A Review of SCADA Anomaly Detection Systems

Iñaki Garitano[1], Roberto Uribeetxeberria[2] and Urko Zurutuza[3]

Abstract The security of critical infrastructures is decreasing due to the apparition of new cyber threats against Supervisory Control and Data Acquisition (SCADA) systems. The evolution they have experienced; the use of standard hardware and software components or the increase of interconnected devices in order to reduce costs and improve efficiency, have contributed to this. This work reviews the research effort done towards the development of anomaly detection for these specific systems. SCADA systems have a number of peculiarities that make anomaly detection perform better than in traditional information and communications technology (ICT) networks. SCADA communications are deterministic, and their operation model is often cyclical. Based on this premise, modeling normal behavior by mining specific features gets feasible.

1 Introduction

For decades Supervisory Control and Data Acquisition (SCADA) systems have been commonly used to continuously monitor and control different kind of processes on Critical Infrastructures (CI) such as industrial processes, power industry, water distribution and oil refineries. Many of them control nations' critical components, like nuclear power generation, public transport, wastewater plants and so on. Based on this, the success of an attack can cause serious consequences. Nowadays, many vulnerabilities have been released on SCADA systems and software, even if hosting Operating Systems keep being the most exploited.

[1] Electronics and Computing Department, Mondragon University. Goiru Kalea, 2, 20500 Arrasate-Mondragon, Spain. igaritano@eps.mondragon.edu

[2] Electronics and Computing Department, Mondragon University. Goiru Kalea, 2, 20500 Arrasate-Mondragon, Spain. ruribeetxeberria@eps.mondragon.edu

[3] Electronics and Computing Department, Mondragon University. Goiru Kalea, 2, 20500 Arrasate-Mondragon, Spain. uzurutuza@mondragon.edu

On July of 2010 the Belarusian company VirusBlokAda discovered an especially designed worm for SCADA systems on a computer in Iran called Stuxnet [1, 2]. Stuxnet was specifically tailored to modify processes under control of Siemens' WinCC/PCS 7 SCADA software. The worm spread to more than fourteen companies attached to USB memory sticks. It was such a complex malware that authors used four zero-day vulnerabilities against Windows, written in different programming languages, and it was signed with two stolen digital certificates. Furthermore, it could be updated through P2P technology. Other well-known cyber attacks against CIs have been reported, like the one at Marrochy Water Services in Australia [3] or the one at Davis-Besse nuclear power plant in Ohio [4]. In the last case the SQL/Slammer worm broke in the nuclear power plant's security management system, leaving it unavailable during five hours.

The SCADA system vulnerabilities allow attackers achieve their goals, causing serious consequences like loss of reputation, economic loss, environmental disasters or even human casualties.

Historically, security measures applied to standard ICT have been also used for SCADA systems. But firewalls do not understand industrial protocols, signature-based intrusion detection systems (IDS) have the lack of specific signatures, so they are not prepared enough. Due to differences between these two kinds of systems, requirements like availability or real-time operability make specially designed security measures necessary.

The main issue of these systems is that they must be permanently available, even if they are attacked with unknown or zero-day attacks. Because of that, especially designed anomaly detection-based IDS have to be developed. This way, detection and protection against new kind of attacks will be possible.

In the next sections a brief description about SCADA systems is given, and after that research on IDS and anomaly detection on SCADA is reviewed.

2 SCADA systems security

In the past, SCADA systems were believed to be secure. They used specially designed hardware and software, proprietary protocols and isolated networks [5]. Nowadays they have evolved into standard platforms, using standard hardware and software and are increasingly interconnected. The interconnection of these systems and the use of standard hardware and software components make ICT vulnerabilities and attack methods target SCADA systems too, even if the standard hardware and software is more tested than ever, and consequently more secure. On the other hand they have made possible to cheapen the implementation costs.

SCADA systems have usually been protected using standard ICT security measures even if ICT security controls and measures such as firewalls and IDS do not suit to specific industrial and CI communication protocols' requirements. These security measures are needed to protect ICT systems against common attacks, as worms, viruses and denial of service (DoS). They are capable of

detecting specific attacks when signatures exist. But measures to detect unknown behavior from the normal operation of CIs are also needed.

