Common Cybersecurity Vulnerabilities in Industrial Control Systems

May 2011

Homeland Security

Control Systems Security Program

National Cyber Security Division

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, or any information, apparatus, product, or process disclosed in this publication, or represents that its use by such third party would not infringe privately owned rights.

ACKNOWLEDGMENTS

Trent Nelson, Project Manager, Idaho National Laboratory;
May Chaffin, Cyber Researcher, Idaho National Laboratory

EXECUTIVE SUMMARY

The U.S. Department of Homeland Security (DHS) National Cyber Security Division's Control Systems Security Program (CSSP) performs cybersecurity vendor assessments, ICS-CERT operations, and asset owner cybersecurity evaluations with the Cyber Security Evaluation Tool (CSET) evaluations for industrial control systems (ICS) to reduce risk and improve the security of ICS and its components used in critical infrastructures throughout the United States. ICS differs from other computer systems because of legacy-inherited cybersecurity weaknesses and the significance of the impact of potential exploitation to the U.S.

In 2009,a report titled "Common Cyber Security Vulnerabilities Observed in DHS Industrial Control Systems Assessments" compiled common vulnerabilities identified during 15 security assessments of new ICS products and production ICS installations from 2004 through 2008. Three additional ICS product assessments were performed in 2009 and 2010. This newer, 2010 version is an update to the 2009 version and has been developed to proactively create greater awareness within the ICS community. Correlated and compiled in this report are vulnerabilities from general knowledge gained from DHS CSSP assessments and Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) activities describing the most common types of cybersecurity vulnerabilities as they relate to ICS. This information is derived from DHS CSSP experiences of the following types:

- Assessments of ICS products
- Published products derived from ICS-CERT operations, including ICS-CERT incident response
- Self-assessments of asset-owner facilities using the Cyber Security Evaluation Tool (CSET).

Cybersecurity vulnerability and mitigation information from authoritative sources is referenced to guide those responsible for securing ICS used in critical infrastructures throughout the United States.

The highest percentage of vulnerabilities identified in ICS product assessments continues to be improper input validation by ICS code. Poor access controls—credentials management and security configuration—were the second most common security weakness identified in new ICS software in 2009–2010. Authentication weaknesses follow in third place. However, vulnerabilities reported from the previous CSSP ICS product assessments include more patch management problems than the more recent findings.

ICS-CERT alerts match 2009–2010 CSSP assessment findings, with most of the published ICS vulnerabilities due to improper input validation, but have a much higher percentage of password weaknesses. See Figure EX-1.

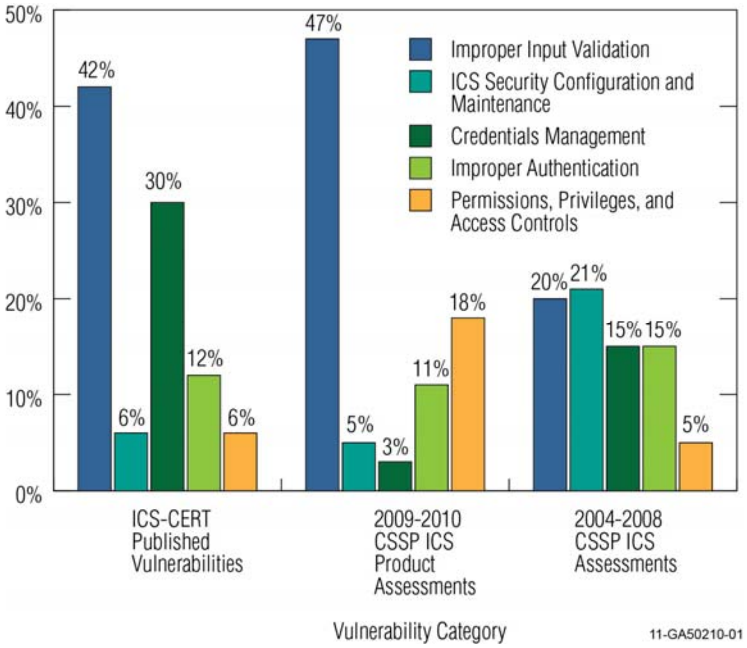
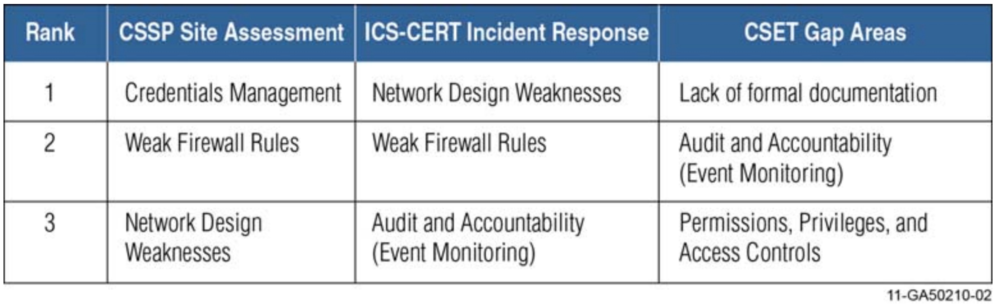


Figure EX-1. Comparison of ICS software security weaknesses.

Production system assessments were performed using the CSET policy-based self-assessment tool in 2009–2010. Individual site vulnerabilities were not recorded from these assessments, but summary reports indicate that the lack of formal documentation is the most common gap identified. ICS-CERT incident response participants have observed an overall lack of defense-in-depth at ICS installations. Prior CSSP site assessments found that the most common configuration problem was credentials management (i.e., weak passwords and insufficiently protected credentials), followed by weak or non-existent firewall rules and network design weaknesses. Table EX-1 ranks the security problem areas identified at production ICS sites.

Table EX-1. Most common weaknesses identified on installed ICS.



Rank	CSSP Site Assessment	ICS-CERT Incident Response	CSET Gap Areas
1	Credentials Management	Network Design Weaknesses	Lack of formal documentation
2	Weak Firewall Rules	Weak Firewall Rules	Audit and Accountability (Event Monitoring)
3	Network Design Weaknesses	Audit and Accountability (Event Monitoring)	Permissions, Privileges, and Access Controls

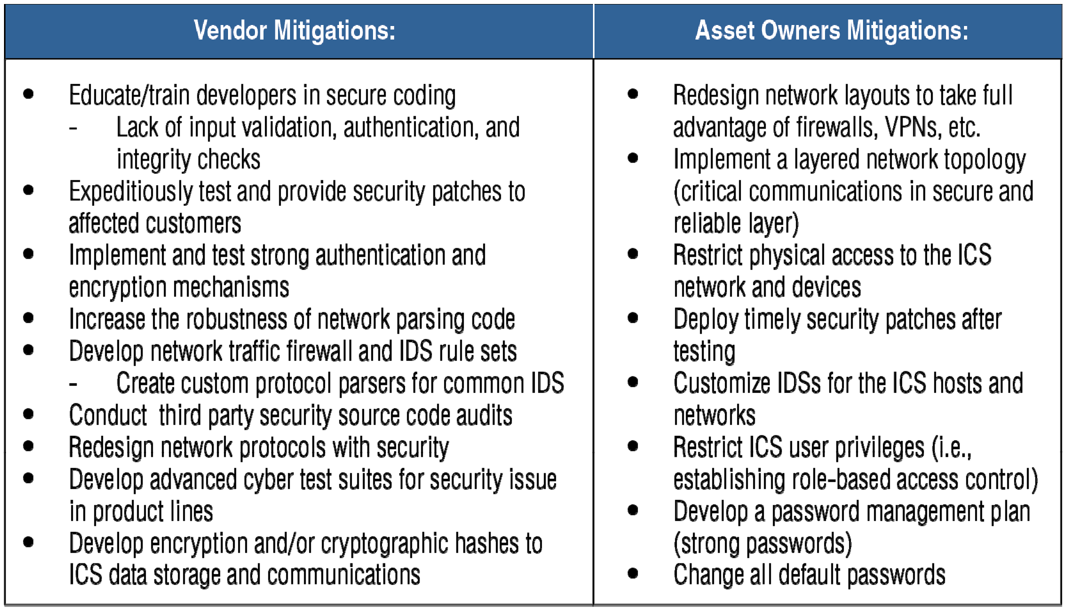
11-GA50210-02

The identified common vulnerabilities from the CSSP assessments are shared here to increase security awareness and mitigation. ICS vendors and owners can learn and apply many common computer-security concepts and practices to secure and protect their systems. Security should be designed and implemented by qualified security and ICS experts who can verify that the solutions are effective and can make sure that the solutions do not impair the system’s reliability and timing requirements. Given the nature of the vulnerabilities found in ICS, asset owners cannot always directly fix them. Thus, as asset owners wait for vendor patches and fixes, the design and implementation of defense-in-depth[a] security strategies that aid in protecting the ICS from attack is part of an effective, proactive security program. Such a program is a necessity because attack strategies are constantly evolving to compensate for increasing defense mechanisms.

To encourage a proactive program, vendors should offer or support security products and features that can be used as layers of defense to help protect ICS installations. Owners should add the additional network perimeter layers of defense and actively update and monitor the system. Increasing the hurdles required to attack a system decreases the chance that attackers will be able to subvert all hurdles and increases the chance that the attackers will give up before accomplishing their goals. Designing security into the system and using secure coding and best practices regarding security can also minimize damage from attacks by insiders, social engineers, or anyone else with access behind the ICS network perimeter.

ICS product vendors are responsible to deliver systems that are able to survive attack without compromising critical functionality. ICS owners must ensure that the physical systems they operate do not put lives, the economy, or the environment at risk by the owners' failing to perform due diligence in procuring, configuring, securing, and protecting the ICS for critical infrastructure. In support of this goal, Table EX-2 presents recommendations for establishing the best possible defense against evolving attack strategies.

Table EX-2. Vendor Mitigations.



Vendor Mitigations:	Asset Owners Mitigations:
• Educate/train developers in secure coding - Lack of input validation, authentication, and integrity checks • Expeditiously test and provide security patches to affected customers • Implement and test strong authentication and encryption mechanisms • Increase the robustness of network parsing code • Develop network traffic firewall and IDS rule sets - Create custom protocol parsers for common IDS • Conduct third party security source code audits • Redesign network protocols with security • Develop advanced cyber test suites for security issue in product lines • Develop encryption and/or cryptographic hashes to ICS data storage and communications	• Redesign network layouts to take full advantage of firewalls, VPNs, etc. • Implement a layered network topology (critical communications in secure and reliable layer) • Restrict physical access to the ICS network and devices • Deploy timely security patches after testing • Customize IDSs for the ICS hosts and networks • Restrict ICS user privileges (i.e., establishing role-based access control) • Develop a password management plan (strong passwords) • Change all default passwords

11-GA50210-13

[a] http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf, Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies

CONTENTS

FIGURES

TABLES

Common Cybersecurity Vulnerabilities Identified in DHS Industrial Control Systems Products

1. INTRODUCTION

The U.S. Department of Homeland Security (DHS) National Cyber Security Division's Control Systems Security Program (CSSP) performs cybersecurity assessments of industrial control systems (ICS) to reduce risk and improve the security of ICS and their components used in critical infrastructures throughout the United States. DHS also sponsors the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) to provide a control system security focus in collaboration with US-CERT (United States Computer Emergency Readiness Team). This report has been developed to share the knowledge and information gained by both of these programs.

This report correlates and compiles vulnerabilities from general knowledge gained from DHS CSSP assessments and ICS-CERT activities and reports the most common types of cybersecurity vulnerabilities as they relate to ICS. DHS CSSP derives the information based on the following activities:

- Cybersecurity assessments of ICS products
- Published products derived from operation of ICS-CERT
- Self-assessments of asset owner facility using the Cyber Security Evaluation Tool (CSET).

The term "ICS," as used throughout this report, includes Supervisory Control and Data Acquisition (SCADA) systems, Process Control Systems, Distributed Control Systems, and other control systems specific to any of the critical infrastructure industry sectors. Although differences in these systems exist, their similarities enable a common framework for discussing and defining security controls. Standard cybersecurity concepts apply to all computer hardware and software, and common issues in ICS can be discussed in general terms.

Common ICS vulnerabilities and associated recommendations are discussed in this report. Insight is gained into the current state of ICS security through high-level analysis of the problem areas by information gathered from CSSP ICS security assessments and ICS-CERT alerts, advisories, and incident response.

This report is organized in three sections. First, the different sources of ICS vulnerability information are summarized. Then the common ICS vulnerabilities are presented according to categories that describe a general problem observed in multiple ICS security assessments. These three general categories are grouped by:

1. Vulnerabilities inherent in the ICS product
2. Vulnerabilities caused during the installation, configuration, and maintenance of the ICS
3. The lack of adequate protection because of poor network design or configuration.

Nonattributable ICS vulnerabilities are listed with the common vulnerability descriptions to aid in understanding the issues. General recommendations based on empirical knowledge gained through performing ICS security assessments are then grouped by software development recommendations for ICS vendors, ICS network configuration, and maintenance recommendations for ICS owners.

2. VULNERABILITY INFORMATION SOURCES

This report is an update of a previous report first published in 2009.[1] The previous document compiled common vulnerabilities identified during cybersecurity assessments of new ICS products and production ICS installations. This report adds the information gained from subsequent ICS cybersecurity assessments with new content from ICS-CERT products, field-knowledge gained by ICS-CERT incident response, and onsite assessments assisting ICS owners in using the CSET self-assessment tool.

These different sources of ICS vulnerability information provide a more complete picture of ICS security: (1) CSSP has performed cybersecurity assessments of ICS software and production installations since 2004, (2) ICS-CERT started publishing vulnerability information and assisting in incident response in 2010, and (3) CSSP has assisted in Control System Cyber Security Self-Assessment Tool (CS2SAT) and CSET policy self-assessments since 2006, Each of these sources is covered in the subsequent sections followed by a discussion of the compiled source information and a comparison against information from past years.

2.1 CSSP ICS Security Assessments

The DHS National Cyber Security Division established the CSSP to help industry and government improve the security of the ICS used in critical infrastructures throughout the United States. A key part of the CSSP mission is the assessment of ICS to identify vulnerabilities that could put critical infrastructures at risk to cyber attack. Once these vulnerabilities are identified, mitigation strategies are developed to enhance ICS security.

CSSP has established a collaborative effort among vendors, owners/operators, industry partners, and other national laboratories to provide an assessment environment where ICS can be evaluated for security vulnerabilities. This controlled environment allows realistic assessments of systems and components without the adverse consequences resulting from potential system failures.

Assessments are performed at Control Systems Analysis Center, located at the Idaho National Laboratory, to evaluate vendors' ICS software. Assessments also are performed at ICS sites in order to assess security issues due to the interdependencies and network design of operational ICS installations. Operational ICS assessments use nonintrusive methods, such as reviewing the production system network diagrams and firewall rules, and performing a hands-on assessment of a duplicate nonproduction installation of the system.

The primary goal of the CSSP cybersecurity assessments is to improve the security of the critical infrastructure by delivering to each industry partner a report of all security problems found during the assessment along with associated recommendations for improving the security of their product or infrastructure (as appropriate). The CSSP has performed assessments on a large variety of systems, and for each assessment, CSSP tailors the assessment plan and methodology to provide the most value to the customer owning the system. System configurations also vary considerably depending on ICS functionality, negotiated objectives, and whether the assessment was conducted in the laboratory or onsite. In all cases, the architecture and boundaries for the system under test are carefully determined. Assessment targets are developed individually for each assessment based on the system configuration and assessment focus in order to address the concerns of the partners. Although a common approach is used for all assessments, the details of each assessment vary; the fact that a vulnerability was not listed on a particular system report does not imply that it did not exist on that system. CSSP vulnerability identification activities focus on enabling the identification and remediation of the highest risk ICS cybersecurity vulnerabilities rather than the collection of data for statistical purposes. One should keep this in mind when interpreting common vulnerability data.

Laboratory assessments are designed to evaluate vendor-specific products and services, such as custom protocols, field equipment, applications, and services. Ideally, the systems are assessed in multiple phases: (1) a baseline system assessment that identifies vulnerabilities in the vendor's default configuration and (2) an evaluation of the system following implementation of mitigation strategies based on baseline assessment results. In some cases, more than two assessments have been performed on different versions of an ICS. Assessment projects typically leverage a full-disclosure approach with the vendor and asset-owner partners. The CSSP focus is on the ICS and its perimeter. By collecting background architecture, policy, and configuration data from a project partner, the team can perform a more thorough assessment of the system. Penetration testing is a security validation process performed by many commercial entities. CSSP does not simulate a blind attack or penetration of the system, but instead works with the project partner to gain the best understanding of security issues obtainable within the time constraints, and provide insight to help mitigate the vulnerabilities found.

2.1.1 Common CSSP ICS Cybersecurity Assessment Vulnerabilities

The previous report[1] presented results from 15 ICS cybersecurity assessments performed by the CSSP from 2004 through 2008. Three additional ICS product assessments are included in this report. Figure 1 shows the categories of vulnerabilities that were identified in the three product assessments performed in 2009 and 2010. Table 1 summarizes these vulnerabilities.

The highest percentage of vulnerabilities identified during ICS product assessments continue to be due to improper input validation by ICS code. Poor access controls are the second most common security weakness identified in ICS software in 2009–2010. Authentication weaknesses follow in third place.

Vulnerabilities reported from the previous CSSP ICS product assessments include more patch management and password problems than the more recent findings. This may be more indicative of the types of systems that were assessed than a change in ICS vulnerability.

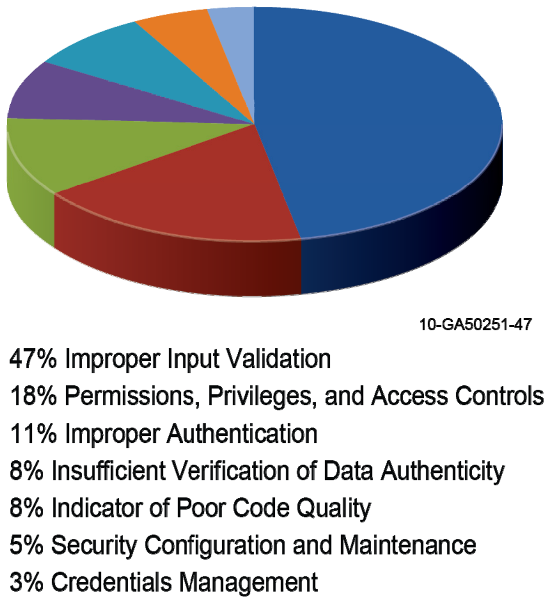
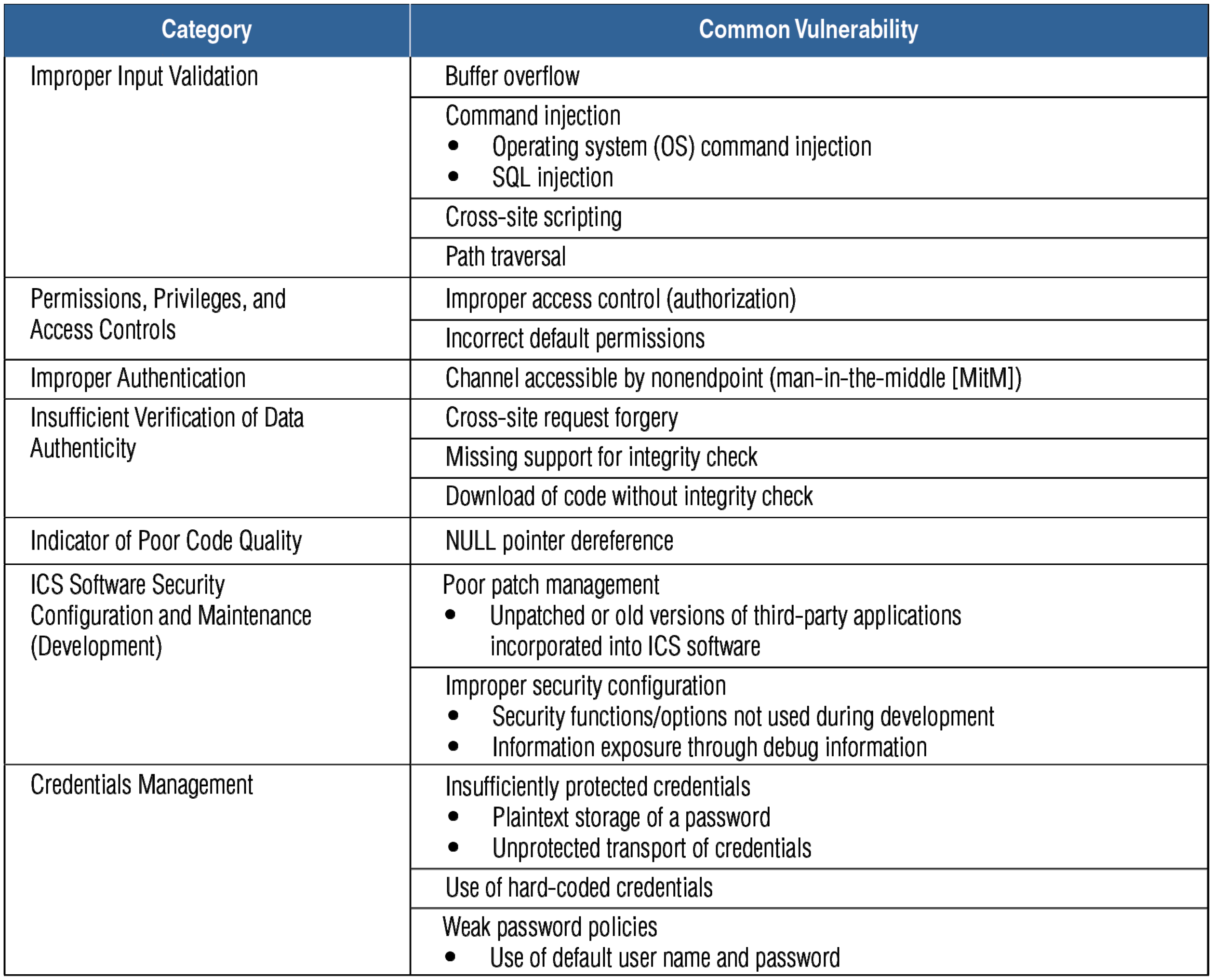


Figure 1. Categories of vulnerabilities identified in 2009–2010 CSSP product assessments.

Table 1. Common security weaknesses identified in 2009–2010 CSSP product assessments.



Category	Common Vulnerability
Improper Input Validation	Buffer overflow
	Command injection • Operating system (OS) command injection • SQL injection
	Cross-site scripting
	Path traversal
Permissions, Privileges, and Access Controls	Improper access control (authorization)
	Incorrect default permissions
Improper Authentication	Channel accessible by nonendpoint (man-in-the-middle [MitM])
Insufficient Verification of Data Authenticity	Cross-site request forgery
	Missing support for integrity check
	Download of code without integrity check
Indicator of Poor Code Quality	NULL pointer dereference
ICS Software Security Configuration and Maintenance (Development)	Poor patch management • Unpatched or old versions of third-party applications incorporated into ICS software
	Improper security configuration • Security functions/options not used during development • Information exposure through debug information
Credentials Management	Insufficiently protected credentials • Plaintext storage of a password • Unprotected transport of credentials
	Use of hard-coded credentials
	Weak password policies • Use of default user name and password

11-GA50210-03

2.2 ICS-CERT Products

ICS-CERT[b] provides a control system security focus in collaboration with US-CERT to:

- Respond to and analyze control systems-related incidents
- Conduct vulnerability and malware analysis
- Provide onsite support for incident response and forensic analysis
- Provide situational awareness in the form of actionable intelligence
- Coordinate the responsible disclosure of vulnerabilities/mitigations
- Share and coordinate vulnerability information and threat analysis through information products and alerts.

ICS-CERT serves as a key component of the Strategy for Securing Control Systems, which outlines a long-term, common vision where effective risk management of control systems security can be realized through successful coordination efforts.

This report uses information gathered from ICS-CERT alerts and advisories published between October 2009 and December 2010. In addition, general knowledge gained from incident

[b] http://www.us-cert.gov/control_systems/ics-cert/

response and forensic analysis is included in this report as well.

