INL/EXT-10-18381

NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses

May 2010

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-10-18381

NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses

May 2010

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Electricity Delivery and Energy Reliability
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

DISCLAIMER

This information 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 of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/EXT-10-18381

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

NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses

May 2010

NSTB

National SCADA Test Bed

Enhancing control systems security in the energy sector

NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses

INL/EXT-10-18381

May 2010

Approved by:

David G. Kuipers
Project Manager
National SCADA Test Bed Program

6/15/2010
Date

EXECUTIVE SUMMARY

Idaho National Laboratory (INL) performs cyber security assessments of Industrial Control Systems (ICSs) under private sector and government programs. This report applies to assessments conducted on behalf of the Department of Energy Office of Electricity Delivery & Energy Reliability (DOE-OE) National Supervisory Control and Data Acquisition (SCADA) Test Bed (NSTB) program. The mission is to help industry and government improve the security of the ICSs used in critical energy infrastructure installations throughout the United States. A key part of this mission is the assessment of ICSs to identify vulnerabilities that could put critical infrastructure at risk from a cyber attack.

Although information found in individual stakeholder ICS vulnerability assessment reports is protected from disclosure, the security of the nation's energy infrastructure as a whole can be improved by sharing information on common security problems with those responsible for developing and operating ICSs. For this reason, vulnerability information was collected, analyzed, and organized to allow the most prevalent issues to be identified and mitigated by those responsible for individual systems without disclosing the identity of the associated ICS product.

Information found in this report can benefit vendors, asset owners, and other stakeholders responsible for securing the systems that control the nation's energy infrastructure. System vendors learn of common weaknesses in ICS applications, services, and protocols, and how to better secure their products. Asset owners can evaluate and better secure their installed system environments and defend against and monitor for exploitation of possible weaknesses in their installed system configurations. Understanding the types of vulnerabilities commonly found and how to mitigate them can serve to help protect the systems currently in development as well as those already installed in ICS applications.

This document presents results from 24 ICS assessments performed under the NSTB program from 2003 through 2009. NSTB assessments reported large ICS attack surfaces created by excessive open ports allowed through firewalls and unsecure and excessive services listening on them. Well-known unsecure coding practices account for most of the ICS software vulnerabilities, which result in system access vulnerability or Denial of Service (DoS). However, poor patch management provides more likely attack targets because the vulnerabilities are public and attack tools are available for them. Once ICS network access is obtained, status data and control commands can be manipulated as they are communicated by unsecured ICS protocols.

Perimeter defenses cannot mitigate threats associated with required services between security zones. Vulnerabilities in Web services, database applications, and data transfer protocols can provide attack paths through firewalls. ICS network protocol applications can also be exploited to gain access to ICS hosts. Weak authentication and integrity checks allow unauthorized control or data manipulation, once ICS network access has been obtained.

NSTB assessments indicate that the biggest security threats to ICS in the energy infrastructure can be mitigated by patch management, eliminating unnecessary and unsafe services, implementing strong authentication and integrity checks to network protocols, and securing applications that accept network traffic. Secure configurations and network layers of defense can then be used to protect these critical assets.

ACKNOWLEDGEMENT

This document was developed for the U.S. Department of Energy to provide a methodology for industry to apply an industry standardized metrics approach to vulnerability classification for industrial control systems. The author team consisted of subject matter expertise from the Idaho National Laboratory

(May Chaffin and Kathleen Lee).

CONTENTS

FIGURES

TABLES

ACRONYMS

ACL	Access Control List
API	Application Programming Interface
ARP	Address Resolution Protocol
ASVS	Application Security Verification Standard
CAPEC	Common Attack Pattern Enumeration and Classification
CVSS v2	Common Vulnerability Scoring System Version 2.0
CWE	Common Weaknesses Enumeration
DMZ	Demilitarized Zone
DNP3	Distributed Network Protocol Version 3
DNS	Domain Name System
DOE	Department of Energy
DOE-OE	Department of Energy-Office of Electricity Delivery and Energy Reliability
DoS	Denial of Service
EMS	Energy Management System
FTP	File Transfer Protocol
HMI	Human-Machine Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
ICCP	Inter-Control Center Communications Protocol
ICS	Industrial Control System
IDS	Intrusion Detection Systems
IKE	Internet Key Exchange
INL	Idaho National Laboratory
IP	Internet Protocol
IPSec	Internet Protocol Security
IT	Information Technology
LAN	Local Area Network
LM	Windows LAN Manager
MAC	Media Access Control
MitM	Man-in-the-Middle
NIST	National Institute of Standards and Technology
NSTB	National Supervisory Control and Data Acquisition Test Bed

NTLM	Windows NT LAN Manager
NTP	Network Time Protocol
OLE	Object Linking and Embedding
OPC	Object Linking and Embedding (OLE) for Process Control
OS	Operating System
OWASP	Open Web Application Security Project
SAMM	Software Assurance Maturity Model
SANS	SysAdmin, Audit, Network, Security
SCADA	Supervisory Control and Data Acquisition
SDL	Security Development Lifecycle
SDLC	Software Development Life Cycle
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
TASE.2	Telecontrol Application Service Element 2.0
URL	Universal Resource Locator
US-CERT	United States Computer Emergency Readiness Team
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

NSTB Assessments Summary

1. INTRODUCTION

The U.S. Department of Energy (DOE) established the National Supervisory Control and Data Acquisition (SCADA) Test Bed (NSTB) program to assist industry and government in improving the security of Industrial Control Systems (ICSs) used in the nation's critical energy infrastructures. The NSTB program is funded and directed by the DOE Office of Electricity Delivery and Energy Reliability (DOE-OE). A key part of the program is assessment research of ICSs to identify and provide mitigation approaches for vulnerabilities that could put the systems at risk to a cyber attack.

The assessment process is highly flexible and may be tailored to the mutual interests of the industry partner and the NSTB program. The typical process includes the following sequence:

- Establish agreement that defines the working relationship (scope, personnel, equipment, facilities, cost sharing) and ensures protection of sensitive information
- Work with partner to establish goals or assessment targets
- Obtain equipment and training from the industry partner
- Set up equipment with support from the industry partner
- Perform assessment to identify cyber vulnerabilities
- Provide detailed assessment report to industry partner
- Issue information suitable for public release to Web sites, conferences, and users' groups.

A key objective of the NSTB program is to share relevant information obtained through security assessments with potentially impacted industry stakeholders, with an emphasis on asset owners and users. However, it is recognized that much of the information obtained in assessments is business sensitive to the industry partner whose system or technology has been assessed. The program works with the industry partner to determine what information obtained or derived from the assessment process is appropriate for disclosure outside the partnership and to identify an appropriate format and forum for disclosure. NSTB does not release attributable information without written concurrence of the industry partner.

The main goal of ICS cyber security is preventing unauthorized manipulation of the system under control. Secondary goals are the availability and integrity of system state data and control commands. Protecting the physical system and its data from malicious manipulation requires protection mechanisms in each of the many networks, applications, and hosts that make up the ICS.

This report presents vulnerabilities at a high level to provide awareness of the common ICS security weakness areas without divulging product-specific information. Vulnerabilities that could be used as part of an attack against an ICS are consolidated into generic common ICS vulnerabilities. Even though ICS functionalities, designs, and configurations vary between vendors, versions, and installations, their vulnerabilities and defensive recommendations are quite similar at a high level.

First, the most significant ICS vulnerabilities are summarized. Next, the NSTB assessment results are presented as categories of finding attributes. Then the common vulnerabilities are discussed with ICS specifics provided along with references to more detailed security information. Finally, recommendations are summarized for ICS developers and administrators to help mitigate the risk of compromise due to vulnerabilities caused by the weaknesses discovered by NSTB ICS security assessments.

2. MOST SIGNIFICANT ICS VULNERABILITIES

The most significant vulnerabilities identified in ICS are those that allow unauthorized control of the physical system. Compromise of the ICS's availability and ability to function correctly may also have significant consequences.

The likelihood of a successful attack must also be considered when assessing risk. Understanding exposure to attack, attacker awareness of vulnerability, and exploitation knowledge help assess the probability of a successful attack.

According to the SysAdmin, Audit, Network, Security (SANS) *2009 Top Cyber Security Risks* report, "During the last few years, the number of vulnerabilities being discovered in applications is far greater than the number of vulnerabilities discovered in operating systems. As a result, more exploitation attempts are recorded on application programs." [1] Overall, operating system (OS) patch management support has improved to the point where OS patching is trivial for most situations. Many ICS vendors now provide timely OS patch test results; however, application patching is often ignored. OS and application patching is not a trivial effort by ICS users since patches need to be tested in a development environment prior to incorporation in their production systems. Application patching can be even more difficult if security fixes change the way that the ICS software must interface with application.

Applications, services, and libraries, not included as part of the OS, tend to not get patched on ICSs. NSTB assessments have not only discovered old and unpatched versions of third-party products on production ICS, but integrated into new ICS product versions as well. Published vulnerabilities in well-known applications and services create the most significant security risks to ICS. Of all the vulnerabilities in an ICS, these vulnerabilities are most likely to be exploited because attackers are likely to be aware of them.

The SANS 2009 report agrees with NSTB reports, that Web vulnerabilities are most critical. "The most 'popular' applications for exploitation tend to change over time since the rationale for targeting a particular application often depends on factors like prevalence or the inability to effectively patch. Due to the current trend of converting trusted Web sites into malicious servers, browsers and client-side applications that can be invoked by browsers seem to be consistently targeted. Automated tools, designed to target custom Web application vulnerabilities, make it easy to discover and infect several thousand Web sites."[1] Due to poor programming practices, ICS Web services are vulnerable to the most popular attack techniques such as Structured Query Language (SQL) injection, cross-site scripting, directory traversal and authentication bypass. ICS Web applications are also more exposed to attack than most ICS components, and may provide the capability to alter ICS data or state.

Web and other applications and services that execute with higher than necessary privileges unnecessarily increase the risk and impact of exploitation. Successful exploitation grants the attacker the same privileges as the compromised application. This includes network privileges of the compromised user and/or host. An attacker may also be able to utilize unnecessary functionality incorporated into applications and services, even if they are disabled. Access controls should be incorporated in ICS components to help prevent and contain compromise.

The ICS environment can be locked-down by providing strong authentication, compartmentalizing functionality, and limiting application, service, and user permissions to only the required access and functionality necessary. Incorporation of least user privileges may require a redesign of some ICS components. Removing unnecessary functionality on ICS hosts includes removing all services and applications that are not necessary for each individual host's role. Develop documentation of required services, communication partners, and direction of communication to lock down the ICS host environments.

Network defenses that utilize specific firewall and Intrusion Detection System (IDS) rules can help control access, even if an attacker has gained access inside the ICS perimeter. The more specific the rules, the more unwanted network traffic are blocked. This requires an understanding of the network protocol, including the valid structure and value ranges. An attacker who has gained access and privileges of a legitimate user often performs actions not typical for that user (or the ICS). Specific firewall and IDS rules can block or detect abnormal activity. An IDS is used without the risk of compromising operations because it is passive and only alerts on suspicious traffic. However, accurate, custom rules along with dedicated and qualified monitoring are necessary for effective intrusion detection.

Unfortunately, ICS vendors do not typically provide enough documentation on required component communications. ICS owners can monitor their own system traffic and create rules that describe their system's behavior. Many ICS protocols are vulnerable to man in the middle (MitM) and spoofing attacks, and require access through firewalls between security zones. Vulnerabilities in services that parse network traffic can allow unauthorized access to their host. These communications cannot be blocked, and the design of the communication protocol determines the degree to which access to these services can be restricted.

Even with good network design with security zones, ICS vulnerabilities are exposed to less-trusted networks to provide remote monitoring, data sharing, historical, and other remote access functions. Vulnerability remediation is necessary because access to ICS software vulnerabilities cannot be prevented entirely. The NSTB has taken care to inform only the responsible ICS vendors of vulnerabilities identified in their products, but other security researchers are starting to announce ICS vulnerabilities in more open formats. Typical hacker behavior is to quietly exploit a newly discovered vulnerability before announcing it publicly. Industry trends show increasing cyber attacks[1]. ICS products must be thoroughly assessed, secured and tested, starting with the most exposed and powerful functions.

The NSTB has seen a significant improvement in OS and network security since 2003. There has been slight improvement in reducing host exposure through services. Little, or spotted, improvement has been seen in vulnerability remediation and secure development of new products. Vulnerabilities, due to unsecure coding practices, are found in new and old products alike, and the introduction of Web applications into ICSs has created more, as well as new, types of vulnerabilities.

Secure design and vulnerability remediation activities have been judged by many companies as undoable due to time, cost, and backward compatibility issues involved. Encryption is in the process of being applied to ICS communications as a mitigation in lieu of remediation. Adding encryption can limit exposure, but does not prevent access through the encrypted channel if an attacker has compromised an encryption endpoint. Encryption can also make system monitoring and trouble-shooting difficult. NSTB experience and feedback have shown that encryption of ICS communications is rarely accomplished successfully. Unsecure encryption configurations have been found. Often encryption was not implemented because it could not be accomplished without disabling ICS communications, or because the communications partner did not support it. In short, encryption can be used correctly as a layer of defense, where appropriate, but not as a mitigation to vulnerabilities.

To rank the most significant issues identified during NSTB assessments, the Common Vulnerability Scoring System version 2.0 (CVSS v2) metrics were applied generically to the common vulnerabilities.[2]

2.1 CVSS v2 Vulnerability Scoring System

Patterned after the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[3], the most critical ICS vulnerability types are summarized using the attribute types in Table 1, when applicable. Attributes assigned to weaknesses that appeared on the SANS/CWE Most Dangerous Programming Errors list were used for associated vulnerabilities.

Table 1. Most critical ICS vulnerability attribute types.

Vulnerability Type	Vulnerability Description
Possible Consequences	When this weakness occurs in software to form a vulnerability, what are the typical consequences of exploiting it?[3]
ICS Impact	ICS specific consequences.
Vulnerable Components	ICS components that may have this vulnerability.
Ease of Detection	How easy it is for an attacker to find this weakness.[3]
Attacker Awareness	The likelihood that an attacker is going to be aware of this particular weakness, methods for detection, and methods for exploitation.[3]
Internet Attack Frequency	How often the weakness occurs in vulnerabilities that are exploited by an attacker.[3]
Remediation Cost	The amount of effort required to fix the weakness.[3]
Weakness Prevalence	How often the issue is encountered in software.[3]
ICS Prevalence	How often the weakness is encountered during assessments.

Generic ICS vulnerabilities are scored in this report using the CVSS v2 and the most common or highest impact characteristics.

> *"The Common Vulnerability Scoring System (CVSS) provides an open framework for communicating the characteristics and impacts of Information Technology (IT) vulnerabilities. CVSS consists of three groups: Base, Temporal, and Environmental. Each group produces a numeric score ranging from 0 to 10, and a Vector, a compressed textual representation that reflects the values used to derive the score. The Base group represents the intrinsic qualities of vulnerability. The Temporal group reflects the characteristics of vulnerability that change over time. The Environmental group represents the characteristics of vulnerability that are unique to any user's environment."*[2]

The individual CVSS v2 metrics are summarized in the tables below. Additional information on CVSS v2 criticality scoring is in Appendix B and at the CVSS Web site.[2]

Table 2 summarizes the Base CVSS metrics. Base metrics are not unique to ICS.

Table 2. CVSS v2 Base scoring metrics.

Base Metric	Measurement	Scoring
Access Vector	How the vulnerability is exploited	The more remote an attacker can be to attack a host, the greater the vulnerability score
Access Complexity	Complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system	The lower the required complexity, the higher the vulnerability score
Authentication	Number of times an attacker must authenticate to a target to exploit a vulnerability	The fewer authentication instances that are required, the higher the vulnerability score
Confidentiality Impact	Impact on confidentiality of a successfully exploited vulnerability	Increased confidentiality impact increases the vulnerability score
Availability Impact	Impact to availability of a successfully exploited vulnerability	Increased availability impact increases the vulnerability score

Table 3 summarizes the Temporal CVSS v2 metrics. The Remediation Level metric may not be valid for an installed system if the available patch or temporary fix cannot be applied without compromising ICS functionality. If patch testing results indicate that a patch cannot be applied or the old version cannot be replaced with a secure version, altering the ICS product to accommodate the patch or otherwise remediate the problem needs to become a top priority for the ICS vendor (weighted by the risk it exposes the system to). Remediation level scoring in this report is left as “Not Defined” for third-party products used by ICSs to reflect this issue and to avoid minimizing the state of the systems that were assessed. If this is the case when scoring a specific vulnerability in an ICS, the appropriate selection is “Unavailable: There is either no solution available or it is impossible to apply.”[2]

Table 3. CVSS v2 Temporal scoring metrics.

Temporal Metric	**Measurement**	**Scoring**
Exploitability	Current state of exploit techniques or code availability	The more easily a vulnerability can be exploited, the higher the vulnerability score
Remediation Level	Level of remediation available	The less official and permanent a fix, the higher the vulnerability score
Report Confidence	Degree of confidence in the existence of the vulnerability and the credibility of the known technical details	The more a vulnerability is validated by the vendor or other reputable sources, the higher the score

Different environments can have an immense bearing on the risk that a vulnerability poses to an organization and its stakeholders. The CVSS v2 environmental metric group captures the characteristics of a vulnerability that are associated with a specific environment (see Table 4). For this report, generic ICS security requirements are used to score generic ICS vulnerabilities.

Security requirements metrics enable ICS owners to customize the CVSS v2 score depending on the importance of the affected component to their own organization, measured in terms of confidentiality, integrity, and availability. For example, denial of service (DoS) vulnerabilities in ICS components that require high availability will receive higher criticality scores than they otherwise would.

Table 4. CVSS v2 environmental scoring metrics.

Environmental Metric	**Measurement**	**Scoring**
Collateral Damage Potential	Potential for loss of life or physical assets through damage or theft of property or equipment	The greater the damage potential, the higher the vulnerability score
Target Distribution	Proportion of vulnerable systems	The greater the proportion of vulnerable systems, the higher the score
Security Requirements	Importance of the affected IT asset to a user’s organization, measured in terms of confidentiality, integrity, and availability	The greater the security requirement, the higher the score

2.2 Top 10 Most Critical ICS Vulnerabilities

The most significant ICS vulnerabilities are ranked in Table 5 using Common Weakness Enumeration (CWE) and CVSS v2 metrics applied generically to the vulnerabilities identified during NSTB assessments. Exposure and security requirements can be adjusted for individual ICS installations.

Table 5. Top 10 most critical ICS vulnerabilities.

Rank	Impact/Vulnerability	Generalized CVSS v2 Score
1	Most Likely Access Vector/ Unpatched Published Vulnerabilities	9.8
2	Supervisory Control Access/ Use of Vulnerable Remote Display Protocols	9.8
3	Supervisory Control Access/ Web HMI Vulnerabilities	9.8
4	ICS Host Access/ Buffer Overflows in ICS services	9.3
5	Access to ICS Applications/ Improper Authentication	9.3
6	Access to ICS Functionality/ Improper Access Control (Authorization)	9.1
7	ICS Credentials Gathering/ Use of Standard IT Protocols with Clear-text Authentication	9.0
8	ICS Credentials Gathering/ Unprotected Transport of ICS Application Credentials	9.0
9	Supervisory Control Access/ ICS Data and Command Message Manipulation and Injection	8.8
10	Data Historian Access/ SQL Injection	8.6

2.2.1 Most Likely Access Vector: Unpatched Published Vulnerabilities

In general, patches are the highest priority because they remediate vulnerabilities with the highest threat. "Public availability of easy-to-use exploit code increases the number of potential attackers by including those who are unskilled, thereby increasing the severity of the vulnerability."[2]

Table 6 summarizes the security relevant attributes of unpatched software and their potential risk to ICSs.

Table 6. Summary of unpatched published vulnerabilities' security characteristics.

Unpatched Published Vulnerabilities	
Possible Consequences	Compromise of ICS hosts and applications. May allow DoS, code execution, data loss, or security bypass.
ICS Impact	Unauthorized access to ICS components: Most likely access vector
Vulnerable Components	Unpatched operating systems, applications, services and libraries on ICS hosts
Ease of Detection	Easy
Attacker Awareness	High
Internet Attack Frequency	High
Remediation Cost	Low
ICS Prevalence	High

Each vulnerability must be scored individually. The criticality of each unpatched vulnerability is different. CVSS v2 scores of published vulnerabilities are available from multiple vulnerability databases, such as the National Vulnerability Database.[a] Base scores can then be tailored to the current temporal values and the particular environment. For a set of vulnerabilities with equal base and environmental impact scores, the known vulnerabilities are higher priority.

[a] http://web.nvd.nist.gov/view/vuln/search

The following example scores represent the most dangerous known vulnerabilities identified on ICS systems for commonly unpatched components.

In general, OS services are network accessible and do not require authentication. Vulnerabilities in OS services can potentially be exploited to gain control of the host. The Access Complexity is “Low” because no additional access or specialized circumstances need to exist for the exploit to be successful.

If an attacker successfully exploits a network service, he may be able to execute arbitrary code with the privileges of the exploited application. If the vulnerable service is executed with administrative (system) privileges, a complete host compromise is possible. If the privileges gained allow access to ICS functionality, a successful exploit may result in catastrophic physical or property damage and loss; or there may be a catastrophic loss of revenue or productivity.

The CVSS v2 score is summarized in Table 7.

Table 7. Generic CVSS score for published vulnerabilities that lead to remote code execution.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	10.0
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	9.5
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.8
Total Score	**9.8**

2.2.2 Supervisory Control Access: Use of Vulnerable Remote Display Protocols

Remote display protocols and applications are used to remotely access a machine, providing the ability to logon and remotely control another machine using the graphical display. Applications and OS services that allow remote display are widely used by ICS to administer ICS hosts remotely or access operator screens and other ICS applications. Remote display protocols used by ICS have been found to accept connections from anywhere, transport credentials in clear text, or use a broken encryption algorithm. Even if strong encryption is used, if the remote display client's host is compromised, the attacker may also have access to the remote ICS host's display.

The use of remote display software for remote access to supervisory control functions could be the most significant vulnerability on an ICS because it allows unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user. (These vulnerabilities are well known.) Table 8 summarizes the relevant security attributes of remote display protocols and their potential risk to ICSs.

Table 8. Summary of remote display protocols' security characteristics.

Use of Vulnerable Remote Display Protocols	
Possible Consequences	May allow DoS, code execution, data loss, or security bypass.
ICS Impact	Unauthorized access to ICS components: Possible unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user.
Vulnerable Components	ICS hosts that allow remote display connections and the applications that the remote users are allowed to access
Ease of Detection	Easy
Attacker Awareness	High
Remediation Cost	Low
ICS Prevalence	High

From an ICS perspective, exposure depends on how and from where the connection is initiated. Connections from within the same local area network (LAN) can only be exploited by someone who has access to that network. Figures 1 and 2 illustrate these different scenarios. Figure 1 illustrates the scenario where the operator screens are displayed using a remote display protocol from a human-machine interface (HMI), or operator screen, server on the local ICS LAN. The exposure for this scenario is the supervisory control LAN because an attacker must gain access to this network before the remote display protocol can be exploited.

