Timing-based Anomaly Detection in SCADA Networks

Chih-Yuan Lin, Simin Nadjm-Tehrani, and Mikael Asplund

Department of Computer and Information Science, Linköping University, Sweden
{chih-yuan.lin, simin.nadjm-tehrani, mikael.asplund}@liu.se

Abstract. Supervisory Control and Data Acquisition (SCADA) systems that operate our critical infrastructures are subject to increased cyber attacks. Due to the use of request-response communication in polling, SCADA traffic exhibits stable and predictable communication patterns. This paper provides a timing-based anomaly detection system that uses the statistical attributes of the communication patterns. This system is validated with three datasets, one generated from real devices and two from emulated networks, and is shown to have a False Positive Rate (FPR) under 1.4%. The tests are performed in the context of three different attack scenarios, which involve valid messages so they cannot be detected by whitelisting mechanisms. The detection accuracy and timing performance are adequate for all the attack scenarios in request-response communications. With other interaction patterns (i.e. spontaneous communications), we found instead that 2 out of 3 attacks are detected.

Keywords: SCADA, Industrial Control System (ICS), Anomaly Detection, Traffic Periodicity

1 Introduction

A SCADA system is an Industrial Control System (ICS) that operates public and private industrial processes including critical infrastructures. Such systems are becoming increasingly dependent on information and communication technologies and being connected to the Internet. This poses new challenges related to cyber security while allowing improved flexibility and ease of use of the systems.

Being less protected and more sensitive to software updates (or malware protection updates) than a typical office environment makes additional security measures in ICS necessary. Also, ease of exposure to cyber attacks once the physical levels of security is breached (e.g. insider attacks) requires a new look at how to protect these critical environments.

Compared with standard information and communication systems, SCADA systems exhibit more stable and persistent communication patterns since the communications are triggered by polling mechanisms. Typically, a master device requests data from field devices such as Programmable Logical Controller (PLC) or Remote Terminal Unit (RTU) periodically in order to provide a real-time view of the industrial processes. This makes anomaly detection based on timing a potentially viable approach to detect unknown adverse events such as those potentially caused by insider threats.

The published litarature regarding timing variation detection in SCADA systems mostly aims at flooding-based attacks, recognizable as dramatic changes of the network throughput [1–3]. Some works also focus on directly modeling the inter-arrival times of periodic messages. These Intrusion Detection Systems (IDS) usually deploy relaxed detection thresholds to avoid high false positive rates. However, there are many attacks that only cause subtle changes on every single inter-arrival period in between communication messages. For example, TCP sequence prediction attacks can be difficult to recognize with existing techniques. To our knowledge, there is no SCADA-specific IDS that has successfully detected this kind of attacks.

In this paper, we propose a solution for timing-based anomaly detection in SCADA networks by monitoring statistical attributes of traffic periodicity. Our approach uses sampling distribution of the mean and the range to model the inter-arrival times of repeated messages in the same master-PLC/RTU flow. This approach has been widely used in statistical process control area to monitor the stability of processes. The contributions of this paper are:

- Analyzing the periodicity of SCADA traffic collected from real and emulated systems and showing that the traffic periodicity exists in both the request and response direction including some asynchronous events.
- Presenting three attack scenarios being formed of only valid requests/responses but breaking the traffic periodicity.
- Exploring the sampling distribution approaches for timing-based anomaly detection and showing that the proposed IDS can detect the timing-based attacks through changes of mean and dispersion in request/response inter-arrival times event though the change is small in every single inter-arrival time.

The rest of the paper is organized as follows. Section 2 provides the background. Section 3 presents the related work. Section 4 describes the threat model. Section 5 elaborates the proposed IDS. Section 6 describes how the attacks are generated for testing and evaluates the detection results. We conclude this paper in section 7.

2 Background

This section provides an overview of SCADA protocols and their communication modes. It also presents a brief introduction of sampling distribution of sample mean and sample range.

2.1 SCADA Protocols

The communication between the master device and field devices relies on SCADA-specific protocols built upon different communication technologies like serial communication and TCP/IP. In this work, we analyze three different protocols: Modbus, Siemens S7 and IEC 60870-5-104. These protocols are widely used in SCADA systems but allow different communication modes.

