2012 IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems

A Formal Authorization Framework for Networked SCADA Systems

Ondrej Rysavy Jaroslav Rab Patrik Halfar
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
Email: {rysavy,rabj,ihalfar}@fit.vutbr.cz

Miroslav Sveda
IT4Innovations Centre of Excellence
Brno University of Technology
Brno, Czech Republic
Email: sveda@fit.vutbr.cz

***Abstract*—In this paper, we propose an application of a formal authorization framework for defining and enforcing security policies in SCADA systems. Current generation of SCADA systems are built as open networked systems often connected to public networks. Thus the security becomes an important issue, which needs to be properly addressed in these systems. The knowledge gained from securing networked computer based systems may help to develop security measures for SCADA systems too. Among such methods, a policy based security methods are the most applied. The contribution of this paper consists of an overview of security issues related to SCADA systems and a proposal to use a logic-based authorization framework in this environment for achieving scalable and efficient authentication.**

***Index Terms*—SCADA systems, authentication, security management, security policy management, formal methods**

I. Introduction

This paper represents work in progress considering the application of a formal authorization framework for implementing authentication of critical requests in Supervisory control and data acquisition (SCADA) systems. An authentication of critical operations in SCADA systems is an important security measure, which is nevertheless very rarely adopted in practice. Because of the character of a SCADA environment, it is impossible to propose a general solution to improve security for all existing systems. Instead, a way to incorporate the security mechanisms as parts of a basic infrastructure seems to be a more viable option. Current proposals target mainly the communication subsystem as one of the potential points to enforce security policies [1], [2].

In our approach we propose to extend the communication subsystem to serve also as an authorization framework protecting critical messages delivered in the system. The inspiration comes from authorization frameworks developed for grid systems [3]. We believe that an advanced authorization framework may represent a useful security option for SCADA applications by providing flexible, efficient and formally verifiable policy-based method for specifying and enforcing the access to critical components and operations. Even preliminary comparison to existing solutions shows that certain security measures can be simplified if implemented in the considered authorization framework. It seems that the key aspect is the presence of the delegation principle.

The structure of the paper is as follows: In section II, we provide background information on the subject of this paper, in section III, we describe the previous work done in this area, in section IV, we present a list of issues we plan to address in our work, in section V, we present a proposal of the authorization framework, and finally, in section VI we conclude the paper discussing the current status of presented work.

II. Background

A. SCADA Systems

SCADA systems are industrial control systems that monitor and control industrial processes and through human-machine interface interact with human operators by presenting measured data and accepting and executing operators' commands.

The SCADA systems provide an instrument for supervising the underlying real-time controlled processes. With the new technology available, the SCADA systems may also incorporate some real-time control functionality. An adequate characterization of SCADA systems can be as follows:

- Data are acquired, stored within the database for further visualization or user-defined processing.
- System reacts upon occurrences of events, which classifies its behavior as reactive. An event may cause operator or automatic intervention.
- Individual subsystems may be spread over different locations. These subsystems communicate with a supervisory station in a soft real-time manner.
- Recovery from a failure and a toleration of data loss is implemented to meet the required reliability.

A SCADA system is a distributed environment consisting of several subsystems. The actual architecture depends on the character of a target application domain and an installation but it usual comprises the following common components:

- A supervisory node (SN), which gathers all relevant data in a system, maintains a database, and processes commands issued through human-machine interface by an operator. In complex systems, this component may consist of multiple servers running distributed application software.
- Remote terminal units (RTU) provide interfaces to sensors and actuators in the system. They perform conversion between electrical signals and its digital representation.

978-0-7695-4664-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ECBS.2012.4

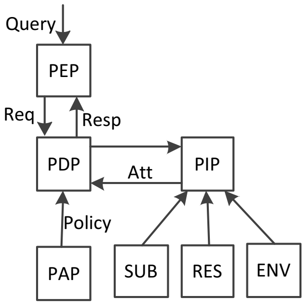

Fig. 1. XACML authorization model
An authorization model contains following components: PEP - policy enforcement point, PDP - policy decision point, PAP - policy administration point, and PIP - policy information point.

- Programmable logic controllers (PLC) implement real-time control loops for particular processes controlled within the system.
- Communication infrastructure (CI) interconnects all components of the system. Usually, multiple communication technologies and protocols are used within a single system.