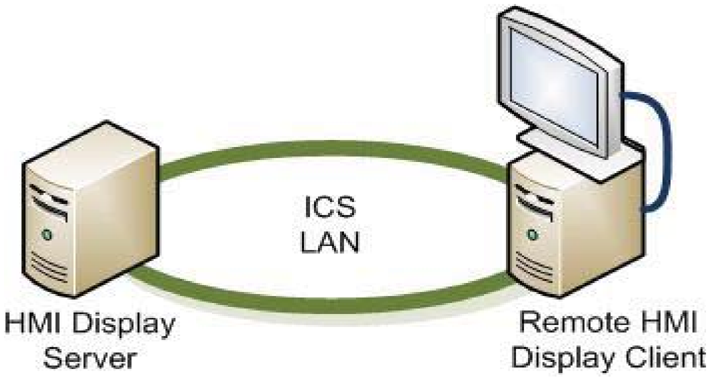

Figure 1. Operator screens are remotely displayed from HMI LAN.

Figure 2 illustrates how remote display connections from outside the ICS LAN create higher exposure to attack. Many sites utilize remote X-servers or other remote display protocols for remote supervisory control access (as well as other remote management capabilities.) This greatly increases the exposure of weaknesses in these protocols. It is also necessary to trust the client, which may not be under the site's management or control. An attacker may be able to gain access to supervisory control functionality by gaining access to the client host or intercepting the remote display connection. This is true of the scenario described by Figure1, but the attack.

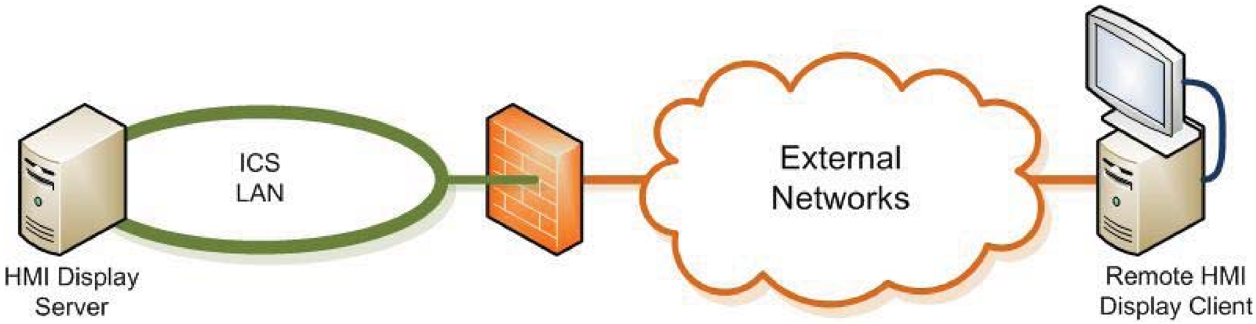

Figure 2. Operator screens are remotely displayed from a remote network.

The Access Complexity is "Low" because no additional access or specialized circumstances need to exist for the exploit to be successful.

Each of the Impact metrics is set to "Complete" because remote display access allows remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user. Assuming that availability is more important than usual for the targeted systems, and depending on the values for Collateral Damage Potential and Target Distribution, the environmental score could vary between 0.0 and 10.

CVSS v2 metrics for the use of remote display protocols on ICSs are summarized in Table 9 using the most common or critical values seen on ICS. See Section 4.3.1.2, "Use of Vulnerable Remote Display Protocols", for more information about this vulnerability.

Table 9. Generic CVSS score for the use of remote display protocols on ICSs.

Metric	Remote Connection	Same-LAN Connection
Base Metric	**Value**	**Value**
Access Vector	Network	Adjacent Network
Access Complexity	Low	Low
Authentication	None	None
Confidentiality Impact	Complete	Complete
Integrity Impact	Complete	Complete
Availability Impact	Complete	Complete
Base Score	10.0	8.3
Temporal Metric		
Exploitability	Functional Exploit Exists	Functional Exploit Exists
Remediation Level	Not Defined	Not Defined
Report Confidence	Confirmed	Confirmed
Temporal Score	9.5	7.9

Metric	Remote Connection	Same-LAN Connection
Environmental Metrics		
Collateral Damage Potential	High	High
Target Distribution	Not Defined	Not Defined
Availability Requirement	Medium	Medium
Integrity Requirement	High	High
Confidentiality Requirement	Medium	Medium
Environmental Score	9.8	9.0
Overall CVSS Score	**9.8**	**9.0**

2.2.3 Supervisory Control Access: Web HMI Vulnerabilities

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 Demilitarized Zones (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.

ICS assessments have also found poor authentication, poor session tracking, SQL injection, and cross-site scripting vulnerabilities that can allow unauthorized access to Web servers and applications. Improper authentication allows an attacker to impersonate another user's identity.

The use of vulnerable Web applications or servers for supervisory control functions poses the same risk to the physical system as remote display protocols because it allows unauthorized remote access to graphical supervisory control software, as well as any other functionality built into the Web application or allowed to the Web server. Table 10 summarizes the security relevant attributes of improper Web HMI authentication and its potential risk to the ICS.

Common Web vulnerability details are given in Section 4.4.3, "Web Vulnerabilities."

Table 10. Summary of ICS Web application security characteristics.

ICS Web Application Vulnerabilities	
Possible Consequences	User accounts compromised or user sessions hijacked
	Exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or allowing execution of arbitrary code.
ICS Impact	Unauthorized access to Web HMI, Web server or other Web applications and functionalities: possible unauthorized remote access to graphical supervisory control software, as well as any other functionality built into the Web application or allowed to the Web server.
Vulnerable Components	ICS Web applications and servers and/or ICS Web clients' and servers' hosts
Ease of Detection	Medium to High
Attacker Awareness	High
Remediation Cost	Low
ICS Prevalence	High

Web attack techniques are well known, even if a particular ICS's Web vulnerabilities are unknown. Therefore, Exploitability is set to "Proof of Concept."

A successful compromise of the Web HMI application or server may result in catastrophic physical or property damage and loss, or there may be a catastrophic loss of revenue or productivity. CVSS v2 metrics for the use of remote display protocols on ICSs are summarized in Table 11 using the most common or critical values seen on ICS.

Table 11. Generic CVSS score for ICS Web application vulnerabilities.

Metric	Values
Base Metric	
Access Vector	Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	10.0
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	9.5
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.8
Overall CVSS Score	**9.8**

2.2.4 ICS Host Access: Buffer Overflows in ICS Services

Buffer overflow vulnerabilities are the most common type of input validation weaknesses reported on ICS assessments. Buffer overflows are the result of programmer oversight. Most exploit code allows the attacker to create an interactive session and send commands with the privileges of the program with the buffer overflow. Any software with network parsing code that does not validate input values may be vulnerable to buffer overflow or other input validation attacks.

Remote code execution through buffer overflow attacks is a common attack method for gaining unauthorized access to hosts. ICS design requires that certain protocols are allowed through firewalls to support external data collection and sharing. These protocols and services should have top priority for vulnerability remediation activities. Vulnerabilities in services that are exposed to less-trusted networks have higher consequences because they may provide a path from the lower security zone to the higher security zone.

Table 12 summarizes the security relevant attributes of buffer overflow vulnerabilities and their potential risk to ICSs.

Table 12. Summary of buffer overflow characteristics.

Buffer Overflows in ICS Services	
Possible Consequences	Compromise of ICS hosts and applications. May allow DoS, code execution, data loss, or security bypass.
ICS Impact	Unauthorized access to ICS components, many times from a different security zone
Vulnerable Components	Services and other applications that parse or accept parsed network traffic
Ease of Detection	Easy
Attacker Awareness	High
Internet Attack Frequency	High
Remediation Cost	Low
Weakness Prevalence	Widespread
ICS Prevalence	Widespread

Known exploits do not currently exist for ICS service vulnerabilities, so Exploitability is “Unproven.” Some ICS vendors have released patches for at least some of the vulnerabilities discovered; so the Remediation Level metric varies between “Unavailable” and “Official-Fix.” Many buffer overflow vulnerabilities still exist in ICS network applications, and mitigation techniques can only reduce their exposure. Therefore, the Remediation Level metric for this generic “common” vulnerability is scored as “Unavailable” in this report.

Report Confidence is scored as “Confirmed” because all ICS vendors review and provide feedback before assessment reports are finalized. Security requirements are dependent on the host functionality and the nature of the ICS. Full compromise of any ICS host is likely to provide an attacker with access to system data or functionality. DoS of the vulnerable service or host has potential to cause an adverse effect. Security requirements are therefore rated as “Medium,” but will range between “Low” and “High” for individual systems and hosts.

A successful compromise of an ICS host may result in catastrophic physical or property damage and loss, or there may be a catastrophic loss of revenue or productivity.

Almost all hosts in an ICS environment are running custom ICS network applications. If they are exploitable, most of the ICS is at risk.

The CVSS v2 values for this generic vulnerability are listed in Table 13 below. Section 4.2.2, “Poor Code Quality,” contains additional guidance and references for avoiding and remediating this type of programming errors.

Table 13. CVSS score for ICS protocol server applications vulnerable to buffer overflow attacks.

Metric	Remote Code Execution Possible	DoS Impact Only
Base Metric	**Value**	**Value**
Access Vector	Network	Network
Access Complexity	Low	Low
Authentication	None	None
Confidentiality Impact	Complete	None
Integrity Impact	Complete	None
Availability Impact	Complete	Complete
Base Score	10	7.8
Temporal Metric		
Exploitability	Unproven	Unproven
Remediation Level	Unavailable	Unavailable
Report Confidence	Confirmed	Confirmed
Temporal Score	9.0	7.0
Environmental Metrics	**Metric Value**	**Metric Value**
Collateral Damage Potential	High	High
Target Distribution	Not Defined	Not Defined
Availability Requirement	Medium	Medium
Integrity Requirement	High	High
Confidentiality Requirement	Medium	Medium
Environmental Score	9.3	8.5
Total Score	**9.3**	**8.5**

2.2.5 Access to ICS Applications: Improper Authentication

Authentication is used to enforce access controls. Weak authentication allows access controls to be subverted. ICS security assessments have shown that access to process data and control functionality can be trivial because authentication is not required, or can be easily circumvented.

Many custom ICS applications implement authentication improperly, or not at all. A common error is known as client side authentication, where the client application authenticates users locally. Since the information needed to authenticate is stored on the client side, it is easy for a moderately skilled hacker to extract that information, or to modify the client to not require authentication.

This is a significant vulnerability because it allows unauthorized access to ICS functionality, possibly the HMI application. Table 14 summarizes the security relevant attributes of improper authentication.

Table 14. Improper authentication to ICS applications.

Improper ICS Application Authentication	
Possible Consequences	Security bypass
ICS Impact	Unauthorized access to ICS applications: Possible unauthorized remote access to supervisory control functionality.
Vulnerable Components	ICS hosts that allow remote display connections and the applications that the remote users are allowed to access
Ease of Detection	Moderate
Attacker Awareness	High
Remediation Cost	Low to High
Attack Frequency	Sometimes
ICS Prevalence	High

Each of the Impact metrics is set to “Complete.” The actual impact depends on the application. CVSS v2 metrics for improper authentication to ICS applications are summarized in Table 15 using the most common or critical values seen on ICS. See Section 4.5, “ICS Application Authentication Vulnerabilities,” for additional information.

Table 15. Generic CVSS score for improper authentication to ICS applications.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	Low
Authentication	Single
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	9.0
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	8.6
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	High
Environmental Score	9.3
Overall CVSS Score	**9.3**

2.2.6 Access to ICS Functionality: Improper Access Control (Authorization)

Access control mechanisms determine which network, host, and ICS resources and services can be accessed, by whom, and under what conditions. The impact of a compromised account, application, or host depends on the privileges it has been granted.

Once an attacker has gained access to a host, compartmentalization and access controls can contain them. 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.

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.

Unnecessary functionality in ICS protocols, services, and applications increases the impact from compromise as well.

This is a significant vulnerability because it allows unauthorized access to ICS networks, hosts, and functionality. Table 16 summarizes the security relevant attributes of improper access control.

Table 16. Improper Access Control.

Improper Access Control	
Possible Consequences	Security bypass: including information leaks, DoS, and arbitrary code execution
ICS Impact	Unauthorized access to ICS functionality
Vulnerable Components	ICS networks, hosts and functionality
Ease of Detection	Moderate
Attacker Awareness	High
Remediation Cost	Low to Medium
Attack Frequency	Often
Weakness Prevalence	High
ICS Prevalence	Widespread

CVSS v2 metrics for least user privileges violations on ICSs are summarized in Table 17 using the most common or critical values seen on ICS. See Section 4.6, “Authorization Vulnerabilities,” for additional information.

Table 17. Generic CVSS score for least user privileges violations.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	Low
Authentication	Single
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	9.0
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	8.1
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.1
Overall CVSS Score	**9.1**

2.2.7 ICS Credentials Gathering: Use of Standard IT Protocols with Clear-text Authentication

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, active unused or obsolete services still exist in many ICSs. Clear-text authentication credentials can be sniffed during transmission and used by an attacker to authenticate to the system. If an attacker is able to capture a username and password, he is able to legitimately log onto the system with that user's privileges. For this reason, plain-text remote login services should be replaced with encrypted services such as Secure Shell (SSH).

The use of unsecure protocols and services to connect to the ICS hosts creates a high-risk access path into the system. This is a significant vulnerability because it allows unauthorized remote access to ICS hosts and the functionality allowed to the remote user. Table 18 summarizes the security relevant attributes of the use of clear-text authentication protocols and their potential risk to ICSs. See Section 4.3.1, "IT Protocols Vulnerable to Spoofing and MitM Attacks," for more information.

Table 18. Summary of clear-text authentication protocols' security characteristics.

Use of Standard IT Protocols with Clear-text Authentication	
Possible Consequences	Lack of identity proofing
ICS Impact	Unauthorized access to ICS components: Possible unauthorized remote access to hosts with privileges to any functionality granted to the compromised remote user.
Vulnerable Components	ICS hosts running clear-text authentication protocol services
Ease of Detection	Easy
Attacker Awareness	High
Remediation Cost	Low
Weakness Prevalence	High
ICS Prevalence	High

CVSS v2 metrics for the use of clear-text authentication protocols on ICSs are summarized in Table 19 using the most common or critical values seen on ICS. The actual impact depends on the privileges of the account whose credentials were stolen.

Table 19. Generic CVSS score for the use of clear-text authentication protocols on ICSs.

Metric	Value
Base Metric	
Access Vector	Adjacent Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	8.3
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	7.9
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	High
Environmental Score	9.0
Overall CVSS Score	**9.0**

2.2.8 ICS Credentials Gathering: Unprotected Transport of ICS Application Credentials

The difference between this vulnerability and use of clear-text authentication protocols in Section 2.2.7, "ICS Credentials Gathering: Use of Standard IT Protocols with Clear-text Authentication," above is how well known the protocols are and what they are used for. Both vulnerabilities are due to the unprotected transportation of credentials. In this case, if the attack is able to capture ICS application credentials, he can then log into the ICS application and gain access to the associated ICS functionality. This may include controlling the physical process, altering data, or reconfiguring ICS devices.

This is a significant vulnerability because it allows unauthorized remote access to ICS functionality, possibly the HMI application (control functionality). Table 20 summarizes the security relevant attributes of transmitting ICS application credentials across the network in clear text.

Table 20. Unprotected transport of ICS application credentials summary.

Unprotected Transport of ICS Application Credentials	
Possible Consequences	Lack of identity proofing
ICS Impact	Unauthorized access to ICS applications: Possible unauthorized remote access to supervisory control functionality.
Vulnerable Components	ICS applications
Ease of Detection	Easy
Attacker Awareness	High
Remediation Cost	Medium
Weakness Prevalence	High
ICS Prevalence	Common

CVSS v2 metrics for unprotected transport of ICS application credentials are summarized in Table 21 using the most common or critical values seen on ICS. See Section 4.3.2, “ICS Protocols Vulnerable to Spoofing and MitM Attacks,” for more information.

Table 21. Generic CVSS score for unprotected transport of ICS application credentials.

Metric	Value
Base Metric	
Access Vector	Adjacent Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	8.3
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	7.9
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	High
Environmental Score	9.0
Overall CVSS Score	**9.0**

2.2.9 Supervisory Control Access: ICS Data and Command Message Manipulation and Injection

ICS network protocols, including those used to send control commands and status data, can be altered, replayed, or spoofed because they lack sufficient access control and integrity checking mechanisms. This vulnerability requires minimal skills to intercept or create the network messages. The ability to intelligently interpret and manipulate process status depends on the level of protocol and process reverse engineering performed. ICS and network programming skills are needed for this attack. The ICS network design and implementation determines the exposure of control protocol vulnerabilities. This vulnerability is exposed to anyone who has gained network access to the supervisory control network, or a network that is allowed access to control equipment.

ICS network protocol vulnerabilities can pose the same risk to the physical system as remote display protocols and vulnerable Web HMI applications because it allows supervisory control abilities. Table 22 summarizes the security relevant attributes of ICS network protocol channel vulnerabilities and their potential risks to the ICS.

Table 22. Summary of ICS network protocols' security characteristics.

ICS Data and Command Message Manipulation and Injection	
Possible Consequences	Data exposure, manipulation, or loss
	Exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or allowing execution of arbitrary code.
ICS Impact	Unauthorized access to network level supervisory control functionalities
Vulnerable Components	ICS communication channels, potentially between security zones
Ease of Detection	Medium to High
Attacker Awareness	High
Remediation Cost	High
ICS Prevalence	Widespread

MitM attack tools exist, and protocol analyzers are available for some control protocols, so Exploitability is "proof-of-concept." Network traffic can be encrypted in some cases; so the Remediation Level metric varies between "Unavailable" and "Temporary-Fix." The Remediation Level metric for this generic "common" vulnerability is scored as "Workaround" in this report. Report Confidence is scored as "Confirmed" because all ICS vendors review and provide feedback before assessment reports are finalized.

Security requirements are dependent on the protocol functionality and the nature of the ICS. Interception of ICS protocol traffic provides access to system data or functionality. DoS of the protocol traffic has potential to cause an adverse effect. Therefore, security requirements are rated as “Medium,” but will range between “Low” and “High” for individual systems and hosts.

Almost all hosts in an ICS environment are communicating using ICS network protocols. If they are vulnerable to MitM attack or spoofing, the ICS is at risk. The Environmental metric values should be modified for individual systems.

The CVSS v2 values for this generic vulnerability are listed in Table 23. See Section 4.3.2, “ICS Protocols Vulnerable to Spoofing and MitM Attacks,” for more information on this vulnerability.

Table 23. Generic CVSS score for ICS protocol MitM vulnerabilities.

Metric	**Value**
Base Metric	
Access Vector	Adjacent Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	8.3
Temporal Metric	
Exploitability	Proof-of-concept
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	7.5
Environmental Metrics	**Metric Value**
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	8.8
Overall CVSS Score	**8.8**

2.2.10 Data Historian Access: SQL Injection

A Historian server is used for data archiving and analysis and is typically an integral part of an ICS. It is usually located in a DMZ or on the corporate network. Threats to the historian include compromise of the historian host and data corruption. ICS historians typically utilize a common SQL server as its backend. The historical data is often made available for viewing via a custom Web interface or application.

The Historian client applications are high-risk components because they are often accessible from the corporate environment and can provide an attacker with a point of entry into the ICS network. Additionally, an attacker may gain access to unauthorized information, which in some cases can be used to cause economic damage.

Historian database applications use SQL queries to retrieve information. An SQL injection vulnerability is caused when an application incorrectly or inadequately filters user input. If an attacker inserts literal escape characters into a database query, they may gain arbitrary read or write access to the database. Attackers could alter the logic of SQL queries in security controls (such as authentication) to bypass security.

According to the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[3] report, SQL injection is the second-most widespread and critical programming error. Table 24 summarizes the security relevant attributes of SQL injection vulnerabilities and their potential risk to ICSs.

Table 24. Summary of SQL injection characteristics.

SQL Injection	
Possible Consequences	Data loss: Unauthorized read or write access to the database Security bypass: DoS of the database service or unauthorized access to the associated host
ICS Impact	Historical data exposure, loss or manipulation Attack path into the ICS network
Vulnerable Components	Historian and other databases and hosts Database-backed Web applications
Ease of Detection	Easy
Attacker Awareness	High
Internet Attack Frequency	Often
Remediation Cost	Low
Weakness Prevalence	High
ICS Prevalence	Common

If the Historian and other ICS databases hold sensitive data, loss of confidentiality will have a high impact. Historian data may also be altered or deleted with a SQL injection attack. This may include authentication and authorization data if it is stored in a database.

A successful compromise of an ICS database may result in a significant loss of revenue or productivity. CVSS v2 metrics for Data Historian SQL injection are summarized in Table 25 using the most common or critical values seen on an ICS. See Section 4.4.2, "Database Vulnerabilities," for general database security recommendations and references.

Table 25. Generic CVSS score for Data Historian SQL injection.

Metric	**Value**
Base Metric	
Access Vector	Network
Access Complexity	Low
Authentication	None
Confidentiality Impact	Partial
Integrity Impact	Partial
Availability Impact	Partial
Base Score	7.5
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	6.8
Environmental Metrics	
Collateral Damage Potential	Medium-High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	High
Environmental Score	8.6
Overall CVSS Score	**8.6**

3. COMMON ICS VULNERABILITY CATEGORIZATION

Common ICS vulnerabilities can be categorized differently based on how the problem is being viewed. Vulnerabilities have been categorized by NSTB assessment targets and component functionality in this section. Detailed descriptions of the vulnerabilities are grouped by the general assessment targets in Section 4. NSTB assessments prioritize attack targets based on the likelihood and impact of compromise. More information about the NSTB assessment methodology is in Appendix A.

Individual ICS components can be evaluated by the risk they contribute to the overall security of the system. The components can be categorized based on their ICS functionality. The ISA reference model[5] describes an ICS as a series of logical levels based on functionality. It is useful to categorize vulnerabilities by this established frame of reference.

When viewing ICS assessment results, it is important to understand the differences between ICS and more common computer and network security priorities and obstacles. ICSs cannot be evaluated using generic security assessment techniques or protected using generic security solutions.

3.1 ICS Security Issues

Cyber security is about securing cyber resources to prevent actions that violate security goals for data confidentiality, integrity, and availability requirements. Cyber security measures in ICS protect the system data and functionality from unauthorized access, use, disclosure, disruption, modification, or destruction.