For example, valve opening and closing orders are common in industrial control SCADA systems. The time for those actions is often deterministic so delaying the closure order can cause serious consequences. If expected time sequences of the valve operations are not taken into account for profiling normal behavior of a SCADA system, some attacks may not be detected. Lets consider a valve that regulates the pressure of substances in a chemical plant. If a valve closure order is delayed, the pressure could be significantly increased causing leaks, although SCADA system will not detect anything abnormal.

For SCADA systems it is not enough to ensure that known traffic goes through the network. It is also necessary to control time periods, parameter values, command's orders and many more variables in order to detect anomalous activity.

3 Intrusion Detection Systems

Several formal definitions of Intrusion Detection systems exist. National Institute of Standards and Technology (NIST) define IDS as the process of monitoring events in a computer system or network and the analysis of such events looking for intrusion traces. In 2006, S. D'Antonio, F. Oliviero and R. Setola [6] define IDS like the art of detecting malicious, uncommon or inappropriate actions of a system in a computer or in a whole network.

Intrusion detection techniques can be classified based on different functional characteristics: Information source, analysis strategy and response.

From the beginning IDS researches have been working with data coming from diverse sources trying to identify the existence of an intrusion. These data can be divided into three main groups: those obtained from a machine or host, those obtained from monitoring a network, and finally data obtained from the execution of applications.

Based on this classification, we can consider host-based intrusion detection systems (HIDS) and network intrusion detection systems (NIDS) as the most used ones. Host-based IDSs analyze information related to the host activities and states such as file-system modifications, applications logs and so on. In the other hand network intrusion detection systems analyze the traffic generated by a set of devices.

NIDS are more frequently used in SCADA networks. Due to the limited resources of the SCADA components, HIDS sensors cannot be installed [7].

If we consider the analysis strategy, we can classify IDS as misuse intrusion detection or anomaly detection systems.

A misuse detection based IDS monitors the activities of a system and compares them with signatures of attacks that are stored in a database. This kind of IDS have high accuracy rates, however, due to the high increase of new attacks and the continuous variants of them it is extremely difficult to have an updated set of rules.

On the other hand, anomaly detection depends greatly on the supposition that users and networks behave in a sufficiently regular way and therefore, any significant deviation from such behavior could be considered as an evidence of an intrusion. The biggest advantage of anomaly detection is that the system is capable of learning the studied object's normal behavior and from that point detects deviations classifying them as intrusions. One of the biggest problems is the high rate of false positives. Another disadvantage is the lack of clarity of the process; it is a fuzzy process. A patient intruder could work slowly and act cautiously in order to modify the profile of the users and make his own actions become acceptable for the IDS not generating any alert as they should (false negatives).

Most of IDSs trigger a basic response method when they detect an attack: a notification. This kind of response is passive and its only aim is to inform the administrator about the occurrence of an attack. During the last years though, automatic response to attacks have been considered and have gain popularity. This is known as active response or automatic response.

In the beginning, NIDS were capable of identifying single packet. But nowadays SCADA attacks may be very complex and few times are composed by a single step, but sequences of single packets or steps. This causes a flood of alerts that the analyst needs to review, resulting in a high cost activity and making difficult the detection process. Thus, a correlation process is needed after detection.

Intrusion detection systems can act passively without disturbing real-time traffic. They can even block traffic that is clearly malicious or alert if something is abnormal. But there is not a unique solution and their functionality can be powered in combination with other security techniques.

4 Anomaly detection

Lane, T. and Brodley, C.E define Anomaly detection as follows [8]: Anomaly detection attempts to quantify the usual of acceptable behavior and flags other irregular behavior as potentially intrusive.

A normal behavior of any system or process must be defined in an attack-free environment. System measures have to be identified as features for every process in order to learn the normal situation.

There exist many techniques used to obtain the model for a normal behavior, and thus develop an anomaly detection system. They can be classified as: knowledge-based methods, statistical methods and machine learning based methods.

Many research works claim that anomaly detection is better than rule based detection for industrial environments [9, 10]. SCADA networks are more predictable than ICT networks as they operate in a regular fashion and often perform same operations continuously.

Oliviero *et al.* [6] present two works to improve security of critical infrastructures. The first one is an IDS based architecture, the second, a method to

extract user's behavior in real time. The IDS architecture of their proposal is composed of a network scanner, a data processor and a classifier. The first scans the network traffic and stores it. The processor transforms the data to easy the feature extraction process. Finally the classifier decides if data is valid taking into account extracted features.

