Interdisciplinary Journal of Information, Knowledge, and Management Volume 3, 2008

Improving Security for SCADA Control Systems

Mariana Hentea
Excelsior College, Albany, NY, USA

mhentea@excelsior.edu

Executive Summary

The continuous growth of cyber security threats and attacks including the increasing sophistication of malware is impacting the security of critical infrastructure, industrial control systems, and Supervisory Control and Data Acquisition (SCADA) control systems. The reliable operation of modern infrastructures depends on computerized systems and SCADA systems. Since the emergence of Internet and World Wide Web technologies, these systems were integrated with business systems and became more exposed to cyber threats. There is a growing concern about the security and safety of the SCADA control systems. The Presidential Decision Directive 63 document established the framework to protect the critical infrastructure and the Presidential document of 2003, the National Strategy to Secure Cyberspace stated that securing SCADA systems is a national priority. The critical infrastructure includes telecommunication, transportation, energy, banking, finance, water supply, emergency services, government services, agriculture, and other fundamental systems and services that are critical to the security, economic prosperity, and social well-being of the public. The critical infrastructure is characterized by interdependencies (physical, cyber, geographic, and logical) and complexity (collections of interacting components). Therefore, information security management principles and processes need to be applied to SCADA systems without exception. Critical infrastructure disruptions can directly and indirectly affect other infrastructures, impact large geographic regions, and send ripples throughout the national and global economy. For example, under normal operating conditions, the electric power infrastructure requires fuels (natural gas and petroleum), transportation, water, banking and finance, telecommunication, and SCADA systems for monitoring and control.

In this paper, we provide an analysis of key developments, architecture, potential vulnerabilities, and security concerns including recommendations toward improving security for SCADA control systems. We discuss the most important issues concerning the security of SCADA systems including a perspective on enhancing security of these systems. We briefly describe the SCADA architecture, and identify the attributes that increase the complexity of these systems including the key developments that mark the evolution of the SCADA control systems along with the growth of potential vulnerabilities and security concerns. Then, we provide recommendations toward an enhanced security for SCADA control systems. More efforts should be planned on reducing the vulnerabilities and improving the security operations of these systems. It is necessary to address not only the individual vulnerabilities, but the breadth of risks that can interfere with critical operations.

We describe key requirements and features needed to improve the security of the current SCADA control systems. For example, in assessing the risk for SCADA systems, use of general methods for risk analysis including specific conditions and characteristics of a con-

Editor: Kathy Lynch

trol system need to be applied. Effective risk analysis for SCADA systems requires a unified definition for mishap and identification of potential harm to safety. As computer systems are more integrated, the distinction between security and safety is beginning to disappear. In bridging the gap between these domains, we propose a unified risk framework which combines a new definition of mishap with an expanded definition of hazard to include the security event.

However, methods for risk management that are based on automated tools and intelligent techniques are more beneficial to SCADA systems because they require minimum or no human intervention in controlling the processes. We also identify a unified security/safety risk framework for control systems. Implementing security features ensures higher security, reliability, and availability of control systems. Thus organizations need to reassess the SCADA control systems and risk model to achieve in depth defense solutions for these systems. The increasing threats against SCADA control systems indicate that there should be more directions in the development of these systems. Therefore, achieving better quality and more secure SCADA control systems is a high priority.

Information security management principles and processes need to be applied to SCADA systems without exception. We conclude with a thought about the future of SCADA control systems. A strategy to deal with cyber attacks against the nation's critical infrastructure requires first understanding the full nature of the threat. A depth defense and proactive solutions to improve the security of SCADA control systems ensures the future of control systems and survivability of critical infrastructure.

Keywords: industrial control system, SCADA control system, cyber security, critical infrastructure, requirements, risk management, security framework.

Introduction

Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other smaller control system configurations including skid-mounted Programmable Logic Controllers (PLC) are often found in the industrial sectors and critical infrastructures. These are also known under a general term, Industrial Control System (ICS). A control system is a device or set of devices to manage, command, direct, or regulate the behavior of other devices or systems. ICSs are typically used in industries such as electrical, water, oil and gas, and chemical including experimental and research facilities such as nuclear fusion laboratories. The reliable operation of modern infrastructures depends on computerized systems and SCADA systems.

The Presidential Decision Directive 63 document established the framework to protect the critical infrastructure and the Presidential document of 2003, the National Strategy to Secure Cyberspace stated that securing SCADA systems is a national priority.

The critical infrastructure includes telecommunication, transportation, energy, banking, finance, water supply, emergency services, government services, agriculture, and other fundamental systems and services that are critical to the security, economic prosperity, and social well-being of the public. The critical infrastructure is characterized by interdependencies (physical, cyber, geographic, and logical) and complexity (collections of interacting components). Cyber interdependencies are a result of the pervasive computerization and automation of infrastructures (Rinaldi, Peerenboom, & Kelly, 2001). The critical infrastructure disruptions can directly and indirectly affect other infrastructures, impact large geographic regions, and send ripples throughout the national and global economy. For example, under normal operating conditions, the electric power infrastructure requires fuels (natural gas and petroleum), transportation, water, banking and finance, telecommunication, and SCADA systems for monitoring and control.

