A Model-based Approach to Self-Protection in SCADA Systems

Qian Chen
Electrical and Computer Engineering
Mississippi State University
qc34@msstate.edu

Sherif Abdelwahed
Electrical and Computer Engineering
Mississippi State University
sherif@ece.msstate.edu

Abstract

Supervisory Control and Data Acquisition (SCADA) systems, which are widely used in monitoring and controlling critical infrastructure sectors, are highly vulnerable to cyber attacks. Current security solutions can protect SCADA systems from known cyber assaults, but most solutions require human intervention. This paper applies autonomic computing technology to monitor SCADA system performance, and proactively estimate upcoming attacks for a given system model of a physical infrastructure. We also present the feasibility of intrusion detection systems for known and unknown attack detection. A dynamic intrusion response system is designed to evaluate recommended responses, and appropriate responses are executed to influence attack impacts. We used a case study of a water storage tank to develop an attack that modifies Modbus messages transmitted between slaves and masters. Experimental results show that, with little or no human intervention, the proposed approach enhances the security of the SCADA system, reduces protection time delays, and maintains water storage tank performance.

1 Introduction

Contemporary Supervisory Control and Data Acquisition (SCADA) systems adopt computer and Internet technology monitor physical system states by collecting data from remote field devices and control critical infrastructures resulting in a feedback loop. The quality and efficiency of industrial processes have been enhanced with the utilization of SCADA systems. However, SCADA systems are exposed to cyber attacks. This is because SCADA systems inherit vulnerabilities of computers and networks, as well as SCADA-specific vulnerabilities of system monitoring and controlling. A successful cyber attack can devastatingly damage properties, result in financial losses, or threaten personal lives. In 2011, Russian attackers compromised SCADA systems of public water utilities and destroyed a pump in Springfield, Illinois [4]. If the SCADA system was not protected in time, attackers might have burnt out all the water pumps in that area, which could result in a break in service to 2,200 rural customers.

Real-world cyber attacks are becoming increasingly sophisticated and organized. For instance, the Stuxnet worm [8] adopting four zero-day vulnerabilities attacked Iran's nuclear enrichment facilities, which damaged one-tenth of the centrifuges and led to the temporal termination of the uranium enrichment. Therefore, utilization of a onefold security control can no longer protect the security of SCADA systems. Autonomic computing technology uses multiple controls in the series (also referenced as the defense in depth strategy) to anticipate the SCADA system security state and regulate system behavior proactively with little or no human intervention. The contribution of this paper is the close integration of system monitoring, intrusion estimation, intrusion detection, live forensics analysis, and intrusion response mechanisms using autonomic computing technology to self-protect SCADA systems from the real-world sophisticated cyber attacks.

As far as protecting the cyberspace of SCADA systems, intrusion detection systems (IDSs) have appeared in the literature. Anomaly detection and signature detection are two major techniques that have been deployed to detect and classify SCADA-specific attacks [13]. The *anomaly detection* comparing real-time system performance with the normal system model can detect known and zero-day attacks. Observations that are deviated from the normal security state are attacks. Methodologies commonly used to establish the normal system model include statistics and machine learning theories. For instance, Wang and Stolfo [11] designed a statistical model using Mahalanobis distance to compare the similarity of observed traffic payloads against normal traffic for the identification of attacks. Neural networks, which are one of machine learning theories, have been applied to train normal datasets of the water storage tank by Gao

et al. [9]. Through experiments, man-in-the-middle attacks, replay attacks, and denial of service attacks have been developed to testify the detection accuracy of the detector. The neural network intrusion detection system provided a high detection accuracy to identify attacks that modify commands or inject malicious responses. The *signature detection* relies on matching observations to misuse patterns despite normal SCADA system behavior. This approach solely identifies and classifies known attacks. As a result, to identify zero-day attacks (attacks that exploit previously unknown vulnerabilities) the signature database must be upgraded frequently. Snort [2] is a widely used signature-based IDS detecting SCADA-specific attacks with pre-defined rules. One example of a successful application of Snort is presented by Yang et. al [12].