– Modbus: There are several Modbus protocols: Modbus RTU and Modbus ASCII are used in serial communication, often RS232. Modbus TCP is used for TCP communication. In this paper, we use a dataset based on Modbus RTU for testing and refer to it as simply Modbus. The Modbus protocol uses a synchronous request-response communication mode. The SCADA master initiates requests/commands stating the request type and starting address. The field device then responds by sending the requested data.
– Siemens S7: We use the S7-0x32 proprietory protocol on top of TCP/IP stack and refer to it as S7 in the rest of the paper. In addition to the synchronous communications, S7-0x32 PLCs may asynchronously send messages from predefined memory areas, called Parameter Items, under certain conditions. Moreover, S7 protocol allows requesting multiple Parameter Items in one message.
– IEC 60870-5-104: This is a standardized application layer protocol built upon TCP/IP stack. The protocol allows balanced/unbalanced communications. In the unbalanced mode, only the master can initiate communications to field devices. On the contrary, both the master and field devices can initiate communications in the balanced mode. This protocol allows both synchronous and asynchronous messages. The field devices can send Spontaneous and Periodic messages from predefined addresses, called Information Object Address (IOA). We will refer to the IEC 60870-5-104 protocol as IEC104 in this paper.

2.2 Sampling Distribution of Sample Mean and Sample Range

Sampling distribution of sample mean and sample range are two metrics widely used in statistical process control to monitor the stability of a production process. In these cases, quality assurance staff take a few sample sets of a certain attribute from produced products (e.g., weight of bottles) and calculate the sample mean and sample range for each sample set $X = \{x_1, \ldots, x_W\}$.

The sample mean is defined as $\bar{X} = \sum x_i/W$. It can provide a measure of central tendency. The distribution of $\bar{X}$ is called sampling distribution of the sample mean. For a finite number of sample means $\bar{X}_j, j = 1, \ldots, k$, one can compute their center of distribution as $\bar{\bar{X}} = \sum \bar{X}_j/k$ and standard deviation $\sigma_{\bar{X}}$ by the Central Limit Theorem (CLT). Based on CLT, for any population with mean μ and standard deviation σ, $\bar{X}$ tends toward being normally distributed with

$$\mu_{\bar{X}} = \mu \tag{1}$$

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{W}} \tag{2}$$

when the sample size increases.

The sample range $R_j = max(X_j) - min(X_j)$ can state the natural variation in a process. The distribution of R_j for finite sets of X_j is called sampling distribution of the sample range. The center of this distribution is $\bar{R} = \sum R_j/k$. People in this area usually assume the population they take samples from follows a normal distribution. Under this assumption, one can estimate the σ_R with $\bar{R}$ and sample size. However, we do not make any specific assumption on the distribution of the population. Instead we use quation (4) in section 5.2 to estimate it.

Sample mean and sample range display variations from their historical distribution when the process is stable. If the variations exceed predefined thresholds, Upper Limitation (UL) and Lower Limitation (LL), it means the system conditions changed. In this paper, we use mean and range to model the message inter-arrival times. For the sake of simplicity, we refer to the sampling distribution of the sample mean and the sample range as the *mean model* and the *range model* in the rest of the paper.

3 Related Work

IDSs that exploit the overall timing attributes such as average packet inter-arrival time and bytes sent in a certain time interval are an active research area. Barbosa et al. [4, 5] investigated the use of spectral analysis methods to uncover traffic periodicity. Udd et al. [6] propose and implement the TCP sequence prediction attack for the IEC104 protocol but this attack is not detected by their IDS, which uses average inter-arrival times of packets grouped by their header type. Other works presented so far are mostly validated with flooding/DDoS attacks [1, 2]. The main limitation of these approaches is the "semantic gap". These approaches demonstrate that traffic patterns can be used for anomaly detection but provide little insights about which packets caused the anomalies.