There is a growing concern about the security and safety of the SCADA control systems in terms of vulnerabilities, lack of protection, and awareness (Byres & Franz, 2005; Byres, Hoffman & Kube, 2006). Therefore, information security management principles and processes need to be applied to SCADA systems without exception.

This paper provides a relevant analysis of most important issues and a perspective on enhancing security of these systems. The rest of this paper is organized in sections as follows: next section provides an overview of the SCADA architecture. Then, in the following section, we describe key developments that mark the evolution of the SCADA control systems along with the increase of potential vulnerabilities and security concerns. In the next section, we provide recommendations toward an enhanced security for SCADA control systems. We describe key requirements and features needed to improve the security of the current SCADA control systems. We conclude with a thought about the future of SCADA control systems.

SCADA Architecture

A SCADA system is a common process automation system which is used to gather data from sensors and instruments located at remote sites and to transmit data at a central site for either control or monitoring purposes. The collected data is usually viewed on one or more SCADA host computers located at the central or master site. Based on information received from remote stations, automated or operator-driven supervisory commands can be pushed to remote station control devices, which are often referred to as field devices.

Generally, a SCADA system includes the following components:

- Instruments that sense process variables
- Operating equipment connected to instruments
- Local processors that collect data and communicate with the site's instruments and operating equipment called Programmable Logic Controller (PLC), Remote Terminal Unit (RTU), Intelligent Electronic Device (IED), or Process Automation Controller (PAC)
- Short range communications between local processors, instruments, and operating equipment
- Host computers as central point of human monitoring and control of the processes, storing databases, and display of statistical control charts, and reports. Host computers are also known as Master Terminal Unit (MTU), the SCADA server, or a PC with Human Machine Interface (HMI)
- Long range communications between local processors and host computers using wired and/or wireless network connections.

SCADA systems differ from DCSs (Distributed Control Systems) which are generally found in plant sites. While DCSs cover the plant site, SCADA systems cover much larger geographic areas. Also, due to the remoteness many of these often require the use of wireless communications. Figure 1 shows an integrated SCADA architecture.

SCADA architecture supports TCP/IP, UDP or other IP-based communications protocols as well as strictly industrial protocols such as Modbus TCP, Modbus over TCP or Modbus over UDP, all working over private radio, cellular or satellite networks.

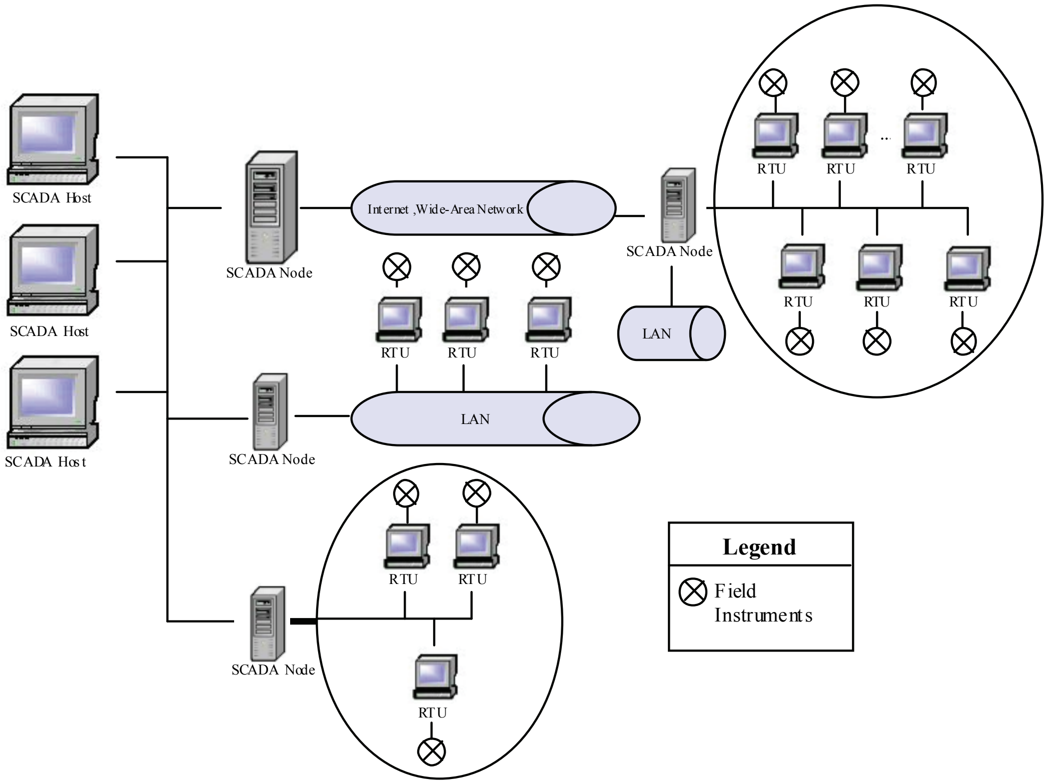

Figure 1. Integrated SCADA Architecture

In complex SCADA architectures, there is a variety of both wired and wireless media and protocols involved in getting data back to the central monitoring site. This enables implementation of powerful IP-based SCADA networks over mixed cellular, satellite, and landline systems. SCADA communications can employ a diverse range of both wired (lease line, dialup line, fiber, ADSL, cable) and wireless media (licensed radio, spread spectrum, cellular, WLAN, or satellite). The choice depends on a number of factors that characterize the existing communication infrastructure. Factors such as existing communications infrastructure, available communications at the remote sites, data rates and polling frequency, remoteness of site, installation budget and ability to accommodate future needs, all impact the final decision for SCADA architecture. In the following section, we discuss key issues in the development of SCADA systems. Therefore, a review of SCADA systems evolution allows us to better understand many security concerns.

