U.S. Department of Energy
Office of Electricity Delivery
and Energy Reliability

September 2006

LESSONS LEARNED FROM CYBER SECURITY ASSESSMENTS OF SCADA AND ENERGY MANAGEMENT SYSTEMS

Raymond K. Fink
David F. Spencer
Rita A. Wells

NSTB

National SCADA Test Bed

Enhancing control systems security in the energy sector

ABSTRACT

Results from ten cyber security vulnerability assessments of process control, SCADA, and energy management systems, or components of those systems, were reviewed to identify common problem areas. The common vulnerabilities identified ranged from conventional IT security issues to specific weaknesses in control system protocols.

In each vulnerability category, relative measures were assigned to the severity of the vulnerability and ease with which an attacker could exploit the vulnerability. Suggested mitigations are identified in each category. Recommended mitigations having the highest impact on reducing vulnerability are listed for asset owners and system vendors.

CONTENTS

ACRONYMS

ARP	Address Resolution Protocol
CRADA	Cooperative Research and Development Agreement
DMZ	De-Militarized Zone
DNS	Domain Name Service
EMS	Energy Management System
IDS	Intrusion Detection System
INL	Idaho National Laboratory
IP	Internet Protocol
IPS	Intrusion Prevention System
IT	Information Technology
LAN	Local Area Network
NSTB	National SCADA Test Bed
OS	Operating System
SCADA	Supervisory Control and Data Acquisition
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

LESSONS LEARNED FROM CYBER SECURITY ASSESSMENTS OF SCADA AND ENERGY MANAGEMENT SYSTEMS

1. INTRODUCTION

The U.S. Department of Energy (DOE) established the National SCADA Test Bed (NSTB) Program to help industry and government improve the security of the control systems used in the nation's critical energy infrastructures. The NSTB Program is funded and directed by the DOE Office of Electricity Delivery and Energy Reliability (DOE-OE). A key part of the program is the assessment of digital control systems to identify vulnerabilities that could put the systems at risk for a cyber attack.

This paper summarizes the findings from cyber security assessments performed by Idaho National Laboratory (INL) as part of the NSTB Program. Findings are also included from INL assessments performed for the Department of Homeland Security (DHS) under the Control System Security Program, which is managed by INL for the DHS National Cyber Security Division.

The systems that were assessed ranged in complexity from a perimeter protection device, to small digital control systems, to large Supervisory Control and Data Acquisition/Energy Management Systems (SCADA/EMS) with complex networks, multiple servers, and millions of lines of code. Assessments were performed in the INL SCADA Test Bed, in an INL process control systems test bed, and in operational installations (examining non-production or off-line systems).

SCADA/EMS were of the greatest interest in the assessments because of their usual interconnections with various data networks and with the networks that monitor and control critical infrastructure equipment, ranging from valves in oil and gas pipelines to switches and breakers in the national electric grid. If compromised, these systems provide a path to many critical end devices and to other SCADA/EMS.

This paper includes information from ten assessments performed within the DOE and DHS programs during late 2004 through early 2006. These assessments were performed under Cooperative Research and Development Agreements (CRADAs) between the system vendors or asset owners and the INL. The vendors and owners provided software, hardware, training, and technical support. The INL performed the cyber assessments and reported the results, including recommendations on ways to mitigate the vulnerabilities found. As noted above, some of these assessments were conducted at INL; others, at asset owners' sites. Under the terms of the CRADAs and associated nondisclosure agreements, proprietary information is withheld from public disclosure. Results are therefore presented in a generic fashion to protect proprietary information, but every effort has been made to be specific enough to benefit those who provide, use, and secure the systems controlling our nation's critical energy infrastructure. The paper focuses on vulnerabilities that were observed across multiple assessments. A fundamental criterion for including a vulnerability or recommendation in this paper was that it had to be identified in at least two independent assessments. The results summarized in this paper describe vulnerabilities that were found to be common in field installations, spanning different control system vendors and asset owner configurations. Asset owners can use these observations, and the corresponding recommendations for mitigation, as a basis for enhancing the security of their control systems. Control system vendors, system integrators, and third-party vendors can use the lessons learned to enhance the security characteristics of current and future products.