To enhance the detection ability for more attack types, IDSs exploiting deep packet inspection technologies and timing models have been proposed. Sayegh et al. [3] model the inter-arrival times between signatures (i.e., a sequence of packets) and validate their approach with large amount of injected signatures. Barbosa et al. [7] propose an approach to model the period of repeated requests in an orderless group. The authors evaluate this approach with Modbus and MMS [1] datasets without attacks and set relaxed thresholds to avoid high false positive rates. This makes it difficult to detect subtle changes within a single period.

More recently, sequence-aware intrusion detection systems are subject of interest. Yang et al. [8], Goldenberg and Wool [9], and Kleinmann and Wool [10–12] use deterministic finite automata to model the message sequences of IEC104, Modbus TCP and S7 respectively. Casselli et al. [13] model the sequence of messages in discrete-time Markov chains in order to detect sequence attacks. These approaches model the order of messages. As a result, they cannot detect timing changes if the order of messages are not changed in the attack.

[1] https://www.iso.org/standard/28059.html

4 Threat Model

Our work focuses on attacks composed of valid messages, which cannot be detected by simple whitelisting. This section introduces three attack scenarios and the expected violation of the mean and range models. This is to provide an intuition for our approach to anomaly detection.

4.1 Flooding-based Attacks

Bhatia et al. [1] propose and implement a Modbus flooding attack. It disrupts the operation of normal commands by sending a huge amount of same commands with different instruction values to the same address on the targeted PLC. In Report to the President's Commission on Critical Infrastructure Protection[2], a similar attack scenario is described. An attacker can cause water hammer effect and damage the system by rapidly opening and closing the major control valves on the water pipeline. These attacks rely on many repetitions of specific messages in a short time, and thus cause decreased average message inter-arrival time and violate the mean model as illustrated in Fig. 1. Blue points represent the sample mean in a sliding window containing four inter-arrival times (i.e., sample size = 4).

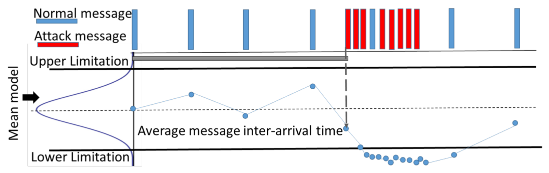

Fig. 1. Illustration of how a flooding attack violates the mean model

4.2 Injection Attacks

Due to lack of authentication mechanisms in many SCADA networks, an attacker can easily capture, modify, and inject a message into the network. Morris et al. [14] provide an overview of different injection attacks in their work. Fig. 2 illustrates the timing changes on the mean and range attributes caused by a single message injection.

4.3 TCP sequence prediction

TCP sequence prediction is an attempt to spoof a trusted host and inject a series of packets in a TCP session. Udd et al. [6] present a SCADA-specific TCP sequence prediction attack. In this work, the attacker can be someone who knows

[2] http://www.sei.cmu.edu/reports/97sr003.pdf

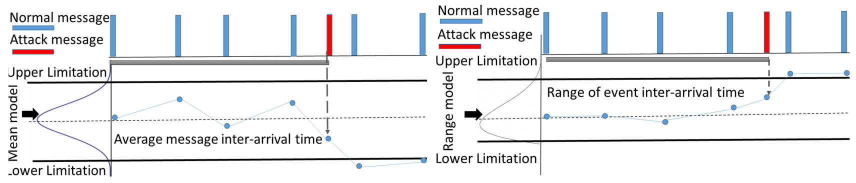

Fig. 2. Illustration of how a single injection attack violates the mean and range model

the traffic periodicity (e.g., an insider). Since the connection in SCADA networks can last for a long time, the attacker can insert a series of packets with predicted sequence numbers just before the real packets come. As a result, the network equipment considers the real packets as TCP retransmission packets and drops them. However, an attack packet may also be dropped if it comes later than the real packet. Fig. 3 presents the changes on range attributes.