In the second work, in order to improve the classification criteria for the real-time extraction and modeling of users behavior it is mandatory to extract a set of parameters from network traffic describing statistical relationships between sessions. Usually monitoring techniques classify packets by grouping them into flows. Traffic flow is defined as a set of packets passing at a network point during a time interval and having common properties. In this work authors propose a monitoring system framework called DiFMon (Distributed Flow Monitoring) [11]. This system is responsible for packet capturing and flow exporting.

In [12] a new kind of an anomaly based IDS is proposed. It defines operation profiles using Stochastic Activity Network (SAN) models. This way, defined profiles can be used as intrusion detection rules. Also protocol messages and time distribution of activity are used to detect attack traces. For the correct operation of this system, SAN models for every process must be developed by identifying all possible operations. After that, Bayesian belief network formalism is applied in order to calculate the distribution probability of each operation. Their IDS is used to detect Modbus memory corruption attacks.

Düssel *et al.* [13] propose a payload based real-time anomaly detection system. Their system is protocol independent and it is able to detect unknown attacks. This method takes into account the similarity of the communication layer messages from a SCADA network.

Four components compose their anomaly detection system: a network sensor, a feature extractor, a similarity processor and the anomaly detection component. The network sensor captures communication layer messages using the known Bro IDS [14]; then, TCP payload data is extracted and sent to the feature extractor. Byte sequences are placed in a multidimensional feature space. The next component finds similarities of byte sequences. This similarity is based on the distance of the vectorial representation of the sequences. Anomaly detection system compares captured byte sequence data with normal behavior byte sequences, looking for dissimilarities that are presented as anomalies. They obtained an unknown attack detection ratio of 88%-92%, with a 0.2% of false positive level in their experiments.

In order to avoid security problems created by the use of TCP/IP protocol in industrial control networks, Gonzalez *et al.* [15] have proposed a passive scanner. This scanner analyzes Modbus protocol communications, in order to get information about network topology and configuration and state of control devices. Their tool allows save activities, detecting intrusions and analyzing taken actions. The information captured by the Modbus scanner is separated into network flows. It then obtains its dynamic data structure, saving status and network topology information.

Cucurull *et al.* propose in [16] a k-means clustering algorithm for anomaly detection. In the training phase, maximum, minimum and threshold of the gathered data are calculated. Then, at testing time, the system's proper behavior is

evaluated, comparing this data with the normal behavior pattern. The thresholds are calculated using Three sigma rule.

Cheung S. *et al.* [10] present three model-based techniques as a prototype implementation for monitoring Modbus TCP networks. They construct models that characterize the expected behavior of the system and detect attacks that cause violations of these models. Protocol-level models have been employed for characterizing Modbus TCP request and responses, based on the Modbus application protocol specification document and the Modbus TCP implementation guide. Snort rules for detecting violations of some of the Modbus specifications have been developed.

They use Prototype Verification System (PVS) language to specify the Modbus behavior. They analyze the regularity of communication patterns to detect attacks. Finally, they have developed two detectors, called EMERALD Bayes sensor and EModbus, to monitor network services and detect service changes in a control network. While EMERALD [17] is designed to discover traditional TCP services, EModbus discovers Modbus supported function codes on the Modbus devices. The last detects new services after some time of system operation.

To prove their proposed system a SCADA testbed has been developed at Sandia National Laboratories (SNL). The demonstration has provided evidence that the model-based intrusion detection is effective for monitoring SCADA systems, being complementary to the signature-based approach.

Bigham J. *et al.* [18] compare two approaches for modeling SCADA data: the first learns normal behavior using data as text features or n-grams, and the last looks for invariants in numerical features, such as mathematical relationships between the numbers (invariant induction).

n-grams are used to classify text independently of errors and language. It can work with data in any format and does not depend upon mathematical relationships. But it has the difficulty of detecting errors that occur close together.

Invariant induction builds a model for normal behavior by looking for relationships between read data. The beliefs that are encapsulated in the invariants can be used to form beliefs about the components of the invariants, but it can only identify incorrect readings by looking at the relationships of the candidates with other correct readings.

In order to test the performance of these two techniques, measurements for a six bus networks have been calculated using a load flow program. Test data has been generated by introducing between 1 and 44 random errors in calculated data files. The results suggest that the best way to detect anomalies is the combination of more than one anomaly detection technique. While n-gram is better on identification of corrupt files, invariant induction has a better overall performance on the identification of errors within files.