The paper is divided into three sections: assessment methodologies, aggregated assessment results and analysis, and recommendations for vendors and asset owners.

2. ASSESSMENT METHODOLOGIES

The configuration of the tested systems varied considerably depending on the vendor's equipment, where the assessment was conducted (laboratory or in the field), and the specific objectives of the assessment. In all cases, the architecture and boundaries for the system under test were carefully determined. Standardized self-assessment tools were not used; targets of evaluation were developed individually for each assessment.

2.1 Field Assessments

Field assessments were conducted on several systems that had been previously assessed in-house at the INL. The field assessment methodology for these systems focused on the security defenses configured. The field assessments did not include review of code, vulnerability scanning, or manipulation of end devices due to the likely impacts on operations. Some network scanning was performed on non-production systems—if they were available and the asset owner could verify no impact on operations. The limited amount of time available for a field assessment constrained the scope of these assessments. They were tightly coordinated with the owners of the systems to ensure no adverse impacts on the operational environment. All of these assessments were performed on non-production configurations.

2.2 Process Control Systems and Component Assessments

The assessment of process control systems and components focused on the architecture and communication paths of the system. These systems were normally loaded on one or a limited number of PCs. These in-house laboratory assessments allowed for more extensive scrutiny than the field assessments since the concern for impact on a production operation system was absent. It was also possible to use vulnerability scanners, code reviews, and data manipulation to affect end devices during these assessments. Test plans were developed to guide the assessments but were frequently modified to adjust for new information gained as the assessments proceeded. This flexibility was available because the system was in the laboratory and not competing for production resources.

2.3 SCADA and SCADA/EMS Assessments

Assessments of several larger SCADA and SCADA/EMS systems were performed in-house at the INL SCADA Test Bed. These systems reside on multiple servers but could be pared down to a single server installation in some cases. These systems generally involve a greater degree of connectivity to other systems or applications than the process control systems. The SCADA, EMS, and process control systems were configured based on the manufacturers' recommendations. These assessments were closely planned and coordinated with the vendors, with whom the results were then shared.

3. AGGREGATED ASSESSMENT RESULTS

The final reports from ten assessments were reviewed to identify common areas of vulnerabilities, characterize the relative risks associated with these vulnerabilities, and determine appropriate mitigation strategies. This review included only those vulnerabilities that were specifically identified in the formal reports; it excluded vulnerabilities that were suggested only through informal discussions with the principal investigators. Reported vulnerabilities are included in this summary regardless of whether or not they were actually exploited during the assessment. All specific data from these assessments are controlled by the owner of the configuration or the vendor. The results are presented only if at least two dissimilar configurations demonstrated the same vulnerability.

3.1 Categorization of Vulnerabilities

Currently there is no commonly accepted taxonomy for vulnerabilities. For the purposes of this aggregated review, vulnerabilities identified during the assessments were grouped into categories. The categories were defined based on the technical characteristics of the vulnerabilities observed. These categories are described in **Table 1**.

Table 1. Characteristic Categories for Vulnerabilities

Category	Description
Clear Text Communications	Clear text (unencrypted) communications were observed in network traffic (through packet sniffing). The clear text revealed user names and passwords, which might permit replay attacks or simplify the process of reverse engineering of the data protocol. In some cases, clear text communications were observed between the control system network and the external corporate network segments.
Account Management	Privileged accounts were found with default or easily guessed user names and passwords; hard-coded usernames and passwords were defined in documentation or extracted from binary executables or configuration files; password protection policies were weak.
Weak or No Authentication	Little or no authentication of host-to-host communications increases system vulnerability to impersonation, replay, or man-in-the-middle attacks.
Coding Practices	Disassembly or decompilation of executable code revealed potentially unsafe coding styles (particularly with respect to string handling and buffer management); applications were vulnerable to crashing on deliberately malicious input.
Unused Services	Services with known vulnerabilities were running on hosts; need for the service was not apparent in the system architecture.
Network Addressing	Network address resolution protocols (DNS, ARP, etc.) were exploitable by spoofing or other bypassing schemes.
Scripting and Interface Programming	Batch files and other script files (Perl, etc.) could be exploited with malicious input or other techniques.
Unpatched Components	Software modules were not current versions and contained known, exploitable vulnerabilities that were required by the configuration.
Web Servers and Clients	Web servers were not securely configured, allowing directory traversal or file modification.

