Towards Experimental Assessment of Security Threats in Protecting the Critical Infrastructure

Janusz Zalewski[1], Steven Drager[2] William McKeever[2] and Andrew J. Kornecki[3]
[1]*Dept. of Software Engineering, Florida Gulf Coast University, Ft. Myers, FL 33965, Florida, U.S.A.*
[2]*Air Force Research Lab, Rome, NY 13441, New York, U.S.A.*
[3]*Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, Florida, U.S.A.*

Keywords: Computer Security, Software Safety, Trustworthy Systems, Automation Systems, Industrial Control Systems, Critical Infrastructure.

Abstract: Security is a system and software property essential in protecting infrastructure critical to the nation's business and everyday operation. It is often related to and overlapping with other trustworthiness properties, such as safety and/or reliability. Mutual relationships of these properties and their interactions in real world systems have been studied by multiple authors in a recent decade; however, they are rarely viewed jointly in the context of critical infrastructure. The objective of this paper is to take a closer look at the relationship of security with safety in computing systems, and present a unified view for further research. In particular, the paper presents an overview of the state-of-the-art and focuses on the discussion of the unifying architecture, which leads to interesting observations how security and safety are related. Preliminary experiments on using safety concepts to assess security in industrial control systems with monitoring tools are discussed.

1 INTRODUCTION

Security as a computer system property has been studied for several decades (Landwehr, 1981). Only in this century it has become an important component of investigating the protection of nation's critical infrastructure (U.S. GAO, 2004). However, it has been mostly considered as separate system attribute rarely associated with other properties contributing to system or software trustworthiness, such as safety or reliability.

A critical infrastructure can be viewed from a number of different perspectives, ranging from plain business viewpoint, on one hand, to a strictly technical point of view, on the other hand. From the technical perspective, the security issues cannot be treated in isolation from other properties of computing systems, because it impacts safety and reliability, and vice versa.

In industrial applications, with a control system in charge of the technological process, which are an essential part of critical infrastructure, typically safety was considered a critical system property. The computer systems were designed such that the behavior of computer software or hardware would not endanger the environment in a sense that equipment's failure would cause death, loss of limbs or large financial losses.

On the other hand, the security of industrial computer control systems was typically limited to the physical plant access and off-line protection of data. With the miniaturization of computing devices, growing sophistication of control, and with the advent of the Internet, multiple functions of industrial control systems have become accessible online, which opens doors to enormous security threats to the entire infrastructure.

Thus, to increase trustworthiness of industrial computer systems, security and safety concerns cannot be treated in isolation, and the mutual relationships of safety and security have to be studied and reconciled. One particular problem, which is the motivation for this work, is that currently there are no standards, or even adequate research, to guide developers and manufacturers through the issues of safety and security combined together. Similarly, the relationship between security and reliability has been complex and is an intrinsic part of this research, but is not discussed in this paper due to space limitations.

The objective of this paper is to look more closely into the relationships of security and safety, and analyze some of the impacts they may have on

Zalewski J., Drager S., McKeever W. and Kornecki A..
Towards Experimental Assessment of Security Threats in Protecting the Critical Infrastructure.
DOI: 10.5220/0004098502070212
In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2012), pages 207-212
ISBN: 978-989-8565-13-6

each other in the context of protecting the critical infrastructure. The ultimate goal would be to develop techniques to measure these properties and enable assessment of system trustworthiness.

The rest of the paper is structured as follows. Section 2 depicts the basic distinction in the roles security, safety and reliability play in the interaction between a computer system and its environment. Section 3 outlines how security has been viewed, historically, in the context of safety. Section 4 presents results of preliminary experiments on relating security with safety, and Section 5 ends the paper with some conclusions.

2 THE ROLES OF SECURITY, SAFETY AND RELIABILITY

While security, safety and reliability are strictly related and intertwined, they can be separated using as a criterion the system's interaction with the environment (Figure 1). Such separation leads to interesting analogies in studying overall trustworthiness properties.

