21, rue d'Artois, F-75008 PARIS
http : //www.cigre.org

D2-111

CIGRE 2010

A case study applying the Cyber Security Modeling Language

TEODOR SOMMESTAD, MATHIAS EKSTEDT, LARS NORDSTRÖM
KTH - Royal Institute of Technology
Sweden
lars.nordstrom@ee.kth.se

SUMMARY

The operation of the power system is today highly dependent on computerized control systems. These SCADA systems resemble the central nervous system of the power system. At the same time as control systems enables more efficient, qualitative, and safe power systems, their vulnerabilities are also vulnerabilities to the power system.

This paper presents a modeling language specifically developed for assessing the cyber security of SCADA systems. The modeling language uses the formalism Probabilistic Relational Models to integrate a mathematical inference engine with the modeling notation. If a SCADA system is modeled using this cyber security modeling language the cyber security of this SCADA system can be assessed probabilistically. Given a graphical description of a system, a quantitative analysis of threats is provided. This makes it possible to use the framework for evaluating the current solution as well as elaborate with what-if scenarios and the trade-offs between these.

This cyber security modeling language could for example be used to model two control centers and the communication between them together with security mechanisms such as access control and communication protection The modeling language can also be used to describe a complete SCADA system and infer its security.

The data associated with the probabilistic inference engine is only preliminary. In this paper we present a case study where cyber security modeling language has been applied to assess the security of a SCADA system. It is demonstrated how the modeling language can be applied and how a value for security can be inferred from architectural models (using the preliminary data). Future work will focus on the quantitative side of the modeling language. Probabilities will be elicited from literature, experiments, and field studies and through the opinion of domain experts. A tool is also being developed to support inference and analysis.

KEYWORDS

Cyber security, SCADA, Information technology, Information security, Probabilistic Relational Models

1 Introduction

Society is increasingly dependent on the proper functioning of the electric power system, which in turn supports most other critical infrastructures: water and sewage systems; telecommunications, internet and computing services; air traffic, railroads and other transportation. Many of these other infrastructures are able to operate without power for shorter periods of time, but larger power outages may be difficult and time consuming to restore. Such outages might thus lead to situations of non-functioning societies with devastating economical and humanitarian consequences.

The operation of the power system is today highly dependent on computerized control systems. These SCADA systems resemble the central nervous system of the power system. At the same time as control systems enables more efficient, qualitative, and safe power systems, their vulnerabilities are also vulnerabilities to the power system.

In addition to the potential severe consequences of a compromised control system, security management of these control systems is a really complex issue. First, security suffers from a weakest-link syndrome. This means that a single misconfiguration in the control system architecture cause a vulnerability that jeopardizes the whole power system. This is truly a challenge since control systems are extremely complex. They contain highly advanced functionality; they are heterogeneous and include several third party components; they are extensively networked, both internally and with external systems, and they depend on the human organization that manages and uses them. Altogether control system security management can be described as keeping track of a moving target that consists of a great number of details that are interrelated in very complex ways.

Cigré [1][2][3][4] and other organizations has acknowledged these potential security risks and several standards (e.g. NERC CIP [5]) as well as guidelines(e.g. NIST's guide [6]) has been developed with a particular focus on SCADA systems. These standards and guidelines are often prescriptive and rarely provide information on the importance of different security mechanisms or how these interrelate. As a consequence they are of limited help if decision makers which cannot implement all security mechanism and need to select a subset of them for implementation. As the standards and guidelines do not prioritize different combinations of security mechanisms they are also of limited use to activities were the security is assessed.

1.1 Scope

This paper presents a modeling language specifically developed for assessing the cyber security of SCADA systems. The modeling language uses the formalism Probabilistic Relational Models [7] to integrate a mathematical inference engine with the modeling notation. If a SCADA system is modeled using this cyber security modeling language the cyber security of this SCADA system can be assessed probabilistically. Given a graphical description of a system, a quantitative analysis of threats is provided. This makes it possible to use the framework for evaluating the current solution as well as elaborate with what-if scenarios and the trade-offs between these.