Most IDSs respond to attacks passively, i.e., only log attack activities but do not mitigate malicious impacts. They may also suffer from the problem of high false alarm rates [10]. These two issues of the utilization of an IDS lead to improper responses and high performance overhead. Therefore, only adopting IDSs to enhance SCADA system security is insufficient. In this paper, we applied autonomic computing approach to designing a front virtual machine (VM), on which autonomic components are installed and closely interact with each other to protect the SCADA system. A monitor is used to collect real-time network and system data (shown in Figure 1). These data are first processed and formatted, and then they are forwarded to intrusion estimation and intrusion detection modules. The intrusion estimation module estimates future system security states. If the estimated security state is abnormal, the intrusion response module will select and implement an appropriate response to eliminate the attacks or mitigate their impacts.

Sophisticated attacks, such as Stuxnet worms employing zero-day vulnerabilities, evade intrusion estimation and prevention processes can be detected by the intrusion detection system in real-time. The live forensics analysis tool located in the intrusion detection module learns signatures of unknown attacks such as evolving attacks or zero-day attacks. Detection algorithms and response mechanisms are updated in real-time with these signatures. Thus, similar attacks will be detected and mitigated in the future. A dynamic intrusion response system maps attacks to responses dynamically, and the most appropriate responses are initiated to eliminate the attacks.

Comparing our approach, which autonomously protects SCADA systems without disrupting normal infrastructure operations, with a comprehensive security design presented by Càrdenas et al. [5], our approach is easier to employ to networked platforms. As all autonomic components are installed on the front VM, only adding the front VM behind the firewall and in front of protected devices can realize a self-protecting SCADA system. The proposed self-protecting SCADA system can also be switched between fully-autonomous and semi-autonomous security modes. Therefore, response mechanisms that have low impacts on personal lives, economics, and property damage are executed without human intervention (e.g., the deployment of one time authentication). Cox [7] also employed autonomic computing technology to improve SCADA cyber security. In his design, responses that have low impacts are executed automatically to shorten the protection time. However, compared with our approach, Cox statically maps responses to the identified attacks. As a result, responses are not sufficient to mitigate attack impacts if the intrusion detection raises a false alarm.

The rest of this paper is organized as follows: Section 2 reviews the design of a front VM using the autonomic computing approach and presents techniques that have been applied to autonomic components. Section 3 provides a case study with the development of exploits on the testbed of a water storage tank to validate self-protection of the proposed approach. We conclude the paper in Section 4.

2 Self-protecting SCADA Systems

Autonomic computing aims at self-protecting SCADA systems from cyber attacks with minimal human intervention. As shown in Figure 1 the autonomic SCADA system estimates upcoming attacks, sends early warnings to system administrators, and autonomously or semi-autonomously implements responses to eliminate cyber attacks. Sophisticated cyber attacks that evade the first line of defense can be detected and classified by intrusion detection systems. Afterward, intrusion response systems assess recommended responses, and optimal ones are selected and implemented by the multi-criteria analysis controller (MAC) to defend the SCADA system against cyber assaults. The closed-loop feedback control design guarantees the system will be regulated to normal behavior. The utilization of live forensics tools analyzes unknown attack signatures. These signatures are used for updating detection algorithms and intrusion responses so that similar attacks will be detected and eliminated in the future.

All modules for establishing an autonomic SCADA system are installed on a front VM shown in Figure 2. Therefore, the configuration and implementation of the autonomic SCADA system are easy to realize. Note that the normal operation region of the SCADA system has been established offline applying expert knowledge, experimental results, literature reviews, and security guidelines. Normal operation regions vary in different physical system models and complexity of the SCADA envi-

ronment (described in Subsection 2.4). Details of techniques and theories used in each module are discussed as follows:

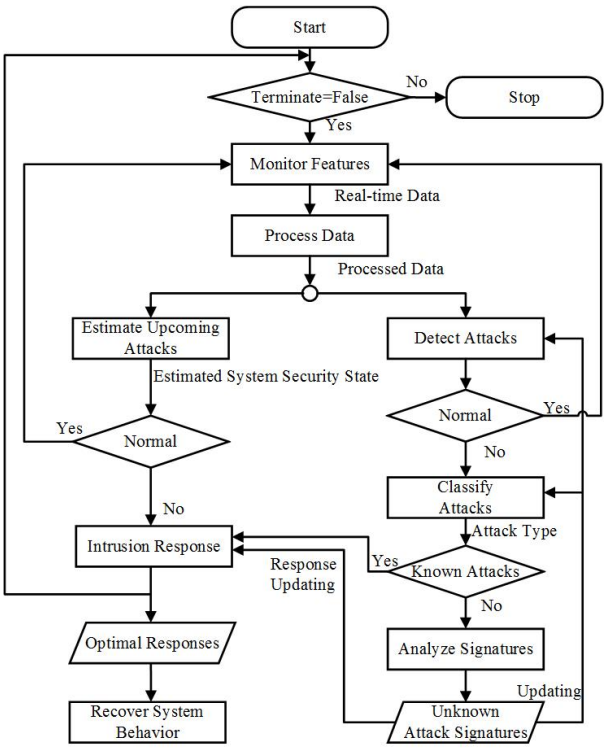


Figure 1: The Outline of the Self-protecting Autonomic SCADA System

2.1 The Monitor

The monitor module collects real-time data of the physical system performance and SCADA system security performance. To this end, we first select features that represent physical system behavior. For example, our approach is applied to autonomously secure a water storage tank from implementing malicious commands in Section 3. The water level of the water storage tank is monitored. Features that represent the security of SCADA systems are also monitored. These features include Modbus TCP/IP packet header, protocol data units, connection rates. Details of selected features can be found in our previous work [6].

2.2 The Data Processing Module

Real-time observations of selected features are directed to the data processing module. This module deletes datasets that contain missing data. Outputs of this module are formatted and pre-processed datasets, which are forwarded to the intrusion estimation module and intrusion detection systems.

2.3 Intrusion Estimation

The estimation module uses the historical observations of controlled variables of a physical model ($\underline{\omega}(k-1,r)$) and selected security features of the SCADA system ($\underline{\lambda}(k-1,r)$) to determine future performance of the physical system. The ARIMA [1] forecasting model has been successfully adopted in our previous work to estimate the future trend of the enterprise system performance [6]. In this work, we use the same forecasting model to predict the future performance of industrial control systems and future security states of the SCADA environment. Estimated values are denoted by hatted letters ($\hat{x}$). Normal letters (x) represent observations and underscored letters ($\underline{x}$) are historical observations. Given the historical observations, the forecasting model has the following form:

$$\hat{\omega}(k) = \phi_k(\underline{\omega}(k-1,r))$$

$$\hat{\lambda}(k) = \phi_k(\underline{\lambda}(k-1,r)$$

where $\hat{\omega}(k)$ and $\hat{\lambda}(k)$ denote estimated values of controlled variables of a control system (e.g. the water level of a water storage tank) and selected security features of the SCADA system (e.g. TCP/IP packet rates and TCP connection rates). $\underline{\omega}(k-1,r))$ and $\underline{\lambda}(k-1,r)$ are sets of previously observed controlled variables and SCADA system security features $\{\omega(k-1),...,\omega(k-r-1)\}$ and $\{\lambda(k-1),...,\lambda(k-r-1)\}$.

The estimation of the future security state of the SCADA system must involve current system security state, estimated values of control variables, and the estimations of selected security features:

$$\hat{x}(k+1) = f(x(k),\hat{\omega}(k),\hat{\lambda}(k))$$

where $x(k)$ is the security state of the SCADA system at time k. The estimated value of security state of the SCADA system $\hat{x}(k+1)$ is compared with the normal operation region, which has been established offline. The future security state of the SCADA system is abnormal if the estimated security state is deviated from the standard region. In this case, the recommended responses will be evaluated by the multi-criteria analysis controller (described in Subsection 2.6). Optimal responses (R) that have the lowest fuzzy scores are initiated to protect the system from the future attacks. As a result, with the implementation of appropriate responses, the estimated security state of the SCADA system at time $k+1$ is:

$$\hat{x}(k+1) = f(x(k),\hat{\omega}(k),\hat{\lambda}(k),R(k))$$

The intrusion estimation module can protect the SCADA environment from most known attacks that significantly influence controlled variables and selected security features. Sophisticated attacks that evade this first line of defense should be detected in real-time by the second line of defense.