2.2.1 Common ICS-CERT Vulnerability Announcements

ICS-CERT alerts and advisories contain information about suspicious cyber activity, incidents, and vulnerabilities affecting critical infrastructure control systems. An ICS-CERT alert discloses information about an ICS-related vulnerability that was reported to them. An ICS-CERT Advisory is intended to provide awareness or solicit feedback from critical infrastructure owners and operators concerning ongoing cyber events or activity with the potential to impact critical infrastructure computing networks.

Figure 2 shows the categories of vulnerabilities that were reported to ICS-CERT in 2009 and 2010. The highest percentage of reported ICS vulnerabilities are buffer overflow vulnerabilities. Credentials management and authentication weaknesses make up the bulk of the remaining published ICS vulnerabilities. Table 2 summarizes the vulnerabilities that were reported to ICS-CERT in 2009 and 2010.

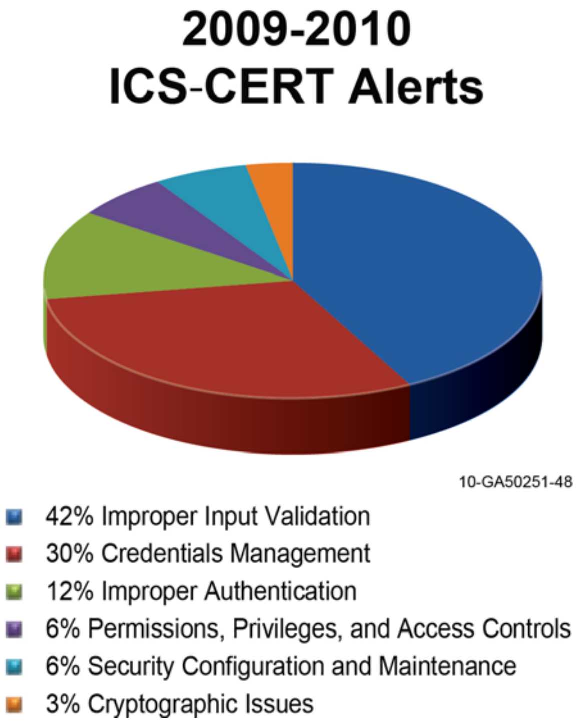
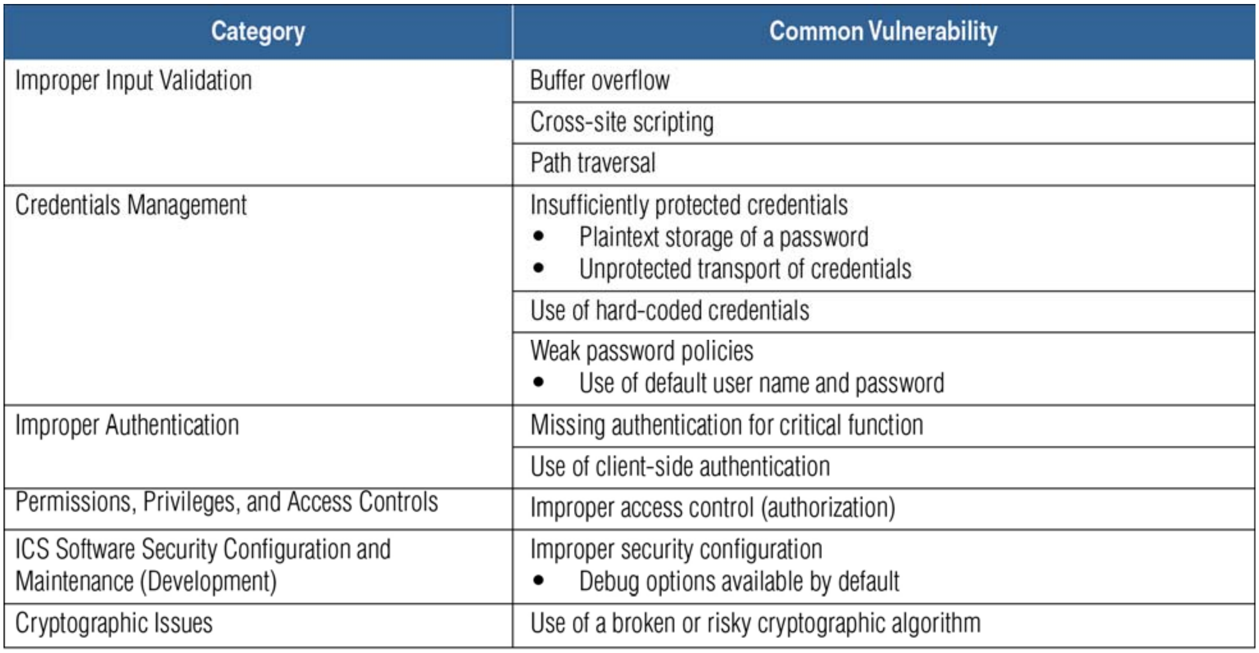


Figure 2. Percentage of 2009–2010 ICS-CERT vulnerability disclosures.

Table 2. Common security weaknesses reported to ICS-CERT in 2009 and 2010.



Category	Common Vulnerability
Improper Input Validation	Buffer overflow
	Cross-site scripting
	Path traversal
Credentials Management	Insufficiently protected credentials • Plaintext storage of a password • Unprotected transport of credentials
	Use of hard-coded credentials
	Weak password policies • Use of default user name and password
Improper Authentication	Missing authentication for critical function
	Use of client-side authentication
Permissions, Privileges, and Access Controls	Improper access control (authorization)
ICS Software Security Configuration and Maintenance (Development)	Improper security configuration • Debug options available by default
Cryptographic Issues	Use of a broken or risky cryptographic algorithm

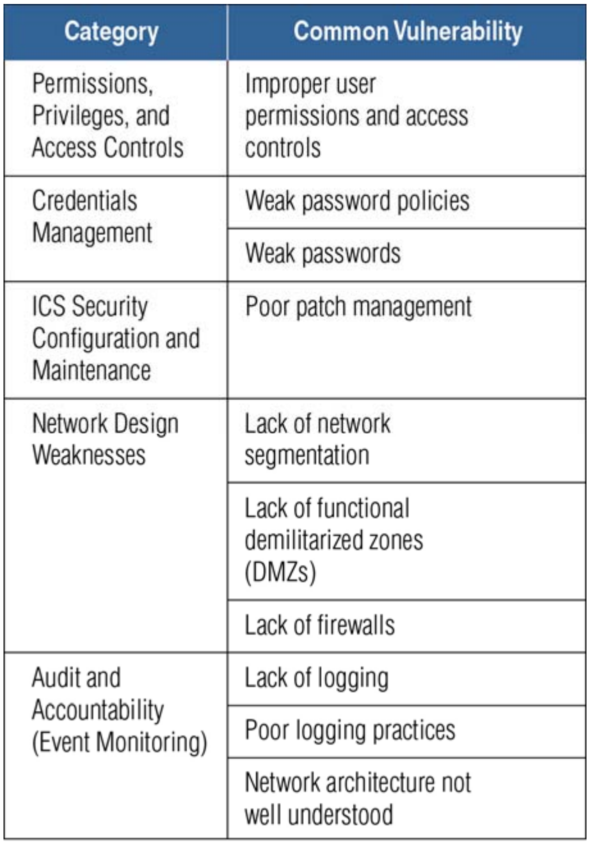
11-GA50210-04

2.2.1.1 Common Incident Response Observations

ICS-CERT incident response activities are performed at the request of owners and operators to assist in the review of network architecture, security practices, and system configurations. ICS-CERT incident response participants have observed an overall lack of defense-in-depth at ICS installations. Table 3 shows the biggest security weaknesses observed at ICS installations.

Some of the sites visited had not segmented the control network and had multiple connections from the control network to the corporate network and to remote sites as one flat network. Many peer and remote site connections were routed over leased networks. Many sites did not limit access between their disparate locations. This means that once any host on the company's network is compromised, there are few access controls preventing malicious intent.

Table 3. Major incident response observations.



Category	Common Vulnerability
Permissions, Privileges, and Access Controls	Improper user permissions and access controls
Credentials Management	Weak password policies
	Weak passwords
ICS Security Configuration and Maintenance	Poor patch management
Network Design Weaknesses	Lack of network segmentation
	Lack of functional demilitarized zones (DMZs)
	Lack of firewalls
Audit and Accountability (Event Monitoring)	Lack of logging
	Poor logging practices
	Network architecture not well understood

11-GA50210-05

Firewalls should be used to filter traffic between security zones. Some sites had implemented network segmentation using VLANs (virtual local area networks) without firewalls. Firewalls should be used to block unauthorized traffic in the case that the VLAN access controls are subverted.

User permissions and access controls should also be limited to those necessary to perform their roles. Some sites trusted all users equally or allowed more access than necessary.

After an incident has occurred, systems logs can be used to help determine the cause of the problem or how the system was attacked. Many sites either did not store system logs or overwrote them within a short period of time. Though not frontline cybersecurity barrier against a threat, event monitory and logging is critical to the capture of forensic data, which ultimately could lead to additional cybersecurity resilience.

2.3 CSET Self-Assessment Tool

The CSET[c] combines the functionality of two earlier tools, the CS2SAT, and the Cyber Security Vulnerability Assessment Tool. The Cyber Security Vulnerability Assessment Tool functionality is called Enterprise Evaluation or EE in CSET.

CSET is a self-assessment software standards application for performing cybersecurity reviews of industrial control and enterprise network systems. The tool may be used by any organization to assess the cybersecurity posture of ICS that manage a physical process or enterprise network. The tool also provides information that assists users in resolving identified weaknesses in their networks and improving their overall security posture.

CSET provides users in all infrastructure sectors with a systematic and repeatable approach for performing assessments against multiple standards, recommended security practices, and industry requirements. CSET provides a flexible question and answer format for performing

[c] http://www.us-cert.gov/control_systems/csetfaq.html

assessments. Users may apply the tool to site-specific configurations, based on user created diagrams and selection of specific standards for each assessment.

CSET is a desktop software tool that guides users through a step-by-step question and answer process to collect facility-specific control and enterprise network information. The questions address topics such as hardware, software, administrative policies, and user obligations. After the user responds to the questions, the tool compares the information provided to relevant security standards and regulations, assesses overall compliance, and provides appropriate recommendations for improving the system's cybersecurity posture. The tool pulls its recommendations from a database of the best available cybersecurity practices, which have been adapted specifically for application to control system and enterprise networks and components. Where appropriate, recommendations are linked to a set of prioritized actions that can be applied to remediate specific security vulnerabilities.

CSET requirements were derived from widely accepted standards such as:

- DHS Catalog of Control Systems Security: Recommendations for Standards Development Revisions 4 and 6
- NIST SP 800-53: National Institute of Standards and Technology (NIST), Special Publication (SP) 800-53, Recommended Security Controls for Federal Information Systems, Revisions 0, 1, 2, and 3 Final Public Draft, June 2009
- NIST SP 800-82: National Institute of Standards and Technology, SP 800-82, Guide to Industrial Control Systems (ICS) Security, Final Public Draft, September 2008
- ISO/IEC 15408 (The Common Criteria): International Organization of Standards/ International Electrotechnical Commission, Version 3.1, September 2007
- DoDI 8500.2: US Department of Defense (DoD) Instruction Number 8500.2, "Information Assurance (IA) Implementation," February 6, 2003
- NERC CIP-002 through CIP-009: North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) (http://www.nerc.com/), Effective June 1, 2006.

2.3.1.1 Common CSET Findings

The CSSP assisted in 50 CSET self-assessments in 2010 at owners and operations facilities within the 18 critical sectors, and in multiple CS2SAT self-assessments between 2006 and 2009. The CSSP provides the following benefits during the CSET evaluations:

- Cyber Security Awareness Briefing
- CSET training and demonstration
- "Over-the Shoulder" guidance to asset owners in using CSET
- Collective knowledge of common issues and good practices to identify vulnerabilities and mitigate risk
- Review assessment findings and provide mitigation techniques.

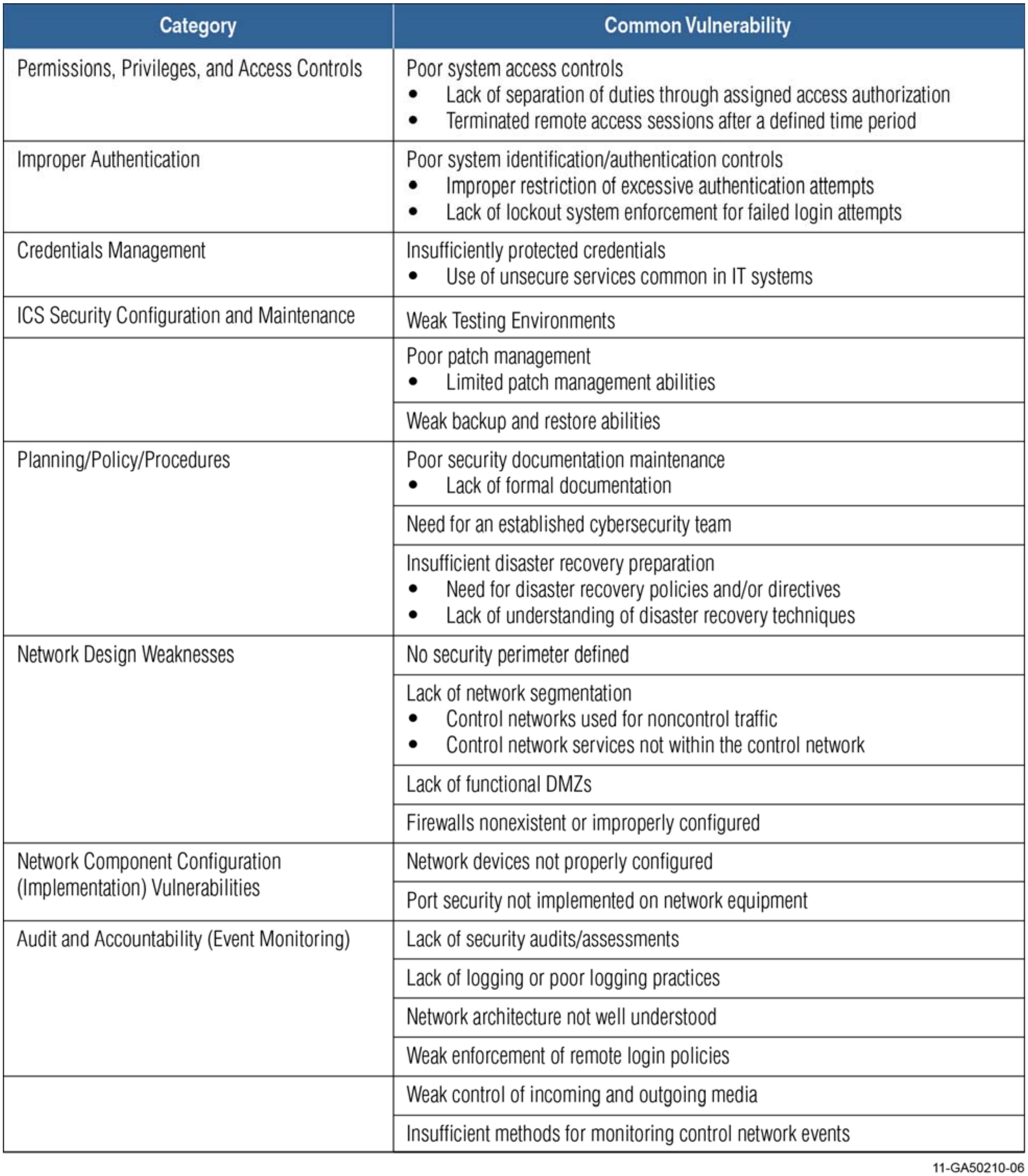
Table 4 summarizes the issues commonly identified as cybersecurity gap by ICS asset owners during onsite CSET assessments.

2.4 Compilation of ICS Vulnerability Information

DHS ICS risk reduction activities have gathered vulnerability information from many different types of ICS components, used by the multiple types of ICS. Information from different assessment approaches and ICS types provides a more complete picture of the security risks to ICS. Common types of vulnerabilities identified through CSSP assessments, ICS-CERT activities, and CSET self-assessments have been named and classified using consistent criteria, such as the Common Weakness Enumeration (CWE)[d] where possible, to enable correlation of vulnerability data. However, one should be careful about drawing conclusions from the data presented in this report.

[d] http://cwe.mitre.org/

Table 4. Common security weaknesses identified during onsite CSET assessments.



Category	Common Vulnerability
Permissions, Privileges, and Access Controls	Poor system access controls - Lack of separation of duties through assigned access authorization - Terminated remote access sessions after a defined time period
Improper Authentication	Poor system identification/authentication controls - Improper restriction of excessive authentication attempts - Lack of lockout system enforcement for failed login attempts
Credentials Management	Insufficiently protected credentials - Use of unsecure services common in IT systems
ICS Security Configuration and Maintenance	Weak Testing Environments
	Poor patch management - Limited patch management abilities
	Weak backup and restore abilities
Planning/Policy/Procedures	Poor security documentation maintenance - Lack of formal documentation
	Need for an established cybersecurity team
	Insufficient disaster recovery preparation - Need for disaster recovery policies and/or directives - Lack of understanding of disaster recovery techniques
Network Design Weaknesses	No security perimeter defined
	Lack of network segmentation - Control networks used for noncontrol traffic - Control network services not within the control network
	Lack of functional DMZs
	Firewalls nonexistent or improperly configured
Network Component Configuration (Implementation) Vulnerabilities	Network devices not properly configured
	Port security not implemented on network equipment
Audit and Accountability (Event Monitoring)	Lack of security audits/assessments
	Lack of logging or poor logging practices
	Network architecture not well understood
	Weak enforcement of remote login policies
	Weak control of incoming and outgoing media
	Insufficient methods for monitoring control network events

11-GA50210-06

All systems were not assessed for the same set of security weaknesses. The lack of vulnerabilities identified by a particular approach does not indicate that systems were found to not be vulnerable to that weakness.

Many of the security weaknesses indentified in installed ICS are not quantifiable because DHS does not keep detailed vulnerability information identified during CSET and incident response activities. This section compiles all common vulnerabilities identified by DHS activities and categorizes quantifiable vulnerabilities by categories and affected component types.

CSSP ICS product assessment reports and ICS-CERT alerts and advisories mainly contain vulnerabilities inherent in ICS software. ICS site assessments and incident response look at the security of the ICS environment.

At a high level, common vulnerabilities are categorized differently based on how the problem is being viewed. Figure 3 groups common ICS vulnerabilities according to eight general security categories that sum up the main weaknesses identified in ICS products by CSSP assessments and ICS-CERT vulnerability disclosures. Figure 3 compares the current cybersecurity issues based on assessment activities within the past eighteen months to the accumulative cybersecurity issues from 2004 to present.

Current vulnerabilities (2009-2010) identified in ICS product assessments continue to be improper input validation by ICS code. Through bad coding practices and improper input validation, access can be granted to an attacker allowing them to have unintended functionality or privilege escalation on the systems. Examples of improper input validation identified are within buffer overflows, boundary checking, and code injection. Other high-level security issues are poor access controls—credentials management and security configuration.

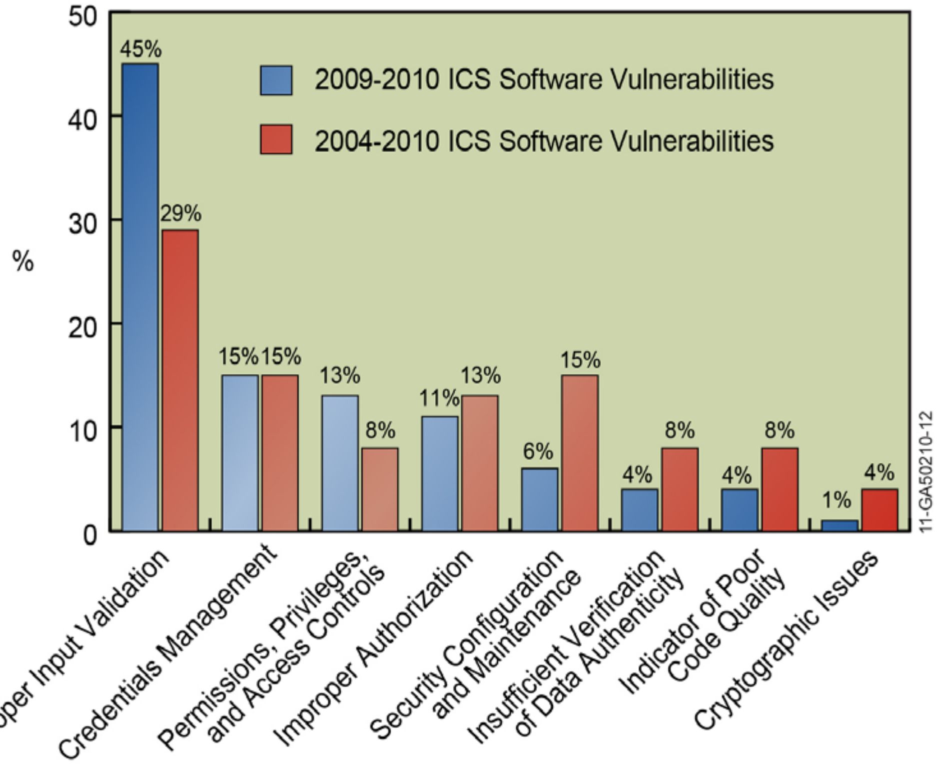


Figure 3. Percentage of 2009–2010 CSSP assessment findings and ICS-CERT vulnerability disclosures.

Based on assessment activities and the industry culture change towards more secured ICS, the vendor and asset owners community has increased in the patch management process and has reduced known vulnerabilities by patching ICS.

These categories summarize the main causes of vulnerabilities that put ICS software at risk to cyber attack.

ICSs are made up of process equipment, process control hardware, network devices, and computers. Vulnerabilities in network devices and protocols, or the operating systems, ICS software, and other software running on the ICS computers could allow an attacker to gather information about, disrupt, or manipulate ICS operations. The percentage of CSSP assessment vulnerabilities that were found in common ICS component types are shown in Figure 4.

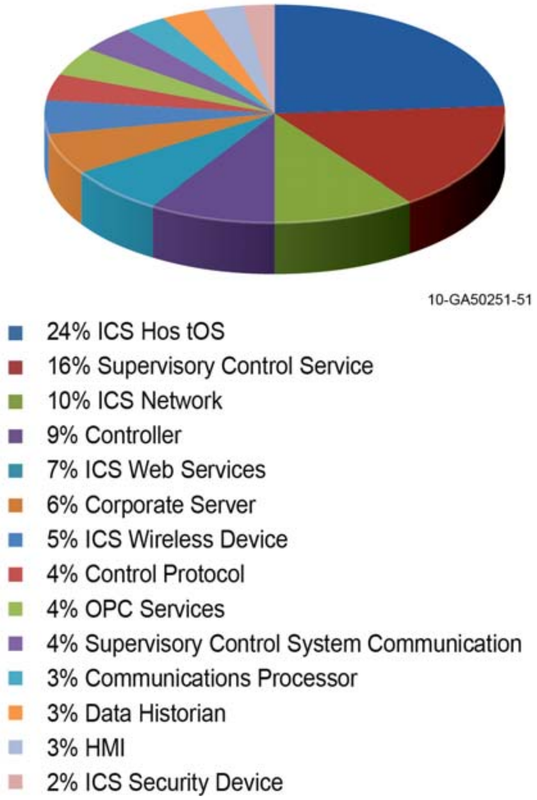


Figure 4. CSSP assessment findings and ICS-CERT vulnerability disclosures per ICS component type.

The International Standards Association (ISA) reference model creates a framework for referencing general Industrial Automation and Control Systems (IACS) network levels.[2] Although all CSSP assessment system networks were not designed consistently, this framework allows the findings to be consistently categorized by logical network layers. Each level represents a class of functionality. Table 5 lists the ISA SP99 reference model levels and associated IACS and SCADA functions.

The majority of functionality evaluated in CSSP assessments was at the supervisory control level. None of the assessments used for this report listed findings at the process level. Figure 5 illustrates the percentage of CSSP assessment findings and ICS-CERT vulnerability disclosures identified in each of the ISA reference model levels.

