Challenges and opportunities in securing industrial control systems

Dina Hadziosmanovic*, Damiano Bolzoni*, Sandro Etalle*† and Pieter Hartel*
*University of Twente and †Eindhoven Technical University
The Netherlands
{dina.hadziosmanovic, damiano.bolzoni, pieter.hartel}@utwente.nl, sandro.etalle@tue.nl

***Abstract*—Industrial Control Systems (ICS) are used for operating and monitoring industrial processes. Recent reports state that current ICS infrastructures are not sufficiently protected against cyber threats. Unfortunately, due to the specific nature of these systems, the application of common security countermeasures is often not effective.**

This paper summarizes experiences over a series of research efforts for building tools and mechanisms to improve the security and awareness in ICS. In particular, we discuss challenges and opportunities identified during an extensive analysis of ICS data resources. We believe that such insights are valuable for further research in the ICS context.

I. Introduction

The security of Industrial Control Systems (ICS) has gained increasing attention recently. There are two main reasons for this. First, ICS systems often control critical infrastructures. For example, ICS can be found in infrastructures such as power plants and power grid systems, water, oil and gas distribution systems or building monitoring (e.g., airports, railway stations), production systems for food, cars, ships and other products. A failure in such systems may endanger people's health and safety, damage industrial facilities and produce financial loss. Secondly, recent reports show a significant number of security incidents in these environments. For example, a security study of 291 utility and energy companies in the U.S. [11] states that 76% of the companies report that they suffered one or more security incidents in 2010.

The increasing number of security incidents in ICS facilities is mainly due to the fact that these systems are not built with security in mind. This manifests in a combination of technological and organizational weaknesses. In the past, ICS were separated from public networks, used proprietary software architectures and communication protocols. Built on the "security by obscurity" paradigm, the systems were less vulnerable to cyber attacks. Nowadays, ICS vendors increasingly use common communication protocols and commercial off-the-shelf software. Also, it is common to deploy remote connection mechanisms to ease the management during off-duty hours, and achieve nearly-unmanned operation. Unfortunately, the stakeholders seldom enforce strong security policies. User credentials are often shared among users to ease day-to-day operations and are seldom updated, resulting in a lack of traceability. Due to these reasons, ICS facilities became more vulnerable to internal and external cyber attacks. Although companies reluctantly disclose incidents, there are several published cases where safety and security of ICS facilities were seriously endangered [16].

There are two general approaches for improving the security in computer systems. Firstly, a detection system can anaylse data resources (host- or network-based) and alert on specific attack signatures. Such approach can only detect known attacks and thus cannot protect against unknown attacks. In the ICS context, the number of known attacks is relatively small, thus there are still few attack signatures available. This implies that the signature-based approach is even less effective in these environments than regular IT systems. Secondly, a detection system can rely on describing common system operations where any behaviour significantly different from common operation implies a potential threat. This (anomaly-based) approach, by design, has potential to detect previously unknown attacks. However, a large number of false positives constraints the wide deployment of such systems in real-world environments of regular IT systems. This is mainly due to a dynamic character of computer systems.

By contrast, ICSs seem to be suitable for this approach (e.g., have static IP addresses, operate a limited number of services, use dedicated devices). However, ICSs are also semi-automated, require continuous production, operate on proprietary architectures and communication protocols. Due to such closed character, the behaviour of real-world ICS facilities and the effectiveness of specific approaches remain largely unknown. To the best of our knowledge, there are no works that present empirical data analyses from real-world ICSs and provide an insight into the behaviour of these systems.

Contribution The main contributions of our work are the following:

- we perform different types of host- and network-based analysis on data traces coming from several real-world ICS facilities,
- we perform a structured evaluation of the results and highlight challenges for performing various data analyses in the ICS context.