2.4 Intrusion Detection

Intrusion detection is the second line of defense. Due to the time delay, the estimations of system security features and physical system behavior may not reflect real system security states. The intrusion detection system adopting anomaly and signature detection techniques can detect real-time known and unknown attacks.

- The anomaly detection technique: A normal region of a secure SCADA system is established using observations of normal system behavior with a Naive Bayesian classifier. Real-time processed datasets are first forwarded to the anomaly-based IDS. Observations that deviate from the normal region are attacks.
- The signature detection technique: The signature-based IDS can only classify known attacks since relevant attack regions have been built, or specific rules of known attacks have been pre-defined.

More information about Naive Classifier, pre-defined rules, and the application of the classifier and rules to IDS can be found in our previous work [6].

2.5 Live Forensics Analysis

It is difficult to select appropriate responses to eliminate or mitigate unknown attacks before signatures of unknown attacks, their causes, and their adverse impacts are revealed. Since the SCADA system must be available 24/7, live forensics analysis learning unknown attack patterns without disrupting system operations is added to protect zero-day and evolving attacks in our approach. The live forensics analysis module monitors and analyzes network traffic, front VM system performance, and auditing files using forensics tools (e.g., Wireshark [3]) and statistical theories (e.g., Naive Bayesian Network). Novel signatures are then applied to update detection algorithms of the IDS. The signatures are also sent to the intrusion response module for helping the multi-criteria analysis controller in selecting appropriate responses.

2.6 Intrusion Response

When estimations of SCADA system security or physical system behavior is abnormal, or real-time observations are detected as attacks, the intrusion response system must select the proper response to recover the physical system behavior back to normal. As the execution of an improper response may have a devastating impact on the external environment, properties, and personal lives, only responses that have low impacts can be executed autonomously. The multi-criteria analysis controller implements the evaluation of recommended responses. The assessment of each response must take into account four criteria, and they are:

- Criterion 1, Enhancement of Security: Confidentiality, integrity and availability are three fundamental facets of the security.The response that efficiently enhances the security of SCADA systems is assigned a fuzzy number 0, otherwise, assigned 1. Normalized values of responses that affect malicious activities can be assigned between the range of [0,1].
- Criterion 2, Operational Costs: The implementation of responses may exhaust computer and human resources. For example, the response "dropping the malicious commands" consumes computer CPU and memory resources to analyze protocol data units of communication messages. It also consumes storage resources for recording all known attack signatures. As a result, the implementation of a response must reduce their operational cost. Values for this criterion can be assigned between the range of [0,1]. The lowest cost responses are assigned 0; otherwise, they are assigned 1.
- Criterion 3, Maintenance of Normal Operations: The execution of responses are not permitted to disrupt normal performance of critical infrastructures. The responses that have no impact on normal operations are assigned 0. The value of Criterion 3 is initialized to 1 if the implementation of selected responses disrupting normal operations of the system; otherwise, values are assigned between the range of [0,1].
- Criterion 4, Impacts on Properties, Finance and Human Lives: SCADA systems controlling critical infrastructures have catastrophic impacts on personal lives, economics, and properties. The responses that greatly affect these features cannot be executed autonomously. Recommended responses such as "the termination of physical processes" and "the isolation of the master terminal unit (MTU) or the remote terminal unit (RTU)" have high impacts.