Category	Description
Perimeter Protection	Connections were initiated from outside the SCADA perimeter; firewalls had unnecessary open ports; access control lists were misconfigured.
Enumeration	Web servers and other network services revealed version information that could be of use to an attacker.

These categories were further subdivided by the types of vulnerabilities observed. The vulnerability classes were defined only when at least two assessments from widely varying configurations exhibited the deficiencies in that class. This requirement eliminates vulnerability classes unique to a single assessment and ensures that the classes reflect deficiencies common across multiple assessments.

3.2 Identification of Recommended Mitigations

After the reported vulnerabilities were categorized as described above, a set of corresponding recommendations for mitigation was developed. These recommendations are based on the assessment reports and reviews by computer security experts. The recommended mitigations tend to be general in nature, with the intent of preserving applicability to the vulnerabilities identified in multiple assessments. As such, they are generic recommendations and require further refinement before implementation on any specific system. A majority of the recommendations will require vendor development, not just a configuration change that can be accomplished by the end user. Based on typical maintenance agreements, changes may have to be approved by the maintenance provider prior to implementation. All changes will have to be tested to determine the impacts on production and operations. Some mitigations would require extensive rewrites and are not feasible for application to current software releases. In these cases, other defensive measures are needed.

3.3 Rating of Vulnerabilities

To help characterize the risk associated with the identified vulnerabilities, the ease of accomplishing a successful attack was rated on a scale (High, Medium, or Low). As shown in Table 2, a “High” rating indicates a greater threat to system security. This rating system considers the relative degree of technical skill that an attacker would need, the extent of system-specific knowledge required, and the length of time needed to exploit the vulnerability. The authors rated the vulnerabilities in consultation with the computer security and control system experts who conducted the original assessments. The likelihood of attack detection is not considered in this measure.

Table 2. Measures for Ease of Attack

Rating	Criteria
NONE (green)	• [not considered exploitable] • An exploit was attempted but did not succeed
LOW (yellow)	• Exploitable only by a highly skilled attacker • Would require days or weeks to exploit • Knowledge of the control system is necessary
MEDIUM (orange)	• Would require a day or less to exploit, or would require the use of multiple scripts or techniques to accomplish the exploit
HIGH (red)	• Exploit tools are available to unskilled attackers • Exploit can be accomplished in less than an hour

It is assumed that an attacker has already gained access to an appropriate point in the system to conduct the exploit. This measure does not address the difficulty an attacker might face in reaching the point at which the vulnerability could be exploited, but focuses instead on how easily an attacker might proceed from that point. That is, this measure assumes that the attacker is inside the security perimeter of the system.

The Ease of Attack is characterized twice, once for the “As Found” condition during the assessment, and again for the “After” case, assuming that the recommended mitigations have been implemented.

3.4 Frequency of Occurrence

As described elsewhere in this document, each assessment had different goals. Not every assessment examined all of the vulnerability areas defined in the matrices. The summary data table (**Table 3**) includes an indicator for how many of the assessments considered and found a particular area of vulnerability. For example, an entry of “3/5” indicates that only five of the assessments considered that vulnerability area, and three of the assessments actually discovered relevant vulnerabilities.