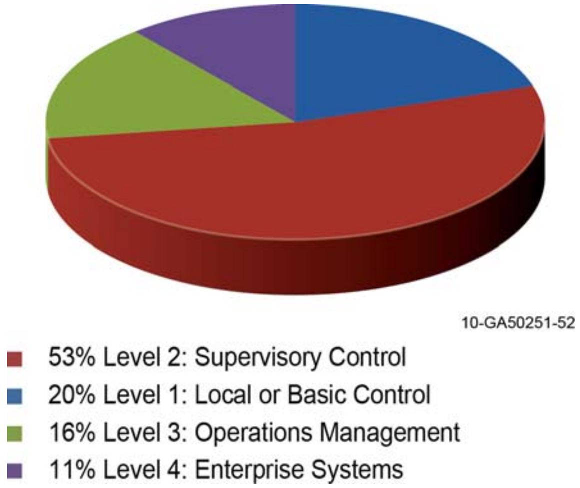
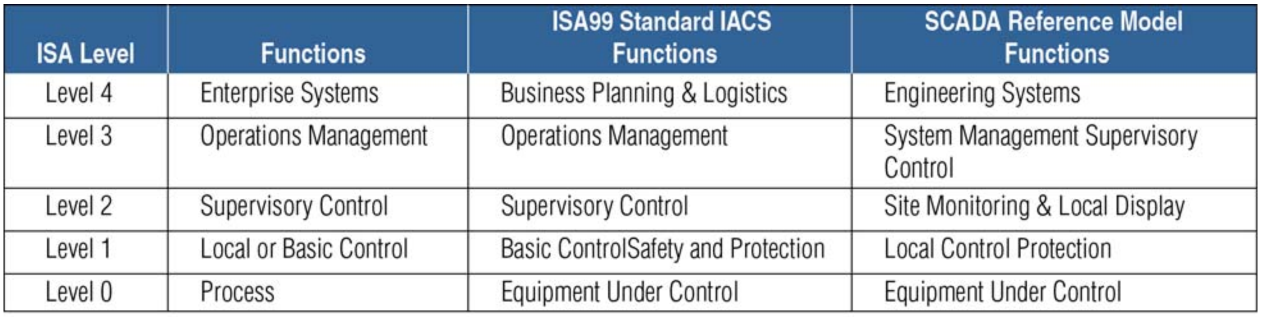


Figure 5. CSSP assessment findings and ICS-CERT vulnerability disclosures by ISA99 reference model levels.

Table 5. Reference model for ISA99 standards.



ISA Level	Functions	ISA99 Standard IACS Functions	SCADA Reference Model Functions
Level 4	Enterprise Systems	Business Planning & Logistics	Engineering Systems
Level 3	Operations Management	Operations Management	System Management Supervisory Control
Level 2	Supervisory Control	Supervisory Control	Site Monitoring & Local Display
Level 1	Local or Basic Control	Basic ControlSafety and Protection	Local Control Protection
Level 0	Process	Equipment Under Control	Equipment Under Control

11-GA50210-08

3. UNDERSTANDING COMMON ICS VULNERABILITIES

A major difference in securing ICS and a typical computer system is in the ICS components that do not use standard information technology (IT) hardware or software. Custom ICS hardware and software have not been scrutinized like common computer products, and refresh rates are typically much lower.

Another difference is the prioritization of security objectives. While adding security measures to ICS components, it is important to keep in mind functional requirements. Unlike typical IT systems, ICS security objectives are typically prioritized as:

1. Availability
2. Integrity
3. Confidentiality.

Violating operational requirements while implementing security features in ICS could cause more damage than a cyber attack.

CSSP ICS security assessments have identified the vulnerabilities described in this section in a majority of the systems. In addition to this subset of these common vulnerabilities, additional vulnerabilities unique to the individual ICS software and implementations were identified. All these vulnerabilities can be mitigated by following secure software design and development principles, and secure platform, software, and network configuration guidelines. References to additional information are included with the common vulnerability descriptions and recommendations. Common weakness areas identified by CSET assessments include a requirements section that contains the standards and guidelines used to identify these security gaps.

3.1 Common ICS Software/ Product Security Weaknesses

The ICS vendor software assessment findings are described in the following sections. Vulnerabilities reported by CSSP assessments and ICS-CERT are generalized to remove attribution details and are listed with each common vulnerability description as examples to aid in understanding the real issues. Multiple assessments and vulnerability announcements may have vulnerabilities that match the same example vulnerability description, and one assessment may have multiple specific vulnerability examples relating to one common vulnerability. Some common vulnerabilities have only one detailed example that describes all findings from the associated assessments. The number of systems that were found at risk to a given vulnerability is not listed in order to avoid any implication that all systems were tested for that vulnerability and to help lend anonymity to the ICS associated with common vulnerabilities and the related specific details listed.

Many ICS have recently incorporated web applications and services to allow remote supervisory control, monitoring, or corporate ICS data analysis. ICS assessments have found unauthorized directory traversal and authentication problems with ICS Web implementations. Many of the poor code quality and input validation findings in this section refer to proprietary web applications.

3.1.1 Improper Input Validation

3.1.1.1 Buffer Overflow

Input validation is used to ensure that the content provided to an application does not grant an attacker access to unintended functionality or privilege escalation.[e] Buffer overflow vulnerabilities are the result of programmer error.[f] This usually happens because the programmer only considered what should happen and what could happen by mistake, but not all the "out of the box" possibilities such as entering a 2,000-character-last name.

Buffer overflows result when a program tries to write more data into a buffer than the space allocated in memory. The "extra" data then overwrite adjacent memory and ultimately result

[e] http://cwe.mitre.org/data/definitions/20.html

[f] http://cwe.mitre.org/data/definitions/119.html

in abnormal operation of the program. A careful and successful memory overwrite can cause the program to begin execution of actual code submitted by the attacker. Most exploit code allows the attacker to create an interactive session and send commands with the privileges of the program with the buffer overflow. When network protocols have been implemented without validating the input values, these protocols can be vulnerable to buffer overflow attacks.

Services written by ICS vendors frequently suffer from coding practices that allow attackers to supply unexpected data and thus modify program execution. Some ICS protocol implementations are vulnerable to packets that are malformed or contain illegal or otherwise unexpected field values. Even though some ICS protocols are commonly used, the services that receive and interpret the protocol traffic are usually customized to the vendor product. Vulnerabilities in these services were a main target of many laboratory assessments because buffer overflows in the ICS services are possible entry points onto the ICS components.

Buffer overflows are the most common type of vulnerability identified in ICS products. The following are example buffer overflow vulnerabilities discovered in ICS products:

- Stack-based buffer overflows allowed remote code execution on ICS hosts
- Heap-based buffer overflows allowed remote code execution on ICS hosts
- A buffer overflow was found in a historian application
- Username and password buffer overflows in Web Human-Machine Interface (HMI) Web server
- Stack-based buffer overflow in ICS Web service
- Stack-based buffer overflow in ICS Web HMI
- Buffer overflow in ICS Web client
- Exploitable stack overflow in OLE for Process Control (OPC) server
- Heap-based buffer overflow in OPC server
- Stack-based buffer overflow in OPC client
- Stack-based buffer overflow caused by the use of the “strcpy” function
- Buffer overflow vulnerability identified in a PLC application
- Multiple buffer overflows identified in network packet parsing application
- Buffer overflows in application that accepts command line and process control arguments over the network
- Heap corruption on communications server
- Multiple stack-based buffer overflows in communications interface.

Recommendation: All code should be written to validate input data. All programmers should be trained in secure coding practices, and all code should be reviewed and tested for input functions that could be susceptible to buffer overflow attacks. All input should be validated, not just those proven to cause buffer overflows. Input should be validated for length, and buffer size should not be determined based on an input value. Length validation is especially important in the C and C++ programming languages, which contain string and memory function calls that can be used insecurely.

Even if values are never input directly by a user, data will not always be correctly formatted, and hardware or operating system protections are not always sufficient. Most buffer overflows identified in CSSP assessments were in the server applications that process ICS protocol traffic. In most cases, values input from network traffic were intercepted and altered in transit. Therefore, network data bounds and integrity checking should be implemented.

Perform a code review of all ICS applications responsible for handling network traffic. Network traffic cannot be trusted; therefore, better security and sanity checks need to be implemented so fuzzing attempts will not cause crashes or a denial of service (DoS).

3.1.1.2 Lack of Bounds Checking

The lack of input validation for values that are expected to be in a certain range, such as array index values, can cause unexpected behavior. For instance, unvalidated input, negative, or too large numbers can be input for array access and cause essential services to crash.

ICS applications frequently suffer from coding practices that allow attackers to supply unexpected data and thus modify program execution. Even though ICS applications pass valid data values during normal operation, a common vulnerability discovery approach is to alter or input unexpected values.

The following are specific assessment findings associated with this vulnerability:

- DoS caused by out-of-range index values:
  - Crashed ICS communications service by altering input value to negative number
  - Crashed proprietary fault tolerant network equipment protocol.

Recommendation: All code should be written to validate input data. Every programmer should be trained in secure coding practices. All code should be reviewed and tested for input functions that could be susceptible to buffer overflow attacks. All input should be validated, not just those proven to cause buffer overflows. Input values should be validated.

Even if values are never input directly by a user, data will not always be correctly formatted, and hardware or operating system protections can be insufficient. Further ICS traffic may be intercepted and altered in transit. Therefore, network data value and integrity checking should be implemented.

3.1.1.3 Command Injection

“Command injection allows for the execution of arbitrary commands and code by the attacker. If a malicious user injects a character (such as a semi-colon) that delimits the end of one command and the beginning of another, it may be possible to then insert an entirely new and unrelated command that was not intended to be executed. Command injection vulnerabilities typically occur when:

1. Data enter the application from an untrusted source.
2. The data are part of a string that is executed as a command by the application.
3. By executing the command, the application gives an attacker a privilege or capability that the attacker would not otherwise have.”[g]

Two types of command injection commonly found in ICS products are OS command injection and Structured Query Language (SQL) injection. ICS applications vulnerable to OS command injection execute OS commands that have been constructed from external input without proper sanitization. SQL injection vulnerabilities, which are more common and generally more exposed to attack, are discussed in the following section.

The following is an example of an ICS command injection vulnerability:

- Web interface on ICS wireless device allows an attacker to inject commands to manipulate data

Recommendation: If possible, use library calls rather than external processes to recreate the desired functionality. Otherwise, ensure that all external commands called from the program are statically created if possible.

Use an “accept known good” input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the backend database, possibly including execution of system commands.

[g] http://cwe.mitre.org/data/definitions/77.html

3.1.1.4 SQL Injection

"SQL command injection has become a common issue with database-driven websites. The flaw is easily detected and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. This flaw depends on the fact that SQL makes no real distinction between the control and data planes."[h]

If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated.

Recommendation: Process SQL queries using prepared statements, parameterized queries, or stored procedures. These features should accept parameters or variables and support strong typing. Do not dynamically construct and execute query strings within these features using "exec" or similar functionality, because it may re-introduce the possibility of SQL injection.

Guidance\references:

- *Attack Methodology Analysis: SQL Injection Attacks*, September 2005, US-CERT secured portal, http://www.us-cert.gov/control_systems/practices/documents/SQL%20Abstract.pdf.

3.1.1.5 Cross-Site Scripting

Cross-site scripting vulnerabilities allow attackers to inject code into the web pages generated by the vulnerable web application. Attack code is executed on the client with the privileges of the web server.

The root cause of a cross-site scripting (XSS) vulnerability is the same as that of an SQL injection, poorly sanitized data. However, a XSS attack is unique in the sense that the web application itself unwittingly sends the malicious code to the user.

An attacker is able to inject malicious script into a link and have a website return it to the victim as though it is legitimate. The victim's web browser will then run the malicious script, because it came from the server, potentially compromising the victim's computer by using one of many browser exploits. Many scenarios allow for this behavior, but they are caused by a lack of data sanitization. Most XSS attacks rely on user interaction and typically come in the form of a link sent by the attacker. Users are usually fooled into clicking on a link since the link probably points to a known and respected entity and has the trust of the user.

The most common attack performed with cross-site scripting involves the disclosure of information stored in user cookies. Because the site requesting to run the script has access to the cookies in question, the malicious script does also.

Some cross-site scripting vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems. Other damaging attacks include:

1. Disclosing end user files
2. Installing Trojan horse programs
3. Redirecting the user to some other page or site
4. Running "Active X" controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy
5. Modifying presentation of content.

Cross-site scripting presents one entry point for attackers to access and manipulate ICS networks. It takes advantage of web servers that return dynamically generated web pages or allow users to post viewable content to execute arbitrary Hypertext Markup Language (HTML) and active content, such as JavaScript, ActiveX, and VBScript, on a remote machine browsing the site within the context of a client-server session. This potentially allows the attacker to redirect the web page to a malicious location, hijack the client-server session, engage in network reconnaissance, and plant backdoor programs.

[h] http://cwe.mitre.org/data/definitions/89.html

The following are examples of ICS XXS vulnerabilities:

- XXS vulnerabilities in multiple web pages
- XXS vulnerabilities in multiple CGI scripts
- XXS vulnerabilities in online help.

Once the malicious script is injected, the attacker can perform a variety of malicious activities. The attacker could transfer private information, such as cookies that may include session information, from the victim's machine to the attacker. The attacker could send malicious requests to a website on behalf of the victim, which could be especially dangerous if the victim has supervisory control privileges through that web application.

Phishing attacks could be used to emulate ICS websites and trick the victim into entering a password, allowing the attacker to gain access to functionality and information to which the victim's account has been given rights.

A script could exploit a vulnerability in the web browser itself, possibly taking over the authorized ICS web client host.

In many cases, the attack can be launched without the victim even being aware of it. Even careful users are susceptible to XXS because attackers frequently use a variety of methods to encode the malicious portion of the attack, such as URL encoding or Unicode, so the request looks less suspicious.

Recommendation: ICS applications should use well-known and tested third-party web servers to serve their web applications. Web applications should be thoroughly tested for malformed input and other vulnerabilities that could lead to a compromise of the ICS web server.

The DHS Recommended Practice Case Study: Cross-Site Scripting[3] suggests the following seven defensive actions:

1. ICS Internet access policy
2. ICS user awareness and training
3. Coordination of security efforts between corporate IT network and ICS network
4. Firewall between the ICS network and the information technology network
5. Up-to-date patches
6. Web browser and e-mail security
7. Secure code.

Guidance\references:

- *Recommended Practice Case Study: Cross-Site Scripting*, February 2007, http://www.us-cert.gov/control_systems/practices/documents/xss_10-24-07_Final.pdf.

3.1.1.6 Improper Limitation of a Pathname to a Restricted Directory (Path Traversal)

Directory traversal vulnerabilities occur when file paths are not validated. Directory traversals are commonly associated with web applications, but all types of applications can have this class of vulnerability. Directory traversals occur when the software uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory. However, the software does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.[i]

The attacker may be able to read, overwrite, or create critical files such as programs, libraries, or important data. This may allow an attacker to:

- Execute unauthorized code or commands
- Read or modify files or directories
- Crash, exit, or restart critical files or programs, potentially causing a DoS.

For example, a directory traversal vulnerability is present in certain ICS devices that can lead to local file disclosure and possible execution of arbitrary commands by uploading malicious code.

[i] http://cwe.mitre.org/data/definitions/22.html

The following are specific ICS vulnerabilities:

- ICS service directory traversal vulnerability allows unrestricted write access
- ICS service directory traversal vulnerability can be exploited to delete any folder
- Directory traversal vulnerability in file upload web form
- Directory traversal vulnerability allows access to system configuration files.

Recommendation: Perform input validation. Use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Inputs should be decoded and converted to the application's current internal representation before being validated.

3.1.2 Poor Code Quality

Poor code quality refers to code issues that are not necessarily vulnerabilities, but indicate that it was not carefully developed or maintained.[j] These products are more likely to contain vulnerabilities than those that were developed using secure development concepts and other good programming practices. "If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code."[4]

ICS code review and reverse engineering exercises indicate that ICS software has not been designed or implemented using secure software development concepts in general. The relatively greater ages of core ICS applications increase the likelihood of development as stand-alone systems with only reliability and efficiency as requirements. However, new ICS applications tend to suffer from the same lack of secure coding principles.

3.1.2.1 Use of Potentially Dangerous Functions

Otherwise known as unsafe function calls, the application calls a potentially dangerous function that could introduce vulnerability if used incorrectly.[k] The problem with using unsafe functions is that the developer is responsible for validating input. The number of publicly announced buffer overflow and other malformed input vulnerabilities is evidence that implementing this validation is a high risk.

Unsafe C/C++ function calls are the most notorious potentially dangerous functions. All have safe counterparts, so there is no reason to use unsafe functions or not replace them in existing code. The strcpy() function in C is an example of a potentially dangerous function because of introducing a buffer overflow vulnerability. If the input to strcpy can in any way be influenced, a chance exists that an attacker can find a way to circumvent the developer's logic. In many cases, the logic is only based on what would normally happen, and a buffer overflow attack is successful because the developer decided that no one would ever create a username longer than 1,024 characters. The attacker simply needs to try a few usernames to figure out that submitting more than 1,024 characters causes problems. The developer can test to make sure nothing larger than the memory buffer he created is sent to strcpy(), but strncpy() eliminates this risk by requiring that the buffer size is specified. The following are specific assessment findings associated with unsafe C/C++ function calls:

- Several instances of unsafe function calls found in communications processing code
- Unsafe C/C++ function calls in ICS code
- Unsafe C/C++ functions in OPC dynamic-link libraries (DLLs)
- Use of potentially dangerous functions in proprietary ICS application.

[j] http://cwe.mitre.org/data/definitions/398.html

[k] http://cwe.mitre.org/data/definitions/676.html

Recommendation: ICS applications tend to suffer from poor code quality. Vendors and asset owners who write custom applications should train developers in secure coding practices. All custom software should undergo thorough code review via both manual and automated processes to identify security issues while the code is still in the development stage. ICS-specific protocols should be redesigned to include strong authentication and integrity checks. IT products deployed on the ICS network should also have passed a security review. Asset owners should explicitly address the security of these products during the procurement process.

3.1.2.2 NULL Pointer Dereference

A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit. NULL pointer dereference issues can occur through a number of flaws, including race conditions, and simple programming omissions.[l]

NULL pointer dereferences usually result in the failure of the process unless exception handling (on some platforms) is available and implemented. Even when exception handling is being used, it can still be very difficult to return the software to a safe state of operation. In very rare circumstances and environments, code execution is possible.

Recommendation: If all pointers that could have been modified are sanity-checked before use, nearly all NULL pointer dereferences can be prevented.

3.1.3 Permissions, Privileges, and Access Controls

Permissions, privileges, and other security features are used to perform access controls on computer systems. Missing or weak access controls can be exploited by attackers to gain unauthorized access to ICS functions.

3.1.3.1 Improper Access Control (Authorization)

If ICS software does not perform or incorrectly performs access control checks across all potential execution paths, users are able to access data or perform actions that they should not be allowed to perform.[m]

The following are specific assessment findings associated with improper access controls:

- Access is not restricted to the objects that require it.
- ICS protocol allowed ICS system hosts to read or overwrite files on other hosts, without any logging.
- Documentation and configuration information was being shared freely (read only).
- Common shares are available on multiple systems.
- Lack of role-based authentication for ICS component communication.
- A remote user can upload a file to any location on the targeted computer.
- Arbitrary file download is allowed on ICS hosts.
- Arbitrary file upload is allowed on ICS hosts.
- Remote client is allowed to launch any process.
- ICS service allows anonymous access.
- Undisclosed “back door” administrative accounts for future vendor access to perform maintenance, updates, or training.

Recommendation: ICS vendors should design their systems to support the least privileges concept, provide the ability to create multiple accounts for functions that require different privileges, and deliver default configurations that only allow the least privileges necessary for each account type. ICS owners can then ensure that each user account is granted the least privileges necessary to perform their functions.

[l] http://cwe.mitre.org/data/definitions/476.html

[m] http://cwe.mitre.org/data/definitions/285.html

3.1.3.2 Execution with Unnecessary Privileges

Services are restricted to the user rights granted through the user account associated with them. Exploitation of any service could allow an attacker a foothold on the ICS network with the exploited service's permissions. Privilege escalation can be accomplished by exploiting a vulnerable service running with more privileges than the attacker has currently obtained. If successfully exploited, services running as a privileged user would allow full access to the exploited host.

This vulnerability is very common. The following are some specific assessment findings associated with this vulnerability:

- Manager account overused
- Remote exploitation of ICS application services allowed root-level access on ICS hosts
- Database service running as administrator.

Recommendation: By default, some ICS installations start services as the root user and root group. Many services do not need to be started with this privilege level, and doing so exposes system resources to preventable risks. By restricting necessary privileges during ICS design and implementation, the window of exposure and criticality of impact is significantly reduced in the event that a flaw is found in that service. Essentially, running with minimum privileges is a recommended practice because it reduces the potential harm that a service can cause in the event of misbehavior due to a bug, accident, or malicious exploit. The most secure service available should be used for a given functionality and then kept patched and up-to-date to help prevent exploitation.

3.1.4 Improper Authentication

Many vulnerabilities identified in ICS products are due to the ICS software failing to sufficiently verify a claim to have a given identity.[n]

Network protocols specify how information is packaged and sent across a computer network. For every network protocol, an application must wait for and process the data off the network. All ICS products use at least one protocol created specifically for ICS component communication. In order to communicate using standard ICS protocols, each ICS vendor must implement his or her own application to process the network traffic.

The protocol specification includes whether and how authentication, integrity checks, and confidentiality will be implemented. Services that employ weak authentication methods can be exploited to gain unauthorized privilege. Poorly protected credentials can be found in documentation or code, sniffed "off the wire," cracked, or guessed.

3.1.4.1 Authentication Bypass Issues

The software does not properly perform authentication, allowing it to be bypassed through various methods.[o]

Web services developed for the ICS tend to be vulnerable to attacks that can exploit the ICS Web server to gain unauthorized access. System architectures often use network DMZs to protect critical systems and to limit exposure of network components. Vulnerabilities in ICS DMZ Web servers may provide the first step in the attack path by allowing access within the ICS exterior boundary. Vulnerabilities in lower level component's web servers can provide more steps in the attack path.

The following are specific assessment findings associated with authentication bypass issues:

- Unauthenticated access to Web HMI Web server
- Web HMI Web server username/password authentication bypass

[n] http://cwe.mitre.org/data/definitions/287.html

[o] http://cwe.mitre.org/data/definitions/592.html

- Web server does not properly authenticate access to several directories
- HMI local area network (LAN) communication protocol authentication by Internet Protocol (IP) address.

Recommendation: ICS applications should use well known and tested third-party web servers to serve their web applications. Web applications should be thoroughly tested with malformed input and for other vulnerabilities that could lead to a compromise of the ICS Web server.

3.1.4.2 Missing Authentication for Critical Function

The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.[p] Many critical ICS functions do not require authentication.

Exposing critical functionality essentially provides an attacker with the privilege level of that functionality. The consequences will depend on the associated functionality, but they can range from reading or modifying sensitive data, access to administrative or other privileged functionality, or execution of arbitrary code.

The following are specific examples of missing authentication for critical ICS functions:

- Web server on controller required no authentication.
- ICS configuration tool allows code upload without authentication.

Recommendation: ICS developers should divide software into anonymous, normal, privileged, and administrative areas. Identify which of these areas require a proven user identity and use a centralized authentication capability.

ICS developers should identify all potential communication channels, or other means of interaction with the software, to ensure that all channels are appropriately protected. Developers sometimes perform authentication at the primary channel, but open up a secondary channel that is assumed to be private. For example, a login mechanism may be listening on one network port, but after successful authentication, it may open up a second port where it waits for the connection, but avoids authentication because it assumes that only the authenticated party will connect to the port.

In general, if the software or protocol allows a single session or user state to persist across multiple connections or channels, authentication and appropriate credential management need to be used throughout.

3.1.4.3 Client-Side Enforcement of Server-Side Security

Applications that authenticate users locally trust the client that is connecting to a server to perform the authentication.[q] Because the information needed to authenticate is stored on the client side, a moderately skilled hacker may easily extract that information or modify the client to not require authentication.

Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

The following are specific assessment findings associated with this vulnerability:

- Client-side validation of HMI application username
- Client-side user and password validation for remote controller configuration
- Unauthorized programming of the controller (authentication bypass).

Recommendation: Implement robust authentication by the server or component that is granting access. For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side.