II. Preliminaries

In this section we explain how a typical ICS system works. An ICS system consists of two main domains: the process field and a control room. Large systems may have more than one control room. The network infrastructure binds the two domains together. ICS users control the industrial process from

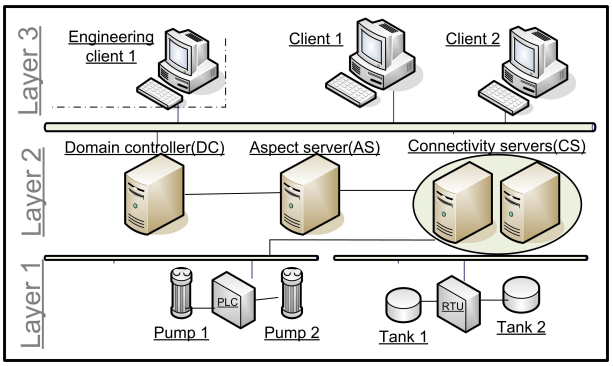

Fig. 1. ICS layered architecture

the control room and are provided with a real-time overview of the process field device parameters (data about tank loads, pump statuses, temperatures, etc). Depending on the underlying process, ICSs differ from each other. For example, a power-related ICS installation contains power switches and transformers while a water-related installation contains water pumps and valves.

A. System architecture

Figure 1 shows a typical ICS layered architecture. Layer 1 consists of physical field devices, PLCs (Programmable Logic Controllers) and RTUs (Remote Terminal Units). The PLCs and RTUs are responsible for controlling the industrial process, receiving signals from the field devices and sending notifications to upper layers. Layer 2 consists of ICS servers that perform several tasks. For example, Connectivity Servers process data from field resources in Layer 1 and present process changes to Layer 3. Aspect Server checks process configuration while Domain Controller performs user authentication. Various clients in Layer 3 represent ICS users. ICSs typically operate on several open-standard and proprietary protocols. For example, the communication between Layer 1 and 2 is typically performed over a vendor implementation of an open-standard ICS protocol, such as IEC [19], MMS [20], Modbus [21]. The communication in Layer 2, and between Later 2 and 3 is typically operated by a high-level ICS specific proprietary protocol. This communication often includes various Windows services (such as RPC).

B. Threats

We classify possible threats against ICS in two groups: system- and process-related threats [10]. As regular IT systems, ICSs are susceptible to threats exploiting software vulnerabilities (e.g., protocol implementation, OS vulnerabilities). We call these threats *system-related threats*. Such threats exploit low-level vulnerabilities, and typically occur at Layer 1 and Layer 2 of the ICS architecture (Figure 1).

However, due to the presence of a production process, ICS facilities are also prone to what we call *process-related threats*. These threats exploit weak process controls, and imply that an attacker obtains (e.g., through social engineering) or has user access rights and issues legitimate ICS commands to cause failures in the industrial process. Based on two possible places where an ICS user can interacts with the process, we distinguish two types of process-related threats: (1) threats that leverage access controls on field devices and (2) threats that leverage vulnerabilities of the centralised application. The first type of threat typically results in sending bad data to the ICS state estimation which can then produce errors in the system status analysis [12]. The second type of threat includes scenarios of performing legitimate user actions (from the control application) that can have negative impact on the process production or devices. These attacks target at high level (i.e., occur via the process control application) and typically relate to Layer 3 (Figure 1). Table I shows examples of system- and process-related threats in an ICS.

TABLE I
EXAMPLES OF SYSTEM- AND PROCESS-RELATED THREATS IN THE ICS CONTEXT

Type of threat	Example
System-related threat	an attacker uses malware to exploit a vulnerability in Windows service [13] an attacker carefully crafts message to cause heap overflow [9]
Process-related threat	an attacker inserts a wrong ratio of chemical components to cause process failure [10] an attacker changes the capacity of a tank to prevent the alarming system form going off [10]

C. Levels of analysis

To address security threats in computer systems, two types of analysis can be performed: host- and network-based. Host-based analysis implies processing of data traces coming from host devices (such as system logs, memory traces). Network-based analysis implies a capture and processing of data packets sent over the network. Due to a less intrusive character, network-based approaches are often preferred. However, host- and network-based analyses are complementary in the detection of different threats. For example, host-based analysis can provide information about internal system state, which is often infeasible to extract from network traces.