Five recommended responses (listed in Table 1) have been installed and configured offline. The multi-criteria analysis controller in Figure 2 evaluates these responses and selects optimal ones to react to cyber attacks. Since all criteria mentioned above are equally important, weights of each criterion for a response R_i is $W_{i,j}$ (j=1,2,3), which are assigned 0.33 (as the sum of the weights is equal to 1). For responses R_i, values of each criterion are represented by $C_{i,j}$. The Criterion 4 decides whether the implementation of the response requires human intervention or not. If the value of Criterion 4 is higher than 0.5, the implementation of such responses must be approved by system administrators since the responses have high impacts. We use "Semi-Auto" to represent a requirement of human intervention and "Auto" to represent autonomous implementation. The total value

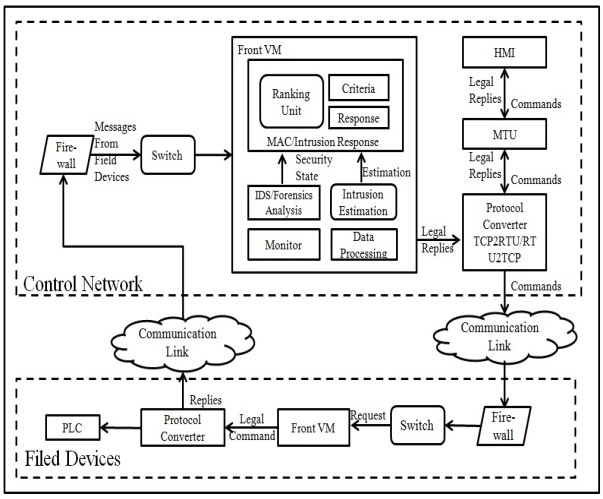


Figure 2: A Testbed of an Autonomic SCADA System

$(S_i, i = 1,2,3,4,5)$ for each response can be calculated using fuzzy-logic theory:

$$S_i = \sum_{j=1}^{5} W_{i,j} * C_{i,j}$$

The optimal response R_o is the one which has the lowest fuzzy score. As an example, R_o for mitigating unauthenticated command attacks is "Replacement of Compromised Devices", which is decried in details in Subsection 3.4. If the selected optimal response is not sufficient to mitigate attacks due to incorrect attack estimations or detection, the second best responses will be executed until the system security is normal.

3 A Case Study of the Water Storage Tank

To demonstrate the proposed approach, we present a case study of a self-protecting SCADA system that monitors and controls a water storage tank even if the SCADA system is compromised by cyber attacks. A front VM was added to the control network to enhance the security of the MTU and the HMI, while the same front VM was added to secure field devices such as RTUs and PLCs. The self-protecting approach reduces the time delay in SCADA system protection, closes the window of vulnerability, and relieves system administrator burden.

3.1 The Virtual Testbed

The water storage tank is modeled by a laboratory-scale control system in Mississippi State University SCADA Security Laboratory. In this control system, the MTU is connected to a Human-Machine-Interface (HMI) server via a RS-232 serial port, and the MTU connects to the RTU wirelessly [9].

We established a testbed with four virtual machines to simulate a similar SCADA system that monitors and controls a water storage tank. Two VMs perform as a RTU and a MTU, and each of them connects to a virtual machine, which performs as protocol converters (shown in Figure 2). Modbus TCP/IP messages were sent from one protocol converter located in front of the field devices. This converter first changed Modbus RTU responses, sent by field devices, to Modbus TCP/IP messages, and delivered the TCP/IP responses to the control network, in which the second protocol converter was placed. The second protocol converter converted Modbus TCP/IP messages to Modbus RTU messages and forwarded the messages to the HMI through the MTU located in the control network. Compared with Modbus serial protocols, Modbus TCP/IP is an open and standardized protocol, which supports long distance transmissions and enables communications between computers and field devices.

3.2 The Development of a SCADA Control System Exploit

To validate our approach, we carried out a malicious command that modified the alarm condition and altered the ladder-logic program of the water storage tank when the water storage tank was set to the "Auto" control mode. Once the water level had reached the low alarm condition (represented by L), the pump was turned on. On the other hand, when the water level increased to the high alarm condition (denoted by H), the pump was turned off automatically. The attack first evaded the authentication process, and then sent an illicit command to change L setpoint from 50.00% to 40.00%, and altered H setpoint from 60.00% to 70.00%. HH (the high high alarm) setpoint was modified to 80.00% from 70.00% and LL (the low low alarm) was changed to 10.00% from 20.00%.

