at the application layer, it allows the detection of a multitude of realistic threat types, which provides more comprehensive detection capabilities compared to existing state-of-the-art methods. It allows for lightweight analysis which is suitable for multiple purposes, such as forensics, SIEM integration, features for enhanced machine learning approaches, and complying with legal requirements.

ACKNOWLEDGEMENTS

This work was funded by EPSRC project ADAMA, reference EP/N022866/1.

REFERENCES

Almalawi, A., Fahad, A., Tari, Z., Alamri, A., AlGhamdi, R., and Zomaya, A. Y. (2016). An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems. *IEEE Transactions on Information Forensics and Security*.

Gonzalez, J. and Papa, M. (2007). Passive scanning in Modbus networks. *Critical Infrastructure Protection*.

Hadziosmanovic, D., Bolzoni, D., Etalle, S., and Hartel, P. H. (2012). Challenges and opportunities in securing industrial control systems. In *Proceedings of the IEEE Workshop on Complexity in Engineering*. IEEE.

Jardine, W., Frey, S., Green, B., and Rashid, A. (2016). SENAMI: Selective Non-Invasive Active Monitoring for ICS Intrusion Detection. In *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy*.

Kleinmann, A., Amichay, O., Wool, A., Tenenbaum, D., Bar, O., and Lev, L. (2017). Stealthy Deception Attacks Against SCADA Systems. In *arXiv CS.CR*.

Lemaire, L., Vossaert, J., Jansen, J., and Naessens, V. (2017). A logic-based framework for the security analysis of Industrial Control Systems. *Automatic Control and Computer Sciences*.

Lemay, A., Fernandez, J. M., and Knight, S. (2016). A Modbus command and control channel. In *IEEE Systems Conference*.

Luchs, M. and Doerr, C. (2017). Last Line of Defense: A Novel IDS Approach Against Advanced Threats in Industrial Control Systems. In *International Conference on Detection of Intrusions and Malware & Vulnerability Assessment*.

Nivethan, J. and Papa, M. (2016). On the Use of Open-source Firewalls in ICS/SCADA Systems. *Inf. Sec. J.: A Global Perspective*.

Payne, S. C. (2006). A guide to security metrics. *SANS institute*.

Pendleton, M., Garcia-Lebron, R., Cho, J.-H., and Xu, S. (2016). A Survey on Systems Security Metrics. *ACM Computing Surveys*.

Rathbun, D. and Homsher, L. (2009). Gathering security metrics and reaping the rewards. *SANS Institute, Oct*.

Robinson, M. (2013). The SCADA Threat Landscape. In *ICS-CSR*.

Rudman, L. and Irwin, B. (2016). Dridex: Analysis of the traffic and automatic generation of IOCs. In *Information Security for South Africa*.

Shostack, A. (2014). *Threat modeling: designing for security*. Wiley.

Spenneberg, R., Brggemann, M., and Schwartke, H. (2016). Plc-blaster: A worm living solely in the plc. *Black Hat Asia, Marina Bay Sands, Singapore*.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., and Hahn, A. (2015). Guide to Industrial Control Systems (ICS) Security. Technical Report NIST SP 800-82r2, National Institute of Standards and Technology.

Terai, A., Abe, S., Kojima, S., Takano, Y., and Koshijima, I. (2017). Cyber-Attack Detection for Industrial Control System Monitoring with Support Vector Machine Based on Communication Profile. *IEEE European Symposium on Security and Privacy Workshops*.

Urbina, D., Cardenas, A., Tippenhauer, N. O., Valente, J., Faisal, M., Ruths, J., Candell, R., and Sandberg, H.

(2016). Limiting The Impact of Stealthy Attacks on Industrial Control Systems. In *NIST*.

Vasilomanolakis, E., Srinivasa, S., Cordero, C. G., and Mhlhuser, M. (2016). Multi-stage attack detection and signature generation with ICS honeypots. In *IEEE/IFIP Network Operations and Management Symposium*.

Wang, L., Jajodia, S., Singhal, A., Cheng, P., and Noel, S. (2014). k-Zero Day Safety: A Network Security Metric for Measuring the Risk of Unknown Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*.

Zhang, M., Wang, L., Jajodia, S., Singhal, A., and Albanese, M. (2016). Network Diversity: A Security Metric for Evaluating the Resilience of Networks Against Zero-Day Attacks. *IEEE Transactions on Information Forensics and Security*.

Zhang, Y., Wang, L., Xiang, Y., and Ten, C. W. (2015). Power System Reliability Evaluation With SCADA Cybersecurity Considerations. *IEEE Transactions on Smart Grid*.