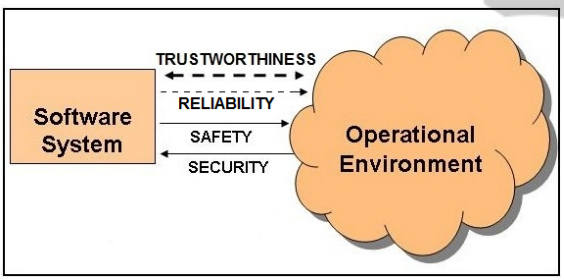

Figure 1: Illustration of trustworthiness properties.

The user or system designer usually views these properties from the perspective of guarantees on system behavior in terms of risk that "nothing bad will happen" or that the risk of "something bad may happen" is low. The risk is usually analyzed involving potential hazards or threats that are related to computer failures (both hardware and software). Thus, in terms of risk and failures, the individual roles of all three properties in the context of the environment, as illustrated in Figure 1, can be briefly described as follows:

- Security: when a failure leads to severe consequences (high risk) to the computer system itself;
- Safety: when a failure leads to severe consequences (high risk) to the environment;
- Reliability: when failure does not lead to severe consequences (high risk) to the environment or a computer system, nevertheless the failure rate is of principal concern.

To state it differently, reliability is relevant to minimizing undesired situations and their effects (to keep the system running), while security and safety are relevant in preventing the computer system and environment, respectively, from undesired situations and their effects. The next section discusses the relationship between security and safety.

3 MODELS RELATING SECURITY TO SAFETY

There is a vast amount of literature discussing jointly safety and security from the broader perspective of placing these properties in the context of system trustworthiness. A thorough literature review reveals multiple entries only in the last decade, discussing both general issues (Schoitsch, 2004; Nordland, 2007; Romanski, 2009; Pietre-Cambacedes and Chaudet, 2010, Goertzel and Winograd, 2011), as well as concerns of specific industries, such as railways (Smith et al., 2003); chemical (Hahn et al., 2005), off-shore (Jaatun et al., 2008), automation (Novak and Treytl, 2008), nuclear (Jalouneix et al., 2009), and industrial control (Kornecki and Zalewski, 2010).

One particular early paper, by Burns, McDermid and Dobson (1992), is worth mentioning, because it's probably the first one, which analyzes issues of mutual dependency of safety and security. It is essential in setting the scene for understanding safety and security, as two complementary system properties. The authors define both concepts implicitly, as follows:

- a *safety critical* system is one whose failure could do us immediate, direct harm;
- a *security critical* system is one whose failure could enable, or increase the ability of, others to harm us;

What Burns et al. call "us" is, in more contemporary terms, the *environment* of a computer system that is safety or security critical.

This view had far reaching consequences for studying mutual relationships of safety and security, probably best expressed in a series of papers by Nordland (2007). Without referring to the original paper by Burns et al., he defines both properties in terms of computer system's relationship with its environment:

- safety – the inability of the system to have an undesired effect on its environment;

- security – the inability of the environment to have an undesired effect on the system.

As is clear from the above definitions, both by Burns et al. and Nordland, safety and security are understood as *negative* properties, that is, to provide either safety or security one has to make sure that certain events *do not* happen. This has severe consequences to and causes significant problems in system design, since the engineers are normally used to designing systems to meet functional requirements, which are expressed in terms "what the system shall do", rather than in terms of "what the system shall not do", as is clearly the case for respective safety and security requirements.

In a view of these definitions it may be surprising that newer publications not necessarily take them into account, possibly assuming implicitly the case. Most recently, for example, Goertzel and Winograd, in their comprehensive survey (2011), seem to have overlooked this fact in definitions of safety and security, concentrating – however – on other important relations between the two properties. They characterize safety following MIL-STD-882D as: "Freedom from those conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment," which is consistent with the understanding outlined above, however, they stop short of defining security in terms of the interaction between the system and its environment.