SCADA Evolution and Security Issues Escalation

A real world SCADA system can monitor and control hundreds to hundreds of thousands of I/O points. SCADA systems evolve rapidly and are now penetrating the market with a number of I/O channels from 100 K up to near 1 M I/O channels currently under development. DataMonitor, a market research firm, predicted growth of revenue from sales of control systems software with a rate from 3.5 to 4 percent per year through 2009 (Geer, 2006).

SCADA systems evolved from hardware and software in the 1970s to current systems that include standard PCs and operating systems, TCP/IP communications, and Internet access. Security of control systems became a concern issue since the advent of Internet and the rise in terrorist threats. In addition, the LANs that these architectures use raise a new set of security concerns, leading to the introduction of features such as encrypted data sets and dedicated access mechanisms in information assurance applications. In the following, we provide a summary of the most important developments of the SCADA control systems.

In the past, control systems were isolated from other Information Technology (IT) systems. Connection to the Internet is new (early 1990s) and debatable among specialists. Many specialists agree that exposing control systems to the Internet is not a good idea. However, without any connection to the Internet these systems are still vulnerable to external or internal attackers that can exploit vulnerabilities in software such as operating systems, custom and vendor software, data storage software, databases, and applications.

These systems evolved from static to dynamic systems. The increased connectivity to Internet and mobile device technology has also a major impact on control systems architectures. Standardization and use of open market technologies are current requirements in control systems. Modern products are often based on component architectures using commercial off-the-shelf products (COTS) elements as units. This architecture leads to control systems that "are becoming very complex software applications" with the following characteristics (Sanz & Arzen, 2003):

- Time critical
- Embedded
- Fault Tolerant
- Distributed
- Intelligent
- Large
- Open
- Heterogeneous.

SCADA systems are exposed to the same cyberspace threats as any business system because they share the common vulnerabilities with the traditional Information Technology (IT) systems. Also, most SCADA systems are not protected with appropriate security safeguards. The operating personnel is lacking the security training and awareness. Threats against SCADA systems are ranked high in the list of government concerns, since terrorists have threatened to attack several SCADA systems of critical infrastructure (Dacey, 2003) and successfully launched near-disastrous attacks. In addition, recent attacks are becoming more sophisticated and the notion of what kind of vulnerabilities actually matter is constantly changing. For example, timing attacks are now common threats, whereas only a few years ago they were considered exotic. The threats are often poorly understood and ignored, and the vast majority of organizations lag in realizing secure infrastructures. In complexly interactive systems whose elements are tightly coupled, great accidents are inevitable. Vulnerabilities and attacks could be at different levels – software controlling or controlled device, application, storage, data access, LAN, enterprise, Internet, communications.

SCADA systems are now adopting Web technology (ActiveX, Java, etc.) and OPC (as a means for communicating internally between the client and server modules). However, Web applications are an interesting target for cyber attacks that are increasingly automated. Web is the dominant development platform for software, but Web-based secure software is immature. In an average month, Web vulnerabilities accounted for 61% of the total vulnerabilities counted during that month of 2006 (Andrews, 2006). New spectrum of Web worms with a high level of sophistication that exploit vulnerabilities in Web applications are growing fast since 2004, when the first worms propagated themselves via blog links (Holz, Marechal, & Raynal, 2006).

All SCADA systems are based on common software that has one or more vulnerabilities. SCADA systems used to run on DOS, VMS and UNIX based operating systems. Lately, several vulnerabilities were discovered and corrected for the UNIX based operating systems. Although UNIX used to be dominant for SCADA systems, now UNIX systems are often displaced by Linux and Microsoft Windows platforms with new versions of Windows operating system updated every few years. These changes increased the threats against SCADA systems and a need for more awareness. The reaction to these new challenges is diverse. Plant people expect their software to run for years without major modification, and certainly without change in operating system. SCADA vendors release one major version and one to two additional minor versions per year. Thus, these products evolve very rapidly so as to take advantage of new market opportunities, to meet new requirements of their customers, and to take advantage of new technologies.

It is well known that Linux and Microsoft Windows have their own set of vulnerabilities. One factor is the huge number of lines of code for an operating system. Recent studies of software reliability estimate that a Linux kernel may have approximately 15,000 bugs for the Linux kernel that has more than 2.5 million lines of code. At the other end, Windows XP has at least double the number of vulnerabilities since its kernel is more than twice as large as the Linux kernel. It has been demonstrated that bugs in the operating systems do more damage than the bugs in application programs (Tanenbaum, Herder, & Bos, 2006). These vulnerabilities are serious concerns because of the 90 percent concentration of Microsoft Windows operating systems for computers. The total number of vulnerabilities logged by organizations in 2006 was 8064, an increase of 35 percent from the previous year (Erickson, 2007).