[p] http://cwe.mitre.org/data/definitions/306.html

[q] http://cwe.mitre.org/data/definitions/603.html

3.1.4.4 Channel Accessible by Nonendpoint (Man-In-The-Middle)

Commands from the HMI cause actions in the ICS. Alarms are sent to the HMI that notify operators of triggered events. The integrity and timely delivery of alarms and commands are critical in an ICS.

MitM is possible if the ICS does not adequately verify the identity of actors at both ends of a communication channel, or does not adequately ensure the integrity of the channel, in a way that allows the channel to be accessed or influenced by an actor that is not an endpoint.

In order to establish secure communication between two parties, it is important to adequately verify the identity of entities at each end of the communication channel. Inadequate or inconsistent verification may result in insufficient or incorrect identification of either communicating entity. This can have negative consequences such as misplaced trust in the entity at the other end of the channel. An attacker can leverage this by interposing between the communicating entities and masquerading as the original entity. In the absence of sufficient verification of identity, such an attacker can eavesdrop and potentially modify the communication between the original entities.[r]

Weak authentication in ICS protocols allows replay or spoof attacks to send unauthorized messages and a possibility of sending messages that update the HMI or remote terminal unit must be considered. The attacker may be able to cause invalid data to be displayed on a console or create invalid commands or alarm messages. Clear-text authentication credentials can be sniffed and used by an attacker to authenticate to the system.

ICS protocols or communication channels vulnerable to MitM attacks were identified on multiple assessments:

- MitM altering of ICS communication is possible between controller and field equipment.
- MitM altering of ICS communication is possible between ICS and controller equipment.
- MitM altering of ICS interprocess communication is possible between ICS components.
- Blind trust relationships are based on the IP address as specified in the /etc/hosts file.
- Lack of secure authentication for session initiation and message authentication means the attacker can initiate sessions or alter established sessions with little difficulty.
- HMI login transmits passwords in clear text, which allows remote attackers to sniff the operator password.
- Remote telnet-style applications with weak authentication run in plain text on the ICS network.
- Lack of packet integrity checking.

MitM is possible when the communication protocol does not ensure the identity of each communication partner or the integrity of the message. If an attacker can pose as a trusted communication partner and formulate the correct integrity check values for a new or altered message, the communication channel is at risk.

Manipulating the communications on a control network requires an in-depth understanding of the protocol to be manipulated. The cyber assessment team is generally able to gather enough information about a network protocol to perform a network layer attack against the system. Most effective network attacks use the address resolution protocol (ARP) MitM attack to achieve their objectives.

The ARP MitM attack is a popular method used by an attacker to gain access to the network flow of a target system. In this style of attack, the network ARP caches of machines on the LAN are targeted, confusing those with whom they think they are communicating. The ARP protocol is used to determine which hardware addresses coincide with the IP addresses on the network. The MitM attack is initiated by sending gratuitous ARP commands to confuse each host. These ARP

[r] http://cwe.mitre.org/data/definitions/300.html

commands tell the two hosts that the attacker computer is really the computer to which they want to send data. When a successful MitM attack is performed, the hosts on each side of the attack are unaware that their network data are taking a different route through the attacker's computer. The attacker's computer then needs to forward all packets to the intended host so the connection stays in sync and does not time out. Figure 6 illustrates a typical MitM attack.

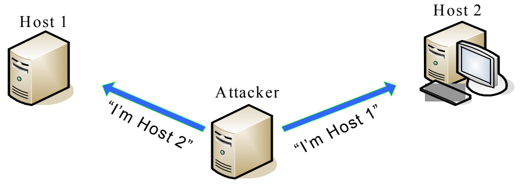


Figure 6. Generic man-in-the-middle attack.

The MitM attack is effective against any switched network because it effectively puts the attacker computer between the two hosts. This means the hosts send their data to the attacker's (compromised) computer, thinking it is the host to which they intended to send the data. The attacker generally needs to be able to compromise a host on (or between) the victim computers' LANs.

With a full ARP MitM attack in place, manipulation of ICS devices and/or modification of data flowing back to the operator's console to give false information of the state of the system (spoofing) can occur. This tampering could allow an attacker to manipulate the system or the operator's response.

Recommendation: ICS vendors should design their systems to fully authenticate both ends of any communications channel. The system design needs to implement strong authentication into ICS communication protocols and encrypt communications if appropriate and possible. Secure authentication and data integrity checks should be used to ensure that process commands and updates have not been altered in transit. These security procedures offer protection against spoofing attacks, in which false information is sent to the operator's console in order to give them an altered view from reality. Authentication also protects against unauthorized commands being sent to the ICS process devices.

Physical access to the controller while the controller is disconnected from a production Ethernet network should be required for firmware updates. Ensuring that updates occur in this environment will help prevent possible exploitation and will prevent the information disclosure of the device's firmware. Authentication and data integrity checks should be used to protect against unauthorized physical access and manipulation of firmware files.

Use hard-coded ARP tables for static IP addresses or dynamic ARP inspection of dynamic IP addresses, if feasible. Monitoring the network traffic for changing media access control (MAC) addresses using an intrusion detection system (IDS), such as ARPWatch, can help detect MitM attacks. Using port security on all network equipment is another good practice, which helps protect against unauthorized physical connections into the network.

The vulnerabilities that were exploited by the assessment team are inherent in the protocols. The only recommended mitigations for field device protocols are to change to a secure alternative protocol or to tunnel the traffic over an encrypted channel that would require "bump-in-the-wire" devices to handle the encryption, at least on the field end.

Reworking the protocol with sequence numbers that are more difficult to predict and incorporating authentication is another option, but this would be expensive and difficult to retrofit to the existing installed base.

3.1.5 Insufficient Verification of Data Authenticity

If ICS protocols and software do not sufficiently verify the origin or authenticity of data, it may accept invalid data. This is a serious risk for systems that rely on data integrity.

3.1.5.1 Cross-Site Request Forgery

When a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, then it might be possible for an attacker to trick a client into making an unintentional request to the web server that will be treated as an authentic request.

If the web interface offers a way to change ICS settings, hijacking credentials using cross-site request forgery (CSRF) could give the ability to perform any task that a legitimate user would be able to do through the web interface.

Recommendation: ICS Web developers should follow available guidelines for secure web development, such as the Open Web Application Security Project.[s]

ICS Web developers should use vetted libraries and frameworks that provide functions for implementing CSRF mitigations. Web developers can add a random token to each form and check that the token provided by the client is the same as that saved on the server for the client's session.[t] Tokens should be long enough to be resistant to brute-force guessing; a length of more than 15 characters is recommended. The goal of this mitigation is to require a user to supply a piece of information that is difficult for an attacker to obtain, thereby adding confidence that the user is legitimate. Limiting tokens' useful lifetime can also make guessing or brute forcing less effective.

Web developers can also identify dangerous operations and send a separate confirmation request to ensure that the user intended to perform that operation. The GET request should not be used for any request that triggers a state change.

ICS developers should also test their web applications for XXS issues that can be exploited to circumvent CSRF mitigations.

A mitigation that can be implemented by asset owners is a policy of not allowing users to connect to any other web servers from the same computer as they use to connect to the ICS Web server.

3.1.5.2 Missing Support for Integrity Check

Many ICS transmission protocols do not include a mechanism for verifying the integrity of the data during transmission.

If integrity check values or "checksums" are omitted from a protocol, there is no way of determining if data have been corrupted in transmission. The lack of checksum functionality in a protocol removes the first application-level check of data that can be used. The end-to-end philosophy of checks states that integrity checks should be performed at the lowest level that they can be completely implemented. Excluding further sanity checks and input validation performed by applications, the protocol's checksum is the most important level of checksum, because it can be performed more completely than at any previous level and takes into account entire messages, as opposed to single packets.[u]

The following are specific assessment findings associated with this vulnerability:

- ICS protocol does not check packet integrity.
- API security setting is configured during development not to check packet integrity.

Recommendation: Add an appropriately sized checksum to the protocol, ensuring that data received may be simply validated before it is parsed and used. Protocol implementers should ensure that the checksums present in the protocol design are properly implemented and added to each message before it is sent.

Simple checksums cannot be relied on to detect malicious alteration during transmission. The message checksum can be recalculated to match the altered message. Even if the checksum algorithm is unpublished, it can be reverse engineered by an attacker with access to the system or its traffic. Encryption or message hashing using a secret key is needed for a high level of assurance that the data have not been altered in transit.

3.1.5.3 Download of Code without Integrity Check

If an ICS component downloads source code or an executable from the network and executes the code without sufficiently verifying the origin and integrity of the code, an attacker may be able

[s] http://www.owasp.org

[t] http://shiflett.org/articles/cross-site-request-forgeries

[u] http://cwe.mitre.org/data/definitions/353.html

to execute malicious code by compromising the host server, spoofing an authorized server, or modifying the code in transit.

A common assessment finding is that firmware updates use weak integrity checks.

Recommendation: ICS vendors can add cryptographic signatures to their updates and modify ICS components to verify the signatures.

Physical access to the controller while the controller is disconnected from a production Ethernet network should be required for firmware updates. Authentication and data integrity checks should also be used to protect against unauthorized physical access and manipulation of firmware files.

3.1.6 Cryptographic Issues

3.1.6.1 Missing Encryption of Sensitive Data

Credentials sent across the network in clear text leave the system at risk to the unauthorized use of a legitimate user’s credentials. If attackers are able to capture usernames and passwords, they will be able to log onto the system with that user’s privileges. Any unencrypted information concerning the ICS source code, topology, or devices is a potential benefit for an attacker and should be limited.

One of the greatest security issues the assessment teams have identified is the widespread use of unencrypted plain-text network communications protocols. Many applications and services use protocols that include human-readable characters and strings. Network sniffing tools, many of which are freely downloadable, can be used to view this type of network traffic. As a result, the content of the ICS communication packets can be intercepted, read, and manipulated. Vulnerable data in this scenario include usernames, passwords, and ICS commands. Examples of these applications and services are proprietary ICS protocols and remote access services, such as telnet, File Transfer Protocol (FTP), and remote shell (rsh), which do not even encrypt the password or obfuscate it with a one-way hash function.

Recommendation: Encryption is a direct answer to information leaks due to clear-text communication. Unfortunately, encryption is not always feasible on ICS networks. Timing concerns may make encryption impractical, and in addition, encryption reduces the ability to monitor network traffic and to troubleshoot the system.

Guidance\references:

- *Control Systems Communications Encryption Primer*, December 2009, http://www.us-cert.gov/control_systems/pdf/Encryption%20Primer%20121109.pdf

3.1.6.2 Use of a Broken or Risky Cryptographic Algorithm

Some standard IT encryption protocols used in assessment systems were exploited due to encryption weaknesses. A published attack was used in multiple assessments to crack a terminal service encryption and view the user credentials during authentication.

The following are common specific assessment findings associated with this vulnerability:

- Remote display application encryption can be cracked.
- LAN Manager (LM) password hashes are found in ICS network traffic.
- Weak hashing algorithm is used in authentication.
- Weakness in its pseudorandom number generation routine.
- Vulnerable (unpatched) secure sockets layer (SSL) libraries deployed with ICS wireless device.

Recommendation: ICS developers and administrators should perform the necessary background research before choosing and properly incorporating an encryption solution. They should stay informed on published vulnerabilities and weaknesses of the deployed protocols and keep patches up-to-date.

The use of LM password hashes is a bad practice due to the easy decoding provided by tools such as John the Ripper and the Rainbow

Tables. Users must assume that any passwords used on the network that were stored as LM hashes are compromised. System administrators should prevent storage of the LM hash if it is not needed for backward compatibility. Windows 2000 and later systems create stronger NT LAN manager (NTLM) hashes, but create LM hashes for interoperability with older Windows systems.

3.1.7 Credentials Management

3.1.7.1 Insufficiently Protected Credentials

Credentials sent across the network in clear text leave the system at risk to the unauthorized use of a legitimate user's credentials. Network sniffing tools, many of which are freely downloadable, can be used to view this type of network traffic. If attackers are able to capture usernames and passwords, they will be able to log onto the system with that user's privileges.

Some ICS applications transport credentials unsecurely, for example:

- Clear-text password sent between the controller and configuration software
- Post-authentication sniffing or hijacking opportunities available on the dial-up connection.

Unsecure services developed for IT systems have been adopted for use in ICS for common IT functionality. Although more secure alternatives exist for most of these services, some ICSs have these services integrated into their applications. Examples of these services are as telnet, FTP, and rsh. The following are specific assessment findings associated with this vulnerability:

- Use of clear-text IT protocols on ICS LAN (e.g., telnet, FTP, "r" services) identified in multiple assessments
- Network file system, which has relatively limited security features, used by the ICS
- Telnet access available on controller.

Recommendation: ICS developers should use cryptography or other secure methods for protecting credentials from unauthorized interception and/or retrieval.

ICS vendors should remove the reliance on unsecure protocols in their products. Unsecure versions of common IT services should be replaced where possible by their secure versions. ICSs use common IT protocols for common IT functionality, such as network device management, remote logins, or file transfers. Because they are not used for real-time functionality, they can be replaced with their secure counterparts in most cases. Secure Shell (SSH) can replace all file transfer and remote login protocols such as FTP, telnet, and rlogin with encrypted versions. Any communication can be "tunneled" through SSH. Hypertext Transfer Protocol (HTTP) can be sent over the Secure Socket Layer (HTTPS). Users of these products should be aware that more secure network file sharing solutions are available. ICS vendors and customers should follow IT security practices and use the current secure versions of common protocols. When replacement is not feasible, access to the services should be minimized, and unencrypted communication should be limited to within the ICS whenever possible. Communications between security zones should be secured as much as possible.

3.1.7.2 Use of Hard-Coded Credentials

Hard-coded credentials have been found in ICS code and configuration scripts for authentication between ICS components.[v] The following are specific assessment findings associated with this vulnerability:

- Authentication is not required to read system configuration file, which contains user accounts details, including passwords.
- Well-known Simple Network Management Protocol (SNMP) community names are hard-coded for both read and write access.

Recommendation: ICS vendors should identify and replace all uses of hard-coded passwords with methods that support secure authentication.

ICS integrators and administrators may have the choice not to enter passwords into

[v] http://cwe.mitre.org/data/definitions/798.html

configuration scripts. If possible, they should choose to use secure protocols and disable the use of services that require hard-coded passwords.

3.1.8 ICS Software Security Configuration and Maintenance (Development)

3.1.8.1 Poor Patch Management during ICS Software Development

Vulnerabilities in ICS can occur because of flaws, misconfigurations, or poor maintenance of their platforms, including hardware, operating systems, and ICS applications. These vulnerabilities can be mitigated through various security controls, such as operating system and application patching, physical access control, and security software (e.g., antivirus software).

A computer system is vulnerable to attack from the time a vulnerability is discovered and publicly disclosed, to when a patch is generated, disseminated, and finally applied. The number of publicly announced vulnerabilities has been steadily increasing over the past decade to the point where patch management is a necessary part of maintaining a computer system. Although patching may be difficult in high-availability environments, unpatched systems are often trivial to exploit due to the ease of recognizing product version and the readiness of exploit code.

It is important for ICS vendors to maintain the operating systems, applications, and services used by their products. ICS developers should document all required applications and services and keep them patched and up to date. This will ensure that the ICS software supports the latest versions and patches.

Unpatched or Old Versions of Third-party Applications Incorporated into ICS Software

In multiple assessments, unpatched or old versions of applications were built into the ICS. Some had newer versions available just for security fixes. These applications possess vulnerabilities that may provide an attack path into the system. The software is well known, and available exploit code makes them an easy target.

The following are examples of unpatched or old versions of third-party applications incorporated into ICS software:

- Vulnerable database version.
- Vulnerable Web server version.
- OPC relies on Remote Procedure Call (RPC) and Distributed Component Object Model (DCOM)—without updated patches, OPC is vulnerable to the known RPC/DCOM vulnerabilities.
- Vulnerable (unpatched) SSL libraries.

Recommendation: The vendor bears responsibility to incorporate the latest versions of third-party (and operating system) software into the current version of the ICS product before delivery. The vendor should also support customers in patch testing and providing patches for their own software.

3.1.8.2 Improper Security Configuration

A common problem found during assessments was that even though secure authentication applications were used, installations and configurations were not correct. Many weaknesses identified in ICS software are because of available security options not being used or enabled.

The following are examples of improper security configurations during ICS development:

- Security functions/options not used during development
- Information exposure through debug information.

3.1.9 Summary of Common ICS Software Vulnerabilities

ICS software mostly suffers from the lack of secure software design and coding practices. ICS network protocols and associated server applications are prone to MitM data viewing and alteration as well as compromise through invalid input. This lack of security culture contributes to poor code quality, network protocol implementations that rely on weak authentication and allow information disclosure, and vulnerable custom ICS Web services.

ICS software generally uses third-party applications such as common web servers, remote access services, and encryption services. Many out-of-date and vulnerable third-party software applications and services have been identified on new ICS versions; this indicates that the ICS vendor is not supporting third-party patch management for their software.

Table 6 lists the ICS software categories and vulnerabilities identified in multiple CSSP assessments.

Table 6. Common ICS software vulnerabilities identified through CSSP and ICS-CERT activities.

Category	Common Vulnerability
Improper Input Validation	Buffer overflow
	Lack of bounds checking
	Command injection • OS command injection • SQL injection
	Cross-site scripting
	Path traversal
Indicator of Poor Code Quality	Use of potentially dangerous function
	NULL pointer dereference
Permissions, Privileges, and Access Controls	Improper access control (authorization)
	Execution with unnecessary privileges • Incorrect default permissions
Improper Authentication	Authentication bypass issues
	Missing authentication for critical function Use of client-side authentication Channel accessible by nonendpoint (MitM)
Insufficient Verification of Data Authenticity	Cross-site request forgery
	Missing support for integrity check
	Download of code without integrity check
Cryptographic Issues	Missing encryption of sensitive data • Clear-text transmission of sensitive information
	Use of a broken or risky cryptographic algorithm
Credentials Management	Insufficiently protected credentials • Plaintext storage of a password • Unprotected transport of credentials
	Use of hard-coded credentials
ICS Software Security Configuration and Maintenance	Poor patch management • Unpatched or Old Versions of Third-party Applications Incorporated into ICS Software
	Improper security configuration • Security functions/options not used during development • Information exposure through debug information

11-GA50210-09

3.2 Common ICS Configuration Weaknesses

Vulnerabilities in the previous section are inherent in the ICS products. Other vulnerabilities can be introduced by the way the ICS is installed and maintained. Each ICS installation is a unique combination of components and functionality offered by an ICS product vendor. ICS are generally such major purchases in time and money that very few systems from each ICS product line are delivered before features are added and a new version is released. Few installations are of the same ICS product version and features, which contribute to a lack of, or insufficient, standard procedures for securely configuring each ICS product.

All vendors have different standard processes for building, testing, and installing an ICS. Some vendors have integrators who work with customers to create and install the system. Other vendors have just a product model. Often, integration consultants with specific ICS product training are available for installation and configuration. All systems are unique; generally, with new features introduced in each one, the level of security in each ICS installation is dependent on those responsible for installing and configuring the operating systems, ICS applications, and third-party applications.

Common security problems that can arise from ICS configuration are unpatched operating system, application, and service vulnerabilities; failure to configure and implement applications and services securely (i.e., selecting security options and protecting credentials); changing all default passwords; setting password policies to require strong passwords; limiting user accounts, applications, and services to only the required permissions; installing or enabling security features correctly; and restricting unnecessary connections.

Assurance of a secure configuration can be increased through automated security configuration packages and detailed instructions provided by the ICS vendor. Automated disabling of unnecessary services, applications, and lists of required applications and services with associated permissions required should be included in instructions. Required ports and components allowed to connect should also be defined. Owners should require this information during the procurement process to ensure the ability to securely configure their systems.

Although some vulnerability is inherent in ICS products, many ICS component vulnerabilities are dependent on how an ICS product was implemented. Even though security configuration can be limited by the design of the ICS, ICS owners can control their risk of cyber attack by securely configuring their systems.

The ICS assessment findings that are due to installation and configuration errors are described below. These issues also apply to the maintenance of the operational ICS.

3.2.1 Permissions, Privileges, and Access Controls

3.2.1.1 Poor System Access Controls

Within access controls, the following common vulnerabilities have been identified during CSET assessments:

- Lack of separation of duties through assigned access authorization
- Lack of lockout system enforcement for failed login attempts
- Terminated remote access sessions after a defined time period.

Recommendation:

1. Account management includes the identification of account types (i.e., individual, group, and system), establishment of conditions for group membership, and assignment of associated authorizations. The organization identifies authorized users of the system and specifies access rights/privileges. The organization grants access to the system based on:
    a. Valid need-to-know/need-to-share that is determined by assigned official duties and satisfying all personnel security criteria
    b. Intended system usage.

2. The ICS organization requires proper identification for requests to establish system accounts and approves all such requests. The organization specifically authorizes and monitors the use of guest/anonymous accounts and removes, disables, or otherwise secures unnecessary accounts. Account managers are notified when system users are terminated or transferred and associated accounts are removed, disabled, or otherwise secured. Account managers are also notified when users' system usage or need-to-know/need-to-share changes.

3. Account management may include additional account types (e.g., role-based, device-based, attribute-based). The organization removes, disables, or otherwise secures default accounts (e.g., accounts used for maintenance) and changes default passwords. In situations where physical access to the ICS (e.g., workstations, hardware components, or field devices) predefines account privileges or where the ICS (e.g., certain remote terminal units, meters, or relays) cannot support account management, the organization employs appropriate compensating controls (e.g., providing increased physical security, personnel security, intrusion detection, and auditing measures) in accordance with the general tailoring guidance.

4. In situations where the ICS (e.g., field devices) cannot support the use of automated mechanisms for the management of information system accounts, the organization employs nonautomated mechanisms or procedures as compensating controls in accordance with the general tailoring guidance.

Requirements: Access control requirements used by the CSET self-assessment tool are summarized below:

1. The ICS organization manages system accounts, including establishing, activating, modifying, reviewing, disabling, and removing accounts.

2. The ICS organization reviews system accounts at least annually.

3. Account management may include additional account types (e.g., role-based, device-based, attribute-based).

4. The ICS organization removes, disables, or otherwise secures default accounts (e.g., accounts used for maintenance) and changes default passwords. In situations where physical access to the ICS (e.g., workstations, hardware components, or field devices) predefines account privileges or where the ICS (e.g., certain remote terminal units, meters, or relays) cannot support account management, the organization employs appropriate compensating controls (e.g., providing increased physical security, personnel security, intrusion detection, and auditing measures) in accordance with the general tailoring guidance.

5. The system enforces separation of duties through assigned access authorizations. ICS Supplemental Guidance: In situations where the ICS cannot support the differentiation of roles or a single individual performs all roles within the ICS, the organization employs appropriate compensating controls (e.g., providing increased personnel security and auditing measures) in accordance with the general tailoring guidance (NIST SP800-53A, AC-5).

6. The system enforces a limit of an organization-defined number of consecutive invalid access attempts by a user during an organization-defined time period. The system automatically locks the account/node for an organization-defined time period and delays next login prompt according to an organization-defined delay algorithm when the maximum number of unsuccessful attempts is exceeded.

   a. In situations where the ICS cannot support account/node locking or delayed login attempts, or the ICS cannot perform account/node locking or delayed logins due to significant adverse impact on performance, safety, or reliability, the organization employs appropriate compensating controls (e.g., logging or recording all unsuccessful login attempts and alerting ICS security personnel though alarms or other means when the number of organization-defined consecutive invalid access attempts is exceeded) in accordance with the general tailoring guidance (NIST SP800-53A, AC-7).

7. The system automatically terminates a session after an organization-defined time period of inactivity.

   a. In situations where the ICS cannot support the automatic termination of remote sessions after a specified period of inactivity, or the ICS cannot automatically terminate remote sessions due to significant adverse impact on performance, safety, or reliability, the organization employs nonautomated mechanisms or procedures as compensating controls (e.g., providing increased auditing measures for remote sessions or limiting remote access privileges to key personnel) in accordance with the general tailoring guidance (NIST SP800-53A, AC-12).

Guidance\references:

- NIST SP800-53A—Guide for Assessing the Security Controls in Federal Information Systems, AC-5, AC-7, AC12
- NIST SP800-82—Guide to Industrial Control Systems (ICS) Security
- ISA-TR99—Security Technologies for Manufacturing and Control Systems.