Instead, they concentrate on some important characteristics of safety and security. For example, the conditions referred to in the MIL-STD-882D definition of safety are called *hazards* and compared, incorrectly, to risks; incorrectly, because risk is involved both with security and safety. What is equivalent in security models to a hazard in safety models is not risk but a *threat*, which is justly pointed out by Goertzel and Winograd as a cause of risk.

This analogy between hazards for safety and threats to security leads to a deeper insight into the relationship between both properties, and brings it to the concept of *dependability*. This is because, as pointed out by Nordland (2007), threats can lead to exploitation of *vulnerabilities* in a system. In a safety critical system, this would be equivalent to activating *faults* in a system to endanger safety. Thus, an analogy exists between vulnerabilities with respect to security and faults with respect to safety.

To summarize, the results of this analysis provided stronger evidence that, while safety and security are strictly related and intertwined, they can be separated using as a criterion the system's interaction with the environment (Figure 1). Such separation leads to interesting analogies in studying both properties within the framework of trustworthiness.

Before proceeding with the analysis, it must be noted that the view of safety and security properties adopted in this work relies on the engineering understanding of both properties. Another view often used in studies of both properties, especially that of safety (or liveness), having its roots in formal specifications (formal methods), although important in itself is out of scope of the current work.

4 PRELIMINARY EXPERIMENTS

Based on the literature surveys, three types of safety and security models have been distinguished:

- analytical models, built with formal theories;
- experimental models, based on measurements;
- computational models, which use simulations.

In this research, we are involved with the latter two models, experimental model being the one covered in the current paper.

4.1 Model Description

All models can be viewed in the context of a particular architecture of real-time computing systems, previously published in the literature (Sanz and Zalewski, 2003). This architecture is based on the typical control system, which interacts with the environment via a number of input/output channels, and involves all essential components of an embedded system or a distributed control system, as illustrated in Figure 2:

- interactions with the controlled process via sensors and actuators;
- user interface;
- communication (network) interface;
- database interface.

The diagram shows an embedded controller interacting with the controlled plant, which can be any controlled device, such as an aircraft, missile, not only a plant in a strict sense, such as a chemical or nuclear power plant. In addition to specific measurement and control signals, through which the controller interacts with the plant, it also has interfaces to interact with other controllers on the network, the operator, and the database. These multiple controller interfaces to the plant, the network, the operator, and the database, are all

subject to security threats. More importantly, to take the analogy further, just like control theory assumes that the plant (controlled object) is subject to disturbances, security theory, if one is built for this model, could assume that known or unknown *threats* play the role of disturbances to the controller.

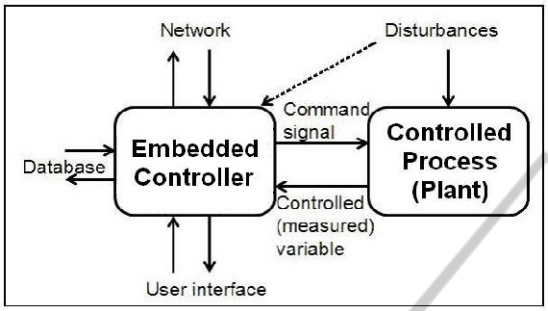

Figure 2: Typical real-time system architecture.

In other words, disturbances affecting the plant in a traditional model of an embedded control system can play a role of unexpected hazards, to which controller's software has to respond without failure, mitigating faults. Threats affecting the embedded controller can be modeled as disturbances impacting controller's behavior. Thus, analyzing security breaches can be viewed in this model as analyzing system failures, essential in the same way as it is done in safety analysis.

4.2 From Safety Shell to Security

Safety analysis of a computer system starts with identifying potential hazards that may be caused by software or hardware failures or external conditions (Leveson, 1995). Analyzing software architecture is particularly helpful, in this respect, because it identifies the major components that may be potential sources of such hazards. Since security analysis originates by identifying potential threats/attacks, it is assumed that techniques developed for safety analysis will be applicable to providing security and its assessment.