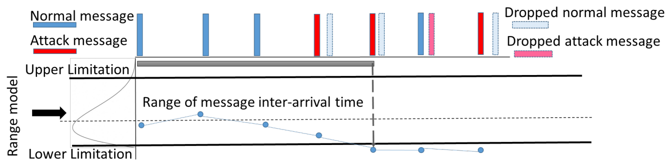

Fig. 3. Illustration of how the TCP prediction violates the range model

5 Proposed Intrusion Detection System

The proposed system contains three main modules: first, the extractor module extracts timestamps of events (i.e., messages having same features) in the same master-PLC/RTU flow. Second, the learner module builds the mean and range models and defines their thresholds. Finally, the detector module runs the testing data and raises alarms when the event inter-arrival times depart from learned models. Fig. 4 illustrates the workflow of the experiments. The solid rectangles are IDS modules. The extractor is written is Python and the others are in R language.

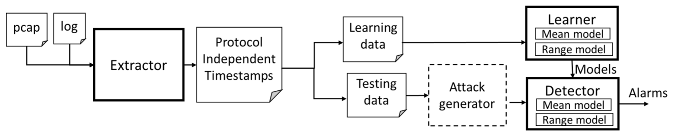

Fig. 4. System components used in our experiment workflow

5.1 Extractor Module

Based on the assumption that the master device periodically reads/writes data from certain memory addresses in a PLC/RTU, the extractor module identifies unique sets of request-response events from two attributes: request type and requested addresses. For the asynchronous events, the extractor only uses their addresses. Each of the attributes can contain a number of features. For example, the requested addresses of S7 comprise the Item Count (number of Parameter Items) and the locations of Parameter Items. The Extractor then outputs the event sets, together with their timestamps, in a protocol-independent text file.

5.2 Learner Module

The learner module is responsible for constructing the mean and range models of event inter-arrival times and setting detection thresholds.

Mean Model. For every event set $E = \{e_1, \ldots, e_{m+1}\}$, there exists a corresponding set of inter-arrival times $T = \{t_1, \ldots, t_m\}$ in the learning dataset. Instead of computing $\bar{\bar{X}}$ as described in section 2, we use CLT to construct the mean model based on equations (1) and (2). Since the μ and σ are unknown, we estimate them with the mean and standard deviation of the subpopulation T.

$$\mu \approx \bar{T} = \frac{1}{m}\sum_{i=1}^{m} t_i \tag{3}$$

$$\sigma \approx S_T = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(t_i - \bar{T})^2} \tag{4}$$

We set detection thresholds, UL and LL, as $\mu_{\bar{X}} \pm N\sigma_{\bar{X}}$, where N is a performance parameter called threshold level. Note that the LL needs to be positive to provide detection ability since all the inter-arrival times are positive.

$$\mu_{\bar{X}} - N\sigma_{\bar{X}} > 0 \tag{5}$$

Equations (1), (2), and (5) imply that the LL is positive when the sample set size $W > (\frac{N\sigma}{\mu})^2$.

Range Model. We continue with the selected sample size W and event set E. For every $W+1$ events, there exists a set of inter-arrival times $T^j = \{t_1^j, \ldots, t_W^j\}$, $j = 1\ldots, \left\lfloor \frac{m+1}{W+1} \right\rfloor$. We calculate the sample range R_j and $\bar{R}$ as described in section 2 and use equation (4) to estimate σ_R. The UL is defined as $\bar{R} + N\sigma_R$ but we set LL as the smallest event inter-arrival time in the learning period since the range model can be asymmetric.

5.3 Detector Module

During the detection phase, the module uses a sliding window which has the same window size as the sample size W. The module calculates the sample mean and sample range in each window and raises an alarm if the mean or range is outside the UL-LL interval. We consider an alarm as a true positive if there is at least one attack event in the window. Otherwise, the alarm would be considered a false positive.

6 Evaluation

To evaluate the effectiveness of proposed IDS, we implement and test the attacks stated in section 4. This section contains the description of datasets and the results regarding the detection accuracy and timing performance of our approach.

6.1 Datasets

The experiments use three datasets: (1) An emulated S7 traffic from 4SICS[3] ICS Lab with real Siemens devices, (2) A Modbus system log from real control networks for building ventilation systems provided by the company Modio, and (3) An IEC104 traffic from the virtual SCADA network RICS-EL that is developed in our project, emulating an electricity utility network extending FOI Cyber Range And Training Environment (CRATE)[4], as shown in Table 1.