3.5 Summary of Assessment Results

Table 3 presents a summary of the relative ease of attack in the different vulnerability categories, both before and after recommendations for mitigation are implemented. Some recommendations require substantial vendor involvement and cannot be implemented simply with a configuration change. Recommended vendor involvement is noted in the comments section.

This table is an aggregation of the results from individual assessments. In some vulnerability classes, the ease of attack varied across individual assessments; this is indicated by multiple columns in the table. Given the range of goals in individual assessments, and the details of specific vulnerabilities, the table reflects some unavoidable mixing of dissimilar elements. However, the vulnerabilities identified and the recommended mitigation approaches do tend to be applicable across the range of systems.

Table 3. Benefit of Recommended Mitigations

Key:

H	Red = Could be attacked by someone of moderate skill level with commonly available tools
M	Orange = Could be attacked with someone of enhanced skill level
L	Yellow = Could only be attacked by someone of high skill level with enough time and resources
-	Green = Not likely to be exploitable

Category	Vulnerability Class	Recommendations for Mitigation	Ease of Attack		Notes / Comments
			Before Mitigation	After Mitigation	
Clear Text Communications	Passwords and accounts (5/6)	Encrypt communication (SSL/SSH), where feasible	L M	-	Encrypted communications would require > 1 week to compromise
		Disable clear text services			Vendor involvement
	Replay possible (6/6)	Encrypt where feasible	L	-	
	Reverse engineer protocol (6/6)	Encrypt where feasible (SSL)	L	L -	Days to weeks to compromise if unencrypted
		Improve robustness of data validation in protocol		L	Vendor involvement
	Inter-network communication (4/6)	Disable service, where feasible	M H	-	
		Encrypt where feasible Use a network DMZ		-	
	Default accounts (3/5/)	Change default accounts and passwords	H	-	
	Hard-coded or documented passwords (3/5)	Avoid storing hard-coded credential information, or store password hashes instead of plain text passwords	H	M	Hashed password still visible but would take longer to compromise
		Obfuscate login information text in source code via conversion routines	H	M	Obfuscated password still visible but would take longer to compromise

Category	Vulnerability Class	Recommendations for Mitigation	Ease of Attack		Notes / Comments
			Before Mitigation	After Mitigation	
	Session weaknesses (3/3)	Rewrite the software to require both username and password before validating credentials	M H	-	Vendor involvement
		Establish a limit on allowable access attempts, where feasible			Substantial increase in difficulty to compromise
	Weak passwords and password operations (5/6/)	Improve password policies and employ password complexity requirements	M H	-	Strong passwords would take weeks to compromise
		Modify software to allow strong passwords		-	
Authentication	No authentication (4/7)	Enforce authentication	L M H	-	
		Upgrade OS to one with better authentication, where feasible		-	Vendor involvement
		Use hardware authentication where feasible		-	
	Weak authentication (3/6)	Eliminate older protocols	L H	-	Vendor involvement
		Use hardware authentication (where feasible) or other host-specific authentication		-	
Coding Practices	Unchecked data stream resulting in buffer overflow (6/6)	Use commercially available tools during development to check for unsafe conditions	L	-	Vendor involvement
		Use robust set of data validation and sanity checking		-	Vendor action
		Eliminate use of language that has unsafe buffer operations, where feasible		-	Vendor involvement
	Miscellaneous (5/6)	Ensure the latest patches are implemented	L	-	Vendor involvement
		Don't hardcode database tables (use configuration files)		-	Vendor involvement
		Use encryption, where feasible		-	Vendor involvement