In general, the most significant difference between the ICS and corporate IT domains is the high availability requirement for monitoring and control functionalities, as illustrated in Figure 3.

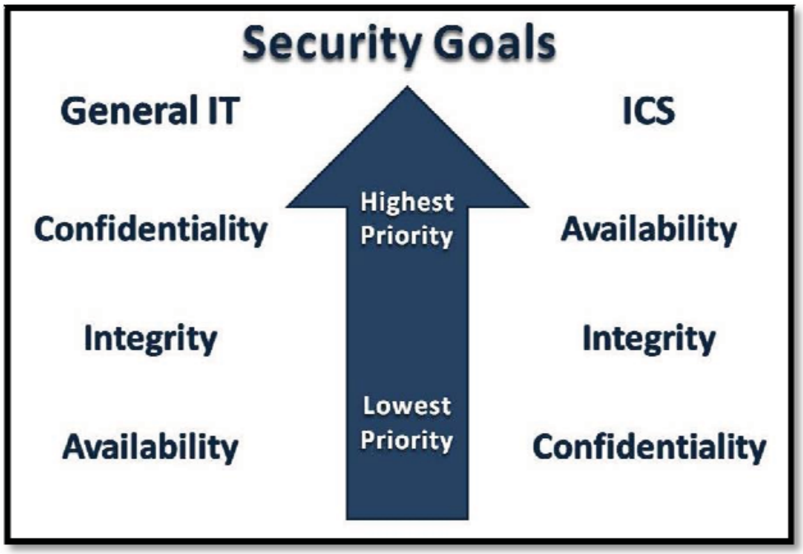

Figure 3. Generic IT security goals versus ICS security goals.

Cyber security is the protection of information transmitted and stored over a computer network. The objectives of cyber security are to:

- Protect confidentiality of private information
- Ensure availability of information to authorized users on a timely basis
- Protect the integrity of information (i.e., accuracy, reliability, and validity).

These objectives can be prioritized differently depending on the physical system under control and the functionality provided by the individual ICS component.

Security policy defines security goals and measures that must be incorporated and enforced through access control mechanisms in design, code, security features, and the host and network environment.

A cyber security vulnerability is a weakness in a computing system that can result in harm to the system or its operation, especially when this weakness is exploited by a hostile person or organization or is present in conjunction with particular events or circumstances. These weaknesses are not usually a problem unless exposed to attack.

There are many different ways the physical process and ICS can be threatened. To fully protect the information and physical system, each component of the ICS must have its own protection mechanisms. The build-up, layering and overlapping of security measures is called defense in depth. The strength of any system is no greater than its weakest link. Using a defense in depth strategy, should one defensive measure fail there are other defensive measures in place that continue to provide protection. All applications, hosts, and networks need to be locked down as much as possible to minimize the chance and potential consequences of compromise.

Defensive measures minimize vulnerability exposure and opportunity. Restricting access and permissions of processes and users minimizes potential damages from successful attacks. Good design can also raise the potential costs of attacking in terms of time and equipment needed to penetrate. Hardening and protective measures should be designed into all critical infrastructure ICSs.

ICS developers and owners can reduce their attack surface by restricting access to functionality, hosts and networks. ICS information and functionality must be restricted to people with authorization to access them. Authorization is also required for applications that perform ICS functions and the computers that host them. Mechanisms must be in place to control access to ICS hosts and functionality.

Access control mechanisms rely on proper identification and authentication (i.e., user name and password). Identification is an assertion of who someone is or what something is. Authentication is the act of verifying a claim of identity.

Access control mechanisms determine system resources a person, program, or computer is allowed to access and which actions are allowed (run, view, create, delete, or change) after successful identification and authentication. Access control mechanism configuration should enforce policies that describe what information and computing services can be accessed, by whom, and under what conditions.

Authentication credentials must be protected from unauthorized access. Encryption can protect confidentiality of authentication credentials. However, cryptography can introduce security problems when not implemented correctly. The keys used for encryption and decryption must be protected.

3.2 Vulnerabilities by NSTB Assessment Target Categories

Table 26 groups common ICS vulnerability types according to the attack patterns that could allow access to core ICS functionality. This view facilitates understanding of the attack paths and consequences of the vulnerabilities. The distribution of NSTB findings per category is shown in Figure 4.

Table 26. Summary of the most common security weaknesses identified in ICS assessments.

Vulnerability Category: Assessment Target	Source of Vulnerability
Published Vulnerabilities: Most Likely Attack Paths	Unpatched or old versions of third-party applications incorporated into ICS products
	Unpatched OS on ICS Hosts
Potential Vulnerabilities: Potential 0-day and Unpatched Vulnerabilities	Excessive ICS host exposure through unnecessary services
	Poor ICS Code Quality
Communication Channel Vulnerabilities: Unauthorized Access to ICS Functionality through Vulnerable Communication Channels	Remote Access Protocols Vulnerable to Spoofing and MitM Attacks
	ICS Protocols Vulnerable to Spoofing and MitM Attacks
Communication Endpoint Vulnerabilities: Unauthorized Access to or DoS of ICS Hosts and Applications	Vulnerable server applications for ICS communication and data transfer protocols
	Database Vulnerabilities
	Web Vulnerabilities
ICS Application Authentication Vulnerabilities: Access to ICS Applications by Exploiting Authentication Mechanisms	Authentication Bypass Issues
	Credentials Management
ICS Host Environment Vulnerabilities: Ability to Cause Harm from an ICS Account	Failure to Secure Host Environment
ICS Network Vulnerabilities: Access to ICS Hosts and Functionality through Available Network Paths	Poor Network Design
	Weak Firewall Rules
	Failure to Secure Network Devices
	Poor Network Monitoring

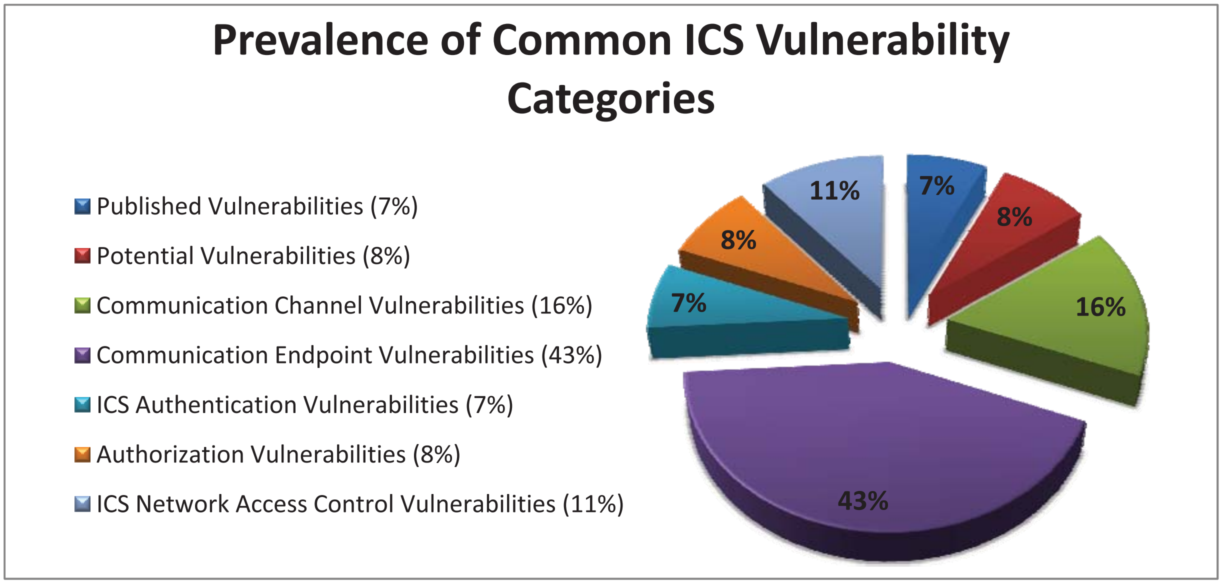

Figure 4. Percentage of NSTB assessment findings per vulnerability category.

3.3 Vulnerabilities by Component Functionality

ICSs are made up of process equipment, process control hardware, network devices, and computers. Vulnerabilities in network devices and protocols, 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 distribution of NSTB assessment vulnerabilities that were found in each ICS component category are shown in Figure 5 below. NSTB assessments have focused on the ICS products to understand the vulnerabilities they are most affected by, and how their design and operational requirements affect host and network security.

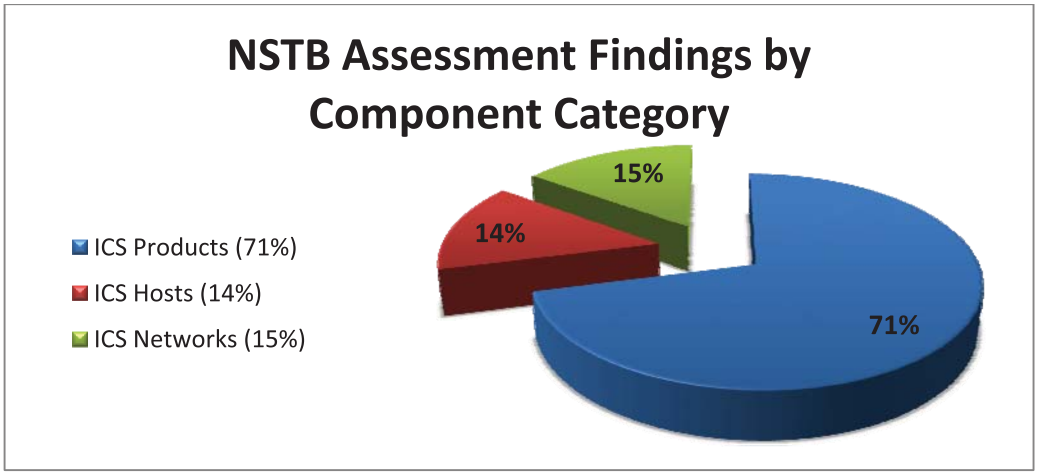

Figure 5. Percentage of NSTB assessment findings per ICS component category.

The distribution of assessment findings in ICS components can be broken down based on functionality, as shown in Figure 6 below. This chart illustrates the high number of vulnerabilities in ICS server applications (services). The distribution is slightly skewed by the ICS products that were selected for evaluation. Supervisory control protocols were available for assessment on almost all NSTB assessments. ICS protocols that are used for external communications are slightly skewed based on their availability for assessment. The Inter-Control Center Communications Protocol (ICCP) was selected for an in-depth assessment,[4] while “basic” or “local” control protocols like Distributed Network Protocol Version 3 (DNP3) were not configured on every assessment system.

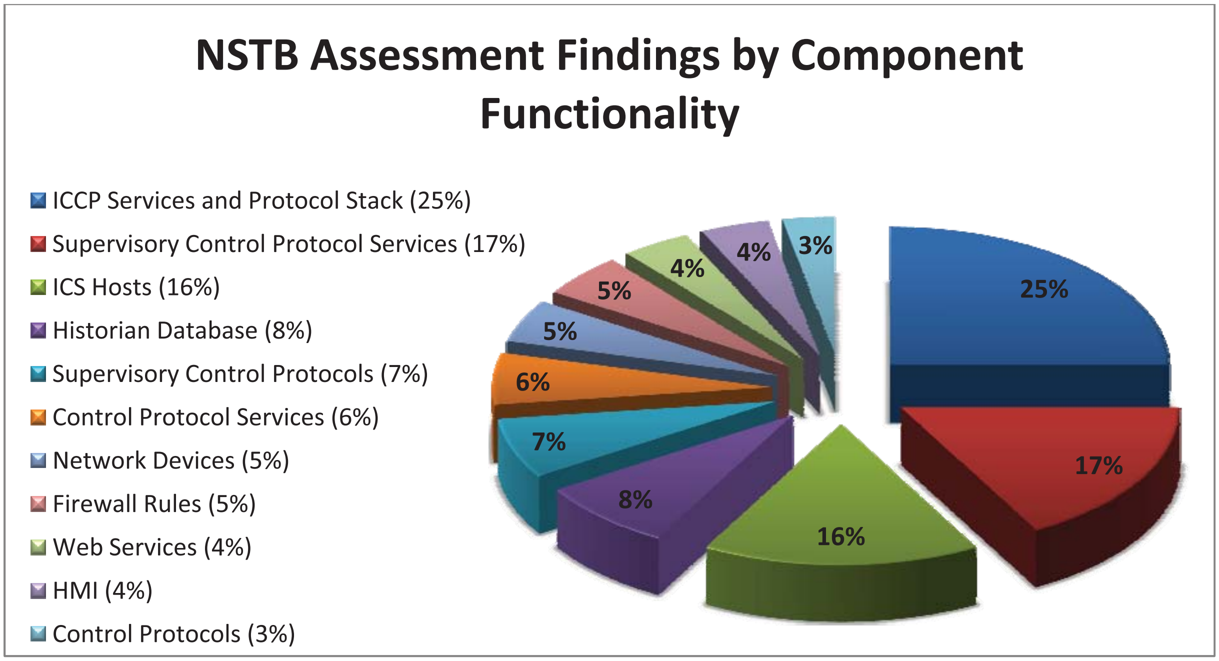

Figure 6. NSTB assessment findings by component functionality.

Component functionality can be grouped using the ISA99 reference model.[5] The NSTB focus on core ICS functionality is evident in Figure 7. The largest portion of products and functionalities tested fit into the supervisory control and operations management categories. The ISA99 levels are shown in Figure 8 for reference.

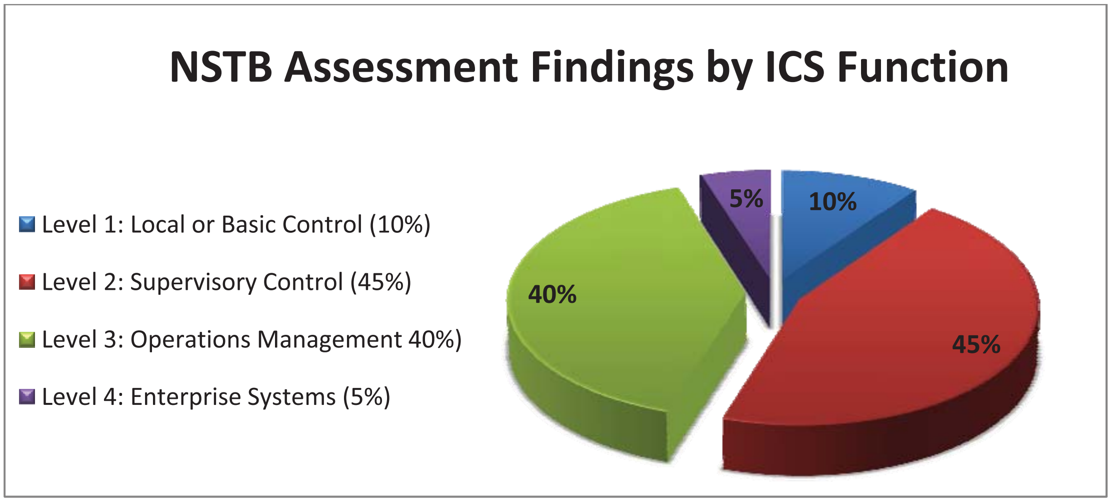

Figure 7. NSTB assessment findings by functionality.

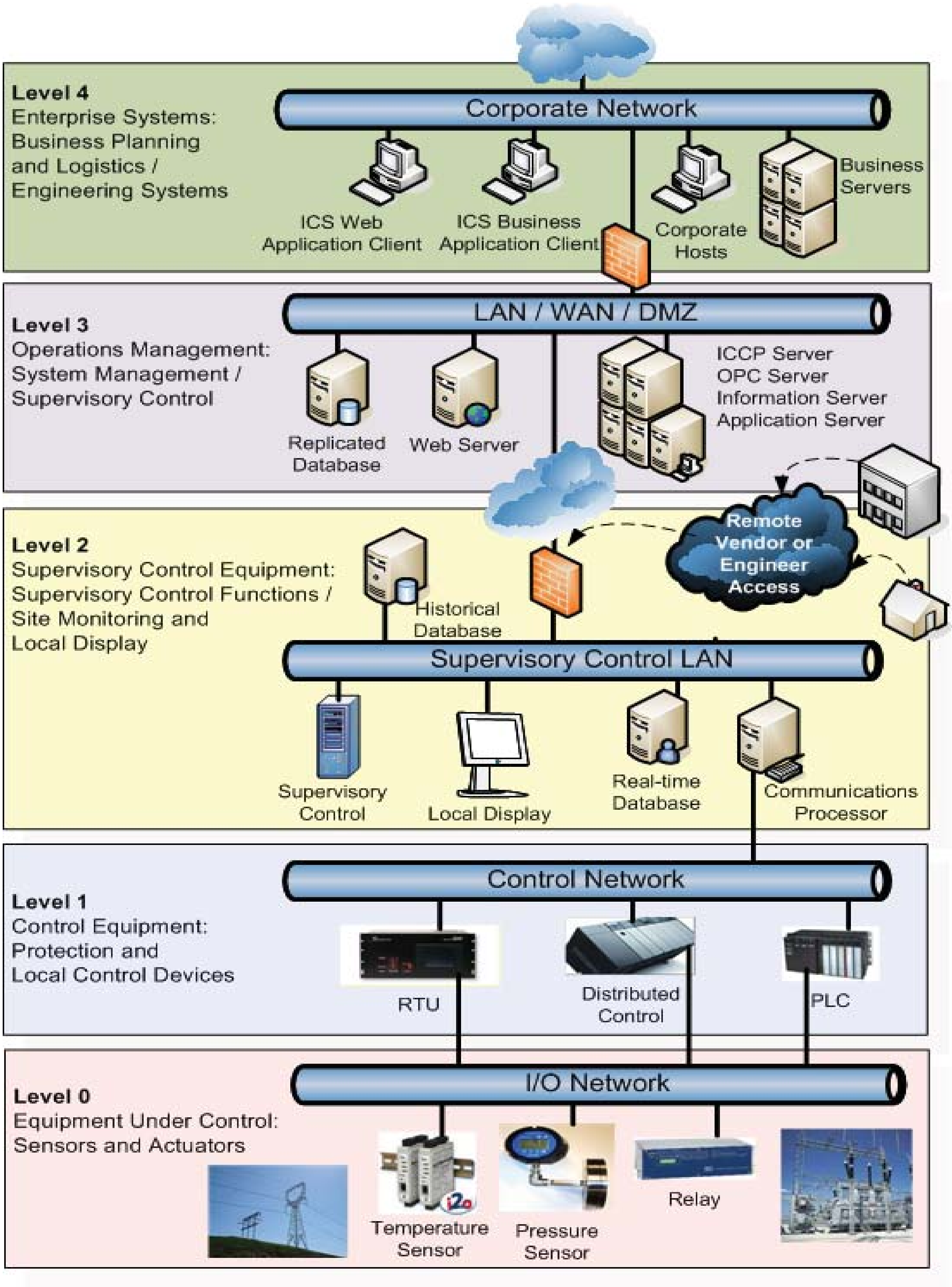

Figure 8. ISA functional reference model.

4. DETAILED NSTB ASSESSMENT RESULTS

The NSTB assessment program priority is identification and analysis of vulnerabilities in ICS that allow unauthorized access to data, functionality, or affect operations. Consequences include:

- Unauthorized user can gain access to ICS hosts, applications, and data
- ICS data and command spoofing and manipulation
- DoS of ICS functionality (communications).

Communication channels and network services are exposed to network attack. Vulnerabilities in ICS systems have become significantly more exposed to attack as they are connected to the Intranet and Internet. Communication channels are of highest interest in ICS assessments because they are often used between network security zones and may possess access rights or functionality to manipulate the ICS. Weak access controls may even allow access to ICS components without impersonating an authorized communication partner.

Network traffic is exposed to MitM attacks that can be used to gather information, alter messages, or drop messages. Credentials sent across a network in clear text or using a broken encryption algorithm can be intercepted, stolen and used by an attacker to gain access to ICS hosts and applications.

Network services at communication endpoints are listening for messages to accept, and are exposed to attacks that exploit input and output validation vulnerabilities. Assessments target ICS programming errors in network parsing code that do not properly validate or "sanity check" input values. These vulnerabilities have the potential to allow access to the associated host through remote code execution, privilege escalation, and authentication bypass. Access to ICS functionality can be gained through the compromised host with the privileges of the compromised account. Additional risks to ICS operations include data loss and DoS of ICS functionality and communications.

Published vulnerabilities in common IT products used by ICSs create opportunity for successful system attack. These vulnerabilities are assessed through use of tools available commercially and in the public domain.

The most common ICS vulnerabilities found by the NSTB assessment program are described in the following sections.

4.1 Published Vulnerabilities

Known vulnerabilities in common IT products installed on ICS hosts create vulnerabilities with a high probability of being attacked. These vulnerabilities may provide an attack path into the system. The software is well known, and available exploit code makes them an easy target.

Figure 9 shows the proportions of each type of vulnerable software.

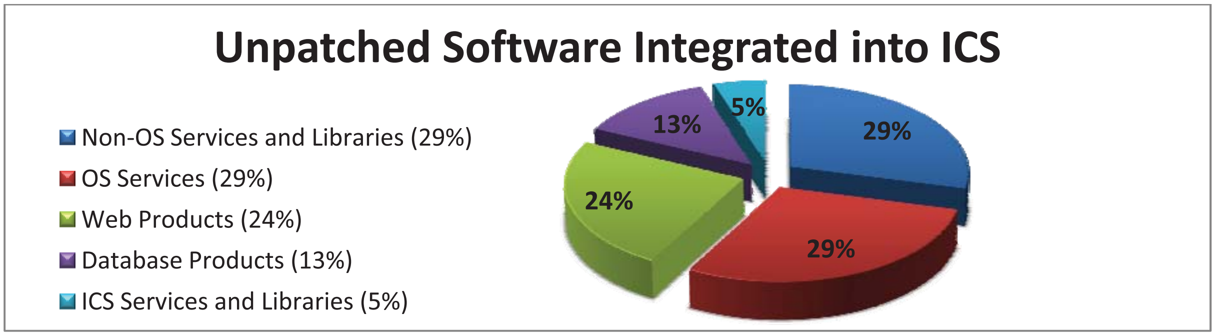

Figure 9. Unpatched components integrated into ICS.

4.1.1 Third-Party Applications Incorporated into ICS Software

ICS software generally uses third-party applications such as common Web servers, database servers, remote access services, and encryption services. Many out-of-date and vulnerable third-party software applications and services are still being incorporated into new ICS software. This indicates that new ICSs are being installed with vulnerable software and that ICS vendors are not supporting third-party patch management for their software.

Unpatched applications and services are probably the biggest threat vector for unauthorized access to the ICS. 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. NSTB assessments have discovered published vulnerabilities in unpatched or old versions of applications, services, and libraries integrated into the latest releases of ICS products.