A real-time control is implemented within RTU and PLC components. Host control functions implemented in SN are restricted to monitoring and supervisory control, e.g., setting parameters to control loops, switching system's operational modes. Data arising at the RTU and PLC are acquired as value-timestamp pairs. They are logged in a (distributed) database offering further analytical reasoning.

The Human Machine Interface (HMI) of SCADA systems visualizes the data to an operator. The operator can change available parameters of the system or manually control specific components. An important part is alarm handling. Alarms are expressed in a form of conditions that the system should satisfy in its normal state. If these conditions are violated an alarm is activated to draw operator's attention.

Various types of communication media are used in SCADA systems. Traditionally, proprietary protocols have been used to communicate with PLC and RTU components, e.g. Modbus, Profibus, RP-570. Currently, standard protocols, e.g., IEC 60870-5-101, DNP3, are preferred. In several application domains, the Internet protocol suite has been deployed allowing for accessing the system from the Internet. This offers to provide a new functionality but also opens new possibilities to attack these systems.

The SCADA architecture evolves from monolithic through distributed to networked [4]. In the last generation of the SACADA systems, standard communication protocols are used and thus related security mechanisms are available for protecting the system. However, many security analyses warn that current systems are often poorly protected to an unauthorized access.

B. Security Issues in SCADA Systems

Security is an important aspect for SCADA applications. SCADA systems are used to control safety critical processes, e.g., water distribution, transmission of electricity, or a critical industrial process. Compromising these systems may have serious consequences. While there is a long list of security issues related to SCADA systems, we focus only at the class of security breach related to insufficient access control enforcement. An overview of security issues related to the SCADA network environment can be found, e.g, in paper by Igure, Laughter and Williams [2]. They identified six classes of security issues that require further research:

- Access Control
- Firewalls and intrusion detection systems
- Protocol vulnerability assessment
- Cryptography and key management
- Device and OS security
- Security management

A properly configured access control ensures that unauthorized persons cannot access system's resources and data. One of the fundamental elements in the access control implementation is a proper authentication usually enforced by assigning login accounts to authorized users. All problems related to the authentication known from other environments hold in SCADA systems too. SCADA systems may involve complicated authentication patterns driven by business needs, e.g., user accounts of a contractor may be temporal and with limited responsibility area and privileges.

According to Alcaraz, Fernandez, Roman et al. in [5], the security management means to define, implement and maintain good security standards. Properly defined security objectives can be enforced by using security policies, which must be clear and comprehensive. Security policies have a wide scope. They define authorized persons or system components for execution of actions as a part of responsibility identification. Also, with respect to security enforcement, the properly implemented security policies should specify how to *react* to incidence, to *detect* security violation by monitoring and auditing system events, and to *protect* the system by using maintenance operations and patch management [6].

Human interaction is necessary in SCADA applications. Human operators also present the danger to SCADA systems. In most of legacy SCADA systems, there is no authentication involved, or a very limited form is implemented, e.g., non per-user authentication. This presents a serious problem for security of SCADA systems. Managing access rights is difficult in SCADA as there are thousands of possibilities because of an enormous number of variables in the system and many roles that different users can take. Some form of a flexible, reasonable complex authorization mechanism is thus required. An overview of security challenges in SCADA systems is provided by Hentea in [7].

C. Authorization Frameworks

Authorization frameworks are widely used in complex and distributed systems to implement access control policies. An industrial standard XACML defines a declarative access control policy language and the model of evaluation. Using the XACML, we will present usual concepts of authorization frameworks. An architecture of the framework is presented in Fig.1.

The PEP analyses requests sent by users and asks the PDP to make access decision according to the security policy. The PAP is an administrator of these security policies. To make a decision the PDP may ask the PIP to obtain additional information and parameters related to analyzed request. While security policy may be statically defined by an administrator, the PIP provides run-time values that correspond to the current system state.

Policies are written in a policy language. The XACML recognizes three kinds of elements, namely, PolicySets, Policies, and Rules. Policies are expressed as collections of associated Rules. A rule is defined by relating an action to subjects and resources. A subject is an entity that requests the access. A resource may be data, a service or a system component. An action represents a type of access permitted on the resource. The environment may provide additional information.

Since the XACML version 3, a delegation is supported. The delegation is an important mechanism that supports the decentralized administration of access policies. Delegation rules are stored in administrative control policies separated from access control policies. Delegation rules increase the flexibility as certain modifications can be done without the need to change root policies.