In the normal case, the water level of the water storage tank was controlled between 50.00% and 60.00% as shown from sample 1 to sample 67 in Figure 3(a). After sample 67, the water level was increased to 82.82% by the malicious command. The highest value of the water level shown in the figure was higher than the HH setpoint. Afterwards, the pump was turned off, and the water level dropped to 40.11%. The control mode of the water storage tank was still "Auto." Thus, at sample 130, the pump was turned on again, and the water level increased to 69.07%. After the water increased to the modified H value, the pump was turned off, and the water level was dropped to 40.0%.

3.3 The Physical Model of the Water Storage Tank

A linear physical model of the water storage tank was established relying on the observations of the physical system when it was automatically controlled. The linear

physical model has the following form:

$$\omega = A * t + BR_o$$

where ω is the value of the controlled water level, t is the sample time, and R_o is the optimal control mechanism to defend the SCADA system against cyber attacks. The water level increases and decreases periodically when it is controlled automatically. From observations of the laboratory-scale water storage tank, we found out that the period of the water level was 80 samples. When $1 \leq t \leq 35$, $A = 0.256$ and $B = 51.181$. When $36 \leq t \leq 39$, $A = -1.976$ and $B = 62.090$. When $40 \leq t \leq 45$, $A = 0.03249$ and $B = 56.71783$. When $46 \leq t \leq 80$, $A = -0.202$ and $B = 56.686$.

We adopted a time series ARIMA model to predict future values of the water level based on the linear physical system model and historical data of the water level. Estimations of the water level are represented by the blue line in Figure 3(a). The green line shows real-time observations of the water level. The ARIMA model can correctly estimate and detect abnormal system behavior as estimations are almost equal to observations, which are deviated from the normal operation region.

3.4 Evaluation of Recommended Responses by the MAC

Table 1 is the rankings of five recommended responses for the protection of the SCADA system from the spoofing attack. The initial fuzzy-values of four criteria for recommended responses were provided based on experimental results and expertise knowledge. The optimal response that evaluated by the MAC was "Replacement of Compromised Devices." As this response may have a high impact on disruption of normal operations, the implementation of the response must be authorized manually. Figure 3(b) shows that, at sample 94, the attack modified alarm conditions, and the water level was abnormally increased to 65.99%. At sample 104 when "Replacement of Compromised Devices" was implemented, a replica PLC containing original ladder-logic programs replied to the MTU and sent commands to control water level of the critical infrastructure. As a result, the water level was regulated back to normal rapidly and efficiently with the application of autonomic computing technology.

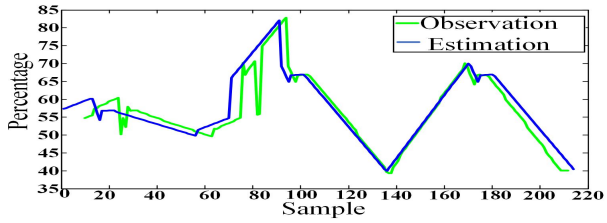
Since adversaries masqueraded as a legitimate user, the signature-based IDS did not identify the attack. The live forensics tool analyzed log files to learn causes of the attack. As a result, the rule containing the modified L,H, LL and HH setpoint values were added to the signature database. Therefore, the similar attack will be eliminated in the future.