Both host- and network-based type of analysis can be performed with different levels of granularity. For example, a network-based analysis can be performed on the flow level (capturing combinations of device IP addresses and TCP port numbers) or at the packet content level (analysing TCP payload). An example of a low level granularity approach in the host-based analysis is the inspection of system logs. Inspection of memory traces represents an example of high granularity host-based analysis.

Generally, a high granularity approach (such as packet content analysis) is considered as more beneficial due to the fact that it provides more details. For example, data injection attacks (e.g., injection of executable code to trigger a buffer overflow sent via network packets) cannot be detected by a low granularity analysis (e.g., flow-based). However, low granularity approaches can give better context information (e.g, flow-based data can model trends of communication and detect DoS and DDoS attacks).

TABLE II
SUMMARY OF PROTOCOLS IN ANALYSED ENVIRONMENTS

Analysed ICS environments	Proprietary protocols	Open ICS-related protocols	Other open protocols
5	3	3[a]	10[b]

[a] *Modbus, MMS, IEC*

[b] *e.g., HTTP, SMB, SSH*

III. APPROACH

We perform an extensive analysis of data traces from real-world ICS facilities to understand the effectiveness of using different data resources and improve the security in the ICS context. We perform our analysis in two main steps:

1) data extraction
2) data interpretation.

Data extraction implies transformation and aggregation operations on raw data resources aiming at the extracting interesting data patterns. For this we use network- and host-based traces with different levels of granularity. The main goal of this step is to understand various aspects of data content, such as: data variability, periodicity and data completeness.

Data interpretation implies the evaluation of the analyses performed during the data extraction step. We base the evaluation on a set of criteria, compiled together with the stakeholders:

- Threat scope:
  1) chances of addressing system-related threats,
  2) chances of addressing process-related threats,
- Inclusion of process semantics,
- Ease of validation,
- Ease of application.

To perform the data interpretation, we evaluate each type of data analysis in step 1 for defined criteria as: low, medium or high. To this end, we present a discussion on the opportunities and tradeoffs for using a specific type of data extraction in the ICS context.

In the next section we describe analysed data resources.

A. Input data description

Our input data consist of two types of traces: (1) network traces and (2) application logs. Network traces origin from computer networks of five real-world facilities that perform gas distribution and water purification. The traces capture full network communication in Layer 1 and 2 of ICS architecture (Figure 1). The traces are sampled over different time periods in several days (ranging from 20 minutes to 50 hours of plant work). All analysed environments represent typical ICS deployments (e.g. facilities operate with up to 160 devices per site). In Table II we present different network protocols as seen in the network trace. We verify that each facility is operated by at least one proprietary and one open standard ICS-related protocol.

ICS application logs represent host-based trace about system activity. Depending on the size of the facility, there could be thousands of events generated per day. Such events describe system status updates, configuration changes, condition changes, user actions, etc. Unfortunately, we were only able to get the application log from one real-world ICS facility. The analysed log consists of 101,025 log entries collected during a 14 day period. The log comprises high-level process information from 153 field devices and 8 user workstations. Table III depicts a sample of a simplified ICS log.

TABLE III
EXAMPLE OF A SIMPLIFIED ICS LOG

W. shift	**Aspect of action**	**Type of action**	**Object path**	**User account**	**SCADA node**
2	-	Process Simple Event	-	-	CS01
2	Layout change	Configuration change	Plant1 /../ layout	Engineer1	EN01
1	-	Operator action	Plant1 / ../ tanks	Operator2	OP03

IV. IMPLEMENTATION

In this section we briefly describe implementation steps for performing data extraction.

The analysis of network traces is performed with different levels of granularity. Firstly, we map common communication patterns in ICS networks. This is implemented by extracting TCP flows from the network trace. Secondly, we perform a coarse analysis of payload content without parsing the underlying protocol. We do this by performing a bytewise analysis of TCP packet content. Finally, we perform a fine grained analysis of packet content by parsing ICS protocol and extracting activity patterns for each ICS device. For this we use Bro framework [15]. We reconstruct data address locations for every request-response pair of packets for a particular protocol. We then map data values found in network packets and build patterns of typical usage for all address locations in an ICS device. The prototype is implemented in C and currently supports Modbus parser only.