III. Previous Work

In this section we overview previous work on securing SCADA systems. Many works have been done on securing communication in networked SCADA systems and enhancing security of individual devices. In general, many approaches tend to redefine concepts and methods from other domains. Here, we present contributions that aim at addressing the specific needs of SCADA systems.

Securing SCADA systems by applying authentication and access control principles are proposed by Xiao, Yen, and Bastani in [8]. The problem of scalability is solved by using a public key cryptography approach based on the grid of certification authorities to reduce the burden of rekeying requests. The integrity of a communication between SCADA components is identified as the important security measure by Fovino, Carcano and Masera in [9]. They proposed to extend communication protocols in the SCADA environments to contain computed signatures that protect the commands and data carried by packets.

A security policy management is considered an important aspect for security enforcement in IT systems. Many of the principles are applicable to SCADA systems. Several works deal with security models for SCADA systems.

Alcaraz, Fernandez and Roman et al. in [5] examine the issues of user identification and authorization. They conclude that currently used authentication mechanisms based on user and password is insufficient for modern SCADA systems. They recommend to follow security standards, e.g., NIST 800-82. Cardenas, Amin and Sastry in their position paper [10] identify and define the problem of secure control in general and propose a set of challenges that need to be addressed. They suggest that the solution needs to be based on the combination of both proactive and reactive security techniques.

Ajos, Brito and Pires [11] proposed the application of Ponder framework [12] for formal specification of policies and rule validation. This idea follows principles defined for policy based network management proposed by IETF [13]. Ponder is a declarative language for specifying security policies. There are following types of policies:

- Authorization policies define subjects that are permitted to perform actions on objects in a target domain.
- Obligation policies specify which actions have to be performed on objects of a target domain in the response to an event.
- Refrain policies define actions that have to be refrained from performing on target objects. The difference to negative policy is that subjects are generally permitted to perform these actions but are blocked from doing so if certain constraints apply.

In the authorization system, policies are stored in a repository and translated into specific device languages. The distribution to devices is performed using appropriate protocols. Devices thus autonomously enforce given security policy. Centralized security policy management is necessary. Ponder offers formal semantics and thus it may be possible to develop tools for consistency and conformity verification of policies. None of that has been done till the present time.

IV. Problem Statement

Current process control devices available in the market cannot adequately enforce security policies. Because of resource limitations and cost constraints it cannot be assumed that much effort will be devoted to extend their functionality by implementing advanced security mechanisms. In this paper, we propose an architecture pattern that introduces new security policy enforcement points. These points require no modification of control devices. The key idea is that security should be enforced mainly at the level of local control network, which serves as a common communication medium between critical system components.

To strengthen security in SCADA systems we propose implementing the following principles:

- A security policy based on Role Based Access Control (RBAC) should be implemented as a part of the underlying authorization framework. The RBAC reduces the complexity and cost of security administration in complex systems by the use of roles, hierarchies, and constraints to organize user access levels.
- Securing communication in Local Control Networks should be supported. Many PLCs are able to communicate using standard protocols, e.g., Ethernet, TCP/IP, within Local Control Network. Such communication is

not secured. An attacker with access to LCN can intercept such messages or generate adversary packets to disrupt system operations.

- Logging of events should be carried in order to provide data for regular security audit and intrusion detection systems. Certain patterns of a system behavior may indicate that there is an security violation attempt. Collecting of events and statistical data provides an input for both online and off-line security analysis. Proper and accurate logs help to determine what caused a security event to occur.

There are other security threats that we did not mention, e.g., inadequate physical protection of network equipment, no malware protection software deployed, etc. These are out of the scope of this paper and our aim to propose a design, which increases the security protection at the level of inter-component communication and logical access.

V. Proposed Approach

Our approach is to modify and extend the communication subsystem for supporting active security enforcement. This is based on an authorization framework deeply embedded with the communication protocol. As an experimental platform we selected Distributed Network Protocol (DNP), which is an open and optimized communication protocol primary developed for the SCADA environments.

