INL/EXT-10-18381

Vulnerability Analysis of Energy Delivery Control Systems

September 2011

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.

EXECUTIVE SUMMARY

A serious and ongoing challenge

Cybersecurity for energy delivery systems has emerged as one of the Nation’s most serious grid modernization and infrastructure protection issues. Cyber adversaries are becoming increasingly targeted, sophisticated, and better financed. The Stuxnet worm—designed to attack a specific control system similar to those found in some energy sector applications—underscores the seriousness of targeted cyber attacks on energy control systems. The energy sector must research, develop and deploy new cybersecurity capabilities faster than the adversary can launch new attack tools and techniques.

> “Industry estimates that the production of malware has reached its highest levels, with an average of 60,000 new pieces identified per day…”[32.]
>
> *-James R. Clapper, Director of National Intelligence, February 10, 2011*

Cybersecurity technologies that are developed to protect business IT computer systems and networks can break an energy delivery control system. The computers and networks that control our Nation’s power grid are very different from those on our desks. Energy delivery control systems are uniquely designed and operated to control real-time physical processes that deliver continuous and reliable power to support national and economic security. These differences must be respected when securing these systems or the cybersecurity protective measures themselves could create a power disruption that rivals that of an intentional cyber-attack.

The goal of the U.S. Department of Energy Office of Electricity Delivery and Energy Reliability (DOE/OE) National Supervisory Control and Data Acquisition (SCADA) Test Bed (NSTB) program is to enhance the reliability and resiliency of the Nation’s energy infrastructure by reducing the risk of energy disruptions due to cyber attacks. A key part of the program is SCADA system vulnerability analysis that identifies and provides mitigation approaches for vulnerabilities that could put these systems at risk. A cybersecurity 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 actor or is present in conjunction with particular events or circumstances.

In 2006, DOE collaborated with energy owners and operators to develop a strategy to secure energy control systems going forward, made available through the *Roadmap to Secure Control Systems in the Energy Sector.* In 2011 the Roadmap was updated to keep pace with advances in technology and the evolving threat landscape, and renamed the *Roadmap to Achieve Energy Delivery Systems Cybersecurity.* The Roadmap lays out a vision that by 2020, resilient energy delivery systems are designed, installed, operated and maintained to survive a cyber incident while sustaining critical functions. One of the Roadmap strategic directions needed to achieve this vision is to assess and monitor risk, which is the subject of this report.

About this report

This *Vulnerability Analysis of Energy Delivery Systems* report describes common vulnerabilities found in assessments performed from 2003 to 2010 by Idaho National Laboratory (INL) on behalf of the DOE/OE NSTB program. INL performs cybersecurity assessments on energy sector supervisory control and data acquisition/energy management systems (SCADA/EMS, hereafter referred to as SCADA) under private sector and government programs. The vulnerabilities described in this report were routinely discovered in NSTB assessments using a variety of typical attack methods to manipulate or disrupt system operations.

The purpose of this report is to provide recommendations to the SCADA vendor and/or owner to identify and reduce the risk of these vulnerabilities in their systems. Vulnerabilities and mitigation recommendations are presented at a high-level to provide awareness of the most common and significant SCADA security vulnerability areas without divulging product-specific information. The common vulnerabilities in this report were found on two or more unique SCADA configurations. Even though SCADA functions, designs, and configurations vary among vendors, versions, and installations, their high-level vulnerabilities and defensive recommendations are similar.

This report begins with a history, purpose, and scope of NSTB vulnerability analysis. Section two describes the standard terminology and proposes a metrics-based approach to evaluate the relative risk associated with common SCADA vulnerabilities and guide risk reduction efforts. Section three discusses several ways to categorize vulnerabilities found in NSTB assessments, and presents the relative frequency of vulnerabilities observed in NSTB assessments using each of these categorizations. Section four presents NSTB detailed vulnerability assessment findings and recommended mitigations with references to further related security information. Finally, section five summarizes general security recommendations for SCADA developers and administrators to help mitigate the risk posed by common vulnerabilities found in NSTB SCADA cybersecurity assessments.

Method of investigation

The NSTB assessment process is highly flexible and is tailored to the mutual interests of the industry partner and the NSTB program. The granularity of report findings depends on the nature of the problem, the time allocated for that target, and how widespread the problem is. More information about the NSTB assessment methodology is provided in Appendix A, "NSTB Assessment Methodology." The following summary describes the main aspects of the process used for conducting the NSTB vulnerability assessments involved in this report:

- SCADA product assessments targeted core components using typical attack vectors to identify and understand the vulnerabilities they may be most affected by, and how their design and operational requirements could affect host and network security. The assessments focused on vulnerabilities that were inherent in the product, and were therefore representative of installed systems. Configuration and password findings were reported only if they were representative of production system settings. Network architecture and firewall rules were only assessed if they were provided as recommended configurations.

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

- NSTB report findings are mapped to software weakness types defined by the Common Weakness Enumeration (CWE) to the extent possible. Findings are reported as CWEs so that SCADA vendors and owners can refer to the CWE for additional guidance in identifying, mitigating, and preventing weaknesses that cause vulnerabilities.[9]

In this report, we introduce a standard metrics-based approach to evaluate the relative risk associated with common SCADA vulnerabilities. We used the Common Vulnerability Scoring

System Version 2 (CVSS) and CWE methodologies to apply standard terminology and metrics-based scores to common SCADA vulnerabilities.

While it is not possible to quantify an absolute measurement of risk, it can be useful to apply a formal, structured process that characterizes relative risk associated with a particular vulnerability taking into account the environment within which that vulnerability exists. The CVSS metrics evaluate vulnerabilities so that higher risk vulnerabilities can be mitigated first, using mitigations that reduce CVSS scores, and the risk these scores reflect, the most. This report recommends mitigations that reduce risk of cyber-attack and will consequently lower CVSS scores in SCADA installations where these mitigations are implemented.