This cyber security modeling language could for example be used to model two control centers and with the communication between them together with security mechanisms such as access control and communication protection, or it could be used to describe a complete SCADA system and infer its security. In this paper we present a case study where cyber security modeling language has been applied to assess the security of a SCADA system. It is demonstrated how the modeling language can be applied and how a value for security can be inferred from architectural models.

1.2 Outline

The outline of this paper is as follows. Chapter two briefly introduces and explains the formalism Probabilistic Relational Models. In chapter three the cyber security modeling language is presented. An overview of the modeling language is given and a subset of its probabilistic inference engine is shown. In chapter four the case study is described. The method and future work is discussed in chapter five.

2 Probabilistic relational models

This chapter briefly explains the formalism *probabilistic relational models* (PRMs*)* [7]. A PRM specifies a template for a probability distribution over an architecture model (or a data model). For a more elaborate description of PRMs the reader is referred to [7].

2.1 Architecture metamodel and architecture model

The architecture metamodel in a PRM describes the concepts expressed in the modeling language. An architecture metamodel describes a set of classes (e.g. firewall, network zone or server). Each class is associated with a set of descriptive attributes. For example, a class *Operating System* might have the descriptive attribute *Patched*, with possible values *{True, False}*. Each class is also associated with a number of relationships. The class *Operating System* can for example be associated with the relationship *Covered By* that associates it to the *Maintenance Process* it is covered by. In the architecture model the architecture is represented using the classes and relationships available in the architecture metamodel. If the value of an attribute is known this value can also be set in the architecture model.

2.2 Probabilistic model over attributes

A PRM *Π* specifies a probability distribution over all architecture models expressed using the metamodel. As a Bayesian network [8][9] it consists of a qualitative dependency structure, and associated quantitative parameters.

The qualitative dependency structure is defined by associating attributes with a set of parents, which influence the value of the attribute (i.e. if it true or false). The parents are defined as using the structure of the metamodel. It can for example be defined that the attribute *Patching Procedures* in the class *Maintenance Process* influence the attribute *Patched* in an *Operating System* if the *Maintenance Process* cover the *Operating System*.

For the quantitative parameters, we can now define a local probability model by associating a conditional probability distribution with the attribute. For instance, the probability that a *OperatingSystem* is *Patched* (*Patched=True*) can be defined as 30 percent if the *Maintenance Process* has *Patching Procedures*. Given an architecture model, a PRM *Π* now specifies a probability distribution over the states in architecture models attributes. A PRM thus constitutes a formal machinery for calculating the probabilities of various architecture instantiations. This allows us to infer the probability that a certain attribute assumes a specific value, given some (possibly incomplete) evidence of the rest of the architecture instantiation.

3 A Cyber Security Modelling Language

The cyber security modelling language presented in this paper is a probabilistic relational model. It thus comprise of an architectural metamodel and a probabilistic model expressing how the attributes in this architectural metamodel depend on each other.

Due to space limitations we cannot present the entire probabilistic relational model in this paper. Instead we here present the parts of the metamodel that would be applicable to a user applying the modeling language and exemplify its inference engine on a subset of this metamodel. Section 3.1 presents the metamodel and section 3.2 presents a subset of its probabilistic inference engine.

3.1 Metamodel

This architectural metamodel of the cyber security modeling language is presented in Figure 1. This figure only depicts the classes, attributes and relationships that need to be represented in an architecture model to enable the computations.

The metamodel does for example specify that hardware should be modeled together with the operating system of that hardware. It also specifies that the services operated by operating systems should be included in the architecture model and that it is of relevance if unnecessary services have been disabled. It further specifies that data flows that are allowed by firewalls should be included and that the server-side of these data flows shall be represented as a service.