A. DNPSec

The Distributed Network Protocol was designed to serve in SCADA applications for efficient message delivery between SCADA components. The protocol is based on the Master-Slave communication schema. The protocol implements features that make it robust, reliable, efficient but not secure from attacks. Several proposals deal with adding security to the DNP. The DNPSec proposed by Majdalawieh at al. in [14] defines a security framework providing authentication, integrity and confidentiality. In addition to the DNP a new communication pattern is introduced in the DNPSec which serves for the key exchange. This is used during the installation and connection setup between the Master and Slave components and after an old session key expires and new keys are distributed. Authors claimed that because a static nature of the SCADA environment a simple key management is sufficient. While the key management always represents a performance penalty, the authors ensures that the delay is not significant for current systems because of sufficient capacity and speed of communication networks and processing power at the end systems.

B. Application Authentication

Recently, the application-layer authentication mechanism for DNP was published as a standard [15]. At application layer it is possible to authenticate individual users. The standard assumes that there are multiple users located at the site of a master node and the authentication is performed for each user separately to others and to the master station itself. The implementation is based on two concepts:

- A challenge response protocol to protect specific critical Application Service Data Unit (ASDU).
- A calculation of a Message Authentication Code (MAC) for each Application Service Data Unit.

The general behavior is that a device performing a critical operation issues the challenge-based authentication to protect this operation. Usually, a master station sends an ASDU carrying the request for performing a critical operation such as setting up new parameters in a control process. On the delivery of this critical ASDU, the receiving device invokes the authentication mechanism by replying with a challenge message. The master device must reply before communication can continue to authenticate the previous operation. The device checks the response from the master station and if the response is valid then the critical ASDU operation is executed. In other case, the authentication failure error is sent back to the master station. There is also an option to use aggressive mode for authentication, which means that the response is precomputed and sent together with a critical ASDU. A user authentication relies on the maintaining a database of active users. While the standard does not specify how such database should be implemented, the character of authentication requires that each unit participating on the authentication process should be able to identify a user by its User Number (UN), which is unique within a particular DNP association. Briefly, a DNP association is a logical connection between particular master and slave stations.

C. Logic-based Access Control

The previous method for authentication requires complex treatment of user identities in order to work properly. The authentication should serve to protect operations from being executed by users that do not have rights to do so. To provide the same level of authentication security but advance the flexibility we propose to use a distributed authorization framework based on formal logic. In this subsection, we shape a possible design of the framework that may be applied in this specific environment.

A high level model is shown in Fig.2. An authorization framework is a top layer of a communication subsystem. Each component composes messages by calling functions of the authorization framework. The types of messages remain the same as defined in the DNP specification. However, the authorization framework is responsible for adding necessary security and authentication information to each message. Similarly to DNPSec, the protocol is modified to carry MAC data used for verification of frames integrity. Moreover, for the purpose of the authorization framework, the critical requests are accompanied with authorization assertions. This is analogous to the approach defined in the DNP authentication standard [15]. The difference is however in the processing of critical requests. By using the logic-based authorization, each critical command is accompanied with security assertions before the command is send to a slave node. If these assertions together

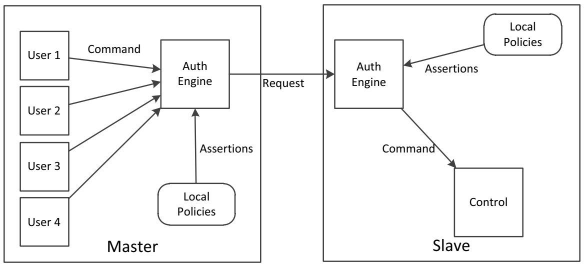

Fig. 2. A model of authorization framework

A communication subsystem is extended with Authorization Engine that manages adding and verifying authorization assertions for requests.

with the slave node's local policy assertions are enough to successfully evaluate the request then the critical command can be executed. In other case the slave node can either refuse the execution or ask for additional assertions either by querying the master station or an external authorization node. In the strictness case, when the node refuses the execution of the command the burden of providing evidence that the user has enough rights to execute the command is on the master station. This represents an efficient one pass authentication.

To demonstrate the way the policy assertions are specified and used we provide a simple example written in SecPAL language [16]. The assertions have a simple form, e.g., the following defines that a person Alice is an operator. Because it is said by the administrator the system trusts this assertion.

$$\textit{admin}\ \text{SAYS}\ \textit{Alice is operator}\,.$$

If Alice wants to execute a command that modifies a control parameter, `set(alertLevel,10)`, then such request has to be sent with a collection of assertions that allow receiving node to verify that this action is permitted. Consider that receiving node allows all operators to perform this action, which means that a local policy contains the following assertions:

$$\textit{node}\ \text{SAYS}\ \textit{admin can say}\ x\ \textit{is operator}\,.$$
$$\textit{node}\ \text{SAYS}\ \textit{operator can set alertLevel}\,.$$

Taking these two assertions and the assertion that comes with a request it is possible to infer that Alice can set the alert level at this node.