Category	Vulnerability Class	Recommendations for Mitigation	Ease of Attack		Notes / Comments
			Before Mitigation	After Mitigation	
		Improve established policies during software development and throughout software life		-	Vendor involvement
		Use code obfuscation if applicable		-	Vendor involvement
	Reverse engineering (4/4)	Use hardware or firmware-based equipment	L	-	Vendor involvement
		Ensure strict file system access		L	Compromise slowed by a few hours; vendor involvement
Unused Services	Exploitable code (5/5)	Disable / remove unused services	H	-	Vendor assistance
		Establish administrative policies to include documentation and periodic review of necessary services		-	Vendor assistance
Network Addressing	Man-in-the-middle (MitM) (5/5)	Encrypt, where feasible	L	-	Vendor assistance
		Apply strict traffic control through routers		-	Vendor assistance
		Hardcode ARP tables where feasible		-	Vendor assistance
		Use IDS to detect unusual network traffic		-	
	Other exploitable addresses (3/3)	Keep patching up to date	L M H	-	Vendor assistance
		Encrypt where feasible		-	Vendor assistance
Scripting or Other Interface Programming	Various (3/6)	Establish administrative policies to disallow browsing from control system network to internet	L M	L	Time to compromise increases from hours to days
		Use DMZ, proxy servers, IDS to watch traffic		L	Increased likelihood of detection
		Filter or block unnecessary traffic		L	Time to compromise increased from hours to day
		Keep patching up to date		-	

Category	Vulnerability Class	Recommendations for Mitigation	Ease of Attack		Notes / Comments
			Before Mitigation	After Mitigation	
		Use robust data validation and sanity checking	L M	-	Vendor assistance
Unpatched Components	Various (4/6)	Update patches	H	-	Vendor assistance
		Establish administrative policies to ensure periodic review		-	Vendor assistance
		Eliminate unneeded services		-	Vendor assistance
Web Servers and Web Clients	Various (5/5)	Ensure latest patched are made	L M H	-	Vendor assistance
		Establish well-defined system configuration controls		-	Vendor assistance
		Establish administrative policies to eliminate default configuration, test services, and configuration directories		-	Vendor involvement
		Use DMZ if server is necessary, otherwise shut down or don't use the server		-	Vendor assistance
		Initiate connections from the most secure to the less secure sub-networks		-	Vendor involvement
	Directory transversal (4/4)	Reconfigure web server	H	-	Vendor assistance
		Set correct file permissions		-	Vendor assistance
Perimeter Protection	Additional connections bypassing protection (2/3)	Establish administrative policies concerning laptop use	L H	-	
		Perform periodic access point audits		-	
		Perform network integrity audits		-	
	Exploitable ports and services (6/6)	Disable or remove any unneeded services	L H	-	Vendor assistance
		Implement detection and monitoring		L	Probable improved chances of detection; vendor assistance

Category	Vulnerability Class	Recommendations for Mitigation	Ease of Attack		Notes / Comments
			Before Mitigation	After Mitigation	
		Use hardware from multiple vendors	L H	L	Compromise slowed but not prevented; vendor assistance
	Misconfigured firewalls (3/3/)	Disable or remove any unneeded services	H	-	Vendor assistance
		Close unnecessary port		-	
		Configure firewall correctly		-	
		Set default-deny on connections		-	Vendor assistance
		Add filtering to disable pinging		-	
Enumeration	Revealing versions (6/6)	Obfuscate banners or eliminate if not needed	H	M	Reduces information available to an attacker, increasing time needed to compromise; vendor assistance

4. RECOMMENDATIONS

Table 3 includes suggested mitigation strategies in each vulnerability category. Some of these actions can be implemented in the field by asset owners; other mitigations involve design changes that must originate with vendors and system integrators. This section identifies high-priority recommendations for each of those groups.

The recommendations address vulnerabilities of high or moderate severity that can be eliminated or made much more difficult to attack successfully. Due to the unique characteristics inherent in every system, it will not be possible to apply every recommended mitigation in any single system design or deployment. However, a defense-in-depth strategy should be used to avoid over-reliance on any one particular security measure.

4.1 Recommendations for Asset Owners

These recommendations primarily apply to the in-the-field configuration. As one might expect, they reflect common practices in conventional IT security. Effective implementation of some of these recommendations will require cooperation from vendors and system integrators.