We use the first 1/10 of data for learning and the remaining for testing. We find that the request-response events (X-req/X-res) come in pairs and have identical $\bar{T}$ and S_T rounded up to first decimal place. In addition, these events have lower Variance to Mean Ratio (VMR) compared with spontaneous events.

Table 1. Overview of datasets and event sets used for experiments

	Duration	# of Events	Event set	# of Events	$\bar{T}$ [s]	S_T [s]
S7	14 hrs	106378	S-req	53189	1.0	0.014
			S-res	53189	1.0	0.014
Modbus	5 days	161062	M-req1	5984	59.9	2.8
			M-res1	5984	59.9	2.8
			M-req2	1197	297.9	25.1
			M-res2	1197	297.9	25.1
IEC104	24 hrs	61714	I-Spont	16831	4.9	2.9

Fig. 5 illustrates the extracted events from the learning period for the three datasets. The Y axes indicate the sequence number of events in each event set

[3] https://download.netresec.com/pcap/4sics-2015/4SCICS-GeekLounge-151022.pcap
[4] Swedish Defense Research Agency (https://www.foi.se)

and X axes present the arrival times of these events. In S7 traffic, there exists only one pair of request-response event sets, S-req and S-res, which are presented as blue and red circles. In Modbus traffic, the extractor identifies 97 pairs of event sets in a master-PLC flow. These 97 pairs of event sets are further grouped by their sizes. The group 1 and 2 event sets present a persistent communication pattern, in which we select one event set in each group and refer to them as listed in Table 1. Though group 3 event sets contain too few events for learning, by manual examination, their events actually arrive around every 12 hours. By contrast, IEC104 traffic lacks persistent request-response periodicities. The request-response events only exists in a short period of time (from 4300-4800 seconds) and there is a rate change between 4400-4500 seconds. Most of the communications are spontaneous events and only event sets from IOA 10091 and 10092 show persistent and stable communication patterns. We select the 10091 event set for testing and refer to it as I-spont. Other spontaneous events are either not persistent (IOA 10001, IOA 5005) or not stable (IOA 10010-10017).

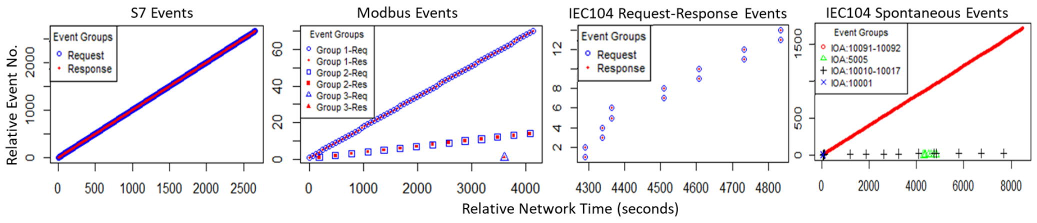

Fig. 5. Event arrival times of identified event sets for S7, Modbus and IEC104 data

6.2 Attack Generation

We implement an attack generator in Python to insert synthetic events for attacks presented in section 4. The generator divides testing data into many segments (see below) and inserts an attack in each segment. This section presents how to insert an attack for each attack type.

- Flooding-based Attack: We identify that the shortest event period in all our datasets is about 1 second. We generate synthetic flooding attack events every 100ms with some random variations. The variations are produced from Gaussian distribution with standard deviation of 10ms. An attack lasts for 1 minute. A segment is 20 minutes long for S7 dataset and 100 minutes long for the others.
- Injection Attack: The generator inserts one event in the beginning of every segment with the random variations sampled from the same Gaussian distribution as above. It also adopts the same setting of segments as above.
- Prediction Attack: The generator uses the learned $\bar{T}$ and S_T of inter-arrival times. It attempts to insert one malicious event in the time that is one S_T ahead of $\bar{T}$. An attack repeats the insertion 100 times in series. A malicious

event will be accepted only when it arrives earlier than the original event. Otherwise, it will be discarded silently. A segment is 1000 events long.