There are multiple, well established methodologies and techniques to address safety concerns during the development process (Leveson, 1995), however, for the model presented in Figure 2, we opted to choose an approach named safety shell, because it fits well into a concept of analogy between safety and security (Figure 3, Gumzej and Halang, 2009).

The shell relies on an architectural model enabling design of control systems. The concept is based on implementing a "test first" design element to prevent dangerous situations from occurring, which is meant to detect a hazardous situation at its beginning. By "testing first" the hardware processor or software shell will either validate or invalidate the current action and/or the desired action. It has been developed further by Gumzej and Halang (2009) to map the design on the UML model.

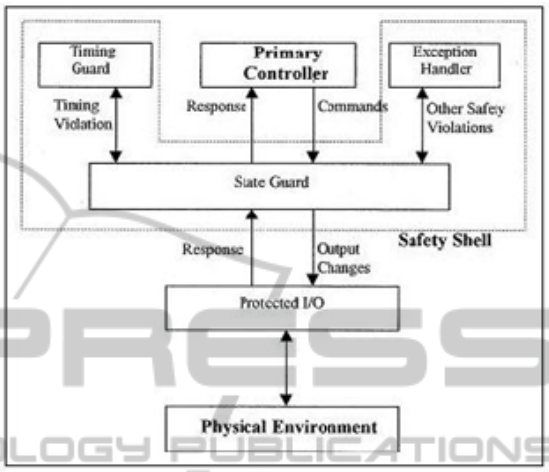

Figure 3: The concept of safety shell (Gumzej/Halang, 2009).

In essence, as shown in Figure 3, the physical environment is separated from the controller by an array of guards forming the shell. A state guard constitutes the core of the shell layer and, with the help of watchdogs performing specific lower-level functions (such as responding to timing violations or handling exceptions), is monitoring the status of all signals interchanged between the controller and the plant, acting as a protecting entity, before any emergency occurs.

The safety shell is built for monitoring signals interchanged between the controller and the plant, that is, for sensors and actuators in the model from Figure 2. However, nothing prevents the designer from using the same concept for network communication between the controller and the environment. Given that the controller's model from Figure 2 is a generic model suitable for a modern control system, the concept of a safety shell has been applied in this research to the controller's communication interface, to monitor its network access.

4.3 Actual Experiments

Thus, the concept of safety shell can be viable in Industrial Control Systems (ICS), where controllers and other computing devices forming a system are

spread over a larger area and external access is provided to and from the enterprise network. This configuration of ICS is typical for systems such as SCADA (SCADA = Supervisory Control And Data Acquisition), commonly used in larger plants, such as water management plants, power plants, etc. A usual ICS configuration is shown in Figure 4 (Schwartz et al., 2010). It includes all components of a generic model from Figure 2, that is:

- central controller, shown as a control system;
- user interface (HMI – Human Machine Interface);
- database (Historian);
- interaction with sensors and actuators (using a number of protocols);
- the network interface.

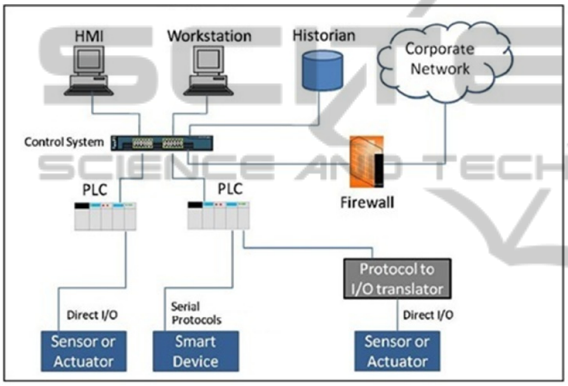

Figure 4: Typical industrial control system (ICS) architecture (Schwartz et al., 2010).