Action #1:	Implement effective patch management policies to ensure that operating systems and installed applications are kept as up-to-date as possible with released patches.
Benefit:	Reduces the exposed attack surface associated with known vulnerabilities. Patches are frequently released in response to publicly identified vulnerabilities.
Considerations:	Close coordination with control system vendors is needed to ensure compatibility of operating system and security patches with control system servers and workstations. Negotiate a flaw remediation approach with the vendor, including the expected time for correction. Coordinate with the vendor to identify defense-in-depth strategies to protect the system prior to implementing the patch or upgrade.

Action #2:	Remove or disable unnecessary services on control system servers and workstations. Only those services required for control system operation should be enabled.
Benefit:	Eliminates the readily exploitable code associated with such services.
Considerations:	Verify list of required services with the control system vendor. During factory acceptance testing and site acceptance testing, vulnerability scans can be conducted on these non-production configurations. The results will identify known vulnerabilities in components and the patching levels needed. This output can also be used to identify all applications on the system and aids in the elimination of unneeded components.

Action #3:	Adopt account management policies reflecting conventional IT best practices. Replace default usernames whenever possible. Establish password policies ensuring appropriate password complexity and prohibiting short or easily guessed passwords.

Benefit:	Reduces or eliminates the vulnerabilities (ranging from default accounts to weak passwords) that provide opportunities for an intruder to gain entry into the system.
Considerations:	Coordinate with the control system vendor on any default usernames. Identify session weaknesses, such as cookies or remembered account names between sessions, and coordinate with the vendor for resolution. Coordinate with your in-house IT department for account, password, and user activity logging to identify areas where resources and policies can be joined.

Action #4:	Isolate the control system perimeter from the corporate network by an appropriate combination of firewalls and DMZs. Configure firewalls to block inbound connections, and limit outbound connections to only those specifically required for operations. Only allow specifically necessary network protocols in the DMZ; block or filter unnecessary protocols. Eliminate network connections that bypass perimeter protection.
Benefit:	Eliminates weaknesses in control system perimeter protection and increases the difficulty for an external attacker to exploit other vulnerabilities. Because of its public visibility and accessibility, the corporate network is vulnerable to intrusion through the use of publicly available hacking tools. Without the added protection of a DMZ or carefully configured firewalls, disruption of the control network is possible by relatively unskilled attackers.
Considerations:	Work with vendor or system integrator to identify all traffic between control system or SCADA and the DMZ. Identify the originator of the communication and the sequence of re-establishing communications upon failure. This information is critical in designing a DMZ. Coordinate with in-house IT departments for relevant expertise to identify areas where resources and policies can be joined.

Action #5:	Implement security hardening of web servers located within the control system network (or having access to the control system network) to establish least-access permissions.
Benefit:	Eliminates directory traversal attacks and other common vulnerabilities.
Considerations:	(none identified)

Asset owners may also wish to determine if vulnerability assessments have been conducted on systems similar to their own; this information would be available through the respective system vendors. If assessments have been conducted, the results should be reviewed to identify any system-specific vulnerabilities that merit additional field mitigations. During the specification and bid evaluation process for new procurements, asset owners should consider the recommendations for system vendors presented below.

4.2 Recommendations for System Vendors

Vendors who have conducted assessments on their own systems will already be familiar with the detailed findings for their systems. The following recommendations primarily apply to system design, rather than field deployment issues.

Action #1:	Establish patch management and review processes to verify compatibility of patches for operating systems and required third-party applications.
Benefit:	Reduces the exposed attack surface by eliminating known vulnerabilities that are often published in the open.
Considerations:	(none identified)

Action #2:	Modify protocols to eliminate clear text network transmission of usernames and passwords. At a minimum, passwords should be transmitted only in a suitable hashed or encrypted format.
Benefit:	Eliminates an attacker’s ability to easily obtain this information and then use it to gain access to the control system.
Considerations:	(See section 4.3.1 below regarding Encrypted Networks)

Action #3:	Modify protocols to include support for authentication of connections between sender and receiver.
Benefit:	Eliminates the vulnerabilities associated with impersonation, replay, and man-in-the-middle attacks.
Considerations:	May require action by industry standards organizations