4 Conclusions and Future Work

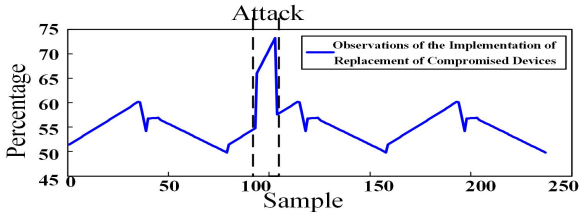
In this paper, autonomic computing technology has been used to self-protect the SCADA system from cyber attacks. This new technology integrates current security solutions so that the system can proactively monitor, estimate, detect, and react to known and unknown attacks with little or no human intervention. It also ensures the SCADA system is accessible 24/7. We applied the proposed approach to enhance the security of a SCADA system, which controls and monitors a water storage tank. Through the experimental result, we validated that the autonomic SCADA system maintained normal infrastructure operations and regulated the water level back to the normal operation region when alarm conditions were changed by attackers. The overhead time for identifying and protecting the SCADA system was short. It cost 22 sample time to regulate the water level back to normal. In the future, we will simulate more sophisticated cyber attacks to validate the efficiency of the approach. In addition, we will also employ autonomic computing to self-protect the next generation SCADA systems from cyber assaults.

Table 1: Assessment of Recommended Responses Example for Unauthenticated Command Attacks

Ranking	Response	C1	C2	C 3	C 4	Total Value (Auto or Semi-Auto)
2	Dropping Malicious Commands	0.5	0.3	0.1	0.2	0.3 (Auto)
4	Termination of Physical Processes	0	0.8	1	1	0.6 (Semi-Auto)
1	Replacement of Compromised Devices	0	0.5	0	0.5	0.17 (Auto or Semi-Auto)
3	One time authentication	0.8	0.3	0.2	0.2	0.43 (Auto)
5	Isolation of Compromised Devices	0.5	0.8	0.8	0.6	0.7 (Semi-Auto)



(a) Observations and Estimations of the Water Level Without Self-Protection



(b) Autonomic SCADA System Self-protects from Unauthorized Attacks

References

[1] The arima procedure. http://www.okstate.edu/sas/v8/saspdf/ets/chap7.pdf.

[2] snort_inline. http://snort-inline.sourceforge.net/oldhome.html.

[3] Wireshark, 2013. http://www.wireshark.org/about.html.

[4] AVILES, M. Scada attack on city water station: What really happened?, 2011. http://blogs.ixiacom.com/ixia-blog/cyber-attack-scada-water-pump-attack/.

[5] CÁRDENAS, A. A., AMIN, S., LIN, Z.-S., HUANG, Y.-L., HUANG, C.-Y., AND SASTRY, S. Attacks against process control systems: Risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (New York, NY, USA, 2011), ASIACCS '11, ACM, pp. 355–366.

[6] CHEN, Q., ABDELWAHED, S., AND ERRADI, A. A model-based approach to self-protection in computing system. In *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference* (New York, NY, USA, 2013), CAC '13, ACM, pp. 16:1–16:10.

[7] COX, D. P. *The Application of Autonomic Computing for the Protection of Industial Control Systems*. PhD thesis, University of Arizona, 2011.

[8] ET AL., M. A. Stuxnet under the microscope. *ESET* (January 2011).

[9] GAO, W., MORRIS, T., REAVES, B., AND RICHEY, D. On scada control system command and response injection and intrusion detection. In *eCrime Researchers Summit (eCrime), 2010* (Oct 2010), pp. 1–9.

[10] STAKHANOVA, N., BASU, S., AND WONG, J. A taxonomy of intrusion response systems. *Int. J. Inf. Comput. Secur. 1*, 1/2 (Jan. 2007), 169–184.

[11] WANG, K., AND STOLFO, S. Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection*, E. Jonsson, A. Valdes, and M. Almgren, Eds., vol. 3224 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 203–222.

[12] YANG, Y., MCLAUGHLIN, K., LITTLER, T., SEZER, S., PRANGGONO, B., AND WANG, H. Intrusion detection system for iec 60870-5-104 based scada networks. In *Power and Energy Society General Meeting (PES), 2013 IEEE* (July 2013), pp. 1–5.

[13] ZHU, B., AND SASTRY, S. Scada-specific intrusion detection/prevention systems: A survey and taxonomy. In *the 1st Workshop on Secure Control Systems, Stockholm, Sweden, 2010.*