The reason that this is such a big problem is that some of these products cannot be patched by the ICS owners. Many times, the Application Programming Interface (API) changes in new versions of applications, services, and libraries, so a product cannot be upgraded without changing the ICS code that interfaces with it.

4.1.1.1 Non-OS Services and Libraries

Overall, OS patch management support has improved to the point where OS patching is trivial for most situations. Many ICS vendors now provide timely OS patch test results. Application patching does not see the same attention, however.

Applications, services, and libraries not included as part of the OS tend not to get patched on many ICSs. NSTB assessments have not only discovered old and unpatched versions of third-party products on production ICSs, but integrated into new ICS product versions as well. Published vulnerabilities in well-known applications and services create the biggest security risks to ICSs. Of all the vulnerabilities in an ICS, these vulnerabilities are most likely to be exploited because all attackers should be aware of them.

Non-OS services and libraries are the most neglected because they are inconspicuous. Many developers and administrators are not aware of them. Statically linked libraries need to be independently kept up-to-date if they are different from the libraries associated with the operating system.

4.1.2 OS Patch Management

Operating system patches repair vulnerabilities in the operating system that could allow an attacker to exploit the computer. The importance of system security to keep operating system patches up-to-date cannot be over emphasized. However, patching ICS machines can present unique challenges. Among the factors to consider are functionality, security, and timeliness.

OS patching has been improving as more and more common IT products are being integrated into ICS. OS patch test results are now provided by some ICS vendors for their newer releases.

4.1.3 Summary of Patch Management Recommendations

The vendor bears responsibility to upgrade and patch third-party (and OS) software that is integrated into their ICS products before it is deployed. ICS owners may not be able to make the ICS code changes required to integrate with the new versions.

Currently, most ICS venders have poor methods of notifying customers about potential security problems and patches. Experience has shown that patches generated as the result of previous security assessments have been slow in being deployed with many end users unaware about the existence of the patches. ICS vendors should create and maintain security mailing lists and also test the procedures needed to notify the end users about security problems.

ICS vendors should create a procedure and have personnel assigned to keep up with the updates and integrate the updates into their products. Vendors should test and approve OS patches, along with all other third-party software incorporated. Products and services used by the ICS 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.

Patches must be tested for adverse affects on system functionality. The system vendor should test operating system patches for compatibility with their system and supply the testing results to users. These results should be made available as soon as possible after the patch release, to limit the length of time the user's system is vulnerable to the operating system exploit.

If vendor support is not available, system owners should test their own patches. This should be done even if the vendor has approved them. Patches should always be tested on a backup system first, before being implemented on an operational system. This testing period should be long enough for any side effects to become apparent.

4.1.3.1 Patch Management References

More detailed guidance on production ICS patch management can be found in the DHS *Recommended Practice for Patch Management of Control Systems,*[7] and the National Institute of Standards and Technology (NIST) *Guide to Industrial Control Systems (ICS) Security.*[8]

4.2 Potential Vulnerabilities

The risk of exploitation increases with the number of applications and services installed on the system. The attack surface is all possible avenues of attacking a system. All open ports and services that can potentially be exploited create the attack surface. New vulnerabilities are constantly discovered in applications, so minimizing the number of installed services and applications minimizes potential vulnerabilities.

Poor code quality and unsecure coding practices create bugs that can potentially be exploited for malicious purposes. These bugs also make ICSs fragile, creating an environment where administrators are afraid to make changes after they are finally able to get everything up and working correctly. Potential vulnerabilities in ICS applications and services can be minimized by following secure coding practices.

4.2.1 Failure to Minimize Services

Open ports and services that are not necessary provide a potential foothold or path for an attacker. The attacker can remotely connect to services listening on ports allowed through a firewall. Once an attacker has a foothold onto a protected network, he can access all services listening on the local network hosts. The more services listening on the ICS hosts, the more exposed it is to attack.

Services or applications running on a system open up different network ports to be able to communicate to the outside world. Each open port can possibly be exploited by an attacker and 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 open, the greater the risk because there are that many more services that may have vulnerabilities.

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 attackers in the community to exploit computers at will, with no patches available to stop them. It follows that having more installed applications translates to an increased risk for the computers running them.

4.2.1.1 Vendor Identification of Necessary Ports and Services

One item of critical importance is ICS vendors' understanding of the ports and services needed to support their systems. NSTB ICS product assessments evaluate open ports and running services on hosts configured by the ICS vendors.

Many times, vendor-published ports and services do not match what was actually installed and running on the system hosts. Another problem is the excessive number of ports that must be opened for some ICS protocols. For example, one range of identified ports required over 21,000 ports to be opened for a single service.

4.2.1.2 Services Recommendations

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.

For each ICS component, ICS vendors should document the necessary services along with the associated port ranges and which components are allowed to initiate a connection to that component. They should then carefully analyze these results, identify and disable all OS or third-party application services that are not explicitly needed for the ICS to operate, and identify all dynamic port services with the respective port ranges. This should be part of the deliverable for every product to help the end user create a network architecture that protects the inner core of their ICS while providing a fully functioning system.

Because required ports and services have been found to disagree with delivered systems, ICS owners should validate the necessity of services installed on new systems before they are deployed. Traffic monitoring between system components during all phases of acceptance testing can be used to identify required communications. Ideally this testing and verification of required ports and services would be laid out by the vendor during system design, and then tested and verified by the owner/operator before during design and testing. This technique eliminates communication not required by a particular ICS configuration. This can create a more secure system, but owners must be sure to exercise all potential functionality so that it will be available when needed.

This same process can be used to minimize risk on an operational ICS, but must be performed with more care. The service can first be removed on a backup or development system to insulate the primary system from any potential damage. Before stopping any services or programs on an operational system, ICS administrators should ask the vendor to confirm that the service is not needed for system functionality. The administrator can also create an IDS rule that watches for the use of installed services until there is sufficient confidence that a service is not necessary. IDS and system logs may also inform administrators when requested services are not available.

4.2.2 Poor Code Quality

In general, ICS software tends to suffer from poor code quality, which leads to stability problems and vulnerabilities. Nearly all ICS code level vulnerabilities were the result of unsecure coding practices and inadequate testing. Secure programming standards and guidelines can be followed to prevent these errors. Automated source code analysis tools can be used to identify existing vulnerabilities for remediation. ICS vendors need to thoroughly test all ICS features to validate ICS stability and security levels before release. ICS customers should require that products are tested by a third party and vulnerabilities are remediated before acceptance of an ICS product.

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 increases the likelihood of unsecure coding practices because they were developed as stand-alone systems with only reliability and efficiency as requirements. However, new ICS applications tend to suffer from many of the same weaknesses in the lack of secure coding principles.

Poor code quality leads to vulnerabilities and bugs in the code that not only make it vulnerable to attack, but also fragile and unstable.

Many secure coding resources are available for all application types and languages. The CWE list[6] provides information about all types of software weaknesses including the most common ICS programming errors listed in Table 27.

Table 27. Most common programming errors found in ICS code.

Weakness Classification	Vulnerability Type
CWE-19: Data Handling	CWE-228: Improper Handling of Syntactically Invalid Structure
	CWE-229: Improper Handling of Values
	CWE-230: Improper Handling of Missing Values
	CWE-20: Improper Input Validation
	CWE-116: Improper Encoding or Escaping of Output
	CWE-195: Signed to Unsigned Conversion Error
	CWE-198: Use of Incorrect Byte Ordering
CWE-119: Failure to Constrain Operations within the Bounds of a Memory Buffer	CWE-120: Buffer Copy without Checking Size of Input ("Classic Buffer Overflow")
	CWE-121: Stack-based Buffer Overflow
	CWE-122: Heap-based Buffer Overflow
	CWE-125: Out-of-bounds Read
	CWE-129: Improper Validation of Array Index
	CWE-131: Incorrect Calculation of Buffer Size
	CWE-170: Improper Null Termination
	CWE-190: Integer Overflow or Wraparound
	CWE-680: Integer Overflow to Buffer Overflow
CWE-398: Indicator of Poor Code Quality	CWE-454: External Initialization of Trusted Variables or Data Stores
	CWE-456: Missing Initialization
	CWE-457: Use of Uninitialized Variable
	CWE-476: NULL Pointer Dereference
	CWE-400: Uncontrolled Resource Consumption ("Resource Exhaustion")
	CWE-252: Unchecked Return Value
	CWE-690: Unchecked Return Value to NULL Pointer Dereference
	CWE-772: Missing Release of Resource after Effective Lifetime
CWE-442: Web Problems	CWE-22: Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal")
	CWE-79: Failure to Preserve Web Page Structure ("Cross-site Scripting")
	CWE-89: Failure to Preserve SQL Query Structure ("SQL Injection")
CWE-703: Failure to Handle Exceptional Conditions	CWE-431: Missing Handler
	CWE-248: Uncaught Exception
	CWE-755: Improper Handling of Exceptional Conditions
	CWE-390: Detection of Error Condition Without Action

4.2.2.1 Use of Potentially Dangerous Functions

Otherwise known as unsafe function calls, the application calls a potentially dangerous function that could introduce a vulnerability if used incorrectly, but the function also can be used safely. The problem with using unsafe functions is that the developer is responsible for validating input. The number of publicly announced buffer overflows and other malformed input vulnerabilities is evidence that relying on the developer to implement this validation is 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 it may introduce a buffer overflow vulnerability. For example, if the input to `strcpy` can in any way be influenced, a potential exists that an attacker will 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 discover that more than 1,024 characters cause problems. The developer can test to make sure nothing larger than the memory buffer he created is sent to a dangerous function, i.e. `strcpy`. Safe functions, i.e. `strncpy,` eliminate this risk by requiring the buffer size to be specified.

Most of the ICS vulnerabilities that allow remote code execution (unauthorized access) are the result of the use of potentially dangerous functions. The assessment team discovered many potential vulnerabilities during code audits and fuzz testing, but it was not prudent or feasible to validate whether every unsafe function call could be reached, or that every crash could be exploited for remote control. The assessment convention has been to demonstrate the existence and impacts of one or more of the coding error instances identified, and then recommend that all unsafe function calls be remediated. As a result, these vulnerabilities are under-represented.

4.2.2.2 Secure Coding Recommendations

ICS applications tend to suffer from poor code quality. Vendors and asset owners who write custom applications should train developers in secure coding practices. Software development procedures should include thorough code reviews 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 and acceptance processes.

Specifically, ICS developers can use static analysis tools to identify and replace dangerous functions. Usages of the unsafe C functions should be replaced with safer alternatives, starting with services that are exposed to less-trusted networks and working inward. The security issues of using C functions have been known since the 1990s and there is no excuse for using them in new products.

4.2.2.3 Secure Coding References

ICS developers can reference the CWE site for additional secure development information and references on the security weaknesses identified in this section as well as other software security weaknesses covered in the Developer's View.[6] ICS administrators can utilize this information for better understanding of the weaknesses to which ICS software is prone. They can then work with their vendors to mitigate the associated risks as much as possible in existing systems, or create procurement requirements that enforce security standards.

All weaknesses listed on the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[3] have been found during ICS assessments. ICS developers should refer to this list of more detailed weaknesses and the associated *2010 CWE/SANS Top 25: Monster Mitigations*[9] for guidance in preventing the most dangerous programming errors.

Secure coding guides and standards are available in a wide range of languages and software types. Some examples include:

- *CERT Secure Coding Standards*[10]
- SAFECode Secure Coding Standards[11, 12]
- *20 Critical Security Controls: Critical Control 7: Application Software Security* [13]

4.3 Communication Channel Vulnerabilities

Network traffic is exposed to MitM attacks that can be used to gather information, alter messages, or drop messages.

Assessments focus on vulnerabilities that allow:

- ICS credentials gathering
- ICS data and command spoofing and manipulation
- DoS of ICS functionality (communications).

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

The Address Resolution Protocol (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 cache of machines on the LAN are targeted, confusing whom they think they are communicating with. The ARP protocol is used to determine which hardware addresses coincide with the Internet Protocol (IP) addresses on the network. The MitM attack is initiated by sending gratuitous ARP commands to confuse each host. These ARP commands tell the two hosts that the attacker computer is really the computer where 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 is taking a different route through the attacker’s computer. The attacker computer then needs to forward all packets to the intended host so the connection stays in sync and does not time out.

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 compromise a host on (or between) the victim computers’ LANs.

4.3.1 IT Protocols Vulnerable to Spoofing and MitM Attacks

Unsecure services developed for IT systems have been adopted for use in ICSs for common IT functionality. Although more secure alternatives exist for most of these services, active unused or obsolete services still exist in many ICSs.

The use of unsecure IT protocols is dangerous because attackers are very aware of them, and have access to script kiddie tools that have been created to exploit them. Network protocol analysis tools are able to intercept and decode most common protocols.[b] Password cracking tools can decode messages and passwords that have been encrypted with broken encryption schemes.[c]

4.3.1.1 Use of Standard IT Protocol with Clear-text Authentication

Clear-text authentication credentials can be sniffed during transmission and used by an attacker to authenticate to the system. If an attacker is able to capture a username and password, he is able to legitimately log onto the system with that user's privileges. For this reason, plain-text remote login services should be replaced with encrypted services such as SSH.

The use of unsecure protocols and services to connect to the ICS hosts creates a high-risk access path into the system. This is a significant vulnerability because it provides remote access to ICS hosts and the functionality allowed to the remote user. Table 28 lists examples of the use of clear-text authentication protocols in ICSs.

Table 28. Sanitized unsecure standard IT protocol findings.

Sanitized Finding	Potential Impact
File Transfer Protocol (FTP) available	Capture of ID and password
The test system is running the plain-text protocols FTP and telnet	
Unencrypted ports were open including FTP, Hypertext Transfer Protocol (HTTP), RPCBIND (Traffic analysis reveals no indication of authentication or encryption)	
rlogin, rsh, FTP, telnet on one workstation	

Recommendation for the Use of Standard IT Protocol with Clear-text Authentication

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. 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 the Secure Socket Layer (HTTPS).

Users of these products should be aware 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 communication should be limited to within the ICS whenever possible. Communications between security zones should be secured as much as possible.

[b] Wireshark, http://www.wireshark.org/

Kismet, http://www.kismetwireless.net/

TCPdump, http://www.tcpdump.org/

Ettercap, http://ettercap.sourceforge.net/

Etherape, http://etherape.sourceforge.net/

[c] Cain, http://www.oxid.it/cain.html

http://www.openwall.com/john/

http://www.thc.org/thc-hydra/

Aircrack, http://www.aircrack-ng.org/

L0phtcrack, http://www.l0phtcrack.com/

4.3.1.2 Use of Vulnerable Remote Display Protocols

Remote display protocols and applications are used to remotely access a machine, providing the ability to logon and remotely control another machine using the graphical display. Remote display protocols used by ICS have been found to accept connections from anywhere, and transport credentials in clear text or by a broken encryption algorithm. Even if strong encryption is used, if the remote display client's host is compromised, the attacker may also have access to the remote ICS host's display and all of the client's functionality including the encrypted channel.

Weaknesses in remote display software can be exploited to gain unauthorized access to the supervisory control user interface software. Table 31 lists common vulnerabilities in remote display software used by ICS for remote access to supervisory control functions. Common remote display services were integrated into the ICS products to provide remote display capability. Remote display software and their weaknesses are well known and easily exploited. Exposure depends on how and where the connection is initiated. The following scenario was accomplished in multiple assessments:

> *"Using a freely available tool called Cain, the team poisoned the ARP caches of the HMI server and a client, telling each that the other was located at the attacker's address. Once the MitM was established, the client connected through the attacker."*

This attack could be accomplished on any network link between the client and server. Multiple free, publicly available tools exist to perform ARP poisoning and MitM attacks, as well as brute-force and dictionary attacks against Terminal Services.

The use of remote display software for remote access to supervisory control functions could be the most significant vulnerability on an ICS because it allows unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user. Table 29 lists examples of vulnerable remote display protocols.

Table 29. Weaknesses in software used by ICS for remote access to supervisory control functions.

Sanitized Finding	Potential Impact
No access controls for remote display	Attacker is able to connect to remote display service from anywhere
Clear-text transmission of remote display credentials	Attacker is able to steal remote display credentials by "sniffing" network traffic while a remote display connection is established
Use of remote display service that uses a broken cryptographic algorithm	Attacker is able to steal remote display credentials by "sniffing" and decrypting network traffic while a remote display connection is established

Remote Display Recommendations

UNIX X server configurations have been found that accepted clients from anywhere. This allows an attacker to connect to it and record keystrokes and screenshots. Access should be restricted and secure authentication should be used to protect login credentials from being stolen.

Vulnerable Windows remote administration services are also used on ICS. All protocols and services used to remotely access ICS hosts should use strong authentication and be kept up to date. When configuring ICS hosts for remote access, administrators and integrators must configure the host to validate the security of the remote access client to protect against unauthorized access through a trusted compromised host.

4.3.2 ICS Protocols Vulnerable to Spoofing and MitM Attacks

Most ICS communication protocols have not been designed and implemented to prevent MitM attacks. They do not adequately verify the identity of actors at both ends of a communication channel, or 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 often important to adequately verify the identity of entities at each end of the communication channel. Failure to do so adequately or consistently 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."*[6]

If an attacker has access to ICS communication paths and reverse engineers the ICS network communications protocol, manipulation is possible of the data flowing between the systems components. This includes commands and messages sent to update operator screens and control field equipment. An attacker may also be able to alter the operator's view of the system by intercepting and manipulating messages received from the ICS. The operator may then be unaware of what the attacker is doing with the system, or tricked into performing dangerous actions. The operator may also just be unaware of the actual system state, and therefore not take the appropriate evasive actions.

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

The lack of, or weak, data integrity checks prevent a protocol from detecting bad data. If integrity check values or "checksums" are omitted from a protocol, there is no way of determining whether data has been corrupted in transmission. Likewise, if integrity check values are easily reverse engineered and duplicated, data manipulation in transmission is invisible upon security inspection.

The lack of, or weak, authentication prevents the protocol from detecting that the message is from an unauthorized sender. Some ICS protocols rely on weak authentication, such as hostname or IP address, which are easily spoofed.

4.3.2.1 ICS Data and Command Message Manipulation and Injection

ICS network protocols, including those used to send control commands and status data, can be altered, replayed, or spoofed because they lack sufficient access control and integrity checking mechanisms. This vulnerability requires minimal skills to intercept or create the network messages. An attacker's ability to intelligently interpret and manipulate process status depends on how much of the ICS protocol and physical process he has been able to discover, or reverse engineer. ICS and network programming skills are needed for this attack.

The ICS network design and implementation determines the exposure of control protocol vulnerabilities. This vulnerability is exposed to anyone who has gained network access to the supervisory control network, or a network that is allowed access to control equipment.

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

Recommendation for ICS Message Manipulation

The system design needs to implement strong authentication into ICS communication protocols. 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 to give them an altered view from reality. Authentication also protects against unauthorized commands being sent to the ICS process devices.

Defenses that reduce exposure to this vulnerability are network access and content filtering rules. IDS monitoring should catch the attacker's presence on the network and MitM activities. Administrators can configure network equipment to prevent MitM attacks, but MitM is not necessary if the attacker has gained access to a host that is allowed to send control messages. Even if the control protocol is encrypted, the attacker can still send control messages if he has gained access to the host that encrypts the packet.

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

4.3.2.2 Unprotected Transport of ICS Application Credentials

Clear-text authentication credentials can be sniffed during transmission and used by an attacker to authenticate to the ICS application. If an attacker is able to capture a username and password, he is able to legitimately log into the application with that user's privileges.

Table 30 lists sanitized examples of unprotected ICS application credentials sent over the network.

Table 30. Sanitized examples of unprotected ICS application credentials.

Sanitized Finding	Potential Impact
Operator and developer applications transmit login information in plain text	Capture of ID and password
Clear-text password traffic	

Recommendation for Unprotected Transport of ICS Application 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 cracked, if necessary, by the attacker. Passwords should be securely encrypted or hashed before being stored or transmitted. When using Web applications with Secure Sockets Layer (SSL), use SSL for the entire session from login to logout, not just for the initial login page.

4.3.2.3 Encryption of ICS Communications

A number of concerns about the mal-effects of cryptography on an ICS have been raised. The four most common concerns are latency, bandwidth, availability, and IDS interaction. The highest priority for distributed real-time ICSs is availability (fault tolerance). For this reason, and because of the difficulty in implementing Internet Protocol Security (IPSec), there seems to be trepidation in implementing encryption on a critical process control network. Another issue is that even if ICS communications are initially configured to use IPSec, it may be turned off for system trouble-shooting and then never turned back on.

IPSec and other cryptography solutions have been viewed as a cure-all for security ailments. IPSec's limitations should be understood to avoid a false sense of security. IPSec provides four main security properties: confidentiality, integrity, authenticity, and replay protection. Unfortunately, IPSec is still difficult to configure and encryption makes system troubleshooting more difficult.

IPSec operates at the host level. Access to a host provides the potential for full control of the IPSec communications. System owners should be aware that unauthorized users may simply utilize the encrypted channel as part of the attack path from a compromised host to its communication partner deeper inside the ICS network. The attacker is even able to hide his activities inside the IPSec encryption.

Even if an attacker cannot eavesdrop or alter packets, he can still prevent IPSec communication. An ARP MitM attack can be performed to intercept and drop packets used for Internet Key Exchange (IKE) authentication. If the default IPSec policy is set to require IPSec and security negotiation is not successful, outgoing traffic is not allowed.

Requiring IPSec increases the risk for a DoS of critical system communications. Even without malicious intent, a DoS can be caused by a failure to identify a trusted host due to clock skew or other authentication criteria. Therefore, necessary communication should not be set to require IPSec. One possible solution is to set the IPSec policy to request IPSec protection between all capable machines. Exceptions must also be added to the policy for each device (controller devices, etc.) that does not support IPSec. It should also be understood that IPSec protection cannot be guaranteed.

The decision for configuring 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 insure integrity, authenticity, or replay protection) or whether it is critical that this communication be available. This can be defined on the firewall at the IP/port level. If the communication requires the security provided by IPSec and can tolerate a DoS, the “require” policy should be implemented. If the hosts must have an available channel for data transfer and the security layers provided by IPSec are secondary, IPSec should be configured with the “request” policy. Unfortunately everything is not this clear cut, but those are the tradeoffs to consider.

Even though IPSec encryption makes network-based intrusion detection difficult, host-based intrusion detection products can potentially see the incoming or outgoing unencrypted messages and make decisions before they are further processed. Also, network IDSs can be used to verify the IPSec policy.

ICS customers should be aware, that encryption only provides protection between the communication endpoints. It cannot protect against attacks or vulnerabilities on the endpoint components by someone who has gained access to the encrypted channel through an endpoint. ICS vendors and owners should become familiar with the encryption endpoint problem and the available encryption solutions before implementing it on their systems.

4.3.3 Summary of Communication Channel Recommendations

Unsecure versions of common IT services should be replaced where possible by their secure versions.