Besides security concerns, the computer systems including SCADA control systems raise the issue of safety causing harm and catastrophic damage when they fail to support applications as intended (Dunn, 2003). In January 2003, the Slammer worm infected the safety monitoring systems at the Davis-Besse nuclear plant in US. In 2003, two hackers gained access to control technology for the US's Amundsen-Scott South Pole Station which ran life-support technology for scientists. This attack disabled the safety monitoring system for nearly five hours (Poulsen, 2003). The infamous breach of SCADA for Maroochy water system in Australia (Gellman, 2002) plagued the wastewater system for two months. This caused a leak of hundreds of thousands of gallons of putrid sludge into parks, rivers, and private properties as a result of which marine life died, the creek water turned black and the stench was unbearable for residents.

The reality is that a growing number of worms and viruses spread by exploiting software design, operations, and human interfaces. The software-intensive system design skills for the construction of control systems are often misunderstood. In the control industry, two separate groups of engineers are typically involved in the development of any nontrivial controller: control engineers and programmers. These two groups tend to have very different perspectives and working practices, and both lack the global picture needed for the task (Rinaldi et al., 2001). In addition, cyber attacks exploit vulnerabilities previously not modeled or unknown to a system. SCADA networks were initially designed with little attention to security. SCADA networks traditionally used dedicated telephone lines to send control messages to field devices from the control station and get the current status of the field equipment. However, the modern SCADA networks, integrated with corporate networks and the Internet, have become far more vulnerable to unauthorized cyber attacks putting the national infrastructures at risk and easy targets of attacks by terrorists. By sending a false control message from a computer connected to the Internet, an unauthorized intruder can manipulate traffic signals, electric-power switching stations, chemical process-control systems, or sewage-water valves, creating major concerns to public safety and health.

Although several documents can make the security evaluation task more efficient and effective, there are no obvious magic-bullet solutions. The worst is that these documents can be used by hackers to refine their own attack techniques.

Recently, the US National Infrastructure Assurance Council Initiative (NIAC) for the Common Vulnerability Scoring System (CVSS) was published in 2005 as a first generation open scoring system designed to address a framework for assessing and quantifying consistent scores that accurately represent the impact of software vulnerabilities. Several other databases contain information about vulnerabilities for IT systems, but there is not so much information provided for SCADA systems. While the exact number of cyber attacks against control systems is not known because many enterprises will not tell the public, a few recent cyber attacks and vulnerabilities are disclosed for SCADA systems. In 2005, CERT posted for the first time a few vulnerabilities for SCADA systems (CERT SCADA, 2007), although many more vulnerabilities are reported via other reports and sources (Byres et al., 2006; Geer, 2006). Although CVSS is an emerging standard, it has some limitations. CVSS generates consistent scores for vulnerabilities in the context of software flaws only (Scarfone & Romanosky, 2006) thus leaving other vulnerabilities uncounted.

On the other hand, risk management for SCADA control systems is not a common practice and it is not based on a unique standard to follow. Several methods for risk assessment are emerging (Vidalis & Jones, 2003). Risk management challenges include limitations of these methods, untrained and unskilled developers in business management and organizational areas, various classifications and non unique taxonomies of vulnerabilities. In the following section, we discuss requirements and enhancing features that should be provided to improve the security and safety of SCADA systems. Also, we suggest a unified security/safety risk framework.

Toward Improving Security

Internet and global e-business application requirements demand that companies increasingly implement computing infrastructures specifically designed for at least 99.999 percent availability. This is the equivalent of less than 5.3 minutes of downtime a year. This is also a requirement for the SCADA networks. In response to these trends, government and SCADA owners need to address increased security and support for high availability.

Lately, the government, NIST, academia, and several SCADA vendors have initiated a strategy to support SCADA security. The CVSS NM-SIG for network monitoring is discussing the Information Systems and SCADA risks (Leivesley, 2005). In addition, the Control Systems Security Event Monitoring (SEM) Working Group at Process Control Systems Forum (PCSF) is working on a method to regularly collect statistics from SCADA and DCS networks that are being monitored for cyber security events. SEM's mission is to accelerate the design, development, and deployment of more secure control and legacy systems. Another goal is quantifying the threat for use in risk calculations (PCSF, 2006). However, more information about SCADA vulnerabilities should be provided on these documents.

More efforts should be planned to reduce the vulnerabilities and improve the security operations of these systems. It is necessary to address not only the individual vulnerabilities, but the breadth of risks that can interfere with critical operations.

Hans-Jurgen Weidemann (Conference Reports, 2006) stressed the complexity of control systems and greater need of compliance for safety, quality of service, and security of systems and data. SCADA security design and information security management can be improved by applying a wide range of control principles and methods as well productivity control, involving decision making under uncertainty with increased levels of decision support. Therefore, the improvements for SCADA security have to be broad - at the systems level - and detailed - at the component level. We discuss the most important issues for improving the security in the next subsection.

Key Requirements

Based on the analysis of the past and current developments, we identified key requirements and features that can improve the security of control systems as follows:

Critical path protection

It is mandatory to protect the critical components from cyber attacks. An approach to security evaluation for the identification of critical components is discussed in Rae, Fidge, and Wildman (2006). In control systems, a component failure greatly increases the likelihood of multiple simultaneous failures. Also, the high-speed of SCADA networks facilitates the quick propagation of malicious code.