6.3 Detection Accuracy

We present the detection accuracy using True Positive Rate (TPR), False Positive Rate (FPR) and Overall Detection Rate (ODR). TPR is defined as number of true alarms divided by number of windows with attacks, FPR is defined as number of false alarms divided by number of windows without attacks, and ODR is number of detected attacks divided by number of attacks.

There are two tunable parameters in the proposed IDS, sample/window size (W) and threshold level (N). We do the following experiments with $W = 5$ and $N = 4$ for the S7 and Modbus datasets; $W = 5$ and $N = 3$ for IEC104 dataset. Fig. 6 shows the FPRs of range detector under different settings when we run the S7 test data without any inserted attacks. The FPRs in other datasets are similar. This shows stable FPRs over different sample sizes.

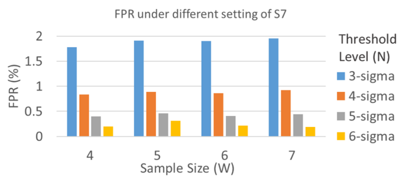

Fig. 6. False positive rates observed when tunning the W parameter (S7 data)

Table 2 is a summary of the detection performance in percentage (%). The results average the measures over segments. For the flooding attack, the mean model performs well (i.e., has high TPR) for all the datasets as we expected. To our surprise, the range model also has over 99% TPRs on flooding attack in the Modbus and IEC104 datasets. This is because the event inter-arrival times and ranges in Modbus and IEC104 datasets are in the order of seconds. However, the event inter-arrival times of flooding attack are in the order of milliseconds. Therefore, the revised range after the attack in milliseconds become lower than LL. For the injection attack, both the mean and range model perform well on request-response event sets. However, it is difficult to detect a single injected event in IEC104-spont events due to the high VMR. For the prediction attack, only the range model can be used to detect this kind of event as expected and its TPR is a bit lower since the IDS spends more time to detect an attack (see next section). However, it still shows a high ODR. All of the cases have FPR under 1.4%.

6.4 Timing Performance

The Mean-Time-to-Detection (MTTD) measurement is based on Average Number of Events (ANE). More specifically, it records the time when the IDS raises

Table 2. Detection performance

	Flooding					Injection					Prediction				
	Mean		Range			Mean		Range			Mean		Range		
	TPR	FPR	TPR	FPR	ODR	TPR	FPR	TPR	FPR	ODR	TPR	FPR	TPR	FPR	ODR
S-req	99.9	0.01	59.1	0.8	100	96.2	0.01	100	0.8	100	0.1	0	90.6	0.8	99.8
S-res	99.9	0.2	56.4	1.1	100	96.6	0.2	99.5	1.1	100	0.2	0.2	91	1.1	99.5
M-req1	99.8	0	99.6	1.1	100	83.3	0	83.3	1.1	100	0	0	92.8	1.3	99.1
M-res1	99.8	0	99.6	1.1	100	83.3	0	83.3	1.2	100	0	0	91.9	1.4	99.1
M-req2	99.8	0	100	0	100	83.1	0	100	0	100	4.9	0	100	0	100
M-res2	99.8	0	100	0.3	100	83.1	0	100	0.3	100	4.9	0	100	0.3	100
I-spont	99.8	0.4	98.1	1.1	100	2.4	0.4	3.7	1.1	13.3	2.8	0.4	72.4	1.1	92.8

its first alarm for an attack and counts how many events have passed since the beginning of the attack. ANE averages the measurements over segments. The MTTD is therefore defined as $ANE \times Avg\_period$. The Avg_period is 100ms for flooding attacks and learned $\bar{T}$ of each dataset for the other attacks. Table 3 shows the ANE and MTTD for different attacks. Most of the time, the proposed IDS can detect attacks immediately but prediction attacks take longer time to detect on average.