3.2.1.2 Open Network Shares on ICS Hosts

The storage of ICS artifacts, such as source code and system configuration on a shared file system, provides significant potential for information mining by an attacker. The design of many ICS requires open network shares on ICS hosts.

The following are examples of assessment findings associated with this vulnerability:

- Publically available network shares on ICS hosts
- Two shares discovered on work station and server computers
- Common shares on multiple systems
- Files available for read access
- Information leak through shared directories
- Large number of publically available network shares on ICS hosts
- The source code for the ICS is shared on ICS hosts. Source code could be downloaded and used to find vulnerabilities.

Recommendation: ICS integrators and administrators should be able to configure ICS hosts to only share files to the computers and accounts that require them. They should restrict the read and write permissions of these shared files and directories to the minimum required for each user. Permission to create network shares should be restricted to the users that need this functionality (generally administrators). ICS network administrators should use network segmentation and firewall rules that block access to file sharing ports (e.g., TCP Port 139 and 445 on Windows systems).

3.2.2 Improper Authentication

3.2.2.1 Poor System Identification/Authentication Controls

Some ICS organizations identified during CSET self-assessments that they have not developed policies or procedures to facilitate the implementation of identification and authentication controls, and do not uniquely identify and authenticate users and specific devices before establishing connections.

Recommendation: The ICS organization must develop, disseminate, and periodically review/update:

- A formal, documented, identification and authentication policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance
- Formal, documented procedures to facilitate the implementation of the identification and

authentication policy and associated identification and authentication controls.

Requirements: Authentication requirements used by the CSET self-assessment tool are summarized below:

1. The identification and authentication policy and procedures are consistent with:
    a. FIPS 201 and Special Publications 800-73, 800-76, and 800-78
    b. Other applicable laws, Executive Orders, directives, policies, regulations, standards, and guidance. The identification and authentication policy can be included as part of the general security policy for the organization. Identification and authentication procedures can be developed for the security program in general, and for a particular system, when required. NIST SP 800-12 provides guidance on security policies and procedures. NIST SP 800-63 provides guidance on remote electronic authentication.
2. The system uniquely identifies and authenticates users (or processes acting on behalf of users).
    a. Where users function as a single group (e.g., control room operators), user identification and authentication may be role-based, group-based, or device-based. For certain ICS, the capability for immediate operator interaction is critical. Local emergency actions for ICS are not hampered by identification or authentication requirements. Access to these systems may be restricted by appropriate physical security controls. In situations where the ICS cannot support user identification and authentication, or the organization determines it is not advisable to perform user identification and authentication due to significant adverse impact on performance, safety, or reliability, the organization employs appropriate compensating controls (e.g., providing increased physical security, personnel security, and auditing measures) in accordance with the general tailoring guidance. For example, manual voice authentication of remote personnel and local, manual actions may be required in order to establish a remote access [see AC-17]. NIST SP 800-82 provides guidance on ICS user identification and authentication.
    b. The system employs multifactor authentication for remote system access that is NIST SP 800-63 organization-defined Level 3, Level 3 using a hardware authentication device, or Level 4 compliant.
    c. Local and remote user access to ICS components is enabled only when necessary, approved, and authenticated. Remote access refers to access to an organizational information system by a user (or an information system) communicating through an external, nonorganization-controlled network. For ICS, the organization is the ICS owner/operator. Thus, remote access to the ICS is access from outside the system boundary defined by the ICS owner/operator. NIST SP 800-82 defines and provides guidance on ICS remote access.
3. The system identifies and authenticates specific devices before establishing a connection. ICS Supplemental Guidance: In situations where the ICS cannot support device identification and authentication (e.g., serial devices), the organization employs compensating controls in accordance with the general tailoring guidance.
4. The organization manages user identifiers by:
    a. Uniquely identifying each user
    b. Verifying the identity of each user
    c. Receiving authorization to issue a user identifier from an appropriate organization official
    d. Issuing the user identifier to the intended party

   e. Disabling the user identifier after an organization-defined time period of inactivity
   f. Archiving user identifiers.

5. The organization manages system authenticators by:
   a. Defining initial authenticator content
   b. Establishing administrative procedures for initial authenticator distribution, for lost/compromised, or damaged authenticators, and for revoking authenticators
   c. Changing default authenticators upon system installation
   d. Changing/refreshing authenticators periodically.

Guidance\references:

- FIPS 201—Personal Identity Verification (PIV) of Federal Employees and Contractors
- NIST SP800-73—Interfaces for Personal Identity Verification
- NIST SP800-76—Biometric Data Specification for Personal Identity Verification
- NIST SP800-78—Cryptographic Standards and Key Sizes for Personal Identity Verification
- NIST SP800-12—An introduction to Computer Security: The NIST Handbook
- NIST SP800-63—Electronic Authentication Guideline: Recommendations of the National Institute of Standards and Technology
- NIST SP800-82 – Guide to Industrial Control systems (ICS) Security.

3.2.3 Credentials Management

3.2.3.1 Insufficiently Protected Credentials

User credentials should be vigorously protected and made inaccessible to an attacker. Whenever credentials are passed in clear text, they are susceptible to being captured and then cracked if necessary by the attacker. If stored password hashes are not properly protected, they may be accessed by an attacker and cracked. In every case, the lack of protection of user credentials may lead to the attacker gaining increased privileges on the ICS and thus being able to more effectively advance the attack.

The following are specific assessment findings associated with this vulnerability:

- Services such as FTP, telnet, and rlogin transmit user credentials in clear text.
- OPC client responds with both newer NTLM and older LM password hashes, making discovery of passwords easier.
- Password hash files are not properly secured.
- LM password hashes are found.
- Database service configuration allowed administrator password to be displayed on web page.

Recommendation: Properly secure password files by making hashed passwords more difficult to acquire (e.g., restrict access by using a shadow password file or equivalent on UNIX systems). Replace or modify services so that all user credentials are passed through an encrypted channel.

LM password hashes are crackable by freely available tools within seconds. All Windows hosts support LM passwords and all versions before Windows Vista and Windows Server 2008 compute and store passwords using the LM hash algorithm by default. LM hashes should be disabled on all Windows hosts and domain controllers. OPC client security policies should be configured so that only the NTLM response is given. Because LM hashing does not support passwords longer than 14 characters, users can prevent a LM hash from being generated for their password by using a password at least 15 characters in length.

Unsecure versions of common IT services should be replaced where possible by their secure versions. ICS use common IT protocols for common IT functionality, such as network device management, remote logins, or file transfers. Because they are not used for real-time functionality, they can be replaced with their secure counterparts in most cases. SSH can replace all file transfer and remote login protocols such as FTP, telnet, and rlogin with encrypted versions. Any communication protocol can be "tunneled" through SSH. HTTP can be sent over HTTPS.

Users of these products should be aware that more secure remote access and file transfer solutions are available. ICS vendors and customers should follow IT security practices and use the current secure versions of common protocols. When replacement is not feasible, access to the services should be minimized, and unencrypted sensitive communication should be limited to within the ICS whenever possible. Communications between security zones should be secured as much as possible.

3.2.3.2 Weak Passwords

Some assessments discovered applications that had been configured without passwords, which means that anyone able to access these applications are guaranteed to be able to authenticate and interact with them.

The following are specific assessment findings where the ICS was designed not to use passwords or delivered with unconfigured third-party applications.

- Database service was configured without a password on multiple assessments.
- NULL connection allows remote hosts to query each system for information without requiring authentication.
- Password length can have zero characters. Any user on the system can have a blank password.

Poorly chosen passwords can easily be guessed by humans or computer algorithms to gain unauthorized access. The longer and more complex a password is, the longer the time it takes to guess or crack the password. Cracking a password can be trivial or virtually impossible depending on the combination of different character types used with larger password length.

Default passwords are generally widely known and can be obtained from system documentation or Internet searches.

The following are specific examples of weak passwords found on production ICS.

- Some ICS hosts had very weak 3-character administrative passwords.
- The weak passwords were recovered and provided root-level access to all system resources.
- Default SNMP community string was used by 89 hosts.
- Several weak passwords were found.
- Default password had not been changed.
- Default administrator level user names and passwords are in use.
- Default credentials are assigned for several predefined user accounts on the device including the administrative user account.
- ICS component is directly accessible from the Internet using the default username and password.
- The length, strength, and complexity of passwords do not follow the general recommendations specified in ISA-TR99.00.02-2004.

Password policies are needed to define when passwords must be used, how strong they must be, and how they must be maintained. Without a password policy, systems might not have appropriate password controls, making unauthorized access to systems more likely. Passwords that are short, simple (e.g., all lower-case letters), or otherwise do not meet typical strength requirements are vulnerable to being cracked. Password strength also depends on whether the specific ICS application was designed to support more stringent passwords.

The following are specific assessment findings associated with weak password policies:

- Many of the accounts, including the administrator account, had no password expiration date.

- Account lockout policy not defined.
- Password complexity disabled.
- Password history set to remember zero previous passwords.

Recommendation: Strong passwords need to be required and deployed on networking, client, and server equipment. Passwords should be implemented on ICS components to prevent unauthorized access.

The length, strength, and complexity of passwords should balance security and operational ease of access within the capabilities of the software and underlying operating system. A policy mandating the use of strong passwords for all cyber assets inside the electronic perimeter with a reasonable lifespan limit needs to be mandated and enforced. Usage of common administrative passwords should be discouraged.

Password policies should be developed as part of an overall ICS security program taking into account the capabilities of the ICS and its personnel to handle more complex passwords. System administrators should enforce the use of strong passwords. A password strength policy should contain the following attributes: (1) minimum and maximum length; (2) require mixed character sets (alpha, numeric, special, mixed case); (3) do not contain user name; (4) expiration; and (5) no password reuse. Authentication mechanisms should always require sufficiently complex passwords and require that they be periodically changed.[4]

Requirements: The following are general recommendations and considerations with regard to the use of passwords. Specific recommendations are presented in ISA-TR99.00.02-2004.

- The length, strength, and complexity of passwords should balance security and operational ease of access within the capabilities of the software and underlying operating system.
- Passwords should have appropriate length and complexity for the required security.
- Passwords should be used with care on operator interface devices such as control consoles on critical processes. Using passwords on these consoles could introduce potential safety issues if operators are locked out or delayed access during critical events. Physical security should supplement operator control consoles when password protection is not feasible.
- The keeper of master passwords should be a trusted employee, available during emergencies. Any copies of the master passwords must be stored in a very secure location with limited access.
- The passwords of privileged users (such as network technicians, electrical or electronics technicians and management, and network designers/operators) should be most secure and be changed frequently. Authority to change master passwords should be limited to trusted employees. A password audit record, especially for master passwords, should be maintained separately from the control system.
- In environments with a high risk of interception or intrusion (such as remote operator interfaces in a facility that lacks local physical security access controls), organizations should consider supplementing password authentication with other forms of authentication such as challenge/response or multifactor authentication using biometric or physical tokens.
- For user authentication purposes, password use is common and generally acceptable for users logging directly into a local device or computer. Passwords should not be sent across any network unless protected by some form of FIPS-approved encryption or salted cryptographic hash specifically designed to prevent replay attacks. It is assumed that the device used to enter a password is connected to the network in a secure manner.
- For network service authentication purposes, passwords should be avoided if possible. There are more secure alternatives available, such as challenge/response or public key authentication.

Guidance\references:

- ISA-TR99.00.02-2004.

3.2.4 ICS Security Configuration and Maintenance

3.2.4.1 Weak Testing Environments

CSET assessments commonly identified maintenance/testing environments as security gap areas. CSSP assessments and ICS-CERT incident response have noted poor patch management on ICS. Backup or test environments are necessary for testing patches before applying them on critical systems.

Patch management is paramount to maintaining the integrity of both IT and ICS. Unpatched software represents one of the greatest vulnerabilities to a system. Software updates on IT systems, including security patches, are typically applied in a timely fashion based on appropriate security policy and procedures. In addition, these procedures are often automated using server-based tools. Software updates on ICS cannot always be implemented on a timely basis because these updates need to be thoroughly tested by the vendor of the industrial control application and the end user of the application before being implemented. ICS outages often must be planned and scheduled days/weeks in advance. The ICS may also require revalidation as part of the update process. Another issue is that many ICS use older versions of operating systems that are no longer supported by the vendor. Consequently, available patches may not be applicable. Change management is also applicable to hardware and firmware. The change management process, when applied to ICS, requires careful assessment by ICS experts (e.g., control engineers) working in conjunction with security and IT personnel.

Vulnerabilities that have had patches available for a long time are still being seen on ICS. Unpatched operating systems open ICS to attack through known operating system service vulnerabilities. For example, in 2003 the Slammer worm disabled an Ohio Davis-Besse nuclear power plant safety monitoring system for nearly 5 hours. The Davis-Besse plant was in a maintenance cycle at this time and not generating power. According to reports, plant computer engineers had not installed the patch for the Microsoft SQL vulnerability that Slammer exploited. In fact, they did not know there was a patch, which Microsoft released 6 months before Slammer struck.[5]

The following are sanitized findings associated with this vulnerability from multiple assessments:

- Operating system vendor patches not applied
- System computers vulnerable to operating system service vulnerabilities
- Vulnerable version of Sendmail
- Sun rpc.cmsd has an integer overflow problem in xdr_array
- Vulnerable version of RPC
- Inconsistent application of current patches on HMIs.

Recommendation: A timely patch management process is critical to reduce vulnerabilities. Operating system patches repair vulnerabilities in the operating system that could allow an attacker to exploit the computer. The importance to system security of keeping operating system patches up-to-date cannot be over emphasized. However, patching ICS machines can present unique challenges. Among the factors to consider are system functionality, security benefit, and timeliness. This process requires elements of IT, IT security, process control engineering, and senior management and incorporates elements of an Incident Response Plan, a Disaster Recovery Plan, testbed testing, and a Configuration Management Plan. Where patching is not an option, work-arounds and defense-in-depth techniques and tactics can be used.[6]

Statically linked libraries need to be independently kept up-to-date if they are different from the libraries associated with the operating system. Database software and other applications also need to be kept patched and up to date.

Guidance\references:

- *Recommended Practice for Patch Management of Control Systems*, December

2008, http://www.us-cert.gov/control_systems/practices/documents/PatchManagementRecommendedPractice_Final.pdf

3.2.4.2 *Limited Patch Management Abilities*

Many ICS facilities, especially smaller facilities, have no test facilities, so security changes must be implemented using the live operational systems.

Recommendation: Because of the complexity of ICS software and possible modifications to the underlying operating system, changes must undergo comprehensive regression testing. The elapsed time for such testing and subsequent distribution of updated software provides a long window of vulnerability.

Patches are additional pieces of code that have been developed to address specific problems or flaws in existing software. Vulnerabilities are flaws that can be exploited, enabling unauthorized access to IT systems or enabling users to have access to greater privileges than authorized.

A systematic approach to managing and using software patches can help organizations to improve the overall security of their IT systems in a cost-effective way. Organizations that actively manage and use software patches can reduce the chances that the vulnerabilities in their IT systems can be exploited. In addition, they can save time and money that might be spent in responding to vulnerability-related incidents.

NIST SP 800-40 Version 2 provides guidance for organizational security managers who are responsible for designing and implementing security patch and vulnerability management programs and for testing the effectiveness of the programs in reducing vulnerabilities. The guidance is also useful to system administrators and operations personnel who are responsible for applying and testing patches and for deploying solutions to vulnerability problems.

Requirements: The following requirements apply to patch management:

1. Establish a testing environment for ICS.
2. Applying patches to operating system components creates another situation where significant care should be exercised in the ICS environment. Patches should be adequately tested (e.g., off-line on a comparable ICS) to determine the acceptability of side effects. Regression testing is advised. It is not uncommon for patches to have an adverse effect on other software. A patch may remove a vulnerability, but it can also introduce a greater risk from a production or safety perspective. Patching the vulnerability may also change the way the operating system or application works with control applications, causing the control application to lose some of its functionality. Another issue is that many ICS use older versions of operating systems that are no longer supported by the vendor. Consequently, available patches may not be applicable. Organizations should implement a systematic, accountable, and documented ICS patch management process for managing exposure to vulnerabilities.

   a. Once the decision is made to deploy a patch, other tools can automate this process from a centralized server and can confirm that the patch has been deployed correctly. Consider separating the automated process for ICS patch management from the automated process for non-ICS applications. Patching should be scheduled to occur during planned ICS outages.

Guidance\references:

- NIST SP 800-82—Guide to Industrial Control Systems (ICS)
- NIST SP 800-40—Creating a Patch and Vulnerability Management Program.

3.2.4.3 *Weak Backup and Restore Abilities*

Backups, restores, and testing environments have been identified as a common issue within the industry for continuity of operations in the event of an incident. Backups are usually made, but usually not stored offsite and rarely exercised and tested.

Recommendation: The frequency of system backups and the transfer rate of backup information to alternate storage sites (if so designated) are consistent with the organization's recovery time objectives and recovery point objectives. While integrity and availability are the primary concerns for system backup information, protecting backup information from unauthorized disclosure is also an important consideration depending on the type of information residing on the backup media.

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. The organization conducts backups of user-level and system-level information (including system state information) contained in the system on an organization-defined frequency and protects backup information at the storage location.
2. The organization tests backup information on an organization-defined frequency to verify media reliability and information integrity.
3. The organization protects system backup information from unauthorized modification (NIST SP800-53A, Sec CP-9).
4. The organization employs mechanisms with supporting procedures to allow the system to be recovered and reconstituted to the system's original known secure state after a disruption or failure.

Guidance\references:

- NIST SP800-52A – Guide for assessing the Security Controls in Federal Information Systems.

3.2.5 Planning/Policy/Procedures

3.2.5.1 Insufficient Security Documentation

A common security gap identified during CSET assessments was that the organization has not developed a formal business case for ICS security.

Recommendation: The first step in implementing a cybersecurity program for ICS is to develop a compelling business case for the unique needs of the organization. The business case should capture the business concerns of senior management while based on the experience of those who are already dealing with many of the same risks. The business case provides the business impact and financial justification for creating an integrated cybersecurity program.

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. The business case should cover the following topics:
   a. List threats that could possibly impact the ICS
   b. Identify consequences related to cybersecurity threats
   c. Prioritize cybersecurity controls
   d. Calculate annual business impact to support control systems security controls
   e. Identify internal and external resources and risk
   f. Phase funding for multi-year cybersecurity program
   g. Integrate cybersecurity policies and procedures with management and operational policies.
2. Senior management fully supports the implementation of the ICS security program and is at a high enough level to make strategic decisions.
3. A Cybersecurity officer has been defined and responsible to maintain and enforce ICS security policies.
4. All internal and external connections are documented and controlled.
5. A guiding charter for the cybersecurity team with roles, responsibilities, and accountabilities are fully defined for system owners and users.
6. A security plan that is formally documented that provides an overview of the security requirements for an ICS and describes the security controls in place or planned for meeting those requirements. The security controls that fall within the NIST SP 800-53

Planning family[w] provide the basis for developing a security plan. These controls also address maintenance issues for periodically updating a security plan. A set of rules describes user responsibilities and expected behavior regarding ICS usage with provision for signed acknowledgment from users indicating that they have read, understand, and agree to abide by the rules of behavior before authorizing access to the ICS.

7. Business continuity planning addresses the overall issue of maintaining or reestablishing production in the case of an interruption. These interruptions may take the form of a natural disaster (e.g., hurricane, tornado, earthquake, flood), an unintentional manmade event (e.g., accidental equipment damage, fire or explosion, operator error), an intentional manmade event (e.g., attack by bomb, firearm or vandalism, attacker or virus), or an equipment failure. From a potential outage perspective, this may involve typical time spans of days, weeks, or months to recover from a natural disaster, or minutes or hours to recover from a malware infection or a mechanical/electrical failure. Because there is often a separate discipline that deals with reliability and electrical/mechanical maintenance, some organizations choose to define business continuity in a way that excludes these sources of failure. Because business continuity also deals primarily with the long-term implications of production outages, some organizations also choose to place a minimum interruption limit on the risks to be considered. For the purposes of ICS cybersecurity, neither of these constraints is recommended. Long-term outages (disaster recovery) and short-term outages (operational recovery) should both be considered. Because some of these potential interruptions involve manmade events, it is important to work collaboratively with the physical security organization to understand the relative risks of these events and the physical security countermeasures that are in place to prevent them. The physical security organization must understand which areas of a production site house data acquisition and control systems that might have higher-level risks.

8. Review threat profiles for the various threat agents (e.g., phishers, botnet operators, criminal groups, terrorists, nation states) and their potential impact on the ICS installation.

9. Define special precautions when using tailored security solutions appropriate to the environment (e.g., DMZs, IDS/IPS, routers, firewalls, logging).

Guidance\references:

- NIST 800-82—Guide to Industrial Control Systems (ICS) Security, Section 4.1–Business Case, Section 6.2.3.1–Business Continuity Planning.

3.2.5.2 Poor Security Documentation Maintenance

A common security gap identified during CSET assessments was that the organization does not develop, implement, disseminate, and periodically review/update policy and procedures to facilitate implementation of security planning controls.

Recommendation: The security plan for an organization is intended to produce the policy and procedures that are required for the effective implementation of selected security controls and control enhancements in the security planning family. The policy and procedures are consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance. Existing organizational policies and procedures may make the need for additional specific policies and procedures unnecessary. The security planning policy addresses the overall policy requirements for confidentiality, integrity, and availability.

The security plan contains sufficient information to enable an implementation that is unambiguously compliant with the intent of the plan and a subsequent determination of risk to organizational operations and assets, individuals, other organizations, and the nation if the plan is

[w] http://csrc.nist.gov/publications/nistpubs/800-53A-rev1/sp800-53A-rev1-final.pdf

implemented as intended. The information in the security plan includes specification of parameters for assignment and selection statements in security controls either explicitly or by reference.

- Control Enhancement 1—The security CONOPS may be included in the security plan for the ICS.
- Control Enhancement 2—Unique security requirements for the ICS include, for example, encryption of key data elements at rest.

The organization considers different sets of rules based on user roles and responsibilities, for example, differentiating between the rules that apply to privileged users and rules that apply to general users. Electronic signatures are acceptable for use in acknowledging rules of behavior.

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. The organization develops, disseminates, and reviews/updates on an organization-defined frequency:
   a. A formal, documented security planning policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance
   b. Formal, documented procedures to facilitate the implementation of the security planning policy and associated security planning controls.
2. The organization:
   a. Develops a security plan for the ICS that:
      i. Is consistent with the organization's enterprise architecture
      ii. Explicitly defines the authorization boundary for the system
      iii. Describes the operational context of the ICS in terms of missions and business processes
      iv. Provides the security category and impact level of the ICS including supporting rationale
      v. Describes the operational environment for the ICS
      vi. Describes relationships with or connections to other ICS
      vii. Provides an overview of the security requirements for the system
      viii. Describes the security controls in place or planned for meeting those requirements including a rationale for the tailoring and supplementation decisions
      ix. Is reviewed and approved by the authorizing official or designated representative prior to plan implementation.
   b. Reviews the security plan for the ICS on an organization-defined frequency
   c. Updates the plan to address changes to the ICS/environment of operation or problems identified during plan implementation or security control assessments
   d. Establishes and makes readily available to all ICS users, the rules that describe their responsibilities and expected behavior with regard to information and ICS usage
   e. Receives signed acknowledgment from users indicating that they have read, understand, and agree to abide by the rules of behavior before authorizing access to information and the ICS.
3. The organization plans and coordinates security-related activities affecting the ICS before conducting such activities in order to reduce the impact on organizational operations (i.e., mission, functions, image, and reputation), organizational assets, and individuals.

Guidance\references:

- NIST 800-82—Guide to Industrial Control Systems (ICS) Security, Section 4.1
- NIST SP800-53 R2—Recommended Security Controls for Federal ICS – Version Rev.2
- NIST SP800-12—An Introduction to Computer Security: The NIST Handbook
- Security Plan Template—NIST 800-18—Guide for Developing Security Plans for Federal ICS.

3.2.6 Audit and Accountability

CSET assessments identified the lack of auditing and logging as a common weakness within industry across the board. Incident response participants identified the lack of logging or poor logging practices as a significant problem.

3.2.6.1 Lack of Security Audits/Assessments

Security audits are not regularly performed to determine the adequacy of security controls within their systems.