Yang D. *et al.* [19] have used a pattern matching method for anomaly detection. First, they create traffic profiles using symptom-specific feature vectors. Then they have classified these profiles based on temporal variables as time of day, day of week, and special days, such as weekends and holidays. In order to predict the correct behavior, predefined features that represent network behavior have been used by an auto-associative kernel regression (AAKR) model. A binary hypothesis technique called sequential probability ratio test (SPRT) is applied to the residuals

to determine if the residual sequence is more probably generated from a normal or anomalous distribution. Alarms are triggered when new traffic data fails to fit within stored profiles.

Finally, Valdes A. *et al.* [20] present a work to demonstrate that anomaly detection, and specifically methods based on adaptive learning, can provide a useful intrusion detection capability in process control network. They describe two anomaly detection techniques, patterns-based anomaly detection and flow-based anomaly detection. In patterns-based anomaly detection they used patterns formed from source and destination IP addresses and destination port. They evaluate patterns against a patterns library in order to find the more similar pattern. The most important feature of this technique is that it does not need attack-free training data.

In the case of flow-based anomaly detection, they define a flow in terms of its source and destination IP address and destination port. Also, they have established that flows are unidirectional. They maintain a database of active and historical flow records and these records are evaluated against learned historical norms.

In order to test two approaches, the have used a test environment that is based on Distributed Control System (DCS) from Invensys Process Systems.

The results that they have obtained indicate that the flow-based anomaly detection technique is able to detect anomalous flows effectively.

The experiments have been done in a simulated SCADA system composed by several SUN servers and workstations on a local network. As a conclusion, the experiments have demonstrated that this methodology can quickly detect anomalous behavior.

5 Conclusions

To protect SCADA systems different types of security mechanisms have been used, such as firewalls, intrusion detection systems, vulnerability scanners, security policy verifiers, patches and so on. In order to create a secure industrial control network, all of them should be combined. A firewall will filter incoming/outgoing connections according to the network services allowed by established security policy. Firewalls should also understand the protocols that SCADA networks use. It is also necessary to verify and assess the vulnerabilities of the different components that form the critical infrastructure, from SCADA servers, network devices, PLCs, RTUs and even traditional ICT components. A vulnerability scanner can help achieving this task. The result helps systems administrators protect their systems by updating existing software, applying patches, changing topology or adding security devices.

Security policies are defined for each system taking into account their requirements. The compliance of them increases the security level of the system. But it is a difficult task to ensure that all system requirements are satisfied, thus security policy checking and reviewing is an important task for every company.

Intrusion detection system will help in SCADA protection as well. Early detection of an intrusion can help adopting measures to avoid consequences. This is imperative in systems that control critical infrastructures.

Rule-based IDS are faster and more reliable than anomaly detection IDS. But signatures must exist for every attack, and even for every variation of attacks. In consequence, new methods of signature development must be researched.

The next table shows a comparison of surveyed anomaly detection works.

Table 5.1 Anomaly Detection works comparison.

Reference	Does it use a specific protocol?	What kind of data does it use?	Does it create a behavioral model?	Real data / Simulated data
[6]	No	Protocol, Source IP, Destination IP, Source Port, Destination Port	No	Simulated data
[12]	Yes (Modbus/TCP)	Packets PDU	Yes (SAN)	-
[13]	No	Features extracted by sliding window over a sequence (n-gram)	No	Real and Simulated data (HTTP)
[15]	Yes (Modbus/TCP)	Master ID, Slave ID, Function Code, Transaction Status, Operation Data, Access Type, Memory Contents, Memory Address	No	Simulated data (Modbus/TCP)
[16]	No	Number of different packets in a time period, Number of packets, between two specific types of packets, Relative difference in the packet rates, Number of different source addresses in a time period.	No	Simulated data
[10]	Yes (Modbus/TCP)	Modbus protocol fields, service discovery	Yes, Modbus features, cross-field relationships	Simulated data (Modbus/TCP)
[18]	No	Features extracted by sliding window over a sequence (n-gram, invariant induction)	No	Simulated data
[19]	No	Link utilization, CPU usage, Login failure	Yes (AAKR)	Simulated data (SCADA data)
[17]	Yes (Modbus/TCP)	Modbus protocol fields, service discovery	Yes, profile-based, safeguarding model	-

Different type of anomaly detection techniques have been proposed in order to improve behavior based Intrusion Detection, but almost all of them have used simulated or another kind of traffic for learning and testing purposes. Traffic

simulation has several risks, such as leaking realism that affects everyday use of SCADA systems. In order to test the proposals as realistic as possible, it is imperative to use real SCADA traffic. Research works like made by Düssel *et al.* [13] are important contributions to anomaly detection based IDS. In this case, the sys-tem is protocol independent and it is able to detect zero day attacks. But these results have been obtained using http traffic, which is not used in CIs.