Stronger safety policies and procedures

Solutions for preventing the attacks are becoming more important to enterprises and software vendors. The best first steps recommended for preventing attacks against control systems are increased awareness of potential vulnerabilities and solutions, as well as implementing stronger safety policies and procedures. It is essential that every enterprise is taking security seriously when developing software, or managing the control systems.

Knowledge management

Organization need to build a strategy that supports knowledge management and training competencies. Security knowledge is likely to include policy, standards, design and attack patterns, threat models, code samples, reference architecture, and secure development framework (Steven, 2006).

System development skills

Specific issues need to be addressed by SCADA systems developers. First, the evaluation of the software is crucial. Software plays an increasingly important role in all types of controllers in SCADA systems. Developers of control systems require programming skills in addition to engineering skills to be able to understand the broad picture and detailed control problems. A common perspective and security solution to the control systems is essential in the design of these systems. New approaches and skills for software development of distributed control systems architectures are required to ensure that software is stable and reliable to avoid hazard conditions (Heck, Wills, & Vatchtsevanos, 2003). Control methods are essential in distributed software development and E-commerce applications (Hellerstein, Diao, Parekh, & Tilbury, 2005).

Enhanced security for device

Since real-time applications alone may never be capable of addressing all security requirements, incorporating security features into a device can further enhance system security. Individual devices erect their own security perimeters and defend their own critical resources, such as a network link or storage media (Cummings, 2002). Although protecting code in embedded systems is sometimes a business decision, a company must weigh the cost of implementation of protection versus the potential loss of service revenue when the vulnerabilities are discovered (Fisher, 2000).

Sensor networks solutions

Security and privacy challenges will require new technological solutions for sensor networks. Sensor networks implementations include monitoring factory instrumentation, pollution levels, etc. Sensor networks consist of hundreds or thousands of sensor nodes. Each node represents a potential point of attack. However securing each node is impractical. Sensor detectors offer one defense against attacks and have the capability to differentiate between the transmissions of authorized and unauthorized sensor networks and other devices (Chan & Perrig, 2003).

Operating system based on microkernel architecture

A more radical method of developing operating systems software based on microkernel architecture to limit the number of lines of code and implicitly the number of bugs is recommended in Tanenbaum et al. (2006). The microkernel technique was used in the past to develop dedicated operating systems for process computer control (Hentea, Balla, Balla, & Rosu, 1978; Soceneantu, Hentea, Balla, Balla, & Rosu, 1978). This technique might be making a comeback to its potentially higher reliability.

Increasing quality of software with security features

Companies such as Klockwork consider that software security is an important and growing aspect of software quality for control systems (Klockwork, 2007). New products are developed to normalize the events in SCADA application logs that are not understood or used in security monitoring. The products from DigitalBond Inc. (DigitalBond, 2007) can detect the security events and provide the pattern a Security Event Monitoring (SEM) can use for SCADA systems.

Security requirements early in the software development cycle

Saydjari (2002) recommends developing a top-down architecture and engineered approach to the problem because current technology is insufficient to defend against cyber warfare. Developing requirements for control systems with security features and use of simulation models based on a framework could improve the definition of requirements and reveal problems early in the software development cycle (Menzies & Richardson, 2006).

Compliance to standards for software development

Control software development can be improved by following documents such as NIST published guidelines SCADA Security (SCADA Security, 2006), NIST Configuration (NIST Configuration, 2005), NIST Guidelines (NIST Guidelines, 2006), general assessment methods and tools for SCADA vulnerabilities (IDAHO Assessment, 2005), and Holzman's rules (Holzmann, 2006).

Integration of different technologies

New languages and platforms such as Java, C#, and CORBA are promising increased ease of use, portability, and safety and contribute to making heterogeneous distributed control system platforms possible. Low-level, real-time technology needs to be combined with high-level aspects, such as programming, networking, security, simulation, and control.

Vulnerability analysis based on proactive, discovery, and adaptation solutions

A changing vulnerability and threat landscape and continuing requirements for compliance are the main drivers for vulnerability management programs to expand. In addition, a strategy to deal with cyber attacks against the nation's critical infrastructure requires first understanding the full nature of the threat.

Vulnerability analysis must focus on identifying (and reducing) the vulnerability of engineered systems to both natural (e.g., weather-related) and man-made (e.g., sabotage, terrorism) disruptions. This implies new tasks related to conceptual and methodological development of risk and vulnerability research in vulnerability modeling, cause mitigation analysis and process component definition including risk and vulnerability assessments for SCADA networks. Also, the focus should be in the development of tools that can provide discovery and training on vulnerability and adaptation. Currently, several vulnerabilities are modeled based on heuristics. In protecting against an attack and maintaining continuous operation, research must focus in vulnerability management.

Vulnerability management consists of a combination of technologies and processes to improve security posture. Targeted threats drive the need for more effective and proactive infrastructure protection solutions. A control system should monitor for cyber attack activities and automatically generate patches to protect an application source code and identify new vulnerabilities (Keromythis, 2004). This assumes that an analysis engine can identify the potential vector attack from the information collected in real time and discover new vulnerabilities.