Recommendation: Periodic audits of the ICS should be performed to validate the following items:

- The security controls present during system validation testing (e.g., factory acceptance testing and site acceptance testing) are still installed and operating correctly in the production system.
- The production system is free from security compromises and provides information on the nature and extent of compromises as feasible, should they occur.
- The management of change program is being rigorously followed with an audit trail of reviews and approvals for all changes.

3.2.6.2 Lack of Logging or Poor Logging Practices

Event logging (applications, events, login activities, security attributes, etc.) is not turned on or monitored for identification of security issues. Where logs and other security sensors are installed, they may not be monitored on a real-time basis, and therefore, security incidents may not be rapidly detected and countered.

Recommendation: Diligent use of auditing and log management tools can provide valuable assistance in maintaining and proving the integrity of the ICS from installation through the system life cycle. The value of these tools in this environment can be calculated by the effort required to re-qualify or otherwise retest the ICS where the integrity due to attack, accident, or error is in question.

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. The system produces audit records that contain sufficient information to establish what events occurred, the sources of the events, and the outcomes of the events.
   a. The system provides the capability to include additional, more detailed information in the audit records for audit events identified by type, location, or subject.
   b. The system provides the capability to centrally manage the content of audit records generated by individual components throughout the system.
2. Audit record content should include:
   a. Date and time of the event
   b. The component of the system (e.g., software component, hardware component) where the event occurred
   c. Type of event
   d. User/subject identity
   e. The outcome (success or failure) of the event (NIST SP 800-92).
3. The system should provide reliable, synchronized time stamps in support of the audit tools
   a. Logging reviewed on a regular basis.
   b. System provides reliable, synchronized time stamps.
   c. Methods are implemented on the ICS to trace all console activities to a user, either manually or automatically.
   d. Policies and procedures are implemented for data to be logged, how logs are stored, how logs are protected, and how/when logs are reviewed.
   e. Logs are maintained by the ICS application and stored at various locations in either encrypted or unencrypted format (NIST SP 800-82, Sec 6.3.3).
4. Network loggings are configured to provide an accurate determination of the security incident.

5. The system protects audit information and audit tools from unauthorized access, modification, and deletion.

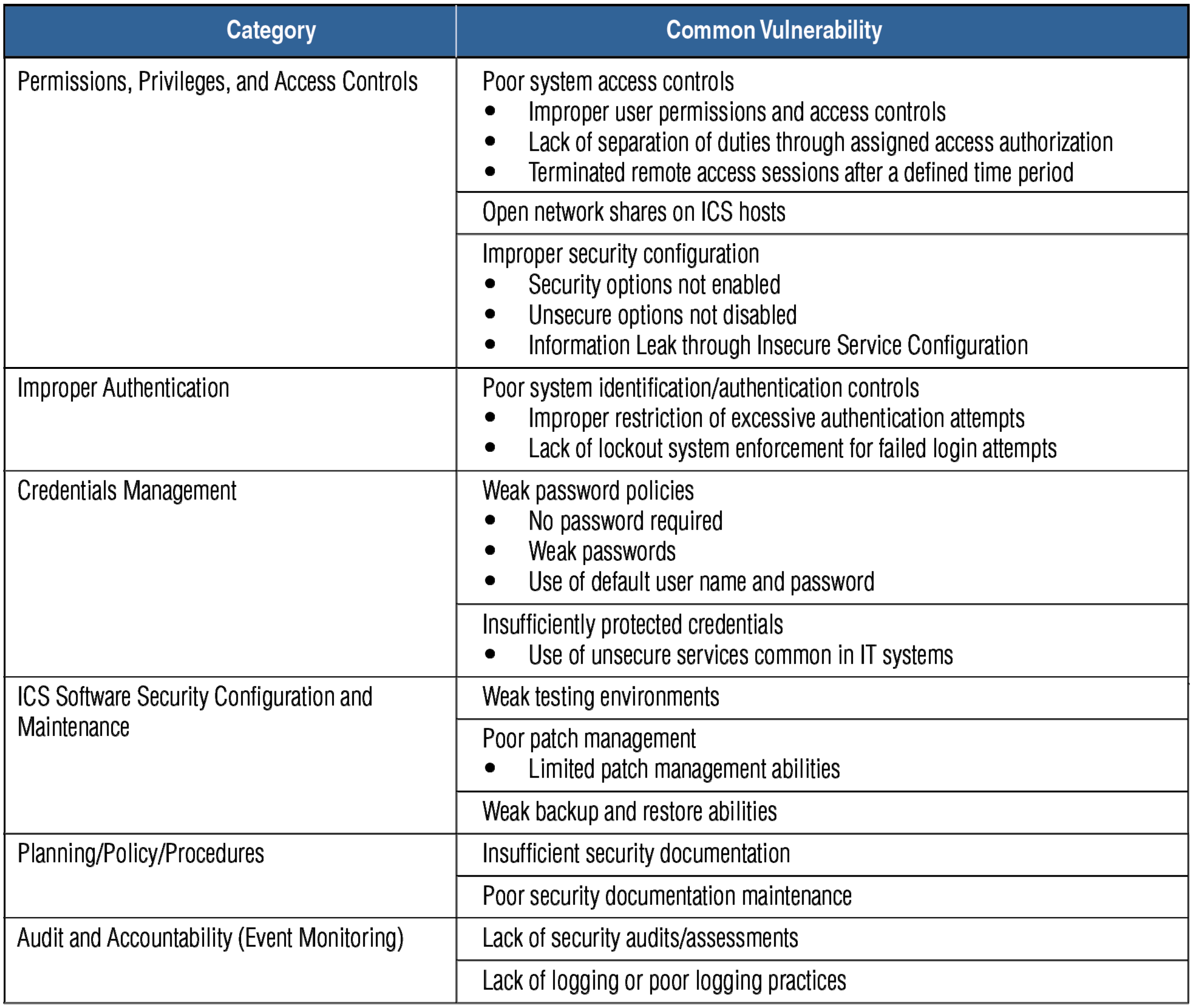
Guidance\references:

- NIST SP 800-82—Guide to Industrial Control Systems (ICS) Security
- NIST SP 800-92—Guide to Computer Security Log Management
- NIST SP 800-53A—Guide for Assessing the Security Controls in Federal Information Systems, AU-9.

3.2.7 Summary of Common ICS Configuration Vulnerabilities

Table 7 lists the common vulnerabilities related to ICS configuration issues that were identified with the assessment activities for the CSSP and by ICS asset owners during onsite CSET assessments.

Table 7. Summary of common ICS configuration findings.



Category	Common Vulnerability
Permissions, Privileges, and Access Controls	Poor system access controls • Improper user permissions and access controls • Lack of separation of duties through assigned access authorization • Terminated remote access sessions after a defined time period
	Open network shares on ICS hosts
	Improper security configuration • Security options not enabled • Unsecure options not disabled • Information Leak through Insecure Service Configuration
Improper Authentication	Poor system identification/authentication controls • Improper restriction of excessive authentication attempts • Lack of lockout system enforcement for failed login attempts
Credentials Management	Weak password policies • No password required • Weak passwords • Use of default user name and password
	Insufficiently protected credentials • Use of unsecure services common in IT systems
ICS Software Security Configuration and Maintenance	Weak testing environments
	Poor patch management • Limited patch management abilities
	Weak backup and restore abilities
Planning/Policy/Procedures	Insufficient security documentation
	Poor security documentation maintenance
Audit and Accountability (Event Monitoring)	Lack of security audits/assessments
	Lack of logging or poor logging practices

11-GA50210-10

3.3 Common ICS Network Security Weaknesses

The network architecture needs to be securely designed and implemented to allow remote control and monitoring of a process and provide process data for business functions while preventing any other traffic from entering or leaving the control network. Security zones with access control rules, which limit the traffic allowed in and out of the zone, will reduce the risk of intentional or unintentional attacks from sources outside the zones to attacks from allowed IP addresses that exploit the protocols allowed through the given security zone's perimeter. The security features built into the protocols used to transfer data in and out of the control network must be relied on to prevent attacks that pass access control requirements. Security features, such as authentication and integrity checks, can be wrapped around unsecure protocols that must be used for communication with the ICS. Understanding the limitations of the protection provided by a security product is essential for proper implementation.

An effective cybersecurity program for an ICS should apply a strategy known as "defense-in-depth," layering security mechanisms such that the impact of a failure in any one mechanism is minimized.

3.3.1 Common ICS Network Design Weaknesses

The network infrastructure environment within the ICS has often been developed and modified based on business and operational requirements, with little consideration for the potential security impacts of the changes. Over time, security gaps may have been inadvertently introduced within particular portions of the infrastructure. Without remediation, these gaps may represent backdoors into the ICS.

During incident response and onsite assessments at asset owner facilities, some ICS network architectures do not deploy any defense-in-depth strategies to protect their environments and use flat networks with no zones, limited to no port security, and weak enforcement of remote access policies. To compound the problem, the ICS networks are directly connected to corporate environments without firewalls and DMZ zones along with direct connections to the Internet.

Recommendation: Good cybersecurity practices for ICS networks include firewalls, the use of DMZs, and intrusion detection capabilities throughout the ICS architecture.

Guidance\references:

- NIST SP800-82—Guide to Industrial Control Systems (ISC) Security: Section 3.3.3. 5.2. 5.5.4, 5.
- *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies*, October 2009, http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf.

3.3.1.1 No Security Perimeter Defined

If the control network does not have a security perimeter clearly defined, then it is not possible to ensure that the necessary security controls are deployed and configured properly. This can lead to unauthorized access to systems and data as well as other problems.

Requirements: The ICS network security perimeter is logically separated from the corporate network on physically separated network devices with documented access points, defined security perimeter, and the necessary network security controls in place to prevent intrusions (NIST 800-82; Sec 5.2).

Guidance\references:

- NIST SP800-82—Guide to Industrial Control Systems (ISC) Security: Section 5.2
- *Backdoors and Holes in Network Perimeters: A Case Study for Improving Your Control System Security*, August 2005, http://www.us-cert.gov/control_systems/pdf/backdoor0503.pdf.

3.3.1.2 Lack of Network Segmentation

Minimal or no security zones allow vulnerabilities and exploitations to gain immediate full control of the systems, which could cause high-level consequences. The following are specific assessment findings associated with the lack of network segmentation:

- Lack of internal segmentation of the ICS production network: Inter-Control Center Communications Protocol (ICCP) servers not on DMZ
- Lack of internal segmentation of the ICS production network: Host with dedicated serial link for data transfer using high-risk application not on DMZ
- Control-related systems are accessible on the corporate LAN
- Incident response and onsite CSET assessments identified the following problems at multiple sites
- Control networks used for noncontrol traffic
- Control network services not within the control network.

Control and noncontrol traffic have different requirements, such as determinism and reliability, so having both types of traffic on a single network makes it more difficult to configure the network so that it meets the requirements of the control traffic. For example, noncontrol traffic could inadvertently consume resources that control traffic needs, causing disruptions in ICS functions.

Where IT services such as Domain Name System (DNS), and/or Dynamic Host Configuration Protocol (DHCP) are used by control networks, they are often implemented in the IT network, causing the ICS network to become dependent on the IT network that may not have the reliability and availability requirements needed by the ICS.

Recommendation: The goal of network segmentation is to create security zones that provide access control by separating systems with different security and access requirements. At a minimum, the ICS network should be separated from the corporate network by a firewall, and a DMZ should be implemented to provide the corporate network access to the required information from the ICS network. The systems located in the DMZ are not production systems and should be treated as hostile. Exceptions between the DMZ and the ICS networks should be kept to an absolute minimum, and exceptions from the corporate to the ICS should be eliminated. Additional security zones can be created within these segments.

Requirements: Use DMZ or VPN connections between the ICS and corporate networks for acceptable communications.

- An acceptable approach to enabling communication between an ICS network and a corporate network is to implement an intermediate DMZ network. The DMZ should be connected to the firewall such that specific (restricted) communication may occur between only the corporate network and the DMZ, and the ICS network and the DMZ. The corporate network and the ICS network should not communicate directly with each other. Additional security may be obtained by implementing a Virtual Private Network (VPN) between the ICS and external networks (NIST 800-41 Draft).
- Creating a DMZ requires that the firewall offer three or more interfaces, rather than the typical public and private interfaces. One of the interfaces is connected to the corporate network, the second to the control network, and the remaining interfaces to the shared or insecure devices such as the data historian server or wireless access points on the DMZ network. By placing corporate-accessible components in the DMZ, no direct communication paths are required from the corporate network to the control network; each path effectively ends in the DMZ. Most firewalls can allow for multiple DMZs, and can specify what type of traffic may be forwarded between zones. The firewall can block arbitrary packets from the corporate network from entering the control network and can regulate traffic from the other network zones including the control network. With well-planned rule sets, a clear separation can be maintained between the control network and other networks, with little or no traffic passing directly between the corporate and control networks. The primary security risk in

this type of architecture is that if a computer in the DMZ is compromised, it can be used to launch an attack against the control network via application traffic permitted from the DMZ to the control network.

3.3.1.3 Lack of Functional DMZs

The use of several DMZs provides the added capability to separate functionalities and access privileges and has proved to be very effective in protecting large architectures composed of networks with different operational mandates.

Recommendation: Firewalls should be used to create DMZs to protect the ICS network. Most firewalls can allow for multiple DMZs and can specify what type of traffic may be forwarded between zones. Different DMZs should be created for separate functionalities/access privileges, such as a peer connection like the ICCP server in SCADA systems, the data historian, the security servers, replicated servers, and development servers. Figure 7 shows this separation into multiple DMZs.

3.3.1.4 Firewalls Nonexistent or Improperly Configured

A lack of properly configured firewalls could permit unnecessary data to pass between networks such as control and corporate networks. This could cause several problems, including allowing attacks and malware to spread between networks, making sensitive data susceptible to monitoring/eavesdropping on the other network, and providing individuals with unauthorized access to systems.

Incident responses and onsite assessments at asset owner facilities have both identified multiple instances where connections to and from remote facilities and the ICS do not pass through a firewall.

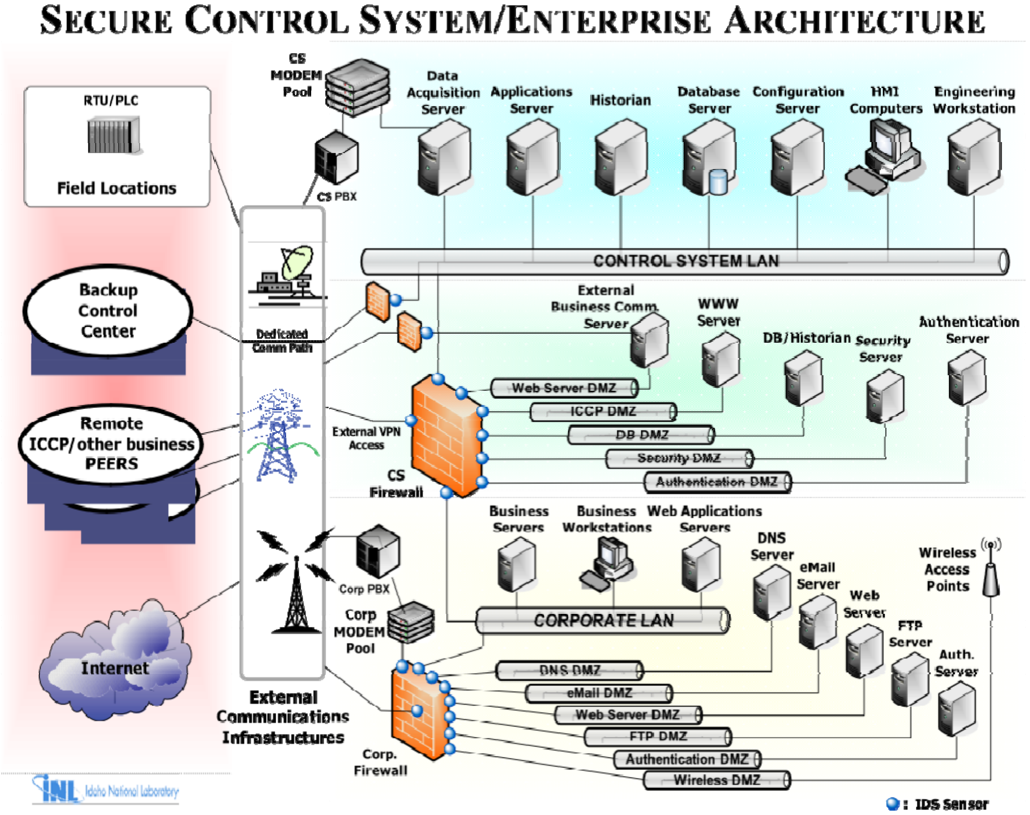


Figure 7. Recommended defense-in-depth ICS architecture.

Recommendation: The ICS network should be separated from the corporate network by a firewall, and a DMZ should be implemented to provide the corporate network access to the required information from the ICS network. The systems located in the DMZ are not production systems and should be treated as hostile. Exceptions between the DMZ and the ICS networks should be kept to an absolute minimum, and exceptions from the corporate to the ICS should be eliminated.

Guidance\references:

- NIST 800-41—Guidelines on Firewalls and Firewall Policy (Draft).

3.3.1.5 Firewall Bypassed

Backdoor network access is not recommended and could cause direct access to ICS for attackers to exploit and take full control of the system. All connections to the ICS LAN should be routed through the firewall. No hardwired connections should be circumventing the firewall.

The following are specific assessment findings associated with this vulnerability:

- Physical cables connected directly to the ICS LAN, bypassing firewall
- SSH server bridges corporate and ICS LANs, bypassing firewall
- Third network card on ICCP server connects directly to ICS LAN.

Recommendation: A firewall should limit access to the different LAN segments to only necessary communication. Each ICS host should be periodically checked for network connections that circumvent the firewalls.

Requirements: The ICS network needs to be continuously monitored for rogue or unknown connections.

3.3.2 Weak Firewall Rules

Firewall rules are the implementation of the network design. Enforcement of network access permissions and allowed message types and content is executed by firewall rules.

Firewall rules determine which network packets are allowed in and out of a network. Packets can be filtered based on IP address, port number, direction, and content. The protection provided by a firewall depends on the rules it is configured to use.

Firewall and router filtering deficiencies allow access to ICS components through external and internal networks. The lack of incoming access restrictions creates access paths into critical networks.

The lack of outgoing access restrictions allows access from internal components that may have been compromised. For an attacker to remotely control exploit code running on the user's computer, a return connection must be established from the victim network. If outbound filtering is implemented correctly, the attacker will not receive this return connection and cannot control the exploited machine.

Firewall rules should restrict traffic flow as much as possible. Connections should normally not be initiated from less-trusted networks.

3.3.2.1 Access to Specific Ports on Host Not Restricted to Required IP Addresses

Detailed findings under this common vulnerability involve firewall rules restricting access to specific ports, but not IP addresses. A common finding was that network device access control lists did not restrict management access to the required IP addresses.

Another common detailed finding was that firewall rules allowed access to unused IP addresses traceable to legacy configuration of the firewall allowed access to unused IP addresses. This finding illuminates an attack path by using this IP address in order to be allowed through the firewall.

The remaining specific assessment details associated with this vulnerability involved access to specific ports being given to either an entire address space or were not restricted by an IP address at all. Assessment findings that fall under this vulnerability are firewall rules that are based on address groups that include a wider range than should be allowed.

The following are specific assessment findings associated with this vulnerability:

- Personal firewalls need to be configured to restrict all unnecessary traffic.
- Router inside and outside interfaces had 24-bit netmask rather than 16-bit.
- Access lists are defined but not applied. No inbound filtering.
- Access lists are incorrect for required ports.
- Access to network printer services on corporate LAN was not restricted by password protection or access control list.
- E-mail client on DMZ had access to corporate LAN and Internet.
- Inadequate outgoing access restrictions.

Recommendations: Firewall rules that apply to functional groups should use defined finite groups that are restricted to required IP addresses. Firewall rules that are no longer needed should be removed as part of a change management procedure or periodic system review or audit.

3.3.2.2 Firewall Rules Are Not Tailored to ICS Traffic

ICS network administrators should restrict communications to only that necessary for system functionality. System traffic should be monitored, and rules should be developed that allow only necessary access. Any exceptions created in the firewall rule set should be as specific as possible, including host, protocol, and port information.

Recommendations: ICS vendors should provide documentation on how the ICS system components use the network so that effective firewall and IDS rules can be created. If ICS network requirements and protocol specifications are not available, owners can monitor network traffic to identify normal system behavior. The network traffic should be validated as required for ICS operations during this process. ICS vendors can document their system requirements using this method as well.

Firewall rules on production ICS should be implemented carefully, slowly working toward a rule set that excludes all traffic, with exceptions for including needed communication. Once the necessary outbound traffic has been determined, a safer configuration can then be created that blocks all traffic with exceptions for necessary communication.

Necessary communication can be determined by monitoring network traffic and implementing with IDS rules first, and then altering the rules, based on alerts from valid traffic, until confidence is gained that the rules will not impair system functionality. Firewall logs should be monitored for indications that legitimate system traffic is being blocked.

3.3.3 ICS Network Component Configuration (Implementation) Vulnerabilities

3.3.3.1 Network Devices Not Securely Configured

A common finding was that network device access control lists did not restrict management access to the required IP addresses. Network devices were also found that were configured to allow remote management over clear-text authentication protocols. Without these restrictions, an attacker can gain control by changing the network device configurations.

Recommendations: Access control lists should be used to limit management access of network equipment to only those who need it. Network devices should be configured to only allow access using secure protocols.

3.3.3.2 Port Security Not Implemented on Network Equipment

Unauthorized network access through physical access to network equipment includes the lack of physical access control to the equipment, including the lack of security configuration functions that limit functionality even if physical access is obtained. The common finding was a lack of port security on network equipment. A malicious user who has physical access to an unsecured port on a network switch could plug into the network behind the firewall to defeat its incoming filtering protection.

Recommendation: Port security should be implemented to limit connectivity to hardware interfaces. Given the static nature of ICS environments, port security may be used to ensure MAC addresses do not change and new devices are not introduced to the network. Actions, such as limiting known MAC addresses to specific interfaces and disabling unused interfaces, should be implemented to assist in network security.

3.3.4 Audit and Accountability

3.3.4.1 Network Architecture Not Well Understood

Incident response and onsite assessments at asset owner facilities review the ICS network diagrams with ICS network administrators. Many times, the current network diagram does not match the current state of the ICS network.

Recommendation: Network administrators should have an accurate network diagram of their ICS LAN and its connections to the other protected subnets, DMZs, corporate network, and external networks.

3.3.4.2 Weak Enforcement of Remote Login Policies

Any connection into the ICS LAN is considered part of the perimeter. Often these perimeters are not well documented, and some connections are neglected.

Recommendation: All entry points into the ICS LAN should be known and strictly managed by a security policy.

3.3.4.3 Weak Control of Incoming and Outgoing Media

Media protections for ICS lack written and approved policies and procedures, lack control of incoming and outgoing media, and lack verification scans of all allowed media into the ICS environment.

Recommendation: System media includes both digital media (e.g., diskettes, magnetic tapes, external/removable hard drives, flash/thumb drives, compact disks, digital video disks) and nondigital media (e.g., paper, microfilm). This control also applies to portable and mobile computing and communications devices with storage capability (e.g., notebook computers, personal digital assistants, cellular telephones, music devices). An organizational assessment of risk guides the selection of media and associated information contained on that media requiring restricted access. Organizations document in policy and procedures, the media requiring restricted access, individuals authorized to access the media, and the specific measures taken to restrict access. The rigor with which this control is applied is commensurate with the FIPS 199 security categorization of the information contained on the media.

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. The ICS organization needs a formal approved media protection policy and procedures that are consistent with applicable laws, Executive Orders, directives, policies, regulations, standards, and guidance. The media protection policy can be included as part of the general security policy for the organization. Media protection procedures can also be developed for the security program in general, and for a particular system, when required (NIST 800-12).
    a. The organization develops, disseminates, and periodically reviews/updates (NIST SP 800-53A, Sec MP-1):
        i. A formal, documented, media protection policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance
        ii. Formal, documented procedures to facilitate the implementation of the media protection policy and associated media protection controls.
2. The ICS organization restricts access to system media to authorized individuals (NIST SP 800-53A, Sec MP-2).
3. The ICS organization affixes external labels to removable system media and system output

indicating the distribution limitations, handling caveats and applicable security markings (NIST SP 800-53A, Sec MP-3).