Table 3. Average number of event and mean-time-to-detection for different attacks

	Flooding		Injection		Prediction	
	ANE	MTTD	ANE	MTTD	ANE	MTTD
S-req	0.05	5ms	0.025	25ms	5.72	5.7s
S-res	0.05	5ms	0.025	25ms	5.72	5.7s
M-req1	0.4	40ms	0.08	4.8s	5	299.5s
M-res1	0.08	8ms	0.08	4.8s	5	299.5s
M-req2	0.08	8ms	0.08	23.8s	4	1190.8s
M-res2	0.08	8ms	0.08	23.8s	4	1190.8s
I-spont	3.87	387ms	3	14.7s	5	24.5s

7 Conclusions

SCADA traffic exhibits persistent and stable communication patterns. This paper studied three attack scenarios formed by valid requests only and then proposed an anomaly detection system, which uses sampling distribution of sample mean and sample range to model the timing of repeated events. We tested and evaluated the proposed IDS with Modbus, S7, and IEC104 traffic. The proposed IDS performs well in all the request-response events, resulting in over 99% for overall detection rate and less than 1.4% for false positive rates. However, it is difficult to detect a single injected message on spontaneous events due to the high VMR of the inter-arrival times.

Acknowledgement

This work was completed within RICS: the research centre on Resilient Information and Control Systems (www.rics.se) financed by Swedish Civil Contingencies Agency (MSB). The authors would also like to thank the support by Modio.

References

1. Bhatia, S., Kush, N., Djamaludin, C., Akane, J., Foo, E.: Practical Modbus Flooding Attack and Detection. Proceedings of the Twelfth Australasian Information Security Conference (AISC) (2014).
2. Valdes, A. Cheung S.: Communication pattern anomaly detection in process control systems. IEEE Conference on Technologies for Homeland Security (HST) (2009).
3. Sayegh, N., Elhajj, H.I., Kayssi, A., Chehab, A.: SCADA Intrusion Detection System Based on Temporal Behavior of Frequent Patterns. 17 th IEEE Mediterranean Electrotechnical Conference (2014).
4. Barbosa, R.R.R., Sadre, R., Pras, A.: A First Look into SCADA Network Traffic. IEEE Network Operations and Management Symposium (NOMS) (2012).
5. Barbosa, R.R.R., Sadre, R., Pras, A.: Towards Periodicity Based Anomaly Detection in SCADA Networks. IEEE Conference on Emerging Technologies & Factory Automation (2012).
6. Udd, R., Asplund, M., Nadjm-Tehrani, S., Kazemtabrizi, M., Ekstedt, M.: Exploiting Bro for Intrusion Detection in a SCADA System. Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security (CPSS) (2016).
7. Barbosa, R.R.R., Sadre, R., Pras, A.: Exploiting traffic periodicity in industrial control networks. International Journal of Critical Infrastructure Protection, vol. 13, pp. 52-62, Elsevier Science Publishers B. V. (2016).
8. Yang, Y., McLaughlin, K., Sezer, S., Yuan, Y., Huang, W.: Stateful Intrusion Detection for IEC 60870-5-104 SCADA Security. IEEE PES General Meeting (2014).
9. Goldenberg, N., Wool, A.: Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. International Journal of Critical Infrastructure Protection, vol. 6, no. 2, pp. 63-7, Elsevier Science Publishers B. V. (2013).
10. Kleinmann, A., Wool, A.: Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensic. The Journal of Digital Forensics, Security and Law, vol. 9, no. 2, open access publisher (2014).
11. Kleinmann, A., Wool, A.: A Statechart-Based Anomaly Detection Model for Multi-Threaded SCADA Systems. International Conference on Critical Information Infrastructures Security (CRITIS) (2016).
12. Kleinmann, A., Wool, A.: Automatic Construction of Statechart-Based Anomaly Detection Models for Multi-Threaded SCADA via Spectral Analysis. Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy (2016).
13. Caselli, M., Zambon, E., Kargl F.: Sequence-aware Intrusion Detection in Industrial Control Systems. Proceedings of the 1st ACM Workshop on Cyber-Physical System Security (CPSS) (2015).
14. Morris, T.H., Gao, W.: Industrial Control System Cyber Attacks. Proceedings of the 1st International Symposium for ICS & SCADA Cyber Security Research (2013).