The analysis of ICS application logs is implemented in two main steps: log preparation and pattern mining. Log preparation implies attribute selection and log transformation (e.g., value aggregation). The pattern mining implies the application of an algorithm for maximal pattern mining on the log. For details please refer to [10].

V. RESULTS

In this section we present findings from the performed experiments. In particular, Section V-A presents empirical results from different types of data extraction analyses. In section V-B we discuss the effectiveness of the performed analyses and evaluate each approach from several aspects.

A. Data extraction

We illustrate interesting observations from the data extraction process through several examples. In particular, first 4 observations relate to the network-based analysis, while the last 2 observations relate to the host-based analysis.

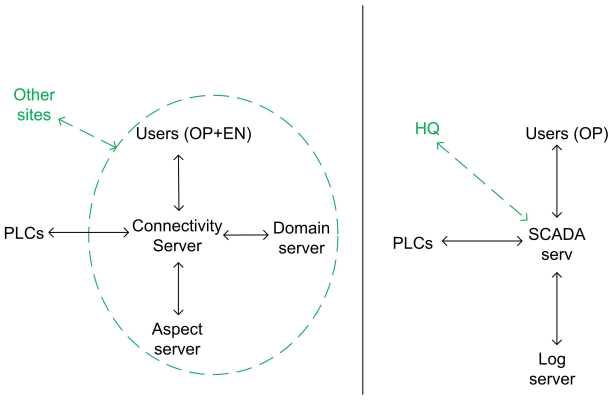

Fig. 2. Simplified communication patterns in two ICS facilities

1) Communication pattern: Figure 2 depicts a simplified representation of communication patterns in two different ICS facilities. The simplification is based on the fact that graph nodes represent clustered devices with the same role in the architecture (e.g., all PLCs represented as one node, all operator machines represented as one node). Figure 2 illustrates that different ICS architectures (e.g., different vendors and communication protocols) have similar high-level communication pattern that is predefined by design. This represents an opportunity to generalise the analysis of communication patterns across the domain of ICSs.

2) Communication character: To understand the communication character we analyse network activity over time. Interestingly, we found out that the ICSs often operate on long flows (e.g., TCP flows last several days), which is not common behaviour in regular computer systems. Also, in [2] authors perform a flow-based analysis of real-world ICS network traces and point out that the number of newly discovered flows over time is low but continuous. We argue that this observation relates to devices that do belong to the ICS architecture but do not directly participate in the production process (e.g., devices in Layer 2 and 3, Figure 1). We verify that the patterns of communication for PLCs (devices in Layer 1, Figure 1) do not change over time. We believe that this observation should be taken into concern when designing security solutions in the ICS context.

We verify that the main communication map is built quickly (e.g., 90% of communication links are seen within 5 minutes of network trace in 3 facilities). This implies that the learning curve of new communication patterns in process-related ICS network is short (which is typically not the case in regular IT systems).

3) Message type: For performing an in-depth type of analysis we consider 2 facilities that operate Modbus protocol. The facilities represent two different installations of an identical production process. To distinguish different types of ICS communication, we use the notion of message type. For us, the message type represents a unique combination of: IP source, IP destination, flow direction and Modbus function code (FC). Flow direction refers to request or response type of message.

TABLE IV
TYPES OF MESSAGES

Facility	# of devices	# of message types	Elapsed time (hours)	# different Modbus FC
A	14	36	5	2
B	16	57	3	3

Modbus function code represents a high level distinction of different types of Modbus messages. Table IV summarises the variability of different message types in two environments (facility *A* and *B*) over 1000000 network packets. Table IV shows that the devices communicate in a limited number of message types (column 3, Table IV) and a low number of protocol message types (column 5, Table IV). We verify similar behaviour in other packet samples of the same facilities over few days. This corroborates the assumption that ICSs behave in a repetitive manner with respect to the message types.