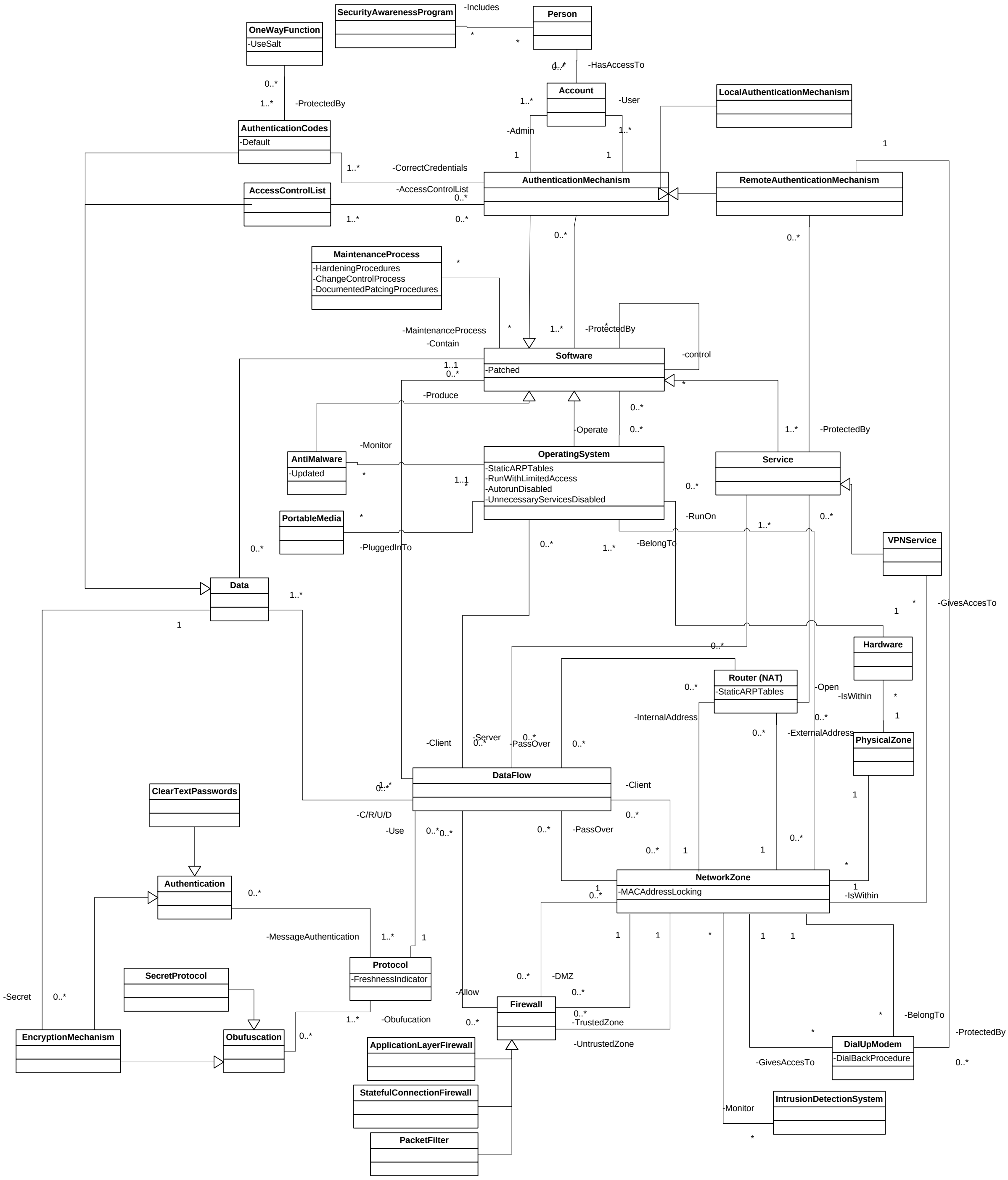


Figure 1 – Architecture metamodel of the cyber security modeling language.