Action #4:	Adopt appropriate software development life-cycle practices to eliminate common coding errors that affect security, particularly with respect to input data validation and buffer management.
Benefit:	Eliminates many common sources of security vulnerabilities that allow uploading of malicious code onto control system servers.
Considerations:	Also applies to system integrators who perform software development for turnkey installations. For legacy software, code can be reviewed to identify the most common security vulnerabilities. Protection of the source code on the installation facility is needed to prevent malicious actors from identifying vulnerabilities once inside an installation. Searching the installed configuration for known accounts and passwords to eliminate those hard-coded values is also recommended.

Action #5:	Adopt a role-based security model, limiting user privileges to only those needed for specific tasks.
Benefit:	Limits the potential damage an attacker could accomplish after exploiting the vulnerabilities.
Considerations:	(none identified)

4.3 Discussion of Selected Protective Measures

4.3.1 Considerations for Encrypted Networks

In several of the vulnerability categories, “encryption, where feasible” is recommended as a mitigation. The benefit of an encrypted data protocol is that it becomes substantially more difficult for an attacker to obtain usernames and passwords, reverse engineer the protocol, or insert malicious data streams. However, encryption of the control system data protocol can present performance issues and increase the complexity of system development and maintenance processes. Furthermore, encrypted data traffic becomes essentially opaque to network monitoring and intrusion detection systems. Although encryption can eliminate some vulnerabilities, it should be implemented only after a careful evaluation of operational constraints and network monitoring policies.

4.3.2 Detection and Monitoring Tools

Malware Detection. Antivirus, spyware, and bot detectors are typically not used within the control system perimeter. The download of the signature-based virus scanners or the process of performing a scan may have the effect of a denial of service on most control system networks. Some vendors supply tested virus protection mechanisms with their systems; others require extensive testing prior to installing on the inner network. These capabilities are typically established at the control system perimeter as a minimum.

Intrusion Detection and Prevention Systems. Anomaly-based intrusion detection systems (IDS), which report deviations from a known traffic baseline, are suitable for control system network environments in which a flood of traffic (such as exception reporting from end devices during bad weather) is not expected or the IDS system has been tuned for such incidents. The signature-based IDS typically work on TCP/IP only and are not tuned for proprietary protocols found in control system networks. However, some work is being done in industry on vendor-independent IDS rules for common protocols, and some control system vendors are now providing guidelines for IDS monitoring.

Host-based IDS (HIDS) can be used to detect new files on a host computer, system administrative access, and the escalation of privileges. Point data files (the status of the endpoints) change frequently and could lead to creation of very large log files, unless these data files are excluded from HIDS coverage. Another type of host-based IDS is a “canary” type of honeypot. (This is a host that performs no function for the process control operations, and no other devices on a static, addressed, inner network would communicate with this host.) If this host does receive a communication from some process, it causes an alarm. This poison box can be used to detect attempts to enumerate the network.

Intrusion prevention systems (IPS) are less common on control systems. If the IPS is not carefully tuned for proprietary control system protocols, the active response may shut down communications. For example, some configurations with many end devices use a fieldbus architecture with the end devices only reporting on exception. During the restoration of end devices (e.g., after a storm), these end devices report their status back to the SCADA network. The User Datagram Protocol (UDP) is common in these architectures. The flood of these packets can be misinterpreted as a denial of service attack on the network, and an IPS not tuned to the network will shut down these connections during the critical stages of restoration.

All of these activities produce logs and are of little use unless the logs are reviewed on a regular basis. These logs aid in identifying what happened during an incident. The network dumps can also be used to verify what is being transmitted through the firewalls and what the normal network traffic looks like. This aids in tuning the firewalls and IDSs. The system logs allow the user to figure out the host activities. These logs can be reviewed together to determine the order of events.