4) Message content: To understand the network message content we analyse TCP packet payload. First, we perform a byte-wise payload analysis. We acknowledge that this approach is suitable for distinguishing different message contents in some cases (e.g., 95% of total TCP packets in facility *A* carry identical payload). However, this behaviour cannot be generalised (e.g., due to high variability, byte-wise analysis in facility *B* performs poorly). We believe that an in-depth analysis of protocol usage can give a better idea about the variability of message content. Table V represents an example of the performed analysis on parsed network communication for one PLC.

The first column represents indexed address locations in a particular ICS device over a long stream of network packets. The remaining columns hold unique values read from a particular PLC from a specific address location. We acknowledge that for some address locations, the stream of data values represents a predictable data type. For example, data values that correspond to the address location *1* in Table V compose a sequence. Similarly, address location *2* holds only constant values. By contrast, address location *3* holds unpredictable, pseudo-random values.

After performing the analysis on all PLCs from both facilities, we derive two conclusions. First, both facilities operate on a large number of variables (e.g., a typical PLC operates on 5000 address locations). This gives an idea about the high complexity of the production process. We believe that the extraction of process states, based on such complex system would be challenging. Secondly, fortunately, a vast amount of address locations in facilities (up to 80% in facility *A*) store predictable data types (sequences or constants). A frequent usage of address locations with constant values contributes to the overall regularity in the packet content in facility *A*.

5) Frequency-based log analysis: To understand the character of application logs we apply pattern mining. In particular, the analysis is based on monitoring the frequency of occur-

TABLE V
MAPPED USAGE OF ADDRESS LOCATIONS FOR PLC 1 IN FACILITY A

Address location	Unique data values read from address location							
0	45	33	180	2	5	100	3	5
1	34	35	36	37	38	39	40	41
2	3	3	3	3	3	3	3	3
3	89	34	35	58	100	1	3	67
...	...	...	...	..	..	...	...	...
9998	...	...	...	...	...	...	..	...
9999	7	6	5	4	3	2	1	0

rence of specific log entries [10].

We verify that ICS log represents a "chatty" system (e.g., one event repeated 1,115 times in 8 hours of plant work in several days of log). Also, similar to the authors in [6], we found a large fraction of events that always appear with the same number of daily occurrences (e.g., timer-triggered event). Thus, a rare event, in a such environment, is likely to be anomalous. For example, an engineer operating from a machine that is usually inactive outside the working hours is considered suspicious. We believe that this feature represents an opportunity for defining a set of rules that will, when applied, trigger an alarm for undesirable frequency-based deviations. For example, the absence of a time-triggered log entry may indicate a device failure.

6) Distribution-based log analysis: In [17] the authors show that observed failures in logs tend to be described in many log entries that occur consecutively forming repetitive patterns. We verified that the high number of frequent patterns that we observed are not such burst of events and are spread through the whole day. Because of this we believe that the analysis of the log over a longer period of time can provide interesting insights in the content of the logs. For example, we could extract patterns that are present every day, and occur with the same (or similar) frequency of occurrence. These observations could define a profile of common ICS behaviour.

B. Data interpretation

Table VI presents the evaluation of the performed data analyses (shown as rows in Table VI). The evaluation is performed together with the stakeholders and is based on the criteria defined in Section III (shown as columns in Table VI). In the remaining of the section we provide a discussion on the evaluation results for each criteria.

1) Threat scope: Generally, network-based approaches have better chances in detecting system-related threats than application logs. In particular, low granularity network-based data analysis can provide an in-depth view on the packet content. therefore, such approach has high chances for detecting system-related threats, such as data injection attack. By contrast, both types of analyses on application logs are simply too high level to capture low level attacks. Therefore, the chances for detecting a system-related attack are low.