3.2 Probabilistic model

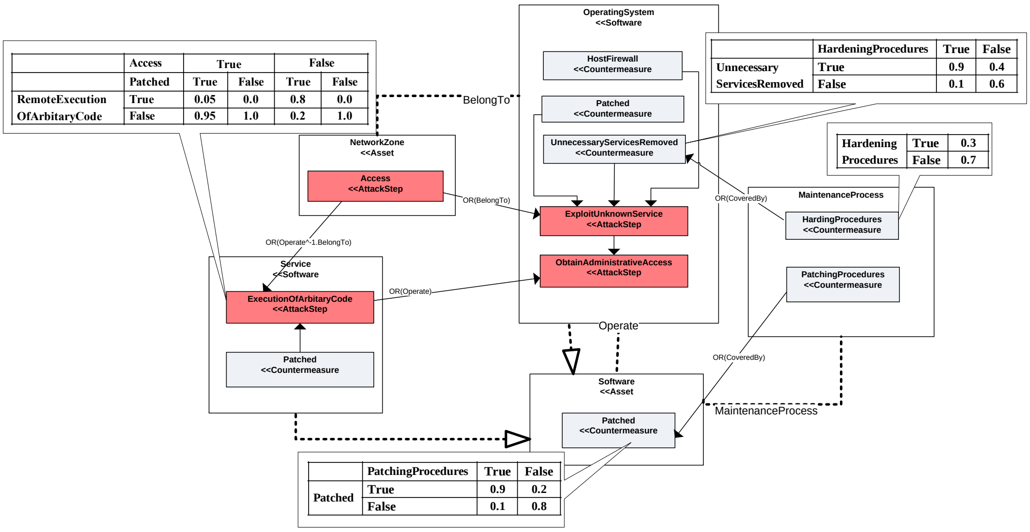
The security assessments supported by this cyber security modelling language makes it possible to assess whether different types attacks can be performed against a SCADA system. The attributes in the metamodel can be divided into two classes: attack steps and countermeasures.

Attack steps are steps that an adversary would take to compromise a SCADA system. These can be linked together so that one attack step so that the accomplishment of one attack step can makes it easier to accomplish another attack step. Some attack steps are necessary to accomplish in order to accomplish other attack steps.

The attacks that can be executed against a SCADA system depend on the architecture of the system. The architecture includes both the configuration of the system in a more general sense (e.g. the services that run on computers and data flows) and the countermeasures that are in place (e.g. antivirus software). The configuration of the system determines if one accomplished attack step makes it easier to accomplish another attack step or not. For example, if no network connection exist between two computers it will not be possible to exploit the network services of one of them from the other one. The countermeasures influence the probability that attack steps can be accomplished or increase the probability that other countermeasures are in place. For example, if antivirus software is used this will influence the probability that a user install malware by mistake.

With these constructs the cyber security modeling language allows inference of the probability that different attack steps can accomplished against a SCADA system. An extensive literature survey form the basis for the selection of classes, relationships, attributes and dependencies included in the language. This literature survey include cyber security standards (e.g. [5]), cyber security guidelines (e.g. [10]), articles (e.g. [11]), technical reports (e.g. [12]) and text books (e.g. [13]).

Figure 2 show an excerpt of the probabilistic model associated with the architecture metamodel in Figure 1. The attribute *OperatingSystem.ObtainAdministrativeAccess* is an attack step that means that the attacker has gained administrative access to a computers operating system. Two other attack steps influence the probability that this can be accomplished: *ExploitUnknownService* in the operating system (a service that the system administrator does not know exists) and *RemoteExecutionOfArbitaryCode* in a service that the operating system operates. For both of these the attacker must first get access to the network zone. The conditional probability table associated with *RemoteExecutionOfArbitaryCode* is shown in Figure 2. This conditional probability table does for instance state that if access is obtained to the network the attack step will succeed with probability 0.8 or 0.05, conditioned that the service is patched or not. If access cannot be obtained to the network zone this probability is zero (it is impossible to perform the attack step).



	Access	True		False	
	Patched	True	False	True	False
RemoteExecution OfArbitaryCode	True	0.05	0.0	0.8	0.0
	False	0.95	1.0	0.2	1.0

	HardeningProcedures	True	False
Unnecessary ServicesRemoved	True	0.9	0.4
	False	0.1	0.6

Hardening Procedures	True	0.3
	False	0.7

	PatchingProcedures	True	False
Patched	True	0.9	0.2
	False	0.1	0.8

Figure 2 – Excerpt from the probabilistic dependency model associated with the modeling language.

Conditional probability tables are also shown for the attributes *Software.Patched*, *MaintenanceProcess.HardeningProcedures* and *OperatingSystem.UnnecessaryServicesRemoved*. These, as well as all other conditional probabilities in the cyber security modeling language, are only based on preliminary data. Efforts will in the future be made to find more accurate data for these conditional probabilities.