Given this analogy, we applied the concepts of a shell to the actual ICS system in a Software Engineering Lab at XYZ University. The role of the watchdog guards of a safety shell is played by network monitoring and penetration tools, in this case: Wireshark and Metasploit, respectively (Top Network Security Tools, 2012). The tools run in real time and the data collected are stored in files analyzed by the shell's State Guard.

Because of limited space, in this paper, we only analyze the Wireshark tests. They were conducted by packet capturing sessions. During each session, over 5000 packets were captured. Most of these packets have little to do with the ICS security testing, however they can be filtered out by looking at specific features. For example, one can see the login attempt packets and draw conclusions regarding potential security violations. Fortunately, the ICS implementation software successfully encrypts the password part of the data packet.

A typical Wireshark packet capture screen (Figure 5) shows multiple packets being highlighted

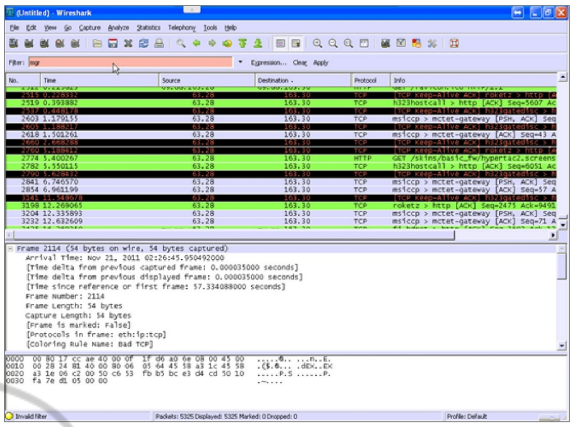

Figure 5: Illustration of broken packets caught by Wireshark.

in black with red font. In Wireshark this type of highlighting is reported as a "bad TCP" packet. With TCP, each packet is sent with a checksum. When the packet arrives at the server end, the checksum is verified, however with these packets, the checksums are not met. This does not necessary report a security threat. Nevertheless, having a constant stream of incomplete packets may be a potential vulnerability that could be exploited. Further investigation into why so many packets are being sent with bad checksum would be recommended.

Similar analysis can be performed by the shell for the Matesploit guard. In this case, however, we only performed manual analysis of data. For the actual real-time operation an automated data analysis is needed, which requires use of artificial intelligence techniques, due to massive amounts of data produced by the tool.

5 CONCLUSIONS

The driving force of this research is that security, safety and reliability properties represent complementary ends of the same problem: system trustworthiness, which is important in protecting the nation's critical infrastructure. While computer safety prevents the environment from being adversely impacted by the computer, computer security prevents the computer system from being adversely affected by the environment.

With a multitude of diverse issues and challenges in trustworthiness of industrial computer systems, including embedded systems, and a lack of a unified approach to security, safety and reliability, we propose a framework for an integrated treatment of trustworthiness properties. The central point of this

framework is a unified architectural model to study mutual relationships among these three properties, based on a controller's interaction with the environment. The model takes advantage of the fact that all practical configurations of control systems have a limited set of categories for input/output.

For the proposed framework, we conducted initial experiments to evaluate the validity and effectiveness of the process. In particular a concept of safety shell was successfully applied in security assessment for an ICS system.

Finally, it is worth noting that at this time no single practice, process, or methodology offers a universal "silver bullet" for evaluating system trustworthiness. However, there exist a number of practices and methodologies, to which the presented approach can be adapted to increase the trustworthiness of the produced software, both in its development and operation.

ACKNOWLEDGEMENTS

This project has been funded in part by a grant SBAHQ-10-I-0250 from the U.S. Small Business Administration (SBA). SBA's funding should not be construed as an endorsement of any products, opinions, or services. The first author acknowledges the AFRL 2011 Summer Faculty Fellowship through the American Society of Engineering Education. Additional funding has been provided by the National Science Foundation Award No. 1129437.