4.4 General Recommendation – A Proactive Security Model

All of the above recommendations can be used individually to improve the security configuration of a system on a piecemeal basis. However, this is a reactive approach that does not necessarily enhance the overall security picture. Recommended practices in computer security support a proactive security model such as the one shown in **Figure 1**; the following description is adapted from Reference 1.

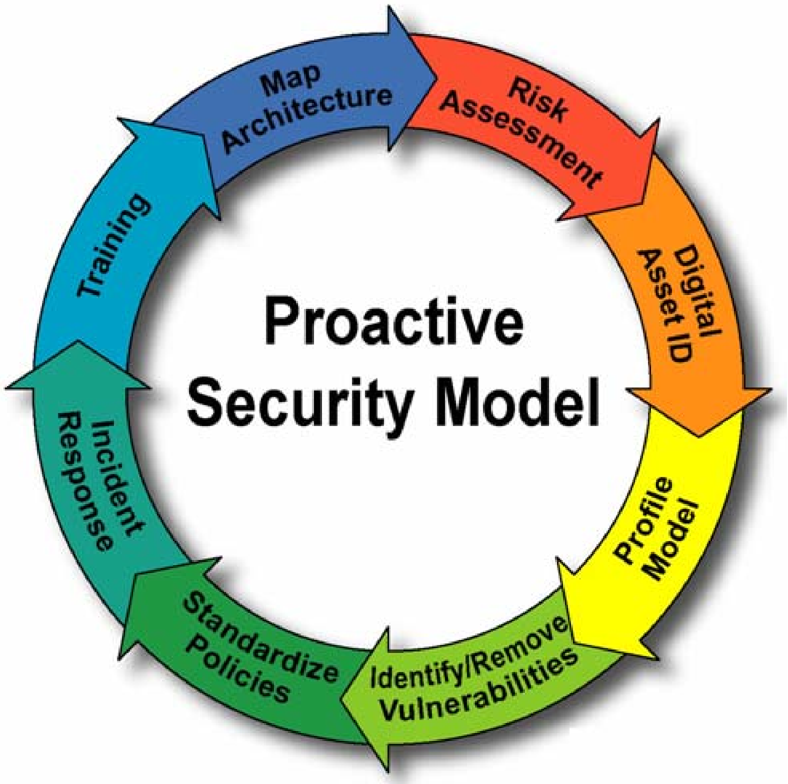

Figure 1. Proactive Security Model

- The first step is to map out the architecture and understand what components are in place and what their communication paths are. This step is essential for identifying all communication paths that need to be addressed in effective perimeter protection.

- The next step is to perform a risk assessment. Understanding the potential impacts of the vulnerabilities and threats is critical to focusing resources appropriately.

- Understanding where the assets are located is required to ensure the physical protection of the assets.

- Creating a protection profile will aid in understanding the type of protection needed for each critical asset. This will aid in the prioritization and allocation of resources.

- Application of available patches and removal of unused services will reduce opportunities for attack.

- Creating a standardized policy ensures that efforts to harden the operating system or segment the network are not undone by poor configuration management practices.

- Implementing effective processes, including ones for monitoring and retention of log files, is important for incident response and recovery. Monitoring of the network logs will help to detect the replay or man-in-the-middle types of attacks used in several of the assessments.

- Training the users will facilitate compliance with policies and help users perform effective response and recovery actions in the event of a cyber intrusion or attack.

As the diagram suggests, a proactive security model is an ongoing process rather than a one-time activity. The typical lifespan of these systems, ten to fifteen years, underscores the need for an ongoing security process.

5. CONCLUSION

The results summarized in this paper describe vulnerabilities that were found to be common in the control systems that were provided by different vendors and configured in different ways by asset owners. Asset owners can use these observations, and the corresponding recommendations for mitigation, as a basis for enhancing the security of their control systems. Control system vendors, system integrators, and third-party vendors can use the lessons learned to enhance the security characteristics of current and future products.

REFERENCE

1. Anonymous, *Maximum Security, Fourth Edition*, Sams Publishing, Indianapolis, 2003.