However, network-based countermeasures cannot detect process-related threats that occur in the ICS context. This is because the anomalies generated by such threats are typically not reflected in communication patterns/data (e.g., executable code data), and can only be detected by analysing data passed through the system at a higher semantic level. Thus, host-based analysis of application logs is suitable for addressing this type of threats. In this context, the distribution-based analysis is considered to provide a more realistic view on the system, thus the chances of detecting a process-related threat are higher than for the frequency-based analysis.

TABLE VI
EVALUATION OF DIFFERENT TYPES OF DATA ANALYSIS

Type of analysis	**Data interpretation criteria**				
	Threat scope		Inclusion of process semantics	Ease of validation	Ease of application
	System-related threats	Process-related threats			
Network-based					
comm. pattern	low	low	low	medium	high
comm. character	medium	low	low	medium	high
message type	medium	low	medium	medium	medium
message content	high	medium	high	low	low
Host-based					
frequency-based	low	medium	high	low	high
distribution-based	low	high	high	low	high

2) Inclusion of process semantics: Application logs represent a high level interpretation of network messages. Therefore, this resource carris the highest level of process semantics. By contrast, communication patterns only model the network architecture design and thus represent a poor resource of process semantic information.

Although application logs carry rich process semantics, the extraction of event consequences is still difficult. This is because the prediction of potential consequences of a performed action and the propagation of such consequences is not straightforward as it implies an in-depth analysis of process dependencies.

3) Ease of validation: Generally, approach validation in the ICS context is challenging. This is because, due to the sensitive character of ICS facilities, there are no publicly available repositories with real-world traces. Also, a realistic validation of a detection approach in the ICS context is hard due to the fact that there are few known attacks. In fact, having a common log attack dataset is infeasible since the consequences of actions relate to each single process differently.

4) Ease of application: Application of low granularity network-based approaches (such as communication pattern) is relatively easy since there are many tools used in regular IT systems that can perform this task. However, the presence of proprietary protocols limits the application of high granularity network-based approaches. For example, many proprietary protocols use dynamic port allocation. Due to this fact, any

generalisations on flow-based analysis become less powerful (since multiple flows cannot be grouped and used for prediction purposes). An alternative approach is to perform unsupervised traffic classification (e.g., [4]) and preferably cluster packets that belong to the same protocol. However, state-of the art approaches in traffic classification do not work well when analysing similar protocols (which is the case in ICS since these environments typically operate on all binary protocols). Finally, the application of an in-depth approach (such as distinguishing message types and understanding message content)in an environment with proprietary protocols becomes challenging of even infeasible due to the lack of knowledge about protocol specification.

VI. Related work

To detect anomalous behaviour in SCADA systems, authors use approaches based on inspecting network traffic [1], [8] validating protocol specifications [3], [18], and analysing data readings [12]. In [2] authors analyse flow-based network traffic from 2 real-world facilities and perform a systematic comparison with internet traffic.

Bigham et al. [5] use periodical snapshots of power load readings in a power grid system to detect if a specific load snapshot significantly varies from expected proportions. This approach is efficient because it reflects the situation in the process in a case of an attack. However, data readings (such as power loads) give a low-level view on the process and do not provide user traceability data.

In [14] authors propose to combine various log resources in a process control environment to detect intrusions. The detection is operator-assisted. To the best of our knowledge, only Balducelli et al. [1] analyse SCADA logs to detect unusual behaviour. There, the authors use case-base reasoning to find sequences of events that do not match sequences of normal behaviour (from the database of known cases).

In [7] authors discuss differences between traditional IT systems and ICS. Also, authors present a linear model of a physical control system and give a formal representation of an attack model on such system.

VII. Conclusion

This paper summarises experiences over a series of research efforts for building tools to improve the security and awareness in ICSs.

We perform the analysis in two steps: data extraction and data interpretation. In the first step we perform a multi-level analysis of network- and host-based data resources. In the second step we leverage stakeholders knowledge to identify relevant criteria for performing the evaluation of each approach from several aspects, such as: level of process semantics inclusion, ease of validation, ease of application. Then, we evaluate the effectiveness of each type of analysis against defined criteria.