Communications between security zones should be secured as much as possible.

The first choice of long-term mitigations is to replace unsecure ICS protocols with protocols that provide strong authentications and integrity checks 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 to give them an altered view from reality. Authentication also protects against unauthorized commands being sent to the ICS process devices.

4.3.3.1 ARP MitM Defenses

One defensive measure against MitM attacks is to hard-code the Media Access Control (MAC) addresses of the communication endpoints in each other’s ARP tables. This causes the systems to ignore the bogus ARPs sent during an ARP MitM attack, which renders it ineffective. The problems with this approach include:

- Some systems do not provide a way to hard-code the ARP tables

- Some systems provide a temporary method, but it must be done every time the system is started (volatile storage)
- Replacement of a remote device requires updating the ARP tables on every system component in the communication path
- This only protects against one MitM attack method (ARP spoofing).

Along with the above, it is also relatively inexpensive to employ all of the features of the installed networking equipment, such as port security on switches and 1-to-1 rules on firewalls. However, firewalls are not commonly used in the basic control communications path.

IDS solutions can be employed to detect ARP MitM activities. Domain Name System (DNS) spoofing can be protected against by using the available DNS security measures.[14]

4.3.3.2 ICS Encryption Issues

A long-term mitigation could be to replace ICS protocols with an encrypted version of the protocol. Unfortunately, there is no drop-in replacement currently available—with the exception of secure ICCP. Even if there were, it would be a huge task to rewrite and certify all of the software involved. Hopefully, the near future will bring such a protocol into common acceptance that future products can employ.

The next best choice is to use IPSec with only the authentication header. This mode does not encrypt the data, but it does provide a cryptographically authenticated wrapper that prevents tampering. Another advantage of this mode is that an IDS can still monitor the communications and detect anomalies in the conversation.

Another option is to use the unsecure protocol through an encrypted protocol tunnel, such as IPSec, SSL, etc. This would require that the components at the communication endpoints have the software installed, configured, and tested to be sure all is setup correctly and securely. This could be a substantially large task, depending on the capabilities of the devices involved. It is also quite possible that some devices may not have the necessary computing power to handle the added burden. Even if the computing resources exist, there is still an issue of labor with key management. Another problem with this approach is that all of the communications between the systems can be blocked as long as the end points are susceptible to ARP cache poisoning. In the case of IPSec and its configuration, it will either stop talking altogether or fall back to unencrypted mode. Obviously, neither is a desirable outcome.

A final option is to employ separate encryption hardware (so-called "bump-in-the-wire") devices to encrypt the traffic for a system. In addition to the cost, these devices have their own configuration and key management problems.

Encryption solutions that tunnel ICS protocols have been tested in NSTB assessments. In some cases, the components were incorrectly configured and the encrypted connections were still vulnerable to a MitM attack. In other cases, the system integrators were not able to successfully implement the encryption solution on the test system. However, the addition of encryption capabilities to ICS products may allow the communication channels to be secured.

Difficulty of implementation and viewing traffic for trouble-shooting are issues that can prevent encryption from being used in operational IDS. ICS designers and customers should refer to current and specific documentation on network cryptography options and implementation instructions. Encryption cannot be used as a fix for vulnerabilities in the ICS system, but it should be considered as a layer of defense.

4.3.3.3 Communication Channel References

More information about communication channel vulnerabilities can be found in the related security weaknesses from the CWE[6]:

- CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
- CWE-285: Improper Access Control (Authorization)
- CWE-311: Missing Encryption of Sensitive Data
- CWE-306: Missing Authentication for Critical Function
- CWE-327: Use of a Broken or Risky Cryptographic Algorithm.

The DHS *Control Systems Communications Encryption Primer*[15] provides ICS specific information on encryption. More detailed information on recommended configurations is available in the NIST Special Publication 800-113.[16]

4.4 Communication Endpoint Vulnerabilities

Network services are listening for messages to accept, and are exposed to attacks that exploit input and output validation vulnerabilities. These services are vulnerable to remote compromise if they do not properly check the size of user input, sanitize user input by filtering out unneeded but potentially malicious character sequences, or initialize and clear variables properly.

Assessments target ICS programming errors in functions that parse network code that do not properly validate or “sanity check” input values. Functions that process data often accept input from another source and expect it to conform to certain requirements. If a function uses the input without testing to make sure it conforms to expectations, it can cause the application to crash or execute commands that were provided as input. An attacker may be able to bypass authentication, access unintended functionality or escalate privileges this way. These vulnerabilities have the potential to allow remote code execution, privilege escalation, authentication bypass, data loss, and DoS.

Authentication systems are also targeted because they can allow authentication bypass if they are not implemented correctly.

4.4.1 Lack of Input Validation and Bounds Checking in ICS Services

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 exceedingly 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.

4.4.1.1 Buffer overflow in ICS service

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 Object Linking and Embedding (OLE) for Process Control (OPC), ICCP, and DNP3. If these applications contain invalid input vulnerabilities such as buffer overflows, exploitation by anyone who is able to gain network access is possible. Such action could allow the attacker to gain access to the host, or cause a communication DoS or other problems for the ICS.

Input validation vulnerabilities have been found in custom server applications written to process ICS protocol messages and other ICS network traffic. These applications are:

- Control protocol services

- Supervisory Control protocol services
- ICCP services.

Buffer overflow vulnerabilities are the most common type of input validation weaknesses reported on ICS assessments. Buffer overflows are the result of programmer oversight, and result when a program tries to write more data into a buffer than the space allocated in memory. The “extra” data then overwrites adjacent memory, and ultimately results in abnormal operation of the program. A carefully planned and executed 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. Network protocol implementations that do not validate input values can be vulnerable to buffer overflow attacks.

Buffer Overflow Recommendations

The use of unsafe C functions, creating buffer overflow vulnerabilities, has been identified throughout ICS code. Buffer overflows can be exploited in any vulnerable functions that accept input that the attacker can control, even if it passes through other functions first. Functions that parse network traffic are at highest risk of attack because they are the most exposed to malicious input.

C-based programs are notorious for their vulnerability to buffer overflows. Older programming languages such as FORTRAN and Pascal are vulnerable as well, but are becoming less common, especially in programs performing network activity.

The interpreted languages such as Java, C#, and Perl, which include most web applications, are generally immune to buffer overflow attacks. However, they are still vulnerable to other types of attacks.

Buffer Overflow References

Five of the *2010 SANS/CWE Top 25 Most Dangerous Programming Erro*rs[3] are types of buffer overflows. Table 31 lists CWE entries related to buffer overflows.

Table 31. Five of the 2010 Top 25 most dangerous programming errors related to buffer overflows.

Rank	Programming Error
3	CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
12	CWE-805: Buffer Access with Incorrect Length Value
15	CWE-129: Improper Validation of Array Index
16	CWE-190: Integer Overflow or Wraparound
18	CWE-131: Incorrect Calculation of Buffer Size

4.4.1.2 ICCP Services

ICCP, also called Telecontrol Application Service Element 2.0 (TASE.2), is an international protocol standard that is used extensively in the electrical power industry. ICCP communications links are used to exchange information among electric utilities, independent system operators, regional transmission organizations, and independent power producers, among others. This information is typically exchanged over private networks or leased lines; in some cases, Virtual Private Network (VPN) connections over the Internet may also be used. Figure 10 shows how these entities could be connected via ICCP.

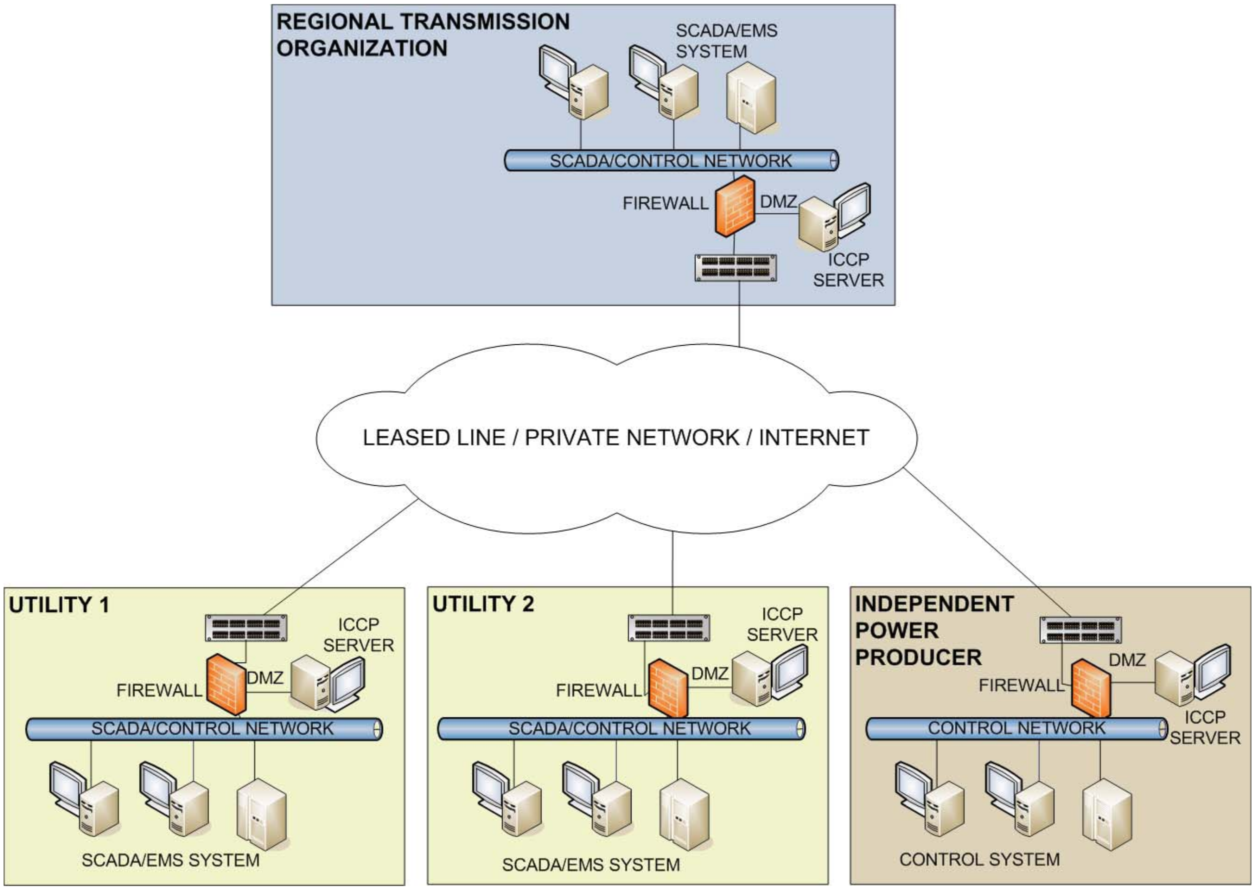

Figure 10. Sample ICCP network.

Two phases of ICCP assessments were conducted on ICCP services. Because of the interconnections these links provide between entities, and the resulting risk of a coordinated cyber attack on multiple entities through these links, the ICCP protocol was chosen as the subject of a cyber security assessment.

Although nearly every major SCADA/Energy Management System (EMS) vendor offers ICCP software as an integrated or standalone part of their overall systems, many of them purchase the underlying protocol layers from a third-party vendor. The NSTB partnered with two major ICCP stack providers and four major SCADA/EMS vendors to assess the security of their ICCP products and implementations. A total of three third-party stacks and five SCADA/EMS ICCP implementations were tested.

The majority of findings were buffer overflow and DoS vulnerabilities. Buffer overflow vulnerabilities can potentially allow an attacker to take control of the ICCP server, providing a possible path to the ICS network. DoS events are less severe, but can be used to cause an outage of the ICCP service. INL found that the complexity of the ICCP protocol contributed to the number of vulnerabilities found.

Some new SCADA/EMS ICCP implementations were found to use older versions of the third-party protocol stack. These older versions contained known vulnerabilities, including multiple DoS weaknesses and vulnerabilities that could lead to remote code execution. SCADA/EMS vendors should integrate the latest ICCP protocol stack into their products. SCADA/EMS owners should validate that their ICCP implementations use the latest versions or patches.

The results from these assessments have been consolidated into a public report, available from the NSTB.[4]

4.4.1.3 ICS Services Recommendations

Vulnerabilities in services that are exposed to less-trusted networks have higher consequences because they may provide a path from the lower security zone to the higher security zone. Remote code execution through buffer overflow attacks is a common attack method for gaining unauthorized access to hosts. ICS design requires that their protocols be allowed through firewalls to support external data collection and sharing. These protocols and services should have top priority for vulnerability remediation activities.

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 unsecurely.

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 NSTB 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.

ICS vendors need to perform code reviews of all ICS applications responsible for handling network traffic. Network traffic cannot be trusted, so better security and sanity checks need to be implemented to prevent crashes and DoS attacks, even if input validation vulnerabilities cannot be exploited for remote access.

4.4.2 Database Vulnerabilities

A Historian server is used for data archiving and analysis and is typically an integral part of an ICS. It is usually located in a DMZ or on the corporate network. Threats to the historian include compromise of the historian host and data corruption. ICS historians typically utilize a common SQL server as its backend. The historical data is often made available for viewing via a custom Web interface or application.

The Historian client applications are high-risk components because they are often accessible from the corporate environment and can provide an attacker with a point of entry to the ICS network. Additionally, an attacker may gain access to unauthorized information which, in some cases, can be used to cause economic damage.

Historian database applications use SQL queries to retrieve information. An SQL injection vulnerability is caused when an application incorrectly or inadequately filters user input. If an attacker inserts literal escape characters into a database query, they may gain arbitrary read or write access to the database. Weak authentication can also be defeated to gain access to the database.

Unsafe function calls have been found in code written to parse historian data messages (see Table 32). Failure to validate input can result in a DoS of the service or unauthorized access to the associated host.

Table 32. Sanitized Historian database findings.

Sanitized Finding	Potential Impact
Multiple SQL injection vulnerabilities	Execution of unauthorized database commands
Database access code susceptible to SQL injection attack of database server	
Vulnerability in Database server when large SQL statement is parsed	
Unsecure C/C++ routines	Unauthorized access to or DoS of Historian database or host
Database server protocol vulnerabilities can be exploited to cause a DoS	Historian DoS
Connection to Historian without user name or password	Unauthorized access to Historian database
Database ports are remotely accessible	
Both the client and server use the same certificate to encrypt/authenticate connections	

4.4.2.1 SQL Injection

SQL injection vulnerabilities are caused by the lack of input validation, or improper sanitization of special elements used in an SQL statement. If attackers can influence the SQL statements used to communicate with the database, they could modify the queries to steal, corrupt, or otherwise change data in the database. If SQL queries are used for security controls, such as authentication, attackers could alter the logic of those queries to bypass security.

Attackers use SQL injections vulnerabilities within client (often Web) applications to attack the SQL server. Even if the vulnerable service is isolated in a DMZ (as shown in Figure 11), SQL injection can still attack the SQL server within the secure network. A successful compromise will give the attacker control of the SQL server within the secure network, even if the firewall prevents communications.

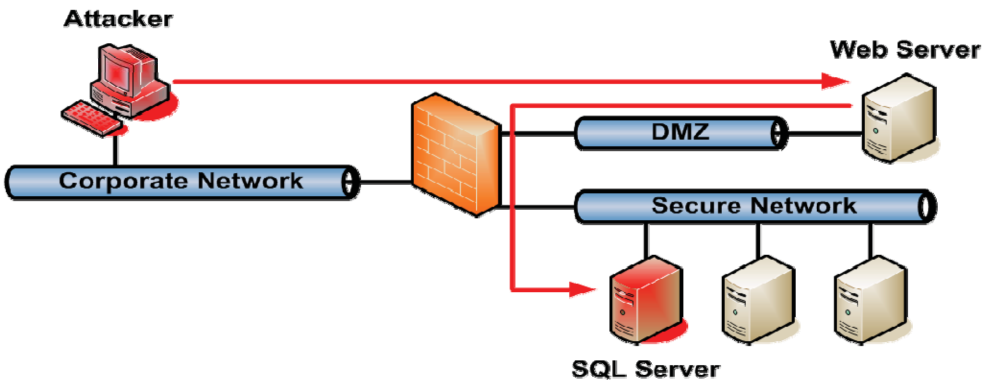

Figure 11. Example of an SQL injection attack via Web applications.

Development Recommendations

Developers should use vetted libraries or frameworks that do not allow SQL injection and cross-site scripting weaknesses to occur or provide constructs that make this weakness easier to avoid. For example, they can use persistence layers such as Hibernate or Enterprise Java Beans, which can provide significant protection against SQL injection if used properly.

Developers should use care when constructing SQL queries, including stored procedures that are located on the SQL server itself. They should follow Web programming security guidelines to help mitigate common mistakes, validate input, and properly encode, escape, and quote output.

Follow the principle of least privilege. Use the strictest permissions possible on all database objects, such as execute-only for stored procedures.

Operational Recommendations

Databases should be replicated out to the DMZ. If an attacker finds and exploits an SQL injection, he will simply own another server in the DMZ rather than jumping into a more secure network.

Administrators of ICS with Web servers should use an application firewall that can detect common Web attacks. This might not catch all attacks, and it might require some effort for customization. However, it is a layer of defense that can be used to help reduce the risk of vulnerabilities in Web applications that expose the ICS historian and Web servers to attack from the Web client's network.

4.4.2.2 Database References

ICS-specific information is available from the ICS-CERT portal.[17]

More detailed information on SQL injection can be found on the Internet:

- *CWE-89: Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection')*[6]
- *SQL Injection Attacks by Example.*[18]

The *Open Web Application Security Project (OWASP) Top Ten for 2010*[19] document provides basic techniques for mitigating the highest Web application risks along with additional references. *Risk A1, Injection*, addresses SQL injection risks. A summary of the recommendations for avoiding injection flaws follows:

1. Avoid the interpreter entirely
2. Use an interface that supports bind variables (e.g., prepared statements, or stored procedures)
    - Bind variables allow the interpreter to distinguish between code and data
3. Encode all user input before passing it to the interpreter
    - Always perform "white list" input validation on all user-supplied input
    - Always minimize database privileges to reduce the impact of a flaw
4. Follow the guidance from the OWASP *SQL Injection Cheat Sheet.*[20]

4.4.3 Web Vulnerabilities

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 report refer to proprietary Web applications.

The major security weaknesses found in ICS Web services along with sanitized assessment findings and associated risks are listed below in Table 33. Like SQL injection vulnerabilities, directory traversal, and XSS vulnerabilities are caused by insufficient or incorrect handling of user input values and can lead to similar consequences. A directory traversal is not a complicated attack and is accomplished by manipulating paths in the Universal Resource Locator (URL). Successful directory traversal attacks allow the attacker to view the contents of directories they not allowed to see. Cross-site scripting attacks come in many different forms and are significantly more complex than directory traversals.

Table 33. Sanitized Web services findings.

Vulnerability	Sanitized Finding	Potential Impact
Poor Authentication	No authentication between corporate clients and Web server on DMZ	Unauthorized access from corporate network to DMZ
Directory Traversals	HTTP Port 80 had no default page. Displayed directory structure.	Unauthorized access to files and directories on the Web server
	Arbitrary files can be read on Web server by adding ../../ or ..\..\ in front of file name.	
	Trivial HTTP used – vulnerable to several exploits	Compromise of Web server
Cross-Site Scripting	Multiple cross-site scripting Vulnerabilities	Compromise of Web client
	Persistent cross-site scripting Vulnerability	
	Cross-site scripting on Login and History Analysis Pages	
Bad Session Tracking	DNS spoof used to redirect to malicious Web page	
Vulnerable Browser Plug-ins	Browser plug-in exploit allowed control of workstation	

4.4.3.1 Web Vulnerabilities: Improper Authentication

Web services developed for ICSs 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 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.

ICS assessments have also found poor authentication, poor session tracking, Structured SQL injection, and cross-site scripting vulnerabilities that can allow unauthorized access to Web servers and applications.

The *OWASP Top Ten*[19] document provides basic techniques for mitigating the highest Web application risks along with additional references. Risk A3, *Broken Authentication and Session Management*, the *Authentication Cheat Sheet* and the *Transport Layer Protection Cheat Sheet* can be referenced for Web authentication information.[20]

CWE categories, *CWE-287: Improper Authentication* and *CWE-442: Web Problems,* contain related authentication and Web programming information as well.[6]

4.4.3.2 Cross-Site Scripting

According to the 2010 *CWE/SANS Top 25 Most Dangerous Programming Errors*[3] report, cross-site scripting is the most widespread and critical programming error.[4] It is dangerous because it allows 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 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.

It is possible for an attacker to inject malicious script into a link and have a Web site return it to the victim as though it is legitimate. The victim's Web browser will then run the malicious script, since it came from the server, potentially compromising the victim's computer by using one of many browser exploits. There are many such scenarios, which allow for this behavior, but they all 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. Since 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 disclosing end user files, installing Trojan horse programs, redirecting the user to some other page or site, running "Active X" controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy, and modifying presentation of content.

Cross-site scripting presents one entry point for attackers to access and manipulate ICSs 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.

Risk to ICS: Web Client Access Control and Authentication Bypass

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 Web site 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 Web sites 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 with careful users, 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.

Cross-Site Scripting 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* 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.[21]

4.4.3.3 Directory Traversal Enabled

Web application 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 developer uses a path provided by the user, but fails to validate the path to ensure that the user can only access the necessary files. For example, the classic HTTP "GET" directory traversal attack is performed by submitting "../" to tell the OS to look up one directory. If the HTTP server was vulnerable to a directory traversal attack, this GET request would cause the HTTP to get the "/etc/passwd" file.

Directory traversal attacks can be used to gather information by downloading files or gain access to the ICS by uploading the exploit code to be executed. Being able to download arbitrary files is more common than being able to upload files. If an attacker can download files, he may be able to obtain important files such as password files or proprietary information about the ICS he is attacking. If an attacker can upload files, he can upload exploits and attempt to compromise the system. For example, an attacker could upload a script to the startup folder, or replace a secure application with a malicious one.

The damage that a directory traversal vulnerability can cause is related to the permission of the application that was vulnerable. If the vulnerable application has limited read/write permissions, the attacker may not be able to do anything of importance. However, when running as system or root, then the damages can be extensive. Table 34 summarizes the potential impacts of directory traversal vulnerabilities.

Table 34. Potential directory traversal impacts.

Vulnerability	Risk
Arbitrary file download	Information disclosure
Arbitrary file upload	System compromise

Directory Traversal Recommendations

The file permissions on the Web server need to be set to grant the least privileges necessary. The system design needs to be evaluated to reduce necessary file access as much as possible. Write permissions are most dangerous, but read permissions may disclose valuable information or information that can be used for an attack.

Features on the Web server, such as unrestricted browsing, need to be disabled and additional security of HTTP can be gained by utilizing the SSL where possible. The Web server should filter input to screen incoming filenames and exclude the ".." string. Disabling unused ports and keeping the Web server patched to current standards are good practices.