4. The ICS organization physically controls and securely stores system media within controlled areas (NIST SP 800-53A, Sec MP-4).
5. The ICS organization protects and controls system media during transport outside of controlled areas and restricts the activities associated with transport of such media to authorized personnel (NIST SP 800-53A, Sec MP-5).
6. The ICS organization sanitizes system media, both digital and nondigital, prior to disposal or release for reuse (NIST SP800-53A, Sec MP-6).

Guidance\references:

- NIST SP 800-53—Recommended Security Controls for Federal Information Systems – Version Rev. 2
- NIST SP 800-53A—Guide for Assessing the Security Controls in Federal Information Systems; Sec MP-1, MP-2, MP-3, MP-4, MP-5, MP-6
- NIST SP 800-12—An introduction to Computer Security: The NIST Handbook (provides guidance on security policies and procedures)
- FIPS PUB 199—Standards for Security Categorization of federal Information and Information Systems.

3.3.4.4 Lack of or Poor Monitoring of IDSs

Recommendation: Good cybersecurity practices for ICS networks include firewalls, the use of DMZs and intrusion detection capabilities throughout the ICS architecture. Intrusion detection deployments apply different rule-sets and signatures unique to each domain being monitored (NIST SP 800-82: Sec 5.4).

Requirements: Requirements used by the CSET self-assessment tool are summarized below:

1. Network-based IDS/IPS capabilities need to be deployed between the ICS and corporate networks with a firewall; and host-based IDS/IPS capabilities should be applied to appropriate ICS devices.
2. An effective IDS deployment typically involves both host-based and network-based IDS. In the current ICS environment, network-based IDS are most often deployed between the control network and the corporate network in conjunction with a firewall. Host-based IDS are most often deployed on the computers that use general-purpose operating systems or applications such as HMIs, SCADA servers, and engineering workstations. Properly configured, an IDS can greatly enhance the security management team's ability to detect attacks entering or leaving the system, thereby improving security. They can also potentially improve a control network's efficiency by detecting nonessential traffic on the network.
3. The ICS network needs to be continuously monitored for rogue or unknown connections.
4. Secure the ICS network from adversaries monitoring ICS network traffic.
5. Adversaries that can monitor the ICS network activity can use a protocol analyzer or other utilities to decode the data transferred by protocols such as telnet, FTP, and Network File System. The use of such protocols also makes it easier for adversaries to perform attacks against the ICS and manipulate ICS network activity.

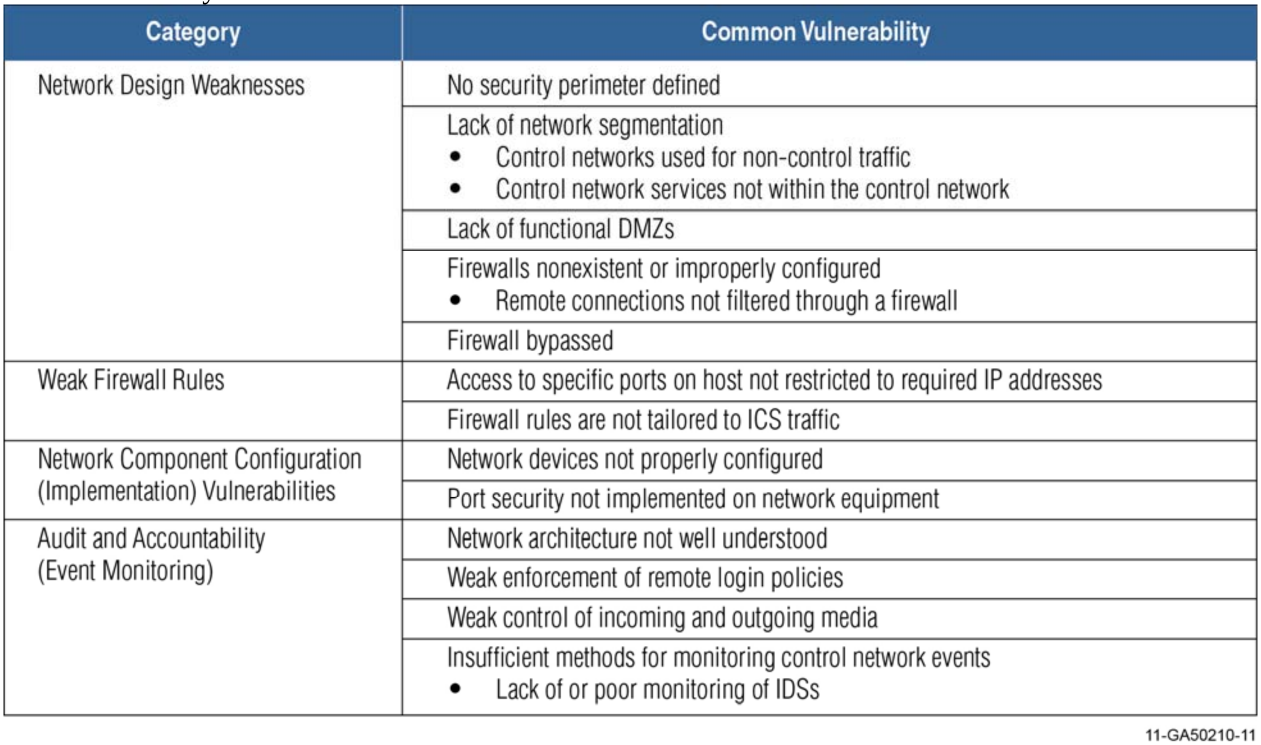
3.3.5 Summary of Common ICS Network Vulnerabilities

Table 8 lists the common vulnerabilities related to ICS network vulnerabilities that were identified with the assessment activities for the CSSP and ICS-CERT activities. Network security guidance and references are listed below.

Guidance\references:

- NIST SP 800-82—Guide to Industrial Control Systems (ISC) Security: Section 3.3.3., 5.2., 5.5.4, 5.4
- NIST SP 800-41—Guidelines on Firewalls and Firewall Policy (Draft)
- Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies.

Table 8. Summary of common ICS network weaknesses.



Category	Common Vulnerability
Network Design Weaknesses	No security perimeter defined
	Lack of network segmentation • Control networks used for non-control traffic • Control network services not within the control network
	Lack of functional DMZs
	Firewalls nonexistent or improperly configured • Remote connections not filtered through a firewall
	Firewall bypassed
Weak Firewall Rules	Access to specific ports on host not restricted to required IP addresses
	Firewall rules are not tailored to ICS traffic
Network Component Configuration (Implementation) Vulnerabilities	Network devices not properly configured
	Port security not implemented on network equipment
Audit and Accountability (Event Monitoring)	Network architecture not well understood
	Weak enforcement of remote login policies
	Weak control of incoming and outgoing media
	Insufficient methods for monitoring control network events • Lack of or poor monitoring of IDSs

11-GA50210-11

4. ICS SECURITY RECOMMENDATIONS

In addition to the specific mitigations and recommendations made for the vulnerabilities called out in the previous sections of this report, several general recommendations are given below.

ICS vendors and owners can learn and apply many common computer security concepts and practices to secure and protect their systems. Security should be designed and implemented by qualified security and ICS experts who are able to verify that the solutions are effective and can make sure that the solutions do not impair the system's reliability and timing requirements.

ICS vendors and asset owners are encouraged to use this report as a guide to help focus further efforts to improve the overall security of their systems. They should investigate whether the identified vulnerabilities affect their systems and if so, follow the recommendations in this report along with more detailed and tailored recommendations from other resources. The classes of vulnerabilities identified in this report can help identify problem areas for self-assessment activities that can be conducted to identify and mitigate vulnerabilities in ICS networks, components, services, and code.

By mitigating the vulnerabilities identified in this report, an ICS can be made more secure, but additional vulnerabilities most likely exist in all systems. The path to a more secure system is a continuous journey and as new attack scenarios are identified or developed, new defenses must be implemented.

ICS have different performance and reliability requirements and use operating systems and applications that may be considered unconventional to typical IT support personnel. Furthermore, the goals of safety and efficiency can sometimes conflict with security in the design and operation of ICS (e.g., requiring password authentication and authorization should not hamper or interfere with emergency actions for ICS.) All security solutions must not compromise critical functionality. All security functions integrated into the ICS must be tested (i.e., offline on a comparable ICS) to prove that they do not compromise normal ICS functionality.

In order to reduce the risk of a successful attack against an ICS, the likelihood of a high-impact incident can be reduced by implementing as many perimeter protection and vulnerability reduction strategies as possible (aka defense-in-depth). A mitigation strategy should not be chosen from a list of possible mitigations for a given identified or possible vulnerability, but rather as many mitigation techniques as reasonably possible should be employed to stand in a line of defense and prevent access to vulnerable components and network traffic. The probability that an attack is able to defeat or circumvent security defenses is increasingly reduced as the number of security measures are implemented and gaps are filled in the line of protection formed by the other security features on the ICS. However, the risk of the layers of defense to the operation of the ICS must be considered and mitigated as well.

The operational and risk differences between ICS and IT systems create the need for increased sophistication in applying cybersecurity and operational strategies. A cross-functional team of control engineers, ICS operators, and IT security professionals needs to work closely together to understand the possible implications of the installation, operation, and maintenance of security solutions in conjunction with ICS operation. IT professionals working with ICS need to understand the reliability impacts of information security technologies before deployment. Some of the operating systems and applications running on ICS may not operate correctly with commercial-off-the-shelf IT cybersecurity solutions because of specialized ICS environment architectures.

4.1 Recommendations for Vendors

Vendors need to incorporate security into every phase of the product development life cycle and rely on manual and automated means to ensure proper bounds checking. Once products are deployed, vendors need to establish a process to manage and mitigate product security defects. The vendor team should consist of representatives of key business functions such as product development, public relations, and legal. A single point of contact leads resolution on reported security issues and must assist asset owners in addressing reported security issues in a timely manner. A common industry practice is the hosting of a “/security” web page off the corporate main domain where information on security issues and the designated contact or team can easily be found. The vendor is responsible for responding to reported security concerns that include issue validation, patch development, patch testing and validation, and response coordination.

ICS security assessment reports show a common need to increase secure coding practices. The top ten ICS vendor recommendations are summarized below:

1. Educate/train developers in secure coding and create a culture that emphasizes security
2. Expeditiously test and provide security patches to affected customers
3. Create the necessary communication paths that are needed to quickly notify customers of security problems, and create the methods needed to provide patches in an effective way
4. Implement and strenuously test strong authentication and encryption mechanisms
5. Dramatically increase the robustness of network parsing code
6. Document how the systems use the network so that effective firewall and IDS rules can be created
7. Pay for a third-party security source code audit, and fix the problems identified during the audit
8. Redesign network protocols to avoid common problems and enhance security
9. Enhance test suites to perform more testing for failure with emphases on testing for potential vulnerabilities
10. Create custom protocol parsers for common IDS so that they can be more effective.

The following sections discuss actions that ICS vendors can take to significantly increase the security of their ICS products.

4.1.1 Create a Security Culture

Educate/train developers in secure coding and create a culture that emphasizes security.

The security development life cycle, created by Microsoft in 2002 as a response to heightened awareness of cybersecurity threats, is a high-visibility example of a security culture change. This process was developed to catch security flaws during the product development life cycle, not just after the product is released. For example, Microsoft has created a culture that promotes safe code development by forcing all new code to pass a set of tests before incorporation into the main product. All developers were put through secure development training to support this new culture. Performance evaluation of software products, as well as the product managers and their teams, also changed to include a focus on security. Although new Microsoft vulnerabilities are still abundant 6 years later, this culture change has made a significant difference in the security level of Microsoft products.[7]

ICS products have gained considerable attention in recent years as the cybersecurity threats due to connection to the Internet have been realized. Microsoft and other hardware, operating system, and software application vendors have experienced the cost and difficulties that arise from public announcement of security flaws to force quicker patch response time. Those companies willing to embrace a security culture change will benefit from fewer security patches for deployed systems and greater customer confidence and loyalty. Public announcements of ICS vulnerabilities are starting to appear and ICS protocol dissectors are becoming available.

ICS vendors must adapt to changing customer needs for security in the products used to control physical systems where compromise can have catastrophic consequences. As Microsoft has experienced, it is difficult to bolt security onto a mature product and impossible to find and prevent all bugs. Security must also compete with functionality for product time and budget. Vendors must accept that security improvements will require an investment. The sooner security is integrated into the product, the better chance it has of competing in a market where ICS products are required to survive cyber attack without compromising critical functionality.

ICS vendors should work toward a culture where software security best practices are adopted throughout the product development organizations and software development life cycles are adjusted to use the best practices. Security practices should be consolidated, integrated, and centralized into a security process that supports the defined strategy for creating the most secure product possible. Security testing and appropriate consequences are essential for creating secure products. ICS vendors can create a security cultural change within their companies by incorporating ICS product security into personnel performance.

Numerous resources are available for information and training on building a security culture and software security best practices. ICS vendors can use the following software security best practices to create more secure products:

1. Develop or acquire the necessary personnel security skills
2. Define security requirements to protect critical functions
3. Identify ICS component designs that violate security
4. Develop secure design or redesign of identified components
5. Require secure source coding handling to protect against malicious vulnerabilities
6. Perform thorough security testing
7. Provide security documentation.

Many ICS vulnerabilities are due to the lack of input validation. Programmers should be trained in secure coding practices to minimize vulnerabilities such as buffer overflows that are due to programmer error. All code should be reviewed and tested for input functions that could be susceptible to buffer overflow attacks. The C and C++ unsafe string and memory function calls should be replaced with their safe counterparts. Input validation should be used to ensure that the content provided to an application does not grant an attacker access to unintended functionality or privilege escalation. All input should be validated, not just those proven to cause buffer overflows. Input should be validated for length and buffer size should not be determined based on an input value. Even if values are never input directly by a user, data are not necessarily correctly formatted, and hardware or operating system protections are not always sufficient. Buffer overflows in applications that process network traffic can be exploited by intercepting and altering input values in transit. Therefore, network data bounds and integrity checking should be implemented as well.

As a layer of defense, compiler protection options should be used when compiling C/C++ code to increase the difficulty for an attacker to execute exploit code. This decreases the impact of a vulnerability from an exploit that allows the attacker to run commands on the computer or use it as a launching point along an attack path into the core of the ICS to a DoS-type attack.

4.1.2 Enhance ICS Test Suites

ICS product test suites should be enhanced to perform testing to failure with an emphasis on potential vulnerabilities. ICS software has historically been tested only within the context of normal operations.

The design and code logic of ICS products should prevent all invalid or unwanted cases, even if they should never occur. ICS experts can be blinded by their goal of creating a system that works reliably and protects against normal failures and mistakes. The connection of ICS to other networks has created the threat of cyber attack. ICS test suites should include “out of the box” scenarios that test all kinds of input values and abnormal conditions. This requires tests built by individuals who can create comprehensive and “out of the box” scenarios and are not involved in the design and implementation of the ICS product.

The CSSP assessment methodology is based on this idea of identifying security weaknesses through an attacker’s perspective and communicating the security issues to the industry partner from this perspective. This testing approach has been very successful in increasing awareness of the “out-of-the-box” attack methods the ICS sector needs to defend against.

Resources such as the Common Attack Pattern Enumeration and Classification project can help in developing test packages:

- Building software with an adequate level of security assurance for its mission becomes more and more challenging every day as the size, complexity, and tempo of software creation increases and the number and the skill level of attackers continues to grow. These factors each exacerbate the issue that, to build secure software, builders must ensure that they have protected every relevant potential vulnerability. Yet, to attack software, attackers often have to find and exploit only a single exposed vulnerability. To identify and mitigate relevant vulnerabilities in software, the development community needs more than just good software engineering and analytical practices, a solid grasp of software security features, and a powerful set of tools. All these things are necessary but not sufficient. To be effective, the community needs to think outside of the box and to have a firm grasp of the attacker's perspective and the approaches used to exploit software.
- Attack patterns are a powerful mechanism to capture and communicate the attacker's perspective. They are descriptions of common methods for exploiting software. They derive from the concept of design patterns applied in a destructive rather than constructive context and are generated from in-depth analysis of specific real-world exploit examples.
- To assist in enhancing security throughout the software development life cycle, and to support the needs of developers, testers and educators, the Common Attack Pattern Enumeration and Classification is sponsored by the Department of Homeland Security as part of the Software Assurance strategic initiative of the National Cyber Security Division. The objective of this effort is to provide a publicly available catalog of attack patterns along with a comprehensive schema and classification taxonomy.[8]

4.1.3 Create and Test Patches

Expeditiously test and provide security patches to affected customers. Create the necessary communication paths that are needed to quickly notify customers of security problems and create the methods needed to provide patches in an effective way. Currently, most ICS vendors have poor methods of notifying customers about potential security problems and patches. Experience has shown that some patches generated as the result of previous security assessments have been slow in being deployed with many end users unaware of the existence of the patches. ICS vendors should create and maintain security mailing lists and test the procedures needed to notify the end users about security problems. Increasing accessibility for end users to obtain the necessary information will greatly increase the use and effectiveness of patching. Many ICS vendors do publish security information, but frequently locate this information in an obscure location on their website that can easily be overlooked. This information should have a more prominent location and should be easy for the users to find. If this advice is followed, ICS vendors will help end users obtain and install the patches more easily. The more difficult it is to find and install the patches, the lower the patching rate will be.

Vendors should test and approve operating system patches, along with all other third-party software. Products and services such as the Network Time Protocol (NTP) should be kept at current version and patch levels prior to deployment at asset owner sites and be included in the patch testing process. ICS products that have third-party services and applications incorporated into their functionality should be designed so that these applications can be updated or replaced as easily as possible.

ICS vendor software vulnerabilities should be patched and made available to affected customers as well.

4.1.4 Redesign Network Protocols for Security

ICS network protocols and the service applications that implement them need to be redesigned for security. Most ICS network protocols were designed with the original ICS code base to be fast and only avoid failure issues and are not designed to provide robust authentication and integrity checks. Many protocol designs contain common security pitfalls. A number of characteristics of a secure protocol are relevant to this discussion.

1. Secure protocols should be simple. The more complex a protocol is, the higher the likelihood of bugs and vulnerabilities within the implementation.
2. Protocols should also minimize duplicate data. If data appear multiple times within the protocol, then portions of the implementation will invariably use one version of the data while other portions use another version. This allows an attacker to put the implementation into an unknown state by sending conflicting versions of the data.
3. Protocols with many optional fields and features are less secure because no two implementations will agree on what is optional and tend to make incorrect assumptions.
4. Secure protocols are also targeted; they contain enough functionality to get the job done and nothing more. If protocols contain seldom used or never used components then those components tend to be more buggy and contain more vulnerabilities than the components that are actually being used because they will be tested to a lesser degree. Secure protocols also have secure authentication methods and options for encryption or data integrity. Security by obscurity cannot be relied on because insider knowledge or reverse engineering can be used to recreate valid network packets. Some ICS protocol analyzers have already been developed, and one should expect to see more given the increasing interest in ICS security.
5. When possible, network protocols should be redesigned to improve security by avoiding common security pitfalls, avoiding designs that lead to implementation issues, and by including secure authentication and encryption methods.

4.1.5 Increase Robustness of Network Parsing Code

The robustness of network parsing code should be dramatically improved. Part of every network protocol is an associated program to build packets or process the traffic off the network. These applications are written by the ICS vendor for their propriety protocols as well as for common ICS protocols such as OPC, ICCP, and Distributed Network Protocol Version 3 (DNP3). If these applications contain input validation vulnerabilities, such as buffer overflows, exploitation by anyone who is able to gain access to the ICS host and port is possible. The lack of input validation can make a system more unstable and makes it vulnerable to attack. Potential consequences are

- Communication DoS
- Unauthorized access to the computer with the privileges granted to the compromised service
- ICS instability
- ICS integrity problems.

Data integrity checks need to be designed and implemented into ICS communication protocols. The lack of or weak data integrity checks prevent a protocol from detecting bad data. An attacker can take advantage of the poor integrity checks to send malformed packets in order to cause DoS attacks or to trigger a buffer overflow and compromise the system. An attacker does not always have to send malformed packets for manipulation of otherwise valid alarm or command messages sent over the wire if the ICS protocol has poor integrity checks.

4.1.6 Create Custom Protocol Parsers for Common IDSs

ICS vendors should create parsers for their custom protocols that can be used by common IDSs. In this manner, intrusion detection monitoring is made more effective by providing the ability to watch for illegal or abnormal values in ICS traffic. The bulk of the current IDS technology is focused on detecting exploits, not vulnerabilities. These systems are not very effective in the ICS environment due to the lack of known exploits to detect. If dissectors for the ICS protocols exist, rules could be written for the IDSs that verify network messages are within reasonable bounds and attempt to detect an exploitation of vulnerability.

4.1.7 Document Necessary Services and Communication Channels

ICS vendors should document how the ICS system components use the network so that effective firewall and IDS rules can be created. For each ICS component, vendors should document the necessary services along with the associated port ranges and which components are allowed to initiate a connection to that component.

ICS vendors should also provide complete documentation and automated setup of security features to allow for quicker, easier, and more consistent implementation of ICS components and security features. Security features that are obtuse or difficult to configure and implement are typically not used or are used incorrectly in the field installations of ICS. Security features that are inconsistently implemented or provide inconsistent results are considered a risk to reliability and availability of the ICS in an operational environment.

4.1.8 Redesign ICS to Use the Least Communication Channels Possible

ICS vendors should redesign their systems for security, reducing the number of services and communication channels required for system operation. Designers should eliminate, minimize, or secure the most unsecure services and communication channels first.

4.1.9 Implement and Test Strong Authentication and Encryption Mechanisms

ICS vendors should implement and strenuously test strong authentication and encryption mechanisms. Applications that process network traffic or accept network connections must use strong authentication to prevent unauthorized access and messages. Weak authentication in network protocols allows replay or spoof attacks to send unauthorized messages. Poor authentication also allows unauthorized users or computers to connect to a device or application. The lack of authentication in most ICS-specific network protocols allows for manipulation of time synchronization and process alarms, commands, and data updates. Poor authentication in protocol server applications allows unauthorized access to ICS components, including ICS hardware. Proven authentication services should be used when available.

Experienced personnel in authentication and encryption systems involved in creating these systems should be a part of any cybersecurity staff. Authentication and encryption systems are complex, and one small mistake or oversight can render the authentication or encryption ineffective. ICS vendors should rigorously test and validate that the authentication and encryption system are working correctly before deploying the solutions.

Where appropriate, ICS vendors should use well-vetted encryption algorithms and select well-tested implementations. ICS developers should design software so that one cryptographic algorithm can be replaced with another, enabling upgrade capability to stronger algorithms. ICS software maintainers should periodically ensure that current methods used have not been broken. Many old algorithms and implementations have become obsolete or discovered to be flawed.

ICS developers, integrators and administrators must securely manage and protect cryptographic keys. Keys should be strong and should not be hard-coded, default, published, or discoverable in any other way.

A remote end-point joins the trusted domain when it is allowed to remotely connect to the ICS network. If VPN endpoints (hosts) are compromised, an attacker can utilize the VPN connection when it is established. Importantly, these hosts must be secured to the maximum extent possible. Endpoint management software can be used to help determine the security posture of the remote device and how it is allowed to connect to the protected network, but should not be the only defense measure. VPN access should only be granted to the minimum set of hosts and users when necessary, and those VPN connections should be restricted to only allow access to the necessary components.

Internet Protocol security (IPSec) and VPN tunneling cannot be used as a replacement for fixing vulnerabilities. A VPN connection extends the attack surface of the system to the VPN client's computer. An attacker may be able to compromise a VPN endpoint computer and use the VPN tunnel as an encrypted pathway to exploit the vulnerabilities.

IPSec can be used for confidentiality, integrity, authenticity, and replay protection. If an attacker intends to disable IPSec or perform a DoS, he may attempt to gain access to any point between two IPSec partners. The implementation of IPSec included with Microsoft Windows XP, Windows Server 2003, and newer uses the identity proofing afforded by Active Directory. This authentication can be intercepted, causing IPSec to fail. This failure can cause a DoS if the IPSec policy is set to require IPSec for communications. If the IPSec policy is set to request, then an attacker can force IPSec to disable itself if they interfere with the communications long enough to fall back onto unencrypted channels. The decision for configuring this implementation of IPSec with a "request" policy versus a "require" policy should be made based on whether the communication between the IPSec partners must be confidential (or ensure integrity, authenticity, or replay protection) or the availability of communication based on criticality.