Finally, for every criteria we highlight challenges that relate to the specific type of analysis. We support our claims with empirical results obtained by analysing data resources from real-world ICS.

References

[1] C. Balducelli, L. Lavalle, and G. Vicoli. Novelty detection and management to safeguard information-intensive critical infrastructures. *Int. J. Emergency Management*, 4(1):88–103, 2007.

[2] R. R. R. Barbosa, R. Sadre, and A. Pras. Difficulties in modeling SCADA traffic: A comparative analysis. In *13th International Conference on Passive and Active Measurement, PAM 2012, Vienna, Austria*, volume 7192 of *LNCS*, pages 126–135, Berlin, March 2012. Springer Verlag.

[3] C. Bellettini and J. Rrushi. Vulnerability analysis of SCADA protocol binaries through detection of memory access taintedness. In LTC John Hill, editor, *Proc. 8th IEEE SMC Information Assurance Workshop*, pages 341–348. IEEE Press, 2007.

[4] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In *Proc. of the ACM CoNEXT conference*, CoNEXT '06, pages 6:1–6:12, New York, NY, USA, 2006. ACM.

[5] J. Bigham, D. Gamez, and N. Lu. Safeguarding SCADA systems with anomaly detection. In *Proc. 2nd International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security*, LNCS 2776, pages 171–182. Springer Verlag, 2003.

[6] L. Burns, J.L. Hellerstein, S. Ma, C.S. Perng, D.A. Rabenhorst, and D.J. Taylor. Towards discovery of event correlation rules. In *Proc. IEEE/IFIP International Symposium on Integrated Network Management*, pages 345 –359, 2001.

[7] Alvaro A. Cárdenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. In *Proc. of the 3rd conference on Hot topics in security*, HOTSEC'08, pages 6:1–6:6, Berkeley, CA, USA, 2008. USENIX Association.

[8] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Sinner, and A. Valdes. Using model-based intrusion detection for SCADA networks. In *Proc. SCADA Security Scientific Symposium 2007*. Digital Bound Press, 2007.

[9] National Vulnerabiltity Database. National Vulnerabiltity Database CVE-2010-4709. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-4709, February 20011.

[10] D. Hadžiosmanović, D. Bolzoni, and P. H. Hartel. A log mining approach for process monitoring in SCADA. *International Journal of Information Security*, 11, 2012.

[11] Ponemon Institute. State of it security: Study of utilities and energy companies, 2011. http://q1labs.com/resource-center/white-papers.aspx.

[12] Y. Liu, P. Ning, and M. Reiter. False data injection attacks against state estimation in electric power grids. In *Proc. 16th ACM conference on Computer and communications security*, CCS '09, pages 21–32, New York, NY, USA, 2009. ACM.

[13] L. Murchu N. Falliere and E. Chien. W32.stuxnet Dossier, 2011.

[14] M. Naedele and O. Biderbost. Human-assisted intrusion detection for process control systems. Accepted for 2nd Int. Conf. on Applied Cryptography and Network Security (ACNS), 2004.

[15] V. Paxson. Bro: a system for detecting network intruders in real-time. *Comput. Netw.*, 31:2435–2463, December 1999.

[16] A. Rege-Patwardhan. Cybercrimes against critical infrastructures: a study of online criminal organization and techniques. *Criminal Justice Studies*, 22(3):261–271, Sep 2009.

[17] F. Salfner, S. Tschirpke, and M. Malek. Comprehensive logfiles for autonomic systems. In *Proc. 18th International Symposium on Parallel and Distributed Processing*, page 211, april 2004.

[18] Alfonso Valdes and Steven Cheung. Communication pattern anomaly detection in process control systems. In *2009 IEEE International Conference on Technologies for Homeland Security*, Waltham, MA, May 11–12, 2009.

[19] IEC 60870-5-104 - protocol specification.

[20] ISO/IEC 9506-2: Manufacturing message specification - part 2: Protocol specification.

[21] MODBUS TCP/IP messaging implementation guide.