4.4.3.4 Web Security References

The OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. OWASP tools and documents can be used to detect and to guard against security-related design and implementation flaws, as well as to add security-related activities into the Software Development Life Cycle (SDLC). The *OWASP Top Ten*[19] ranks the most critical Web application security flaws.

The CWE can also be referenced for information about Web security weaknesses. Table 35 lists related OWASP and CWE resources.

Table 35. OWASP and CWE Web security resources.

Web Security Reference Title	Location
OWASP Developer's Guide OWASP Testing Guide OWASP Code Review Guide Application Security Verification Standard (ASVS) Open Software Assurance Maturity Model (SAMM) OWASP Prevention Cheat Sheet Series Top 10-2010 The Ten Most Critical Web Application Security Risks	http://www.owasp.org/
CWE-442: Web Problems CWE-79: Failure to Preserve Web Page Structure ('Cross-site Scripting') CWE-89: Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection') CWE-98: Improper Control of Filename for Include/Require Statement in PHP Program ('PHP File Inclusion') CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	http://cwe.mitre.org

4.4.4 Other Input Validation Errors

Like other software products, the biggest security weakness in ICS code is poor input validation. 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.

Improper input validation is a high-level root cause of many types of vulnerabilities. It also describes most of the vulnerabilities found in ICS software. Many different kinds of input validation errors were identified on ICS assessments, and only the most significant ones are specifically addressed in this section under the ICS components most affected by them. All of the weaknesses in the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[3] can be associated with improper input validation.

4.4.4.1 Input Validation Recommendations

Input validation vulnerabilities have been found in server applications written to process ICS protocol traffic. Most result in access to the host on the server was running. The impact of these vulnerabilities can be reduced by limiting the server's privileges. The attacker will inherit the rights of the exploited process, so service privileges should be minimized as much as possible.

Message values and format should be validated to prevent exploitation. Sanity checks of incoming messages can ensure that the lengths and counts seem reasonable, if the data in the message is valid, and if the message is valid given the state of the connection. Network parsing code should be reviewed, starting with exterior services and moving inward. Add additional sanity checking to insure that malformed messages are gracefully rejected.

The top recommendation in the CWE/SANS "Monster Mitigation" list for making more secure software addresses input validation:

> *"M1: Establish and maintain control over all of your inputs.*
>
> *Improper input validation is the number one killer of healthy software, so you're just asking for trouble if you don't ensure that your input conforms to expectations. Many of today's most common vulnerabilities can be eliminated, or at least reduced, using proper input validation.*
>
> *Use a standard input validation mechanism to validate all input for:*
>
> - *Length*

- *Type of input*
- *Syntax*
- *Missing or extra inputs*
- *Consistency across related fields*
- *Business rules.*

Where possible, use stringent white lists that limit the character set based on the expected value of the parameter in the request. This can have indirect benefits, such as reducing or eliminating weaknesses that may exist elsewhere in the product.

Do not accept any inputs that violate these rules, or convert the inputs to safe values.

Understand all the potential areas where untrusted inputs can enter your software: parameters or arguments, cookies, anything read from the network, environment variables, reverse DNS lookups, query results, request headers, URL components, e-mail, files, databases, and any external systems that provide data to the application. Remember that such inputs may be obtained indirectly through API calls.

Be careful to properly decode the inputs and convert them to your internal representation before performing validation."[9]

4.4.5 Communication Endpoint Vulnerabilities Summary

All ICS components that handle data from other components should be evaluated, starting with services that are exposed to less-trusted networks and working inward.

Weak or missing security features in ICS software leave the system components vulnerable to manipulation by any threats they are exposed. For the best defense, each component of the ICS must have its own protection mechanisms. The identification of critical components with corresponding risk analysis and mitigation strategies is a must for both operations and security.

Software that does not properly check the size of user input, fails to sanitize user input by filtering out unneeded but potentially malicious character sequences, or does not initialize and clear variables properly could be vulnerable to remote compromise. Attackers can inject specific exploits, including buffer overflows, SQL injection attacks, and cross-site scripting code to gain control over vulnerable machines.

To avoid such attacks, both internally developed and third-party application software must be carefully tested to find security flaws. For third-party application software, enterprises should verify that vendors have conducted detailed security testing of their products. ICS vendors must conduct such testing themselves or engage an outside firm to conduct such testing.

See Section 4.2.2.3, "Secure Coding References," for additional information.

4.5 ICS Application Authentication Vulnerabilities

Many of the input and output validation vulnerabilities described in Section 4.4 above have the potential to bypass authentication. Authentication is used to enforce access controls. Weak authentication allows access controls to be subverted. ICS security assessments have shown that access to process data and control functionality can be trivial because authentication is not required, or can be circumvented.

4.5.1 Authentication Bypass Issues

ICS applications, like the operator's user interface, must be protected from unauthorized access because they possess the functionalities and permissions to affect the physical process. The operator interface, or HMI, provides graphical monitor and control of the physical system. Table 36 lists assessment findings relating to HMI authentication bypass vulnerabilities.

Table 36. Sanitized HMI authentication findings.

Sanitized Finding	Risk
Login information remembered	Authentication without credentials
Kerberos authentication always succeeds	
No limit on authentication attempts	Password guessing or cracking

4.5.1.1 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. 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.

ICS developers should implement robust authentication by the server or component that is granting access.

4.5.1.2 Authentication Recommendations

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side. 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.

4.5.2 Credentials Management

Passwords are often the weakest link in an authentication architecture. Typically, this is due to human and policy factors and can only be partially addressed by technical remedies. ICS systems have many levels of passwords that could provide the weak link an attacker needs to gain access to the system.

OS-level passwords are used when the user logs onto a machine, and for authenticating OS-level services, like network file systems. With Windows computers, administrators must ensure that both the local accounts and the domain accounts have good passwords.

User IDs and passwords are shared among the different operators of the ICS, in some cases, because of the criticality of the system the operators are running. Nonetheless effective passwords that meet minimum security requirements and are frequently changed are key.

Application passwords must be managed as well. This includes Web applications, ICS applications, etc.

Backend services, such as SQL services, are frequently forgotten because they are usually not directly exposed to the user. This can be dangerous because many of these services will provide full server access to anyone who connects to them. For example, strong passwords at the OS level do not provide much protection if the database still has the default accounts and passwords, and allows a remote connection to execute shell commands as the system user. NSTB assessments have identified many cases where third party products were delivered with the ICS without passwords or with default passwords. The accounts can remain unconfigured, sometimes because ICS owners are unaware that they exist.

4.5.2.1 Weak Passwords

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 time to guess or crack the password increases. Cracking a password can be trivial or virtually impossible depending on the combination of different character types used with larger password length.

A policy mandating the use of strong passwords for all assets inside the electronic perimeter with a reasonable lifespan limit needs to be mandated and enforced. Usage of common passwords, especially administrative, needs to be discouraged.

Default Passwords

A common problem found during assessments was that even though secure authentication applications were used, installations and configurations were not correct.

Default database accounts are often found without passwords. This may be due to oversight while configuring a test system versus a problem with the default settings on a newly deployed system, but will always be a possibility until configuration procedures are put in place to ensure secure and consistent default configurations.

ICS and networking equipment should not be left with the default password from the manufacturer. Default passwords can give an attacker easy access to the equipment that controls the process. Unless required by the ICS software, the default password should always be changed to a robust, unpublished password.

Exploiting a system with default accounts would only require access to the documentation, or access to a sample system, in which case an attacker can discover the accounts themselves. In many cases default passwords can be found globally available on the Internet.

Hosts are exposed to attack by anyone able to connect and authenticate using the default accounts and passwords.

Remove the default accounts, or at least ensure that each installation uses different passwords. In addition, ensure that the password used is a strong password. By having different accounts and passwords, an attacker will not be able to translate knowledge learned on one system to another system. Documentation about the default accounts should be distributed to all users so they know that the accounts exist and can take the initiative in removing and/or changing their passwords.

No Password Configured

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.

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.

Table 37 lists sanitized ICS application authentication findings.

Table 37. No password findings.

Sanitized Finding	Risk
No authentication between corporate clients and Web server on DMZ	Unauthorized access to DMZ from corporate network
Connection to Historian without user name or password	Unauthorized access to Historian

Strong Password Recommendations

Strong passwords are easy to remember and hard to guess. The two most important password criteria are length and complexity. This means that passwords should be created from a large character set, as long as possible, and easy to remember. There are many tricks for accomplishing these objectives. The following guidance can be used to create strong passwords.

Use sufficient length and complexity:

- Use 8 or more characters whenever possible (14 or more characters on Windows systems)
- Pull from all characters on the keyboard for the largest character set.

Make it memorable by using a formula. Many people use the following formula to create unique strong passwords:

1. Think of a memorable sentence, lyrics, poem or saying
2. Use the first letter of each word
3. Create a scheme for creating a mixture of upper and lower case letters
4. Insert numbers in a meaningful (memorable) way
5. Insert punctuation and symbols in a meaningful (memorable) way
6. Test the password using a secure password checker[d]

Make it difficult to guess by avoiding:

- Words in any language, including common misspellings, abbreviations, backward spellings, etc.
- Sequences or repeated characters
- Keyboard patterns
- Personal information such as name, birthday, children, address, driver's license, passport number, etc.

A strong password can only stay strong if it is protected. Users must not reveal their passwords to anyone or write it down where it may be discovered.

Strong Password References

Tips for creating strong passwords are widely available. A few examples are listed below:

- *Secrets to the Best Passwords*, http://www.computerworld.com/s/article/82883/Secrets_to_the_best_passwords
- *Password Security*, Red Hat, http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/s1-wstation-pass.html
- *Minimum Password Complexity Standard*, https://security.berkeley.edu/MinStds/Passwords.html
- The SANS Institute's sample password policies.[22,23]

4.5.2.2 Weak Password Requirements

Passwords exist at multiple locations, many of which don't have automated policies that can be applied.

[d] https://www.microsoft.com/protect/fraud/passwords/checker.aspx?WT.mc_id=Site_Link

Cain, http://www.oxid.it/cain.html

If a product does not require that users have strong passwords, it makes it easier for attackers to compromise user accounts.

An authentication mechanism is only as strong as its credentials. For this reason, it is important to require users to have strong passwords. Lack of password complexity significantly reduces the search space when trying to guess user’s passwords, making brute-force attacks easier.

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 it changes too often, users will undermine their security 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 auto-generated, but instead created from passphrases or other memorable means.

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. Table 38 shows general weak password findings.

Table 38. Weak password requirements findings.

Vulnerability	Risk
Password was found on the device it was meant to protect	Unauthorized access
Maximum password length is too short	Password guessing or cracking
Minimum password length is too short	
No minimum length for user interface password	

Password Policy Recommendations

Implement a password policy that enforces strong passwords to prevent password cracking. One password security concern is that if a password is too complicated and difficult to remember, users will undermine their security by writing the password down on small pieces of paper and placing them on the back of the keyboard or in a drawer. 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.

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 usage of strong passwords.

Authentication mechanisms should always require sufficiently complex passwords and require that they be periodically changed.

The SANS Institute’s sample password policies provide guidance on creating, protecting, and changing passwords.[22,23]

4.5.2.3 Weak Protection of User 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.

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.

LAN Manager (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.[24] Client security policies should be configured so that only the NTLM response is given.

If LM authentication is required, update the configuration settings so that only the new Windows NT (NTLM) network authentication is used. 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. Table 39 shows examples of weak protection of user credentials.

Table 39. Weak protection of user credentials.

Vulnerability	Impact
User names and passwords are stored in database	Discovery of ID and password
Database user name and password found in documentation	

Shared Accounts

The reality of working on an ICS is that most user IDs and passwords are shared among the different operators of the system. This sharing exists, in many cases, because of the continuous operational criticality of the system the operators are running. The cost of an outage because of a locked user ID or a forgotten password may be too high.

If user-level authentication is not an option, using different user IDs and passwords for the DMZ, as well as different user IDs and passwords for the business LAN, can help increase security. This prevents an attacker from using a user ID and password obtained from the business LAN to gain access to the ICS DMZ and/or the ICS LAN.

4.5.2.4 Credentials Management Recommendations

A common problem found during assessments was that even though secure authentication applications were used, installations and configurations were not correct.

Instructions for secure installation and proper configuration for each application need to be followed and tested. Do not allow login information to be hard coded into scripts and user programs, or stored so that reauthentication on that computer is never required again.

ICS vendors should deploy systems with default non-guessable passwords. All users and their default passwords should be documented along with instructions for changing these passwords once ownership transfers to the end user of the system.

The users of the system should change all default passwords to secure passwords. The users of the system should also ensure that all users on any system are documented and have secure passwords.

Common practice for an attacker once access is gained is to create backup administrative accounts in case the compromised account is detected. Therefore, regular polling of all usernames will not only help ensure that accounts have passwords, but also help detect compromised systems.

4.5.3 Authentication and Credentials Summary

Users are responsible for creating and protecting authentication credentials. Application developers are responsible for supporting strong passwords and protecting authentication credentials in the software.

System integrators and administrators are responsible for configuring the systems to require and protect strong passwords as well. High risk or common ICS problem areas are listed below:

- Outward facing services (that allow access from another network)
- Default accounts
- Support services, such as SQL services

In some ICS operations, operators share user IDs and passwords. This sharing exists, 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/or 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 hardcoded passwords, ICS owners can work with the vendor to fix this vulnerability. They can then implement a password policy that enforces strong passwords to greatly impede password cracking and guessing.

4.6 Authorization Vulnerabilities

If an attacker gains full access to a host, all functions that the server can execute are now under the attacker's control. In addition, the attacker now has access to the resources as the compromised server, including communications with other devices and servers.

All applications, hosts and networks need to be locked down as much as possible to limit the consequences of compromise as much as possible. Once an attacker has gained access to a host, compartmentalization and access controls can contain them.

4.6.1 User Accounts with Unnecessary Privileges

Many ICS user accounts are given administrator or root privileges. This means that an authenticated user has full access over the host. User accounts used for interactive logon should be carefully evaluated for the proper set of permissions.

Configure the OS access control capabilities with Access Control Lists (ACLs) using a "default deny" policy.

4.6.2 File Permissions

A related issue is file permissions. File shares should be restricted to only those users who require access and the access level they require. For example, if ICS info is shared to everyone on the network, even if the ICS network is segmented, an intruder gaining access to the ICS network will have access to all that ICS-specific information.

Share files to only the computers and accounts that require them. Restrict the read and write permissions of these shared files and directories to the minimum required for each user. Restrict ability to create network shares to the users that need this functionality (generally administrators). Use network segmentation and firewall rules that block access to file sharing ports.

Directory traversal vulnerabilities are authorization weaknesses. See Section 4.4.3.3, “Directory Traversal Enabled.”

4.6.3 Web Server Access Control

For Web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

4.6.4 Database Access Control

Database access controls may need to be more specific than the generic roles-based rules. Access control checks should be based on the ICS’s functionality and business logic. For example, database access should be based on the record being accessed, not just by database user.

4.6.5 Execution with Unnecessary Privileges

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.

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.

4.6.6 Unnecessary Functionality

ICS applications, services, and protocols with unnecessary functionality prevent the implementation of least user privileges. Compartmentalization of functionality can help restrict individual ICS functions to the applications and users that require them. This can also help reduce required privileges by separating out the functionalities that do require elevated privileges.

4.6.7 Lack of Host Configuration Procedure

NSTB assessments are still encountering quality control issues related to configuration errors. Host configurations are still inconsistently deployed by ICS vendors. The installation, configuration, and patching of OSs, applications, services, and libraries varies by integrator or system administrator. When secure configuration documentation does exist, it is not always sufficiently detailed or followed.

A methodical and documented procedure should be created and used for configuring ICS components. Procedures should be customized for specific ICS components and functionality.

4.6.8 Recommendations for Permissions, Privileges, and Access Controls

Lock down the host environments as much as possible by individually restricting the privileges granted to user accounts, applications and services. Follow the principle of least privilege when assigning access rights to entities in a software system. Restrict allowable communication to that which is necessary and lower permission levels of users and applications to that necessary for their functions. Base access decisions on permission rather than exclusion. This means that, by default, access is denied and the protection scheme identifies conditions under which access is permitted.

ICS vendors can aid in this effort by following the principle of least privileges when designing and implementing their products.

Complete documentation and/or automated setup of security features should be provided 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.

4.6.8.1 References for Permissions, Privileges, and Access Controls

Improper Access Control (Authorization) is fifth on the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors* list.[3]

Execution with unnecessary privileges is a high-level root cause of many vulnerabilities. The third recommendation in the SANS/CWE software "Monster Mitigation" list addresses unnecessary privileges.[9]

4.7 Network Access Control Vulnerabilities

The lack of network segmentation into security zones passes up the opportunity to contain or slow network attacks as much as possible. Good network designs lock down the network environment by restricting host and user network permissions and access rights as much as possible, and segregating components into network security zones. Each security zone should include components that need to communicate and can be allowed the same trust levels. Components on the same network segment are effectively given the same level of trust.

Connections to ICS components located on the business network represent the same threat as any other business host connection. Host security levels may vary, but communication channels between network security zones are exposed to threats on both networks (and any intermediate networks).

4.7.1 Failure to Secure Network Device

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.

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. Table 40 shows network device configuration weaknesses.

Table 40. Network device configuration weaknesses.

Vulnerability	Impact
Network device configured for management over unsecure protocols	Management credentials can be sniffed off the network
Network device ACLs do not restrict by IP addresses	Device management access is not restricted
Network switch not configured with port security	Switch is not protected against physical connections

4.7.1.1 Network Device Recommendations

Network devices should be managed using two-factor authentication and encrypted sessions. Only true two-factor authentication mechanisms should be used, such as a password and a hardware token, or a password and biometric device. Requiring two different passwords for accessing a system is not two-factor authentication.

The network infrastructure should be managed across network connections that are separated from the business use of that network, relying on separate Virtual Local Area Networks (VLANs) or preferably relying on entirely different physical connectivity for management sessions for network devices.

Port security should be implemented to limit connectivity to hardware interfaces. Given the static nature of ICS environments, port security can 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.

4.7.2 Poor Network Design

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 rules should restrict traffic flow as much as possible. 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.

No outbound restrictions make the system vulnerable to indirect attack on connections that originated from the ICS.

4.7.2.1 Lack of Network Segmentation

The goal of network segmentation is to create security zones that provide access control by separating systems with different security and access requirements. Minimal or no security zones allow vulnerabilities and exploitations to gain immediate full control of the systems, which could cause high-level consequences. Backdoor network access is also not recommended and could cause direct access to ICS for attackers to exploit and take full control of the system. Table 41 shows problems found with ICS network designs.

Table 41. Unsecure network design findings.

Vulnerability	Impact
Single Point of Failure	Network DoS
Historian Server is on the Corporate LAN	Unnecessary exposure
Firewall Bypass (circumvented)	Unprotected attack path

Recommendation

Firewalls should be used to create DMZs to protect the ICS network. 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 12 shows this separation into multiple DMZs. All connections to the ICS LAN should be routed through the firewall. There should be no hardwired connections circumventing the firewall. Network administrators need to have an accurate network diagram of their ICS LAN and its connections to the other protected subnets, DMZs, corporate network, and external networks.

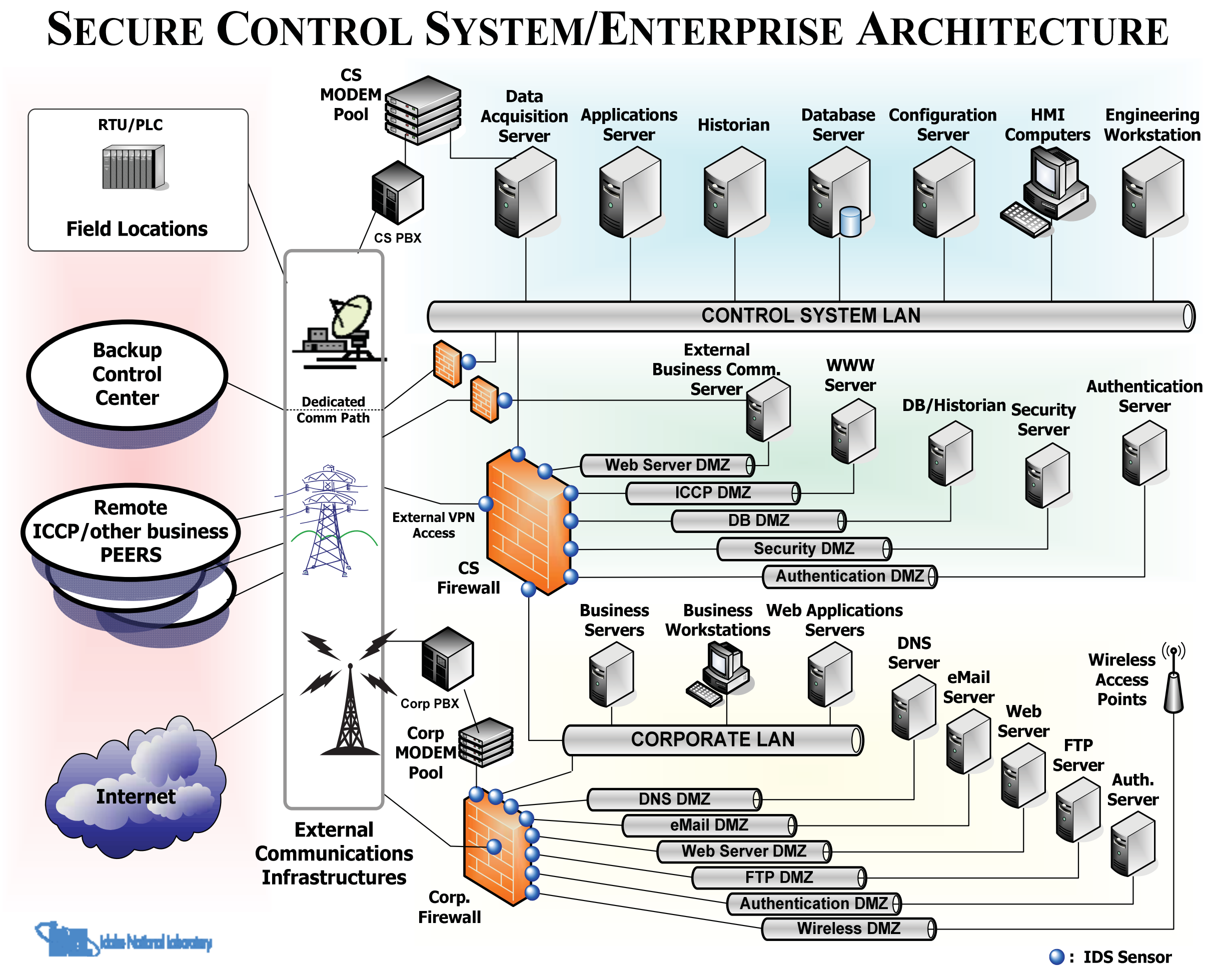

Figure 12. Recommended defense-in-depth ICS architecture.