Vulnerabilities in software (bugs and flaws) can be grouped together by central characteristics and give rise to particular attack patterns (Hoglund & McGraw, 2004). This is based on the premise that related programming errors give rise to similar exploit techniques. A particular exploit usually amounts to the extension of a standard attack pattern to a new target. An attack pattern is a blue print for exploiting software vulnerability. Solutions such as identifying attack patterns to provide general descriptions of vulnerabilities are discussed in (Gegick & Williams, 2005; Jiwnami & Zelkowitz, 2002).

Innovative risk management approaches

Information security management is a continual cyclical process comprised of key phases such as assessment (risk management approach), policy, implementation, training, and auditing. Risk management characterizes an overall process to identify, measure, control, and minimize losses associated with uncertain events or risks. The first phase, called risk assessment, includes analyzing assets, identifying vulnerabilities and potential risks due to threats, risk reducing measures, and decisions related to the acceptance, avoidance, or transfer of risk. Risk assessment characterizes both the process and the result of analyzing and assessing risk. The second phase includes the process of assigning priority to, budgeting, implementing, and maintaining appropriate risk-reducing measures. General models for managing risk through its various phases are available for IT systems (Craft, Wyss, Vandewart, & Funkhouser, 1998; Glavin, 2003). However, methods for risk management that are based on automated tools (Ozier, 1999), and intelligent techniques are more beneficial to SCADA systems because they require minimum or no human intervention in controlling the processes (Hentea, 2006).

The key issue in managing the risk is reducing the vulnerabilities and causes of the vulnerabilities. A vulnerability is a problem that can be exploited by an attacker. To measure risk in a system, it is necessary to identify the vulnerabilities, threats, and asset values. In assessing the risk for SCADA systems, use of general methods for risk analysis including specific conditions and characteristics of a control system need to be applied.

Another important issue is applying vulnerability management life cycle that offers guidance on design and operational processes and technologies needed to find and remediate security weaknesses before they are exploited. It is imperative to analyze risk as a function of asset value, threat and vulnerability. New concepts to analyze the threats and vulnerabilities have to be applied regularly and uniformly. Methods for SCADA security risk analysis based on combining concepts of vulnerability tree analysis, fault tree analysis, attack tree analysis, and the cause-consequence are discussed in (Hentea, 2008; Patel, Graham, & Ralston, 2006).

Ensure authentication, confidentiality, integrity, availability, and non-repudiation

Security of SCADA control systems must ensure far more than the confidentiality of information in transit. It must also ensure that only authorized parties have access to such information, a task that will require abuse-resistant methods for identifying such parties. In the information security context, the SCADA control systems require features that support key security concepts such as authentication, authorization, confidentiality, integrity, availability, and non repudiation. Also, control protocols should be improved to include security features.

Calculate risk as impact to security and safety

Effective risk analysis for SCADA systems requires a unified definition for mishap and identification of potential harm to safety. As computer systems are more integrated, the distinction between security and safety is beginning to disappear. In bridging the gap between these domains, we propose a unified risk framework which combines a new definition of mishap with an expanded definition of hazard to include the security event. We modify the unified definition for mishap described in Stoneburner (2006) by including the hazard due to security event in a model depicted in Figure 2. The figure shows the unified security/safety risk framework. Also, the figure shows that the risk should be calculated as impact of hazard multiplied by likelihood of mishap event.

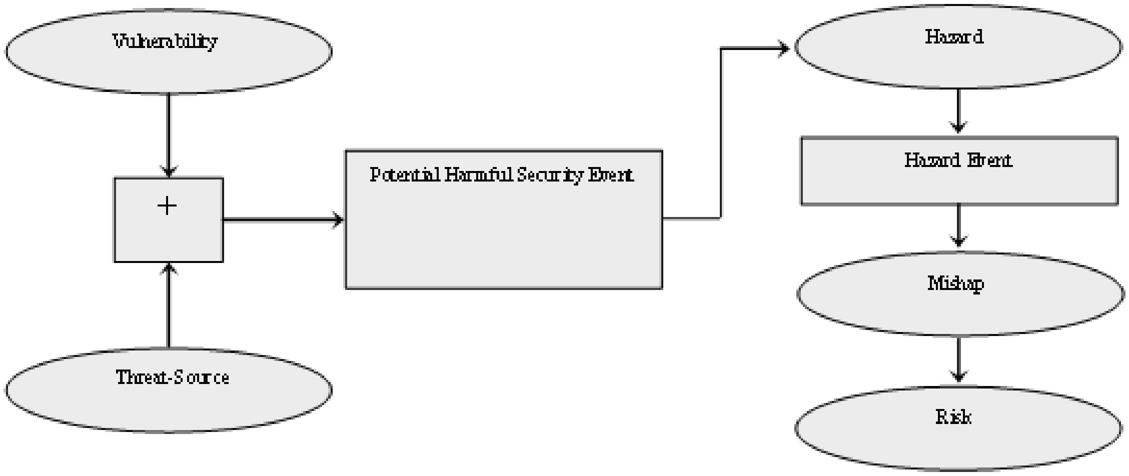

Figure 2. Unified security framework

Nanotechnology: Promise in the future

Nanotechnology can help in building security applications (Arazi, 2006), however research and results in these directions are far away.