Combination of techniques used for anomaly detection systems that operate in similar conditions but taking into account the protocol features will increase the detection ratio.

Acknowledgments Iñaki Garitano is supported by the grant BFI05.454 of the Department of Research, Education and Universities of the Basque Government. The work has been developed by the embedded systems group supported by the Department of Education, Universities and Research of the Basque Government.

References

1. McMillan R (2010) Siemens: Stuxnet worm hit industrial systems. http://www.computerworld.com/s/article/print/9185419/Siemens_Stuxnet_worm_hit_industrial_systems?taxonomyName=Network+Security&taxonomyId=142. Accessed 14 September 2010
2. Richmond R (2010) Malware hits computerized industrial equipment. http://bits.blogs.nytimes.com/2010/09/24/malware-hits-computerized-industrial-equipment/?ref=middleeast. Accessed 24 September 2010
3. Christiansson H, Luiijf E (2007) Creating a European SCADA Security Testbed. Critical Infrastructure Protection. 237-247
4. Byres E, Creery A (2005) Industrial cybersecurity for power system and SCADA networks. Petroleum and Chemical Industry Conference. 303-309
5. Barbosa R, Pras A (2010) Intrusion Detection in SCADA Networks. Mechanisms for Autonomous Management of Networks and Services. 163-166
6. D'Antonio S, Oliviero F Setola R (2006) High-speed intrusion detection in support of critical infrastructure protection. Critical Information Infrastructures Security. 222-34
7. Barbara D,Wu N, Jajodia S (2001) Detecting novel network intrusions using bayes estimators. First SIAM Conference on Data Mining
8. Lane T, Brodley C (1997) An application of machine learning to anomaly detection. Proceedings of the 20th National Information Systems Security Conference. 366–377.
9. Valdes A, Cheung S, Lindqvist U et al (2007) Securing Current and Future Process Control Systems. International Federation for Information Processing Digital Library. 99-115
10. Valdes A, Cheung S, Dutertre B et al (2006) Using model-based intrusion detection for SCADA networks. Proceedings of the SCADA Security Scientific Symposium.
11. Salvi D, Mazzariello C, Oliviero F D'Antonio S (2005) A Distributed multi-purpose IP flow monitor. 3° International Workshop on Internet Performance, Simulation, Monitoring and Measurement IPS-MoMe. 9
12. Rrushi J, Campbell R (2009) Detecting Cyber Attacks On Nuclear Power Plants. Critical Infrastructure Protection. 41-54
13. Düssel P, Gehl C, Laskov P et al (2010) Cyber-Critical Infrastructure Protection Using Real-time Payload-based Anomaly Detection. Critical Information Infrastructures Security. 85-97
14. Lawrence Berkeley National Laboratory (2010) Bro intrusion detection system. http//www.bro-ids.org. Accessed 17 September 2010

15. Papa M, Gonzalez J (2007) Passive Scanning in Modbus Networks. International Federation for Information Processing Digital Library. 175-187
16. Cucurull J, Asplund M, Nadjm-Tehrani S (2010) Anomaly detection and mitigation for disaster area networks. Recent Advances in Intrusion Detection. 339-359.
17. Porras P A, Neumann P G (1997) EMERALD: Event monitoring enabling responses to anomalous live disturbances. Proceedings of the 20th National Information Systems Security Conference. 353–365.
18. Bigham J, Gamez D, Lu N (2003) Safeguarding SCADA systems with anomaly detection. Computer Network Security. 171-182
19. Yang D, Usynin A, Hines J W (2005) Anomaly-based intrusion detection for SCADA systems. 5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies. 12-16.
20. Valdes A, Cheung S (2009) Communication pattern anomaly detection in process control systems. IEEE Conference on Technologies for Homeland Security. 22-29.