4.7.2.2 Connections across Security Zones

Business applications require connections from the corporate network into to ICS. These connections create potential attack paths from the Internet onto the corporate network and then into the ICS networks. Exposure to the business network can be minimized to the necessary connections using perimeter protection techniques, but vulnerabilities in the protocols and services used for business functions can be exploited to gain access inside the ICS perimeter.

Business applications generally require connections to the historian database for access to historical data. They may also connect to the Web HMI server to allow real-time viewing of the process. Any connection to ICS functions extends the exposure of associated vulnerabilities to the corporate network (or where ever the connection is initiated from).

Data sharing protocols such as ICCP and OPC are utilized to send and receive data from remote sites and peer utilities. These connections must be treated as untrusted if the remote site or intermediate pathway networks are unknown.

The design of ICS protocols can force sub-optimal network designs and implementations. The use of protocols that require access to wide port ranges limits the ability to prevent unauthorized system access with firewall rules.

Vulnerabilities in services that must be allowed to accept connections from less-trusted networks must be left exposed to possible exploitation from these networks. Table 42 lists typical types of ICS services that must be exposed to possible attack from external networks.

Table 42. Major security weakness created by external communications.

Vulnerability	Impact
Business applications require holes through the firewall from the corporate network into the ICS networks	Increases exposure to the corporate network
Connections to remote sites	Increases exposure to less-trusted remote networks and the networks that provide the pathway
Data sharing protocols require connections to networks the ICS owner has no control over	Increases exposure to unknown external networks
ICS vendor and administrator VPN connections	

Protocol Design Reduces Firewall Effectiveness

Network protocols specify how information is packaged and sent across a computer network. Client and server applications are used to send and receive data that conforms to a given protocol. For every network protocol, an application (known as server) must wait for and process the data off the network. Corresponding client applications initiate communication sessions for that protocol. The client is able to identify the correct server to connect to by the port number on which it is listening. For example, common IT protocols use standard port numbers that all versions of server applications listen on. An FTP client knows to request a connection on Port 21.

Firewalls can restrict access on a host by specifying the port numbers of the applications that are allowed to accept connections. A host can be configured to only accept connections on the SSH Port 22, for example. This means that if an attacker wants to attack this host, it will have to be done by exploiting the SSH protocol, the SSH server installed on that machine, or an account that has privileges to establish SSH connections with that host.

Some ICS vendor proprietary protocols use ranges of port numbers for their servers. Firewall rules must then allow connections to system hosts on any of the port numbers in this range. If an attacker is able to gain access to the host, he could potentially download his own server and configure it to listen on one of these open ports.

Protocol Design Recommendations

Protocols and services that connect to less-trusted networks should have top priority in ICS software vulnerability remediation activities. Focus on input validation vulnerabilities.

ICS owners can reduce the risk of business application connections by minimizing exposure to the business network and closely monitoring the necessary communication paths. Web servers, Historian Databases, and other servers required for business functions should be placed on DMZs that have been segmented into security zones. Access rules should be as restrictive as possible. Restrict access to the required port numbers and IP addresses. Rules should be directional to prevent activities such as database connections from being initiated from the corporate network.

One way to prevent direct access to the ICS LAN from the corporate clients is by using a replicated data server in a DMZ as shown in Figure 13.

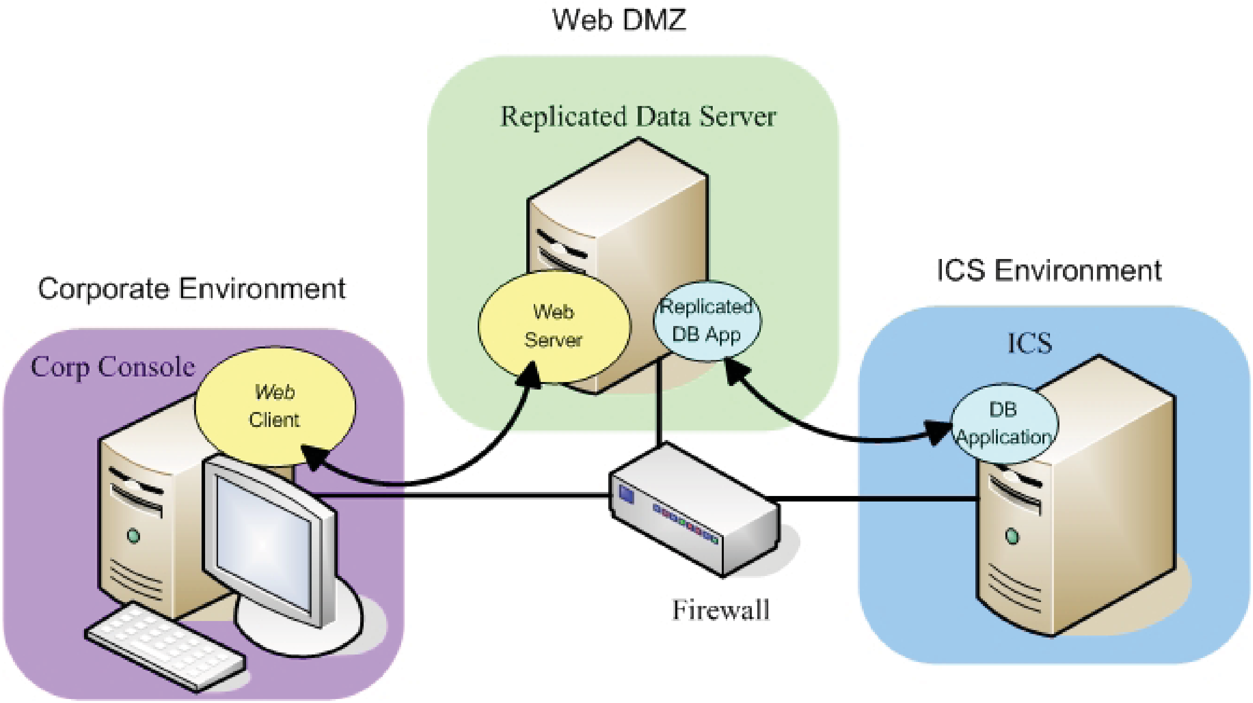

Figure 13. Replicated data server.

In this architecture, a Web server is located in a DMZ between the ICS and corporate networks. Replication of data from the ICS is accomplished by the data application running on the ICS server and the data application running on the Web server. The Web server then becomes a replicated data server, allowing corporate clients read-only access to ICS data.

4.7.3 Weak Firewall Rules

Firewall rules implement network segmentation. 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 with which it is configured. Enforcement of network access permissions, allowed message types, and content is executed by firewall rules.

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 sufficient outbound restrictions make the system vulnerable to indirect attack on connections that originated from the ICS. 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 that restrict access to specific ports, but not IP addresses, provide little protection. Another common detailed finding was that firewall rules allowed access to unused IP addresses traceable to legacy configuration of the firewall. This creates an attack path by using this IP address 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 not restricted by an IP address at all.

Assessment findings that fall under this vulnerability are firewall rules that are based on address groups, which include a wider range than should be allowed.

Table 43 lists specific assessment findings associated with overly permissive firewall rules.

Table 43. Unnecessary exposure from firewall rules.

Vulnerability	Impact
Lack of or improper segmentation into security zones	Unnecessary exposure from connected networks
Access to excessive number of ports is allowed	
Access to excessive number of IPs is allowed	
Lack of directional rules	
Out of date access control rules	
Lack of egress filtering	

4.7.3.1 Firewall Recommendations

A well configured firewall is critical to ICS security. Communications should be restricted to only what is 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.

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. 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. 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.

Not all assessment findings were related to system functionality. Many findings related to the IP addresses allowed to initiate connections between networks. 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. Access control lists should be used to limit management access of network equipment to only those who need it.

Rules should also consider the direction of network packets. Connections should normally not be initiated from less-trusted networks. Outbound connections should be filtered as well.

4.7.4 Poor Network Monitoring

Intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity, or availability of a resource. Intrusion detection does not include prevention of intrusions. Intrusion detection can be performed manually or automatically. Manual intrusion detection is done by examining log files or other evidence for signs of intrusions, including off normal network traffic. Automated approaches use intrusion detection systems to monitor system logs, network traffic flow, and packets. When a “probable intrusion” is identified, an alert is sent. Determining what the probable intrusion actually is and taking some form of action to stop it or prevent it from happening is outside the scope of intrusion detection systems. Intrusion prevention systems are not generally recommended along paths of critical functionality.

Intrusion detection was not a focus for laboratory assessments, but in many cases it was noted whether the system gave any indication of abnormal conditions, such as alerts on the operator screen. There was a general lack of adequate ICS indicators of abnormal conditions. Onsite assessments also found IDSs to be lacking in their installation, monitoring, and/or updating. Better indicators of abnormal system traffic and behavior should be built into operator screens. IDS systems should be deployed, tailored to the ICS architecture and traffic, and continuously monitored. ICS networks are generally static in nature and IDS rules can therefore be developed to look for abnormal behavior such as a protocol that should not be used between two computers.

4.7.5 Network Access Control Summary

Attackers take advantage of the fact that network devices may become less securely configured over time as the users demand exceptions for specific and temporary business needs, the exceptions are deployed, and those exceptions are often not removed when the business need is no longer applicable. Making matters worse, in some cases, the security risk of the exception is never properly analyzed, nor is this risk measured against the associated business need. Attackers search for electronic holes in firewalls, routers, and switches and use those to penetrate defenses. Attackers have exploited flaws in these network devices to gain access to target networks, redirect traffic on a network (to a malicious system masquerading as a trusted system), and to intercept and alter information while in transmission. Through such actions, the attacker gains access to sensitive data, alters important information, or even uses one compromised machine to pose as another trusted system on the network.

ICS customers need better and more concise information on how their system operates to guide the development of effective network isolation architectures and configurations. This is necessary to mitigate some of the identified vulnerabilities and others that may evolve. To this end, ICS vendors need to better identify and delineate all required ports and services necessary to support their product. This will better equip the end user with the tools needed for effective network isolation of their implementation of the ICS product.

4.7.5.1 Network Access Control References

ICS-specific network security recommendations can be found in the following references:

- 21 Steps to Improve Cyber Security of SCADA Networks[25]
- NIST SP 800-82, *Guide to Industrial Control Systems (ICS) Security*,[8] pages 5-1 to 5-19
- Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies[26]
- Control Systems Cyber Security: Defense in Depth Strategies[27]
- Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks.[28]

General network access control recommendations can be found in the *Twenty Critical Security Controls for Effective Cyber Defense*[13]

- Critical Control 4: Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
- Critical Control 5: Boundary Defense
- Critical Control 6: Maintenance, Monitoring, and Analysis of Audit Logs.

4.8 ICS Vulnerabilities Summary

NSTB assessments look for access paths to ICS resources and functionality. Failures to provide or configure defensive functionality in ICS applications, hosts, and networks are summarized in this section.

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; all indications show that the ICS vendor is not supporting third-party patch management for their software.

Weak or missing security features in ICS software leave the system components vulnerable to manipulation by any threats they are exposed. For the best defense, each component of the ICS must have its own protection mechanisms.

Table 44 lists the ICS software categories and vulnerabilities identified in multiple NSTB assessments.

Table 44. Summary of ICS software weaknesses.

ICS software weakness	Affected Components	Mitigations
Vulnerable third-party products integrated into ICS	ICS components that utilize third-party software (Web servers, databases, remote access services, protocol libraries, etc.)	Make sure ICS supports the latest OS, application, service, and library versions. Use vulnerability scanners and regularly apply security patches to ensure that the ICS product is not delivered with known vulnerabilities.
Improper Input Validation	All ICS components that handle external data (including data from other components)	Use a standard input validation mechanism. Start with services that are exposed to less-trusted networks and work inward.
Buffer Overflows	ICS software written in languages without memory management support such as C and C++	Validate input. Start with services that are exposed to less-trusted networks and work inward.
Poor Code Quality	All ICS software	ICS developers can use static analysis tools to identify and replace dangerous functions. ICS vendors should educate developers in secure coding and thoroughly test all components.
Permissions, Privileges, and Access Controls (Authorization)	ICS hosts, applications, and services	Restrict privileges needed by ICS users and services as much as possible and isolate functionalities. Deliver ICS with secured OS and application configurations. Document applications, services, and permissions required for each ICS component or functionality.
Lack of or Weak Authentication	Applications, protocols, and their network services	Authentication mechanisms should always require sufficiently complex passwords and require that they be periodically changed.
Lack of or Weak Integrity Checks	ICS network communication channels (status data and command messages, configuration downloads, etc.)	Design and implement integrity checks into ICS communication protocols. Monitor communication channels. Secure communication endpoints.

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 required that very few systems from each ICS product line are delivered before features are added and a new version is released. A large investment of financial and personnel resources needed for ICS upgrade contributes 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 OS, 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 and 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 insure 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.

In implementing the mitigations and prioritizing efforts for enhancing security, risk and consequence need to be considered. Security should balance the risk of system compromise by an intruder with the risk of potentially degrading system operability. Also, security solutions need to be practical enough for busy system administrators to implement and maintain. Above all, ICS must be reliable. Therefore, the suggested approach is to add security in small increments, using backup configurations, so that if any security measure conflicts with system operation it can quickly be reversed.

Security assessments of ICS products have identified problem areas associated with a lack of ICS vendor support in applying basic security best practices. Better vendor support is needed to remediate the unnecessary exposure and vulnerabilities caused by excessive services and unpatched systems. ICS software has not been designed for security, in general, which decreases the ability to reduce exposure by implementing least user privileges and firewall rules. The following common ICS security risks cannot be minimized by ICS owners alone. Table 45 summarizes the security vulnerabilities that cannot be completely remediated through perimeter defenses.

Table 45. Limitations of ICS security through perimeter defenses.

Common ICS Security Risks	Limitations of Perimeter Defenses
Poor Code Quality	Necessary protocols cannot be blocked Encryption does not fix vulnerabilities
Unpatched OS, Third Party Products, and Third Party Libraries	The ICS may not be compatible with the newer, or patched, versions
Least Privileges Violations	Privileges are required by the ICS products
Unneeded/Unused/Unsafe Services	Unnecessary services may be hard to infer if they have not been defined Unused services may not be unneeded in some circumstances
Poor Network Layout due to ICS Protocol Requirements	Protocol designs limit the effectiveness of network security mechanisms (i.e. large port ranges)
Unsecure Protocols	Necessary protocols cannot be blocked

Security best practices for the configuration and maintenance of applications, hosts, and networks are defense in depth measures to help prevent harm through access to ICS resources and functionality. These protective measures should be implemented wherever possible, but cannot be used to make up for vulnerabilities in the ICS design and implementation.

5. SUMMARY OF ICS SECURITY RECOMMENDATIONS

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 recognize 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. 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 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 (i.e., requiring password authentication and authorization should not hamper or interfere with emergency actions for ICS). All security solutions must not compromise critical functionality. Also, 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.

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. As many mitigation techniques as reasonably possible should be employed to stand in a line of defense to prevent access to vulnerable components and network traffic. The probability that an attack is able to defeat or circumvent security defenses is 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 cyber security 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 OSs and applications running on ICSs may not operate correctly with commercial-off-the-shelf IT cyber security solutions because of specialized ICS environment architectures.

Additional ICS-specific security resources can be found on the NSTB, United States Computer Emergency Readiness Team (US-CERT), and other Web sites:

- http://www.inl.gov/scada/publications/index.shtml
- http://www.us-cert.gov/control_systems/csdocuments.html

General cyber security resources are listed in Table 46. These and other resources can be used as references in securing an ICS. These references are not endorsed by DOE. This list is intended to help provide additional guidance and is not an inclusive list of cyber security resources. The following sections address ICS specific priorities and issues that should also be considered.

Table 46. General cyber security resources.

Resource	Location
Common Weakness Enumeration (CWE)[6]	http://cwe.mitre.org
2010 CWE/SANS Top 25 Most Dangerous Programming Errors[4]	http://cwe.mitre.org/top25
2010 CWE/SANS Top 25: Monster Mitigations[9]	http://cwe.mitre.org/top25/mitigations.html
Open Web Application Security Project (OWASP)	http://www.owasp.org
CERT Secure Coding Standards including[10] Top 10 Secure Coding Practices	https://www.securecoding.cert.org
Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines[13]	http://www.sans.org/critical-security-controls/

5.1 ICS Software Security Recommendations

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 from key business functions, such as product development, public relations, and legal. Common industry practice is to host 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 three most common problems are the lack of input validation, authentication, and access controls. The top 10 ICS vendor recommendations are listed, along with related references, and then discussed 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 IDSs so that they can be more effective.

5.1.1 Create a Security Culture

ICS vendors need to educate/train developers in secure coding and create a culture that emphasizes security.

The security development lifecycle (SDL), created by Microsoft in 2002 as a response to heightened awareness of cyber security threats, is a high-visibility example of a security culture change. This process was developed to catch security flaws during the product development lifecycle, 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 six years later, this culture change has made a significant difference in the security level of Microsoft products.

ICS products have gained considerable attention in recent years as the cyber security threats have been realized due to connection to the Internet. 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. Most important is a change in attitudes to a realization that security is important because it is associated with consequences for everyone. 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:

- Develop or acquire the necessary personnel security skills
- Define security requirements to protect critical functions
- Identify ICS component designs that violate security
- Develop secure design or redesign of identified components
- Require secure source coding handling to protect against malicious vulnerabilities
- Perform thorough security testing
- 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 oversight. 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.

5.1.2 Enhance ICS Test Suites

ICS product test suites should be enhanced to perform testing to failure with an emphasis on potential vulnerabilities. ICS software code logic has been found to only test for failures and other problems that may occur during normal operations.

The design and code logic of ICS products should handle 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 attacks that can cause errors that would never occur naturally or by accident. The possibility of malicious input requires logic that handles every possible error condition.

ICS test suites should include unconventional 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 NSTB 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 unconventional 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 of 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 lifecycle, and to support the needs of developers, testers and educators, the Common Attack Pattern Enumeration and Classification (CAPEC) 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."*[29]

5.1.3 Create and Test Patches

Security patches should be expeditiously tested and provided to affected customers. The necessary communication paths should be created that are needed to quickly notify customers of security problems and the methods needed to provide patches in an effective way should also be created. Currently, most ICS venders have insufficient methods of notifying customers about potential security problems and patches. Experience has shown that patches generated as the result of previous security assessments have been slow in being deployed with many end users unaware about the existence of the patches. ICS vendors should create and maintain security mailing lists and also 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. Owners should be proactive as well; actively inquiring about and watching for security notices.

Vendors should test and approve OS 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 quickly and easily as possible.

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

5.1.4 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 of protocol designs contain common security pitfalls.

A number of characteristics of a secure protocol are relevant to this discussion. These characteristics are:

- Secure protocols should be simple. The more complex a protocol is, the higher the likelihood of bugs and vulnerabilities within the implementation.
- Protocols should 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.
- 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.

- 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.
- 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.

5.1.5 Increase Robustness of Network Parsing Code

ICS developers need to dramatically increase the robustness of network parsing code. 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 DNP3. If these applications contain invalid input vulnerabilities such as buffer overflows, exploitation by anyone who is able to gain access to the ICS host and port is possible. Such action could cause a communication DoS, with an attacker gaining access to the computer with the privileges the account service was running, or other problems for the ICS.

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 to cause DoS attacks or 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.

5.1.6 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.

5.1.7 Document Necessary Services and Communication Channels

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

Complete documentation and/or automated setup of security features should be provided 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.

5.1.8 Implement and Test Strong Authentication and Encryption Mechanisms

Strong authentication and encryption mechanisms should be implemented and strenuously tested. 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.

5.1.8.1 Authentication and Encryption Development

Experienced personnel in authentication and encryption systems should be involved in creating authentication and encryption mechanisms. Authentication and encryption systems are complex and one small mistake or oversight can render the authentication or encryption ineffective. The authentication and encryption system should be tested rigorously to ensure the systems are working correctly before deploying the solutions.

A well-vetted encryption algorithm should be used that is currently considered to be strong by experts in the field, and select well-tested implementations. Software should be designed so that one cryptographic algorithm can be replaced with another, improving upgrade capability to stronger algorithms. ICS owners should periodically ensure that the current methods used have not been broken. Many old algorithms and implementations have become obsolete or discovered to be flawed.

5.1.8.2 Limitations and Risks of Encryption

Encryption solutions, such as IPSec and VPN tunneling, can be used for confidentiality, integrity, authenticity, and/or replay protection. They 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 cannot be prevented from compromising a VPN endpoint computer and using the VPN tunnel as an encrypted pathway to exploit vulnerabilities in the other endpoint host. This is true for any encrypted channel.

Encryption poses a risk to network throughput, bandwidth, availability, and IDS capabilities. ICS designers and administrators should carefully consider the priorities of each communication channel when implementing encryption. Difficulty of implementation and the inability to view traffic for trouble-shooting purposes are other issues that can prevent encryption from being used in operational ICS. Still, encryption should be used as a layer of defense where confidentiality or integrity is a higher priority than availability, i.e. external and non-critical connections.

5.1.8.3 Encryption Configuration and Maintenance

An appropriate encryption solution should be selected for each ICS communication channel that can handle the associated risk and support encryption. It needs to be configured securely and safely to support the ICS’s priorities.

Administrators need to securely manage and protect cryptographic keys. Keys should be strong and not 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. End-point 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.

A recent “trend” in the ICS industry has been to encrypt core ICS communications with IPSec. IPSec must be configured not to jeopardized critical communications. ICS hosts that require high availability must be configured with an IPSec “request” policy instead of “require”. This means that encryption is requested, not required. This is especially important in the configuration of the IPSec implementation included with Microsoft Windows XP and Windows Server 2003 and newer, because the identity proofing afforded by Active Directory 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. The decision for configuring this implementation of IPSec with a “request” policy versus a “require” policy should be made based on whether the highest priority for the communication between the IPSec partners is provided by encryption (confidentiality, integrity, authenticity, or replay protection) or is high availability.

5.1.9 Improve Security through External Software Security Assessments

ICS software vendors should conduct third-party security source code audits and then remediate the problems identified during the audits. 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, 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 publicly announced.

5.2 Secure ICS Installation and Maintenance

An effective cyber security 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 upon 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, it should be tested on a backup system first to allow quick recovery to the previous configuration in the event any security measure affects system operation. The risks should always be weighed and the appropriate amount of security measures added for the specific situation.