4.1.10 Improve Security through External Software Security Assessments

ICS software vendors should pay for a third-party security source code audit and fix the problems identified during the audit. Independent source code auditing can help ensure quality and security in software products. An outside professional opinion of software design and implementation based on the actual source code and build process of the ICS product will greatly enhance quality and security, or confirm the security of the product.

ICS software can have large, complicated, and legacy codebases. ICS operations require high availability, and update scenarios are complicated. Unlike the standard off-the-shelf computer software model, the cost of security fixes and support and maintenance has traditionally been transferred to the ICS customer. With the new focus and requirements for ICS security, including ICS product vulnerabilities starting to be publicly announced, vendors may find the cost of code audits and associated code changes to be very cost effective versus fixing single vulnerabilities as they are publically announced.

4.2 Recommendations for ICS Owners and Operators

An effective cybersecurity program for ICS should apply a strategy known as defense-in-depth, layering security mechanisms such that the impact of a failure in any one mechanism is minimized. Implementing security controls, such as intrusion detection software, antivirus software, and file integrity checking software, where technically feasible, will prevent, deter, detect, and mitigate the introduction, exposure, and propagation of malicious software to, within, and from the ICS.

The most successful method for securing an ICS is to gather industry- recommended practices and engage in a proactive, collaborative effort between management, the controls engineer and operator, the IT organization, and a trusted automation advisor. This team should draw on the wealth of information available from ongoing federal government, industry groups, vendor, and standards organizational activities. ICS owners should perform risk-based assessments on their systems and tailor the recommended guidelines and solutions to meet their specific security, business, and operational requirements.

Planning efforts need to be implemented for prioritization of the tasks necessary to enhance ICS security. Important considerations in this process are cost, probability, and consequence. Decisions concerning methods of mitigating cyber vulnerabilities include balancing the risk of system compromise by an intruder with the risk of potentially degrading system operability. Above all, the ICS must be reliable and perform its required mission. Therefore, the suggested approach is to build security into a system before it is put into production or add security into an existing system in small increments. When adding security to a production system, test on a backup system first to allow quick recovery to the previous configuration in the event any security measure affects system operation. Always weigh the risks and add the appropriate amount of security measures for the specific situation.

Asset owners must use procurement specifications to ensure that security development life-cycle requirements are met by the vendor. Asset owners also may hire independent security assessment teams to review demonstration vendor products for security issues prior to purchase. Vulnerability and patch management programs and policies must be established and enforced.

Good defense-in-depth perimeter protections should be used to help prevent access to vulnerable components and communication on ICS networks. Part of a good defense-in-depth strategy is identifying and mitigating known vulnerabilities and weaknesses in the system that may help an attacker manipulate or cause damage to the system. Continuous monitoring of IDS logs can allow system administrators to catch and block attempts to circumvent these defenses before serious damage is done.

Firewalls, IDSs, and antivirus solutions should be deployed and properly configured at all appropriate locations. Asset owners must identify and deploy security workarounds, defense-in-depth strategies, and use monitoring (access logs and IDSs) to mitigate risk introduced by the presence of unpatched vulnerabilities until patches can be properly tested and deployed.

Owners/operators are recommended to increase the security of their systems by completing the recommendations in the following sections. These recommendations are summarized below:

1. Redesign network layouts to take full advantage of firewalls, VPNs, etc.
2. Implement a network topology for the ICS that has multiple layers, with the most critical communications occurring in the most secure and reliable layer
3. Restrict physical access to the ICS network and devices
4. Expeditiously deploy security patches after testing all patches under field conditions on a test system if possible, before installation on the ICS
5. Work with vendor to test and apply patches for all operating systems and software on the ICS networks
6. Customize IDSs for the ICS hosts and networks
7. Restrict ICS user privileges to only those that are required to perform each person's job (i.e., establishing role-based access control and configuring each role based on the principle of least privilege)
8. Develop a password management plan to enforce strong passwords with minimum length, mixed character sets, expiration, no password reuse, etc., and change all default passwords.

4.2.1 Restrict ICS User Privileges to only those Required

A common problem with applications and services is that they are run with system or root-level privileges. If this case is applicable, and an attacker is able to redirect execution, exploit code will run with those same privileges giving the attacker full access to that device. A number of software products run with these super user permissions by default even though their functions do not require them. Therefore, permission levels of applications and services should be lowered to that necessary for their required functions.

Another common problem is allowing users to operate a computer system (consoles, servers, etc.) with more permissions than necessary. User accounts used for interactive logon should be carefully evaluated for the lowest set of permissions necessary.

File access should then be restricted to those who require access. If network access to a file is necessary, restrict access as much as possible and require strong authentication.

4.2.2 Change All Default Passwords and Require Strong Passwords

In some ICS operations, user IDs, and passwords are shared among the different operators of the system. This sharing must exist in many cases because of the criticality of the system operation. Unacceptable consequences might occur because of a locked user ID or a forgotten password. Typical continual manning of operating consoles provides additional physical security that reduces the need for distinct operator user IDs and passwords. If user-level authentication is not an option for operators, ensure all users have separate accounts for all other account types in the ICS to help increase security and accountability. These prudent actions can prevent an attacker from using a user ID and password obtained from the business LAN to gain access to the ICS DMZ and the ICS LAN and also prevent authorized users from performing actions that cannot easily be attributed to them.

ICS and networking equipment should not be left with the default manufacturer passwords. Default passwords can give an attacker easy access to the equipment that controls the process. Unless required by the ICS software, default passwords should always be changed to robust, unpublished passwords. In the case that the software uses hard-coded passwords, work with the vendor to fix this vulnerability. Implement a password policy that enforces strong passwords to greatly impede password cracking and guessing.

Passwords have been found in control rooms on small pieces of paper on the bottom of the keyboard, in a drawer, etc. If a password is too complicated and difficult to remember, or changes too often, users will undermine their security in order to remember them. Complex passwords do protect against some of the advanced password cracking attacks, but they create a physical and social engineering vulnerability that could be exploited by an attacker. Therefore, passwords should not be autogenerated, but instead created from passphrases or other memorable means.

4.2.3 Test and Apply Patches

ICS owners must rely on their ICS vendor in some part for validation of patch compatibility before applying them to their operational system. One way to reduce this problem is to reduce the number of applications that need to be patched.

Services or applications running on a system open up different network ports to be able to communicate to the outside world. Each open port provides a possible access path for an attacker that can be used to send exploits and receive data. An attacker can only gain access to and receive information from the ICS through an open port. The more ports and services that are accessible, the greater the risk of successful exploits due to existing vulnerabilities in the services.

New vulnerabilities are found every day in the applications and services that run on computers. Some of these vulnerabilities are published shortly after their discovery, and some are kept a close secret, allowing a few hackers to exploit computers at will, with no patches available to stop them. Decreasing the number of installed applications and services decreases the likelihood of an attacker finding a vulnerability on the computer. Therefore, all unneeded applications and services should be removed. Also, adequate resources must be allocated to ensure that all services and applications are completely patched and up-to-date using the process described in the preceding patches section.

The patching process should be worked closely with vendor support to ensure ICS application integrity is maintained. Before stopping any services or programs, the vendor should confirm that the service is not needed for system functionality. For confirmation, any patch process test should be performed on a backup or development system first, to isolate the primary system from any potential damage. For example, a standard security measure is to shut off the auxiliary services such as echo, chargen, daytime, discard, and finger. However, if the echo port is being used as the system pulse to confirm that the system is up and running, shutting off these services would disable the entire system.

4.2.4 Protect Critical Functions with Network Security Zones and Layers

In many cases, the individuals in charge of the ICS network do not have adequate security training. This situation is generally due to a lack of funding or appreciation for the importance of this training. Training provides an understanding of the security implications of a given network architecture and how to design a more secure network. Educating or hiring network administrators with skills to design and manage the ICS network and its perimeter defenses with the most current security techniques is essential. Network attacks must be prevented, detected, or stopped before they have the opportunity to affect critical ICS functions. ICS security is largely dependent on the effectiveness of the network design to prevent unauthorized access. Network administrators need to understand security concepts such as layering, security, and functionality zones, and specific access rules to restrict all communication to only that which is necessary for system functionality. If the network administrator has designed the network correctly, an attacker is limited to finding vulnerabilities in the authorized users/systems, protocols, or associated applications/servers allowed into each network segment, without being detected.

To provide defense-in-depth, firewalls can be used to separate different layers of the ICS network (i.e., the HMI level LAN from the ICS DMZ from the Enterprise network). These layers can be further segregated into security zones to protect systems from attack through compromised systems on that layer. Multiple DMZs, or security zones, should be created for separate functionalities and access privileges, such as peer connections, the data historian, the OPC server or ICCP server in SCADA systems, the security servers, replicated servers, and development servers.

Any connection into the ICS LAN is considered part of the perimeter. Often these perimeters are not well documented and some connections are neglected. All entry points into the ICS LAN should be known and strictly managed by a security policy. Route all connections to the ICS LAN through the firewall, with no connections circumventing it. Network administrators need to keep an accurate network diagram of their ICS LAN and its connections to other protected subnets, DMZs, the corporate network, and the outside.

Well-configured firewalls are critical to ICS security. Communications should be restricted to that necessary for system functionality. ICS traffic should be monitored, and rules should be developed that allow only necessary access. Any exceptions created in the firewall rule set should be as specific as possible, including host, protocol, and port information. All rules should be concise and well documented. The IDS sensors can then be used to audit the firewall rule set.

A common oversight is not restricting outbound traffic. Firewall rules should consider both directions through the firewall. An exploit that cannot connect back to the attacker is limited to blind attacks. An attacker needs to obtain information from and send files and commands to the ICS network. To remotely control exploit code running on an ICS computer, a return connection must be established from the ICS network. Because of the nature of most vulnerabilities, exploit code must be small and contain just enough code to get an attacker onto the computer; insufficient space is present to add expensive logic for the attacker to get advanced functionality. Therefore, additional instructions are needed from the attacker to continue with the discovery portion of the attack. If outbound filtering is implemented correctly, the attacker will not receive this return connection and cannot discover and control the exploited machine.

The top priority of most ICS installations is availability. The risk to availability of any security feature must be weighed against the expected added security benefit (lowered risk). ICS network administrators may not want to risk the chance of impacting ICS functionality by redesigning the network or updating rules as components are added or removed. In this case, network traffic can be monitored for a long enough period to be confident all possible scenarios have occurred. Rules can then be created starting with the standard restrictions; working toward a rule set that excludes all unnecessary traffic. Once the necessary traffic has been determined, a safer configuration can then be created that blocks all traffic with exceptions for the specific host, protocol, and port combinations that require access in each direction through the firewall.

Greater assurance that network security changes will not affect operations can be obtained by implementing changes as IDS rules. IDS logs can be monitored for alerts identifying traffic that would have been prevented by the new segmentation or access rules. All proposed network changes can be tested as IDS rules for as long as necessary to provide assurance that they will not affect critical functions. Because IDSs do not prevent access, ICS administrators or network security personnel should closely monitor IDS logs during this period and immediately investigate unexpected communication.

4.2.5 Customize IDS Rules for the ICS and Closely Monitor Logs

The configuration and deployment of IDS for an ICS is not as straightforward as it is for typical computer networks. IDS signatures are available to detect a wide range of attacks, but the signatures required to monitor for malicious traffic in control networks are not adequate. When looking at the unique communications protocols used in ICS, such as Modbus or DNP3, specific payload and port numbers have traditionally not been a part of the signatures seen in a contemporary IDS. In short, modern IDSs deployed on ICS networks may be blind to the types of attacks that an ICS would experience.

When deploying IDS in an ICS network, the ability to add unique signatures must be used. Removal of some default signatures and response capability is commonplace, as it may have no relevance to ICS network. However, analysis must be made to ensure some of the inherent capability of the IDS is leveraged with some of the capability refined and augmented. Many security vendors, including those specializing in ICS security, have created signatures for the IDS that are deployed in control architectures. Rules sets and signatures unique to the traffic on the network being monitored are imperative when deploying IDSs on ICS networks. Developing security signatures and rules in a cooperative relationship with the ICS vendor are shown through study as very advantageous.

One of the common problems observed in industry is that tools deployed for network monitoring are implemented but improperly updated, monitored, or validated. Assigned individuals should be trained and given the responsibility of monitoring system data logs and keeping the various tool configurations current.

IDS logs can also be used to identify normal communication between each of the ICS components. All unexpected traffic can be investigated and either added to the required communication list or blocked by firewalls.

A one-to-one mapping of firewall rules and IDS signatures should exist so when a firewall rule is not successfully applied, the IDS sensor will alert and allow administrators to take corrective action on the firewall.

The external IDS sensor is used for notification of malicious attempts on the firewall and for monitoring egress rules from the ICS out to the DMZ or corporate networks. The internal IDS sensor and the DMZ IDS sensor are used to closely monitor the exceptions in the firewall for malicious activity.

Intrusion detection is not a single product or technology. A comprehensive set of tools providing network monitoring can give an administrator a complete picture of how the network is being used. Implementing a variety of these tools will help create a defense-in-depth architecture that will be more effective in identifying attacker activities.

4.2.6 Force Security through External Software Security Assessments

ICS customers can require a security audit of an ICS product and fixes in order to meet specified security levels as part of the procurement process. This allows the ICS customers to identify security risks of the products and determine whether they are acceptable or able to be mitigated. ICS owners can also have external security audits on their existing systems to identify risks that need to be mitigated. Security audits also help fulfill regulatory requirements, but the audit should be used to help secure the ICS as much as possible, not just to fill a requirement.

As ICS industry security requirements have begun to be created, some facilities have learned that they can get away with documenting exceptions to the rules. The requirements developed in an effort to help ICS owners increase their security levels have failed in some cases. ICS owners should look at the development of standards as an opportunity to obtain assistance in securing their assets. Requirements such as yearly security audits can be viewed by those responsible for ICS systems as help in convincing management to spend money on security.

5. REFERENCES

1. DHS CSSP, *Common Cyber Security Vulnerabilities Observed in DHS Industrial Control Systems Assessments,* July 2009, http://www.us-cert.gov/control_systems/pdf/DHS_Common_Vulnerabilities_R1_08-14750_Final_7-1-09.pdf.

2. ANSI/ISA–99.00.01–2007, *Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models*, October 2007, pages 69–73.

3. DHS, *DHS Recommended Practice Case Study: Cross-Site Scripting,* February 2007, http://www.us-cert.gov/control_systems/practices/documents/xss_10-24-07_Final.pdf, Web page last accessed December 2010.

4. MITRE, *CWE (Common Weaknesses Enumeration)*, http://cwe.mitre.org/, Web page last accessed January 2011.

5. Kevin Poulsen, *Slammer worm crashed Ohio nuke plant network*, August 2003, http://www.securityfocus.com/news/6767, Web page last accessed January 2011.

6. DHS CSSP, *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies*, October 2009, http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf, Web page last accessed January 2011.

7. Jaikumar Vijayan, *Gates pushed change in security culture at Microsoft*, June 2008, http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9102998, Web page last accessed January 2011.

8. MITRE, *Common Attack Pattern Enumeration and Classification (CAPEC)*, http://capec.mitre.org/, Web page last accessed February 2011.

Appendix A

Terms and Definitions

Appendix A

Terms and Definitions

Access Authorization

Access authorization restricts access to or from a computer, server, website, or network to a group of users through the application of authentication systems. These systems can protect either the whole computer, such as through an interactive logon screen, or individual services, such as an FTP server. Many methods are available for identifying and authenticating users, such as passwords, identification cards, smart cards, and biometric systems.

Access Control List

An access control list is a list of permissions attached to a firewall, server, or other device on a network. The list specifies who or what is allowed to access the device and what operations are allowed to be performed on the device.

Antivirus Software

Antivirus software consists of a computer program that attempts to identify, neutralize, or eliminate malicious software (i.e., viruses, Trojan horses, malware, spyware).

ARP

Address resolution protocol (ARP) is the standard method for finding a host's hardware address when only its network layer address is known.

Buffer Overflow

There are two types: stack buffer overflow and heap buffer overflow. Both types of overflow occur when an amount of data larger than the target data buffer area is written to that buffer. The extra data overwrite adjacent memory locations in either the stack (temporary memory) or the heap (dynamic memory) with corrupt data values causing erroneous program results or malicious code to be executed.

Change Management

The change management process is the process of requesting, determining attainability, planning, implementing, and evaluation of changes to a system. It has two main goals: supporting the processing of changes and enabling traceability of changes.

DMZ

A demilitarized zone (DMZ), more appropriately known as demarcation zone or perimeter network, is a physical or logical subnetwork that interfaces an organization's external services to a larger, untrusted network, usually the Internet. The DMZ adds an additional layer of security to an organization's Local Area Network (LAN).

Encryption

Encryption is the process of transforming information (referred to as plaintext or clear text) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key.

Exploit

An exploit (from the same word in the French language, meaning "achievement" or "accomplishment") is a piece of software, a chunk of data, or sequence of commands that take advantage of a bug, glitch, or vulnerability in order to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). This frequently includes such things as gaining control of a computer system or allowing privilege escalation or a denial-of-service attack.

Finding

An item identified during an assessment. It can be a vulnerability, an observation, a weakness, a flaw, a code error, or a concern.

Firewall

Firewalls can either be hardware devices or software programs. They provide some protection from online intrusion. They are systems that help protect computers and computer networks from attack and subsequent intrusion by restricting the network traffic that can pass through them, based on a set of system administrator defined rules.

Fuzzing or Fuzz Testing

A software testing technique that uses random data, also known as "fuzz," as input to the software. This technique attempts to exercise code by using values that may be outside the normal range of values for which the software was designed. By doing this testing, it will uncover areas of the code that were inadequate in handling input values outside the normally desired ranges.[x]

ICCP

The Inter-Control Center Communications Protocol (ICCP or IEC 60870-6/TASE.2) is being specified by utility organizations throughout the world to provide data exchange over wide-area networks between utility control centers, utilities, power pools, regional control centers, and Non-Utility Generators. ICCP is also an international standard: International Electrotechnical Commission (IEC) Telecontrol Application Service Element 2 (TASE.2).

Industrial Control System

A device or set of devices to manage, command, direct, or regulate the behavior of other devices or systems.

ICS-CERT Advisory

An ICS-CERT Advisory is intended to provide awareness or solicit feedback from critical infrastructure owners and operators concerning ongoing cyber events or activity with the potential to impact critical infrastructure computing networks.

Ground Truthing

The technique of verifying that results obtained from lab testing or simulations are repeatable in real-world situations. An example: lab results show a particular configuration creates a vulnerability. Ground truthing of this is accomplished by checking the production system and verifying that indeed a vulnerability exists.

[x] See *The Open Web Application Security Project*: http://www.owasp.org/index.php/Fuzzing

Information Leaks

Inside information that is carelessly disseminated such as passwords written on sticky notes or shared among users. This can also include information items such as user IDs, passwords, and other system information that is not encrypted when transmitted or when stored.

Least Privileges

The technique of assigning privileges for doing certain functions to only those that require them. For example, restricting the ability to create new user accounts to only the system administrator or a user that should only be able to query a database, but has privileges to delete the folder containing the database file.

Man-in-the-Middle Attack

The man-in-the-middle (MitM) attack or bucket-brigade attack is a form of active eavesdropping in which the attacker makes independent connections with computers that communicate with one another and relays messages between them, making them believe that they are talking directly to each other over a private connection when in fact the entire conversation is controlled by the attacker.

OPC

Object Linking and Embedding (OLE) is a technology that allows embedding and linking to documents and other objects developed by Microsoft. OLE for Process Control (OPC) is the standards specification for the communication of real-time plant data between control devices from different manufacturers.

Protocol

A protocol is the set of standard rules for data representation, signaling, authentication, and error detection required to send information over a communications channel.

Reliability

Reliability is the ability of a system to perform and maintain its functions in routine circumstances as well as hostile or unexpected circumstances.

Safety System

A Safety System or Safety Instrumented System (SIS) is a control system consisting of sensors, one or more controllers, and final elements. An SIS monitors an industrial process for potentially dangerous conditions and alarms or executes preprogrammed action to either prevent a hazardous event from occurring or mitigate the consequences of such an event should it occur.

Social Engineering Awareness

Keeping employees aware of the dangers of social engineering and having a policy in place to prevent social engineering can reduce successful breaches of the network and servers.

Taxonomy

The science, laws, or principles of classification.

Appendix B

CSET Self Assessment Activities

Appendix B

CSET Self Assessment Activities

The CSET self-assessments consist of the following six activities in order to provide the users with a systematic and repeatable approach for assessing the cybersecurity posture of their ICS.

Form Team: A team is formed by selecting cross-functional resources consisting of personnel familiar with the various operational areas in an organization. For example, in the ICS environment, teams typically include representatives that are familiar with the ICS details such as senior management, operations, information technology, ICS engineers, and security (physical and cyber). Organizations may add additional team members depending on the skills and expertise required to complete the assessment process.

Select Standards: Users are given the option to select one, several, or all the following industry and government recognized cybersecurity standards.

- DHS Catalog of Control Systems Security: Recommendations for Standards Developers, Revisions 4 and 6
- NIST SP800-82
- NIST SP800-53, Revisions 2 and 3
- NERC CIP-002-009 Revisions 1 and 2
- ISO/IEC 15408 Revision 3.1
- DoDI 8500.2
- Consensus Audit Guidelines 2.3.

After the user selects the applicable standards, CSET will generate questions that are specific for those requirements.

Determine Assurance Level: The Security Assurance Level is based on the user's answers to a series of questions related to the potential worst-case consequences of a successful cyber attack. CSET will calculate a recommended Security Assurance Level for the facility or subsystem being assessed and then provide the level of security rigor needed to protect against a worst-case event. For NIST-based standards and guidance, CSET also supports the Federal Information Processing Standards (FIPS) 199 guidelines for determining the security categorization of a system. The system will determine and report security gaps based on comparing the answers with the different assurance levels.

Create Diagram and Analyze Network Topology: CSET contains a graphical user interface that allows users to build the control system network topology (including criticality levels) into the CSET software. By creating a network architecture diagram, which is based on components deemed critical to the organization, users are able to define the organizations cybersecurity boundary and posture. An icon palette is provided for the various system and network components, allowing users to build a network architecture diagram by dragging and dropping components onto the screen. Specific questions are then generated for each component.

Answer Questions: CSET generates questions based on the specified network topology and the security standards that were selected. The assessment team then selects the best answer to each question based on the system's network configuration and implemented security practices. CSET compares the answers provided by the assessment team with the recommended security standards and generates a list of recognized good practices and security gaps.

Review Reports: CSET generates both interactive (on-screen) and printed reports. The reports provide a summary of security level gaps or areas that did not meet the recommendations of the selected standards. The assessment team may then use this information to plan and prioritize mitigation strategies.

Appendix C

Acronyms

Appendix C

Acronyms

ARP	address resolution protocol
CIP	Critical Infrastructure Protection
CRADA	Cooperative Research and Development Agreement
CS2SAT	Control System Cyber Security Self-Assessment Tool
CSET	Cyber Security Evaluation Tool
CSRF	cross-site request forgery
CSSP	Control Systems Security Program
DCOM	Distributed Component Object Model
DHS	U.S. Department of Homeland Security
DMZ	demilitarized zone
DNP	distributed network protocol
DoS	denial-of-service
FIPS	Federal Information Processing Standards
FTP	File Transfer Protocol
HMI	human-machine interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IACS	Industrial Automation and Control Systems
ICCP	Inter-Control Center Communications Protocol
ICS	industrial control system(s)
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
IDS	intrusion detection system(s)
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPSec	Internet Protocol security
ISA	International Standards Association
IT	information technology
LAN	local area network
LM	LAN Manager (password hash)
MAC	media access control

MitM	man-in-the-middle
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
NTLM	NT LAN Manager
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OS	operating system
RPC	Remote Procedure Call
rsh	remote shell
SCADA	Supervisory Control and Data Acquisition
SIS	Safety Instrumented System
SNMP	Simple Network Management Protocol
SP	Special Publications
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
TASE	Telecontrol Application Service Element
VLAN	virtual local area network
VPN	virtual private network
XSS	cross-site scripting