Conclusion

Implementing security features as those described above ensures higher security, reliability, and availability of control systems. Thus organizations need to reassess the SCADA control systems and risk model to achieve in depth defense solutions for these systems. The increasing threats against SCADA control systems indicate that there should be more directions in the development of these systems. The above analysis suggests that achieving better quality and more secure SCADA control systems is a high priority.

However, a strategy to deal with cyber attacks against the nation's critical infrastructure requires first understanding the full nature of the threat. A depth defense and proactive solutions to improve the security of SCADA control systems ensures the future of control systems and survivability of critical infrastructure. However, "what the future brings depends on two factors: available technology and societal concern" (Bell, Dooling & Fouke, 1999). Perrow (2006) said that "the public is unaware of our basic (US) vulnerabilities in the chemical industry, electric power industry including nuclear plants."

References

Andrews, M. (2006). The state of web security. *IEEE Security & Privacy*, *4*(4), 14-15.

Arazi, B. (2006). Enhancing security with nanotechnology. *IEEE Computer*, *39*(10), 106-107.

Bell, T. E., Dooling, D., & Fouke, J. (1999). Threshold of the new millennium. *IEEE Spectrum*, *36*(10), 59-64.

Byres, E. J., & Franz, M. (2005). Finding the security holes before the hackers do vulnerability discovery in industrial control systems. *ISA Technical Conference, Instrumentation Systems and Automation Society*, Chicago, October 2005. Retrieved March 12, 2007, from http://www.byressecurity.com/pages/publications/technical-papers/

Byres, E. J., Hoffman, D., & Kube, N. (2006). On shaky ground – A study of security vulnerabilities in control protocols. *5th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology*, *American Nuclear Society,* Albuquerque, NM, November 2006. Retrieved March 15, 2007, from http://www.byressecurity.com/pages/publications/technical-papers/

CERT SCADA. (2007). Retrieved January 27, 2007 from http://www.kb.cert.org/vuls/byid?searchview

Chan, H., & Perrig, A. (2003). Security and privacy in sensor networks. *IEEE Computer*, *36*(12), 103-105.

Conference Reports. (2006). Four focused forums. *IEEE Control Systems Magazine*, *26*(4), 93-98.

Craft, R., Wyss, G., Vandewart, R., & Funkhouser, D. (1998). An open framework for risk management. *21st National Information Systems Security Conference Proceedings.* Retrieved February 21, 2007, from http://csrc.nist.gov/nissc/1998/proceedings/paperE6.pdf

Cummings, R. (2002). The evolution of information assurance. *IEEE Computer*, *35*(12), 65-72.

Dacey, R. F. (2003). *Information security progress made, but challenges remain to protect federal systems and the nation's critical infrastructures*. Retrieved February 10, 2007, from http://world.std.com/~goldberg/daceysecurity.pdf

DigitalBond. (2007). Retrieved February 22, 2007, from http://www.digitalbond.com/index.php/research/

Dunn, W. R. (2003). Designing safety-critical computer systems. *IEEE Computer*, *36*(11), 40-46.

Erickson, J. (2007). Vulnerabilities: What's wrong with this picture?. *Dr. Dobb's Journal.* Retrieved June 15, 2007, from http://www.ddj.com/blog/securityblog/archives/2007/01/whats_wrong_wit.html

Fisher, M. (2000). Protecting binary executables. *Embedded Systems Programming*, *13*(2), 24-30.

Geer, D. (2006). Security of critical control systems sparks concern. *IEEE Computer*, *39*(1), 21-23.

Gegick, M., & Williams, L. (2005). Matching attack patterns to security vulnerabilities in software-intensive system designs. *Proceedings of the 2005 Workshop on Software Engineering for Secure Systems, International Conference on Software Engineering,* St. Louis, Missouri, 1-7.

Gellman, B. (2002, June 22). Cyber-attacks by Al Qaeda feared terrorists at threshold of using internet as tool of bloodshed, experts say. *Washington Post,* p. AD1. Retrieved February 6, 2007, from http://www.washingtonpost.com/ac2/wp-dyn/A50765-2002Jun26?language=printer

Glavin, H. E. (2003). A risk modeling methodology. *Computer Security Journal*, *XIX*(3), 1-29.

Heck, B. S., Wills, L. M., & Vatchtsevanos, G. J. (2003). Software technology for implementing reusable, distributed control systems. *IEEE Control Systems Magazine*, *23*(2), 21-35.

Hellerstein, J. L., Diao, Y., Parekh, S., & Tilbury, D. M. (2005). Control engineering for computing systems. *IEEE Control Systems Magazine*, *25*(6), 56-68.

Hentea, M. (2006). Enhancing information security risk management with data mining and fuzzy logic techniques. *Proceedings of 19th International Conference on Computer Applications in Industry and Engineering*, November 2006, Las Vegas, Nevada, 132-139.

Hentea, M. (2008). A perspective on security risk management of SCADA control systems. *Proceedings of 23rd International Conference on Computers and Their Applications,* April 9-11, 2008, Cancun, Mexico.

Hentea, M., Balla, I., Balla, E., & Rosu, M. (1978). An operating system for process control computers written in concurrent Pascal. *Proceedings of IV-th Symposium on Computer Science*, Cluj-Napoca, Romania.