Asset owners must use procurement specifications to ensure that security development life-cycle requirements are met by the vendor. Also, asset owners may hire independent security assessment teams to review 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, IDS, 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 intrusion detection systems) 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 following recommendations:

- Redesign network layouts to take full advantage of firewalls, VPNs, etc.
- Implement a network topology for the ICS that has multiple layers, with the most critical communications occurring in the most secure and reliable layer
- Restrict physical access to the ICS network and devices
- Expeditiously deploy security patches after testing all patches under field conditions on a test system if possible, before installation on the ICS
- Work with vendor to test and apply patches for all operating systems and software on the ICS networks
- Customize IDSs for the ICS hosts and networks
- 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)
- 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.

5.2.1 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 him 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 required. 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.

5.2.2 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, integrators or administrators should 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/or the ICS LAN and also prevent authorized users from performing actions that cannot easily be attributed to them.

Administrators should not leave the default manufacturer passwords on ICS and networking equipment. Default passwords can give an attacker easy access to the equipment that controls the process. Owners should always change default passwords to robust, unpublished passwords. In the case that the software uses hardcoded passwords, ICS owners can work with the vendor to fix this vulnerability. They can then 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. Users will undermine the security of complicated or frequently changing passwords because they are too difficult to remember. 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 auto-generated, but instead created from passphrases or other memorable means.

5.2.3 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 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. Reducing 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 Section 5.1.3, “Create and Test Patches.”

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 conformation, any patch process test should be first performed on a backup or development system 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.

5.2.4 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. All connections should be routed 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.

Open ports and services that are not necessary provide a potential foothold or path for an attacker. The attacker can remotely connect to services listening on ports allowed through a firewall. All unneeded applications and services should be removed and then blocked by the firewall as well. In the event that a service is installed or enabled, this layer of defense will prevent connections to unauthorized services through the firewall.

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, closely monitor IDS logs during this period and immediately investigate unexpected communication.

5.2.5 Customize IDS Rules for the ICS and Closely Monitor Logs

The configuration and deployment of an 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 an 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 an 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 that domain are imperative when deploying IDS 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 utilized. Implementing a variety of these tools will help create a defense-in-depth architecture that will be more effective in identifying attacker activities.

5.2.6 Force Security through External Software Security Assessments

ICS customers can require a security audit of an ICS product and fixes 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 and/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 succeed at 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.

6. CONCLUSION

NSTB ICS security assessments evaluate ICS products and production configurations. ICS product assessments focus on vulnerabilities that are inherent in the product, and are therefore representative of installed systems. Production ICS assessments concentrate on the aspects of the ICS that the system owner is able to control, such as secure configurations and layers of defense.

An attacker must be able to access the ICS to do harm. From a cyber security perspective, this means that they must create an attack path from their attack computer to the ICS. An attack could potentially start from any point between the Internet and the physical equipment that the ICS is monitoring. Layers of defense are necessary for protection against multiple threat vectors, but perimeter protection cannot fully mitigate vulnerabilities that exist in the ICS.

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 version; all indications show that the ICS vendor is not supporting third-party patch management for their products.

Vendor support is needed to remediate the unnecessary exposure and vulnerabilities caused by excessive services and unpatched systems. ICS software has not been designed for security, in general, which decreases the ability to reduce exposure by implementing least user privileges and firewall rules. The following common ICS security risks cannot be minimized by ICS owners alone:

- Unpatched OS, third-party products, and third-party libraries
- Unneeded/unused/unsafe services
- Poor network layout due to ICS protocol requirements
- Privilege levels.

A defense in depth approach to securing ICSs includes identifying and remediating existing vulnerabilities in current ICS products, developing secure new products, and supporting patching and secure configurations of ICS components. ICS owners can then install, maintain, and monitor secure OS, software, ICS, and network configurations.

7. REFERENCES

1. SANS, *The Top Cyber Security Risks*, SysAdmin, Audit, Network, Security, September 2009, http://www.sans.org/top-cyber-security-risks/, Web page accessed May 2010.

2. Mell, Peter, Scarfone, Karen, and Romanosky, Sasha, *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*, Forum of Incident Response and Security Teams, June 2007, http://www.first.org/cvss/cvss-guide.html, Web page accessed May 2010.

3. Christey, Steve, *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*, MITRE, April 5, 2010, http://cwe.mitre.org/top25/, Web page accessed May 2010.

4. Lee, Kathy, et al., *NSTB ICCP Security Assessment*, U.S. Department of Energy Office of Electricity Delivery and Energy Reliability, February 2010.

5. ISA, *ANSI/ISA–99.00.01–2007 Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models*, International Society for Automation, October 29, 2007, http://www.isa.org/filestore/expo/2009/PressKit/Information%20about%20ISA/Membership/Samples%20of%20Free%20ISA%20Standards%20and%20Technical%20Papers/ANSI%20ISA%2099-00-01%20%202007.pdf, Web page accessed May 2010.

6. MITRE*, CWE (Common Weaknesses Enumeration),* Department of Homeland Security, January 11, 2009, http://cwe.mitre.org/, Web page accessed May 2010.

7. DHS, *Recommended Practice for Patch Management of Control Systems*, Department of Homeland Security, http://csrp.inl.gov/Documents/PatchManagementRecommendedPractice_Final.pdf, December 2008.

8. NIST, *NIST SP 800-82, Guide to Industrial Control Systems (ICS) Security, Final Public Draft*, National Institute of Standards and Technology, September 29, 2008.

9. SANS/CWE, *2010 CWE/SANS Top 25: Monster Mitigations*, SysAdmin, Audit, Network, Security, February 15, 2010, http://cwe.mitre.org/top25/mitigations.html, Web page accessed May 2010.

10. Seacord, Robert*, CERT Secure Coding Standards*, Carnegie Mellon University, June 7, 2010, https://www.securecoding.cert.org, Web page accessed May 2010.

11. SAFECode, *Software Assurance: An Overview of Current Industry Best Practices*, Software Assurance Forum for Excellence in Code, February 2008, http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf, Web page accessed May 2010.

12. SAFECode, *Fundamental Practices for Secure Software Development*, Software Assurance Forum for Excellence in Code, October 2008, http://www.safecode.org/publications/SAFECode_Dev_Practices1108.pdf, Web page accessed May 2010.

13. CSIS, *Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines*, Center for Strategic and International Studies, August 10, 2009, http://csis.org/files/publication/Twenty_Critical_Controls_for_Effective_Cyber_Defense_CAG.pdf, Web page accessed May 2010.

14. Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S., *DNS Security Introduction and Requirements*, RFC 4033, March 2005, http://www.faqs.org/rfcs/rfc4033.html, Web page accessed May 2010.

15. Wright, Jason, *Control Systems Communications Encryption Primer*, Department of Homeland Security, December 2009, http://www.us-cert.gov/control_systems/pdf/Encryption%20Primer%20121109.pdf, Web page accessed May 2010.

16. Frankel, S., et al., *Guide to SSL VPNs*, National Institute of Standards and Technology, July 2008, http://csrc.nist.gov/publications/nistpubs/800-113/SP800-113.pdf, Web page accessed May 2010.

17. Rolston, Bri, *Attack Methodology Analysis: SQL Injection Attacks*, United States Computer Emergency Readiness Team, September 2005, http://www.inl.gov/technicalpublications/Documents/3395025.pdf, Web page accessed May 2010.

18. Friedl, Steven, *SQL Injection Attacks by Example*, October 10, 2007, http://www.unixwiz.net/techtips/sql-injection.html, Web page accessed May 2010.

19. OWASP, *OWASP Top 10-2010 The Ten Most Critical Web Application Security Risks*, Open Web Application Security Project, April 2010, http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project, Web page accessed May 2010.

20. OWASP, *OWASP Cheat Sheets,* Open Web Application Security Project, June 22, 2009, http://www.owasp.org/index.php/Category:Cheatsheets, Web page accessed May 2010.

21. DHS, *DHS Recommended Practice Case Study: Cross-Site Scripting,* Department of Homeland Security, February 2007, http://www.us-cert.gov/control_systems/practices/documents/xss_10-24-07_Final.pdf, Web page accessed May 2010.

22. SANS, *Password Policy*, SysAdmin, Audit, Network, Security, 2006, http://www.sans.org/security-resources/policies/Password_Policy.pdf, Web page accessed May 2010.

23. SANS, *DB Password Policy*, SysAdmin, Audit, Network, Security, 2006, http://www.sans.org/security-resources/policies/DB_Credentials_Policy.doc, Web page accessed May 2010.

24. Microsoft, *How to Prevent Windows from Storing a LAN Manager Hash of Your Password in Active Directory and Local SAM Databases*, December 3, 2007, http://support.microsoft.com/kb/299656, Web page accessed May 2010.

25. DOE-OE, *21 Steps to Improve Cyber Security of SCADA Networks*, U.S. Department of Energy Office of Electricity Delivery and Energy Reliability, http://www.oe.netl.doe.gov/docs/prepare/21stepsbooklet.pdf, Web page accessed May 2010.

26. DHS, *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies*, Department of Homeland Security, October 2009, http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf, Web page accessed May 2010.

27. Idaho National Laboratory, *Control Systems Cyber Security: Defense in Depth Strategies*, Homeland Security External Report # INL/EXT-06-11478, May 2006, http://csrp.inl.gov/Documents/Defense%20in%20Depth%20Strategies.pdf, Web page accessed May 2010.

28. *CPNI, Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks*, National Infrastructure Security Coordination Centre, London, 2005, http://www.cpni.gov.uk/docs/re-20050223-00157.pdf, Web page accessed May 2010.

29. MITRE, *Common Attack Pattern Enumeration and Classification (CAPEC)*, http://capec.mitre.org/, May 18, 2010.

Appendix A

NSTB Assessment Methodology

Appendix A

NSTB Assessment Methodology

NSTB assessments target core supervisory control components using typical attack vectors. NSTB assessments have focused on the ICS products to identify and understand the vulnerabilities they are most affected by, and how their design and operational requirements affect host and network security.

NSTB product assessments focus on the core components of new ICS products. This includes the custom software components that relay commands to control hardware, provide system state data, store historical data, and provide other supervisory control and management functions. Common computer software products are integrated into these complex systems, such as Web servers, database applications, and remote access and file transfer services. Because supervisory control software usually only supports one or two operating systems, which are generally installed and configured by a vendor integrator, operating systems can also be thought of as integrated into the ICS.

Security information about common IT operating systems, applications, services, and network protocols is widely available, as well as secure configuration guides. NSTB assessments look for known vulnerabilities in these components or configuration errors that can be exploited to gain access to ICS components or manipulate the system. Widely known vulnerabilities and configuration errors represent the most likely attack paths of an ICS because the information and tools for discovering and exploiting them are publicly available.

To cause damage, an ICS cyber threat must compromise an ICS component or network traffic with the ability to control the physical system or alter, insert, or delete system operational status data.

ICS software is evaluated for vulnerabilities that would allow access to critical ICS functionalities. Unsecured protocols that transfer system state data and commands, or are used for communication channels between security zones, are evaluated for vulnerabilities that would allow manipulation or spoofing of system communication messages, DoS of system communication, or information gathering.

Programming errors are identified in ICS applications that can be exploited for unauthorized access, privilege escalation, data manipulation, DoS attacks, etc. Server applications, that parse network traffic, are top priority because of their exposure to the network.

ICS user interface applications are evaluated for weak authentication or other vulnerabilities that would allow unauthorized access to system diagrams and monitoring and control functionalities.

ICS and OS user accounts, services, and applications are evaluated for unnecessary privileges to files and ICS and OS commands. The lack of compartmentalization of ICS functionalities and user accounts makes it hard to contain an attacker who has gained access to a system component.

NSTB assessments evaluate ICS installation network defenses, as well as ICS vendor network recommendations. They test the effectiveness of network designs and implementations at preventing unauthorized traffic to and from ICS networks. The ability of the network defense strategy to effectively filter and monitor traffic, given the ICS system design, is also evaluated.

The most common and significant ICS vulnerability types are described in this report. The information is presented at a high level to facilitate reporting and understanding of the major ICS security issues without disclosing system-specific details. Vulnerabilities are derived from NSTB ICS security assessments of varying subsets of components and functionalities, ranging from minimal supervisory control test systems to full production systems used for electric power generation and transmission. Assessment results are the vulnerabilities discovered using typical attack methodologies, in the allotted timeframe. Attack targets vary, but always support the goal of creating an attack path through necessary communication channels and manipulating or disrupting system operations. Table A1 shows high level, generic ICS security assessment targets.

Table A1. Generic ICS security assessment targets.

Assessment Target	Targeted Component	Methods
Identify Known Vulnerabilities and Listening Services	Unauthorized access to ICS hosts and applications	Vulnerability and port scans Common attack tools
Evaluate Communication Channels	Network traffic	MitM Analyze network traffic Reverse engineer protocol Spoof, drop, or alter messages
Evaluate Network Services	Server applications (aka protocol implementations)	Network fuzzing Reverse engineer binaries Code reviews
Evaluate Authentication Mechanisms	Applications and services used for ICS operations	Penetration testing Analyze network traffic
Evaluate Security Configurations	User accounts, services, and applications	Evaluate user accounts Evaluate permissions and access controls Evaluate credentials management
Evaluate Network Defenses	Network device configurations and firewall rules	Traffic captures and analysis Production network diagrams, ACLs, firewall rules and Intrusion Detection System (IDS) signatures are reviewed and discussed with the network administrator

A-1. Reporting Methodology

ICS product assessments focus on vulnerabilities that are inherent in the product, and are therefore representative of installed systems. The reporting standard is to only report configuration and password findings if they are representative of production system settings. Network architecture and firewall rules are only assessed if they are provided as recommended configurations.

The attacker must be able to access the ICS to do harm. From a cyber security perspective, this means that they must create an attack path from their attack computer to the ICS. An attack could potentially start from any point between the Internet and the physical equipment that the ICS is monitoring. Layers of defense are necessary for protection against multiple threat vectors.

Any computer that is connected to the Internet, directly or indirectly, is a potential risk for an attack from viruses or external attackers. An attack initiated from the Internet must create a path to the ICS network. The number of possible paths to the target is the system's exposure. ICSs are generically exposed to attack through connections to the corporate network for business functions, connections to peers (i.e., ICCP connections), connections to remote sites, remote access allowed to vendors, system administrators and operators, and connections to field equipment. Insider threats have a shorter attack path based on their access level.

Production ICS assessments (i.e. on-site assessments) concentrate on the aspects of the ICS that the system owner is able to control, such as secure configurations and layers of defense. The assessment team only performs penetration testing on disconnected backup or development systems.

The ICS network administrators review and discuss production network diagrams, ACLs, firewall rules, and IDS signatures with the assessment team. They can then perform hands-on assessments of ICS and network component configurations together. This includes a review and tour of the production system to help identify through documentation, observation, and conversation any possible security problems with the production system and network configuration without putting the operational (production) system at risk. This is a learning opportunity for both the assessment team and the asset owner personnel.

The NSTB approach has always been to assess ICS security and educate vendors and owners on how they can make their systems more secure. The granularity of report findings depends on the nature of the problem, the time allocated for that target, and how widespread the problem is. For example, some NSTB ICS security assessments identified general security problems, such as the use of unsecure C functions, and then demonstrated that they could be exploited by creating an exploit for at least one example of the problem. The wording used in reports for this type of finding is similar to:

> *"Buffer overflow in the specified application allows a remote attacker to execute arbitrary code and gain full control of the ICS host it runs on. This is caused by the use of unsecure C functions such as strcpy, etc. Other buffer overflow vulnerabilities were identified in this and other applications. Replace all instances of dangerous C functions with their safe alternatives."*

NSTB report findings are mapped to software weakness types defined by the CWE to the extent possible. Findings are reported as CWEs to aid in the understanding of ICS vulnerabilities. ICS vendors and asset owners can refer to the CWE for additional guidance in identifying, mitigating, and preventing weaknesses that cause vulnerabilities.[6]

The common weaknesses in this report are similar security weaknesses found on two or more unique ICS configurations. Findings that mapped to very specific CWEs are reported as a higher level CWE that describes multiple similar weaknesses. Weaknesses are then categorized in various ways to illustrate when they were created and the types of ICS components they were found in.

Appendix B

Vulnerability Scoring

Appendix B

Vulnerability Scoring

The most significant vulnerabilities identified in ICS are those that allow unauthorized control of the physical system. Compromise of the ICS's availability and ability to function correctly may also have significant consequences.

Likelihood of a successful attack must also be considered when assessing risk. Exposure to attack, attacker awareness of the vulnerability, and exploitation knowledge help assess the probability of a successful attack.

B-1. CVSS Version 2.0 Metrics

Generic ICS vulnerabilities are scored in this report using the Common Vulnerability Scoring System Version 2.0 (CVSS v2) and the most common or highest impact characteristics. CWE characterization of weaknesses were used where appropriate as well. The following CVSS v2 scoring criteria are taken from the CVSS Scoring Guide.[2]

B-1.1 CVSS v2 Base Metrics

The base metric group captures the characteristics of a vulnerability that are constant with time and across user environments. The Access Vector, Access Complexity, and Authentication metrics capture how the vulnerability is accessed and whether or not extra conditions are required to exploit it. The three impact metrics measure how a vulnerability, if exploited, will directly affect an Information Technology (IT) asset, where the impacts are independently defined as the degree of loss of confidentiality, integrity, and availability. CVSS v2 base scoring metrics are summarized in Table B1.

Table B1. CVSS v2 base scoring metrics.

Base Metrics	Metric Value	Metric Description
Access Vector	Local	Requires the attacker to have either physical access to the vulnerable system or a local (shell) account.
	Adjacent Network	Requires the attacker to have access to either the broadcast or collision domain of the vulnerable software, local IP subnet, for example.
	Network	The vulnerable software is bound to the network stack and the attacker does not require local network access or local access, aka "remotely exploitable."
Access Complexity	High	Specialized access conditions exist.
	Medium	The access conditions are somewhat specialized.
	Low	Specialized access conditions or extenuating circumstances do not exist.
Authentication	Multiple	Exploiting the vulnerability requires that the attacker authenticate two or more times, even if the same credentials are used each time.
	Single	The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or Web interface).
	None	Authentication is not required to exploit the vulnerability.
Confidentiality Impact	None	There is no impact to the confidentiality of the system.
	Partial	There is considerable informational disclosure.
	Complete	There is total information disclosure, resulting in all system files being revealed.
Integrity Impact	None	There is no impact to the integrity of the system.
	Partial	Modification of some system files or information is possible, but the attacker

Table B1. (continued).

Base Metrics	Metric Value	Metric Description
		does not have control over what can be modified, or the scope of what the attacker can affect is limited.
	Complete	There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.
Availability Impact	None	There is no impact to the availability of the system.
	Partial	There is reduced performance or interruptions in resource availability. An example is a network-based flood attack that permits a limited number of successful connections to an Internet service.
	Complete	There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.

B-1.2 CVSS v2 Temporal Metrics

The temporal exploitability metric measures the current state of exploit techniques or code availability. Public availability of easy-to-use exploit code increases the number of potential attackers by including those who are unskilled, thereby increasing the severity of the vulnerability.

The effectiveness of available work-around mitigations is used to adjust the temporal score. CVSS temporal scoring metrics are summarized in Table B2.

Table B2. CVSS v2 temporal scoring metrics.

Temporal Metrics	Metric Value	Metric Description
Exploitability	Unproven	No exploit code is available, or an exploit is entirely theoretical.
	Proof-of-Concept	Proof-of-concept exploit code or an attack demonstration that is not practical for most systems is available. The code or technique is not functional in all situations and may require substantial modification by a skilled attacker.
	Functional	Functional exploit code is available. The code works in most situations where the vulnerability exists.
	High	Either the vulnerability is exploitable by functional mobile autonomous code, or no exploit is required (manual trigger) and details are widely available. The code works in every situation, or is actively being delivered via a mobile autonomous agent (such as a worm or virus).
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.
Remediation Level	Official Fix	A complete vendor solution is available. Either the vendor has issued an official patch, or an upgrade is available.
	Temporary Fix	There is an official but temporary fix available. This includes instances where the vendor issues a temporary hotfix, tool, or workaround.
	Workaround	There is an unofficial, non-vendor solution available. In some cases, users of the affected technology will create a patch of their own or provide steps to work around or otherwise mitigate the vulnerability.
	Unavailable	There is either no solution available or it is impossible to apply.
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.
Report Confidence	Unconfirmed	There is a single unconfirmed source or possibly multiple conflicting reports. There is little confidence in the validity of the reports. An example is a rumor that surfaces from the hacker underground.

Table B2. (continued).

Temporal Metrics	Metric Value	Metric Description
	Uncorroborated	There are multiple non-official sources, possibly including independent security companies or research organizations. At this point there may be conflicting technical details or some other lingering ambiguity.
	Confirmed	The vulnerability has been acknowledged by the vendor or author of the affected technology. The vulnerability may also be Confirmed when its existence is confirmed from an external event such as publication of functional or proof-of-concept exploit code or widespread exploitation.
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.

B-1.3 CVSS v2 Environmental Metrics

Different environments can have an immense bearing on the risk that a vulnerability poses to an organization and its stakeholders. The CVSS v2 environmental metric group captures the characteristics of a vulnerability that are associated with a specific environment. For this report, generic ICS security requirements are used to score generic ICS vulnerabilities.

Security requirements metrics enable ICS owners to customize the CVSS v2 score depending on the importance of the affected component to their own organization, measured in terms of confidentiality, integrity, and availability. DoS vulnerabilities in ICS components that require high availability will receive higher criticality scores than they otherwise would. The effectiveness of available work-around mitigations is used to adjust the temporal score. CVSS v2 environmental scoring metrics are summarized in Table B3.

Table B3. CVSS v2 environmental scoring metrics.

Environmental Metrics	Metric Value	Metric Description
Collateral Damage Potential	None	There is no potential for loss of life, physical assets, productivity or revenue.
	Low	A successful exploit of this vulnerability may result in slight physical or property damage. Or, there may be a slight loss of revenue or productivity to the organization.
	Low-Medium	A successful exploit of this vulnerability may result in moderate physical or property damage. Or, there may be a moderate loss of revenue or productivity to the organization.
	Medium-High	A successful exploit of this vulnerability may result in significant physical or property damage or loss. Or, there may be a significant loss of revenue or productivity.
	High	A successful exploit of this vulnerability may result in catastrophic physical or property damage and loss. Or, there may be a catastrophic loss of revenue or productivity.
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.
Target Distribution	None	No target systems exist, or targets are so highly specialized that they only exist in a laboratory setting. Effectively 0% of the environment is at risk.
	Low	Targets exist inside the environment, but on a small scale. Between 1% and 25% of the total environment is at risk.
	Medium	Targets exist inside the environment, but on a medium scale. Between 26% and 75% of the total environment is at risk.

Table B3. (continued).

Environmental Metrics	Metric Value	Metric Description
	High	Targets exist inside the environment on a considerable scale. Between 76% and 100% of the total environment is considered at risk.
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.
Security Requirements	Low	Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
	Medium	Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
	High	Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).
	Not Defined	Assigning this value to the metric will not influence the score. It is a signal to the equation to skip this metric.