4 Case study

The cyber security modeling language has been applied to model a SCADA system and its environment. The control center, its interface to the office network as well as substation communication was included in this case study. The case study was performed during fall 2009 and involved a number of interviews with a senior system architect. Interviews were also conducted with a security expert within the organization to complement the architecture models with details concerning security solutions.

Confidentiality prevents publication of the full empirical data set and publication of analysis results. Figure 3 depict an excerpt of the architecture model and some of the probabilities inferred based on it. These probabilities do for example indicate that there is a two percent probability that an attacker can obtain administrative access to this particular operating system installation. They also show that there is 20 percents probability that the power system applications service and the operating system are patched. As indicated by the arrow from *PatchingProcedures* to *Patched* the former influence the latter. According to the conditional probability in Figure 2 a maintenance process that includes patching procedures would increase the probability from 20 percent to 90 percent. This would indirectly influence the probability that different attack steps can be accomplished. A decision maker can use this type information to carry out what-if analysis, and to identify possible improvements to security.

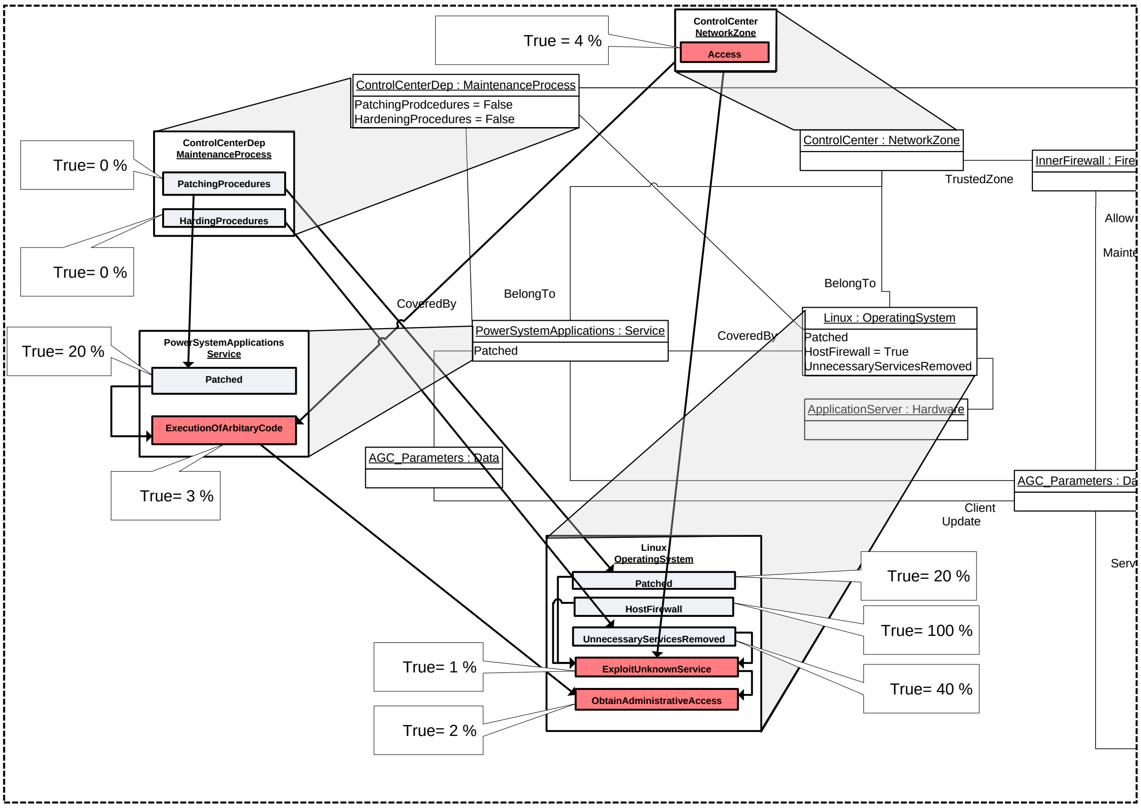


Figure 3 – Excerpt from architecture model.