Hoglund, G., & McGraw, G. (2004). Attack patterns. *Computer Security Journal*, *XX*(2), 15-32.

Holz, T., Marechal, S., & Raynal, F. (2006). New threats and attacks on the world wide web. *IEEE Security & Privacy*, *4*(2), 72-75.

Holzmann, G. J. (2006). The power of 10: Rules for developing safety-critical code. *IEEE Computer, 39*(6), 95-97.

IDAHO Assessment. (2005). Retrieved February 3, 2007, from http://www.oe.energy.gov/DocumentsandMedia/Cyber_Assessment_Methods_for_SCADA_Security_Mays_ISA_Paper.pdf

Jiwnami, K., & Zelkowitz, M. (2002). Maintaining software with a security perspective. *Proceedings of the International Conference on Software Maintenance*. October 3 – 6, Montreal, Quebec, Canada, 194- 203.

Keromythis, A. D. (2004). Patch on demand saves even more time. *IEEE Computer*, *37*(8), 94-96.

Klockwork. (2007). Retrieved March 26, 2007 from http://www.klocwork.com/products/k7_security.asp

Leivesley, S. (2005). *Global terrorism: Governance, business continuity and competitive markets*. Retrieved January 12, 2007, from http://www.first.org/conference/2005/cep/abstracts/june30-05.html

Menzies, T., & Richardson, J. (2006). Making sense of requirements, sooner. *IEEE Computer*, *39*(10), 112-114.

NIST Configuration. (2005). Retrieved January 11, 2007, from http://checklists.nist.gov/download_sp800-70.html

NIST Guidelines. (2006). Retrieved January 28, 2007 from http://csrc.nist.gov/publications/drafts/800-82/Draft-SP800-82.pdf

Ozier, W. (1999). *A framework for an automated risk assessment tool*. Retrieved January 25, 2007, from http://www.theiia.org/itaudit/index.cfm?fuseaction=forum&fid=228

Patel, S. C., Graham, J. H., & Ralston, P. A. S. (2006). Security enhancement for SCADA communication protocols using augmented vulnerability trees. *Proceedings of 19th International Conference on Computer Applications in Industry and Engineering*, Las Vegas, Nevada, 244-251.

PCSF. (2006). Retrieved January 18, 2007, from http://www.controlglobal.com/industrynews/2006/062.html

Perrow, C. (2006). Shrink the targets. *IEEE Spectrum*, *43*(9), 46-49.

Poulsen, K. (2003). Slammer worm crashed Ohio nuke plant network. *Security Focus*. Retrieved on January 16, 2007, from http://www.securityfocus.com/news/6767

Rae, A., Fidge, C., & Wildman, L. (2006). Fault evaluation for security-critical communications devices. *IEEE Computer*, *39*(3), 61-68.

Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine, 21*(6), 11-25.

Sanz, R., & Arzen, K. E. (2003). Trends in software and control. *IEEE Control Systems Magazine*, *23*(3), 12-15.

Saydjari, O. S. (2002). Defending cyberspace. *IEEE Computer*, *35*(12), 125.

SCADA Security. (2006). Retrieved January 15, 2007, from http://www.oe.netl.doe.gov/docs/prepare/21stepsbooklet.pdf

Scarfone, P. M., & Romanosky, K. S. (2006). Common vulnerability scoring system. *IEEE Security & Privacy*, *4*(6), 85-89.

Soceneantu, A., Hentea, M., Balla, I., Balla, E., & Rosu, M. (1978). A concurrent program for real-time resources management. *Bulletin IPTVT*, *Electrotechnica*, *23*(37).

Steven, J. (2006). Adopting an enterprise software security framework. *IEEE Security & Privacy*, *4*(2), 84-87.

Stoneburner, G. (2006). Toward a unified security/safety model. *IEEE Computer*, *39*(8), 96-97.

Tanenbaum, A. S., Herder, J. N., & Bos, H. (2006). Can we make operating systems reliable and secure? *IEEE Computer*, *39*(5), 44-51.

Vidalis, S. & Jones, A. (2003). Using vulnerability trees for decision making in threat assessment. Retrieved January 7, 2007, from http://www.glam.ac.uk/socschool/research/publications/technical/CS-03-2.pdf

Biography

Mariana Hentea is a Faculty and Director of Academic Programs at Excelsior College, Albany, New York, USA. She is a member of ISI, IEEE, ACM, ISCA, IRMA, CSI, (ISC)$^{2®}$, and SWE. She received a MS and Ph.D. in Computer Science from the Illinois Institute of Technology at Chicago, and a B.S. in Electrical Engineering and MS in Computer Engineering from Polytechnic Institute of Timisoara, Romania. She has published papers in a broad spectrum of computer software and engineering applications for telecommunications, steel, and chemical industries. Mariana engineered networks and security systems for telecommunications industry and government. Also, she has been involved in the research and development of novel products based on various telecommunications technologies such as ATM switch, Voice over IP, wireless, broadband access (xDSL, cable), network management, and residential gateway. Her research focuses in computer and network security, network design and architecture, wireless technologies, multimedia systems, home networking, broadband access, and use of Artificial Intelligence techniques for information security management, intrusion and prevention systems, risk management, network management, quality of service, and computer process control in manufacturing.