Students Michael Humphries (FGCU) and Wendy Stevenson (ERAU) are gratefully acknowledged for assistance in the use of tools and conducting the experiments.

REFERENCES

Burns A., J. McDermid, J. Dobson (1992), On the Meaning of Safety and Security, *The Computer Journal*, Vol. 35, No. 1, pp. 3-15.

Critical Infrastructure Protection: Challenges and Efforts to Secure Control Systems (2004), Report to Congressional Requesters, GAO-04-354, U.S. Government Accounting Office, Washington, DC.

Goertzel K. M., T. Winograd (2011), *Safety and Security Considerations for Component- Based Engineering of Software-Intensive Systems*, Booz Allen Hamilton.

Gumzej R., W. Halang (2009), A Safety Shell for UML-RT Projects Structure and Methods of UML Pattern, *Innovations in Systems and Software Engineering: A NASA Journal*, Vol. 5, No. 2, pp. 97-105.

Hahn J., D. P. Guillen, T. Anderson (2005), Process Control Systems in the Chemical Industry: Safety vs. Security, *Proc. 20th Annual CCPS International Conf.*, Report INL/CON-05-00001.

Jaatun M. G., T. O. Grotan, M. B. Line (2008), Secure Safety: Secure Remote Access to Critical Safety Systems in Offshore Installations, *Proc. ATC 2008, 5th Intern. Conf. on Autonomic and Trusted Computing*, Oslo, Norway, June 23-25, pp. 121-133.

Jalouneix J., P. Cousinou, J. Couturier, D. Winter (2009), *A Comparative Approach to Nuclear Safety and Nuclear Security*, IRSN, Tech. Rep. 2009/117.

Kornecki A., J. Zalewski (2010), Safety and Security in Industrial Control, *Proc. CSIIRW 2010, 6th Annual Workshop on Cyber Security and Information Intelligence Research*, Oak Ridge, Tenn., April 21-23.

Landwehr C. E. (1981), Formal Models for Security, *ACM Computing Surveys*, Vol. 13, No. 3, pp. 247-278.

Leveson N. (1995), *Safeware: System Safety and Computers*. Addison-Wesley, Boston.

Nordland O. (2007), Safety and Security – Two Sides of the Same Medal, *European CIIP Newsletter*, Vol. 3, No. 2, pp. 20-22, May/June.

Novak T., A. Treytl (2008), Functional Safety and System Security in Automation Systems, *Proc. ETFA'08, 13th IEEE Conf. on Emerging Technologies and Factory Automation*, Hamburg, Germany, pp. 311-318.

Pietre-Cambacedes L., C. Chaudet (2010), The SEMA Referential Framework: Avoiding Ambiguities in the Terms "Security" and "Safety", *Intern. Journal of Critical Infrastructure Protection*, Vol. 3, pp. 55-66.

Romanski G. (2009), Safe and Secure Partitioned Systems and Their Certification, *Proc. WRTP 2009, 30th IFAC Workshop on Real-Time Programming*, Mragowo, Poland, October 12-14.

Sanz R., J. Zalewski (2003), Pattern Based Control Systems Engineering, *IEEE Control Systems*, Vol. 23, No. 3, pp. 43-60.

Schoitsch E. (2004), Design for Safety and Security of Complex Embedded Systems: A Unified Approach, *Proc. NATO Advanced Research Workshop on Cyberspace Security and Defense: Research Issues*, Gdansk, Poland, September 6-9, pp. 161-174.

Schwartz M. D. et al. (2010), *Control System Devices: Architectures and Supply Channels Overview*, Report SAND2010-5183, Sandia National Laboratories, Albuquerque, NM.

Smith J., S. Russell, M. Looi (2003), Security as a Safety Issue in Rail Communications, *Proc. SCS 2003, 8th Australian Workshop on Safety Critical Systems and Software*, Canberra, October 9-10, pp. 79-88.

Top 125 Network Security Tools (2012). URL: http://sectools.org/