The practical experience from this case study when it comes to the modeling language and the method is that the modeling language provides a close to ideal separation of concerns. The system owner can create models of different architectures by representing relevant objects and relationships in diagrammatic descriptions and from these the security risk associated with the architectures can be assessed programmatically. Architecture models can assess the security risk associated with different network architectures, or to assess the impact of different password policies on the overall security risk. This type of application relies on the data expressed in the conditional probabilities of the cyber security modeling language. As stated above the data used now is only preliminary.

However, even without these the modeling language provides support to the cyber security assessment process. The metamodel associated with the modeling language guide the person assessing security to only focus on the concepts that are of relevance to security. By reviewing an architecture model expressed with this metamodel a cyber security expert can assess the cyber security.

5 Discussion and future work

A software tool is being extended to support the creation and instantiation of PRMs, including support to automatically infer different attack probability of success given an instance model. To allow quantitative assessments the quantitative part cyber security modeling language presented herein must be specified with a reasonable level of accuracy. The dependencies among variables in the security field can be obtained from sources such as domain experts, literature, empirical tests, vulnerability statistics, or a combination of sources like these. Models can also be updated as new threats or countermeasures emerge. To create a cyber security modeling language where the qualitative structure is optimal with regard to security risk prediction and the conditional probabilities represent an undisputable truth is of course extremely difficult, if not impossible. However, to be of practical use it is sufficient if the cyber security modeling language captures the knowledge that is available in the security field and thereby provides the decision maker with a tool that improve security risk analysis activities.

The mere possibility to express and quantify how security theories relate to different architectures is another feature offered by the cyber security modeling language. Furthermore, since PRMs in their pure form are versatile, the cyber security modeling language can be integrated with modeling languages that express theories from other fields in terms of PRMs. For example theories on how costs, business value or modifiability relates to different architectures. Thus allowing decision maker to make informed decisions regarding the security risk associated with different architectures, while at the same time take other concerns into consideration.

Based on the qualitative structure presented in this paper future work will focus on the quantitative side of the modeling language. Conditional probabilities will be elicited from literature, experiments, and field studies and through the opinion of domain experts.

BIBLIOGRAPHY

[1] JWG D2/B3/C2-01, "Cyber Security Considerations in Power System Operations," *Electra*, vol. 218, pp. 15-22, Feb. 2005.

[2] JWG D2/B3/C2-01, "Cyber Security Risk Assessment in the Electric Power Industry," *Electra*, vol. 24, pp. 36-43, Feb. 2006.

[3] JWG D2/B3/C2-01, "Managing Information Security in an Electric Utility," *Electra*, no. 26, pp. 20-27, Oct. 2004.

[4] JWG D2/B3/C2.01., "Technical Considerations for building secure Substation Automation Systems," *Electra*, vol. 229, pp. 28-33, Dec. 2006.

[5] NERC, "Standard series CIP 002-009," 2006.

[6] K. Stouffer, J. Falco, and K. Kent, "Guide to Supervisory Control and Data Acquistion (SCADA) and Industrial Control System Security," 2008.

[7] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *In IJCAI*, 1999, pp. 1300-1309.

[8] F. V. Jensen, *An Introduction to Bayesian Networks*. New York: Springer-Verlag, 1996.

[9] F. V. Jensen, *Bayesian Networks and Decision Graphs*. Springer New York, Secaucus, NJ, USA, 2001.

[10] Critical Infrastructure Protection Board, "21 steps to improve security of SCADA networks," 2002.

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

[12] J. Stamp, M. Berg, and M. Baca, "Reference Model for Control and Automation Systems in Electrical Power," 2005.

[13] M. Howard and D. C. LeBlanc, *Writing Secure Code*. Redmond, WA, USA: Microsoft Press, 2002.

[14] G. Stoneburner, "Underlying Technical Models for Information Technology Security," NIST Special Publication NIST 800-33, Dec. 2001.

[15] TeodorSommestad, M. Ekstedt, and P. Johnson, "Cyber Security Risks Assessment with Bayesian Defense Graphs and Architectural Models," in *Proceedings of the Hawaii International Conference on System Sciences*, Hawaii, USA, 2009.

[16] R. L. Krutz, *Securing SCADA systems*. Indianapolis, Indiana, USA: Wiley Publishing, 2006.