To implement this kind of authentication it is necessary that every node maintains a predefined set of local policies. Also every assertion must be signed by its issuer, which guarantees its authenticity.

VI. Conclusions

In this paper, we proposed to employ a logic-based authorization framework for implementing a flexible application layer authentication in SCADA systems. The current security issues of SCADA were briefly reviewed. From the presented review it is evident that authentication and secure communication are necessary to be implemented in order to ensure an adequate protection level of critical processes in SCADA applications. We discussed how this proposal fits with the current architecture. In particular, we focus at the DNP communication infrastructure, which seems to be a suitable underlying technology for the experimental implementation. Our future steps include experimental design and implementation of the proposed authorization framework through extending the DNP following the similar approach as proposed for the DNPSec and in the DNP3 Secure Authentication Standard.

Acknowledgment

This work was supported by the Technology Agency of the Czech Republic under grant No. TA01010632, the research programme MSM 0021630528, the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070 and by the internal university grant FIT-S-11-1.

References

[1] P. S. M. Pires and L. A. Oliveira, "Security aspects of SCADA and corporate network interconnection: An Overview," in *Proceedings of the International Conference on Dependability of Computer Systems*. IEEE Computer Society, 2006, p. 8.

[2] V. Igure, S. Laughter, and R. Williams, "Security issues in SCADA networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, Oct. 2006.

[3] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman, "A multipolicy authorization framework for grid security," in *Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on*. IEEE, 2006, pp. 269–272.

[4] B. Atlagic, D. Milinkov, M. Sagi, and B. Bogovac, "High-Performance Networked SCADA Architecture for Safety-Critical Systems," *2011 Second Eastern European Regional Conference on the Engineering of Computer Based Systems*, pp. 147–148, Sep. 2011.

[5] C. Alcaraz, G. Fernandez, R. Roman, A. Balastegui, and J. Lopez, "Secure Management of SCADA Networks," *New Trends in Network Management, Cepis UPGRADE*, vol. 9, no. 6, pp. 22–28, 2008.

[6] M. Brändle and M. Naedele, "Security for Process Control Systems," *IEEE Security and Privacy*, vol. 6, no. 6, pp. 24–29, Jan. 2008.

[7] M. Hentea, "Improving security for SCADA control systems," *Journal of Information, Knowledge, and Management*, vol. 3, 2008.

[8] L. Xiao, I.-L. Yen, and F. Bastani, "Scalable Authentication and Key Management in SCADA," *2010 IEEE 16th International Conference on Parallel and Distributed Systems*, pp. 172–179, Dec. 2010.

[9] I. N. Fovino, A. Carcano, and M. Masera, "A Secure and Survivable Architecture for SCADA Systems," *2009 Second International Conference on Dependability*, pp. 34–39, Jun. 2009.

[10] A. a. Cardenas, S. Amin, and S. Sastry, "Secure Control: Towards Survivable Cyber-Physical Systems," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*. Ieee, Jun. 2008, pp. 495–500.

[11] I. dos Anjos, A. Brito, and P. Motta Pires, "A Model for Security Management of SCADA Systems," in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. IEEE, 2008, pp. 448–451.

[12] E. Lupu, M. Sloman, and N. Dulay, "Ponder: Realising enterprise viewpoint concepts," in *Proceedings of the 4th Enterprise Distributed Object Computing*. IEEE Comput. Soc, 2000, pp. 66–75.

[13] B. Moore, E. Ellesson, J. Strassner, and A. Westerin, "RFC 3060: Policy Core Information Model," 2001.

[14] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, "DNPSec: Distributed Network Protocol Version 3 (DNP3) Security Framework," *Advances in Computer, Information, and System Science, and Engineering*, vol. 3, 2006.

[15] DNP, "DNP3 Secure Authentication Version 5," DNP Users Group, Tech. Rep., Oct. 2011.

[16] M. Becker, C. Fournet, and A. Gordon, "Design and Semantics of a Decentralized Authorization Language," *20th IEEE Computer Security Foundations Symposium (CSF'07)*, pp. 3–15, Jul. 2007.