In appendix C, we have scored each of the ten vulnerabilities identified by SANS, The Top Cyber Security Risks,[1] according to the CVSS process. We used input values to the CVSS metrics that we found to be the most common values observed in NSTB assessments to create a generic score.

It is important to recognize that the generic CVSS scores found in Appendix C do not describe the risk to a particular SCADA configuration because each system installation is unique and the same vulnerability in different environments poses different risks. Technical mitigations, operating procedures and cybersecurity policies in place in different operational environments will result in different, in some cases very different, CVSS scores.

Key findings: common SCADA vulnerabilities identified in NSTB assessments

The NSTB has seen a significant improvement in operating system (OS) and network security since 2003. Some improvement has been observed in reducing host exposure by reducing the number of available ports and services on SCADA hosts and in vulnerability remediation and secure development of new products. However, vulnerabilities caused by less secure coding practices can be found in new and old products alike, and the introduction of Web applications into SCADA systems has created more, as well as new, types of vulnerabilities. The 10 most significant cybersecurity risks identified during NSTB software and production SCADA assessments are listed in Table EX-1.

Table EX-1. Ten common vulnerabilities identified in NSTB assessments.

Common vulnerability	Reason for concern
Unpatched published known vulnerabilities	Most likely attack vector
Web Human-Machine Interface (HMI) vulnerabilities	Supervisory control access
Use of vulnerable remote display protocols	Supervisory control access
Improper access control (authorization)	SCADA functionality access
Improper authentication	SCADA applications access
Buffer overflows in SCADA services	SCADA host access

Common vulnerability	Reason for concern
SCADA data and command message manipulation and injection	Supervisory control access
SQL injection	Data historian access
Use of standard IT protocols with clear-text authentication	SCADA host access
Unprotected transport of application credentials	SCADA credentials gathering

Recommendations for mitigating vulnerabilities identified in NSTB assessments

NSTB assessments identify and analyze SCADA system vulnerabilities that could allow unauthorized access to SCADA hosts, applications, and data, or unauthorized manipulations that affect operations, that spoof or manipulate SCADA data and commands or that impose a DoS that could impede communications and jeopardize SCADA functionality. Recommended mitigations, organized according to the location of the vulnerability within the energy delivery control system architecture, include:

- **The SCADA cyber-attack surface** is protected through secure code and removal of unneeded ports and services. The attack surface comprises all possible avenues of attacking a system. All open ports, installed services and applications that can potentially be exploited create the attack surface. Recommendations:
    - Design and implement secure code to protect against vulnerabilities, such as buffer overflow, structured query language (SQL) injection, cross-site scripting (XSS), and directory traversal
        - Replace potentially dangerous functions with safe counterparts
        - Validate input data
    - Minimize the number of open ports, installed services and applications
- **Known vulnerabilities** are mitigated through effective patch management and removal of unneeded applications and services. New vulnerabilities in computer applications and services are found every day. Some are published shortly after their discovery. Others are kept a close secret by those who discover them. These remain unpatched and can be exploited at will by their discoverers. Recommendations:
    - Implement effective patch management
    - Remove all unneeded applications and services
- **Communication channel vulnerabilities** are mitigated through protected transmission of authentication credentials, secure control of local and remote access and SCADA data integrity checks**.** As SCADA systems become increasingly connected to company intranets and to the external Internet, they can also become more exposed to cyber attack. SCADA communication channels may use common IT communication protocols that provide common IT functionality in SCADA systems, as well as SCADA communication protocols to transmit SCADA data and command messages. They often connect different network security

zones and may have access rights and functionality to manipulate the SCADA system. Recommendations:

  - Rigorously protect authentication credentials during transmission
  - Implement secure communications practices in utilization of common IT protocols for local and remote access
  - Detect data corruption or manipulation by performing SCADA data integrity checks that are designed to prevent recalculation
  - Consider the benefits and challenges of implementing data encryption for SCADA systems

- **Communication endpoint vulnerabilities** are mitigated through secure coding practices that enforce rigorous input data validation in SCADA and ICCP services, database applications and web services**.** Network services at communication end-points listen for messages to accept, and can be exposed to attacks that exploit input and output validation vulnerabilities to gain unauthorized access to their host. Recommendations:
  - Implement secure coding practices that enforce rigorous input data validation in SCADA and ICCP services, database applications, and web services
- **Authentication vulnerabilities** are mitigated through authentication at both the server and the client, and effective management of authentication credentials. Authentication is the act of validating the identity of a person or a process requesting access, and is used to enforce access controls. Recommendations:
  - Implement authentication at both the server and the client
  - Manage authentication credentials, that is, change default passwords, use strong passwords, implement an effective password policy, and rigorously protect authentication credentials
- **Authorization vulnerabilities** are mitigated through least-privileges access control, removal of unneeded functionality and secure configuration of SCADA components. Authorization is the act of validating access rights through controls that restrict access to entities in a network, host, or software system and can be incorporated into SCADA components to help prevent and contain compromise. If an attacker gains full access to a host, all functions that the server can execute can be under the attacker's control. In addition, host access gives the attacker access to the resources of the compromised server, including communications with other devices and servers. Recommendations:
  - Implement least-privilege access control
  - Limit functionality only to that required, which enables the ability to implement least-privilege access control
  - Implement a secure design of SCADA components that compartmentalizes functionality
- **Network access vulnerabilities** are mitigated through network segmentation, strong firewall rules that enforce secure connections across security zones and intrusion detection**.** Attackers can search for vulnerabilities in firewalls, routers, and switches and use those to gain access to sensitive data and target networks, redirect traffic on a network to a malicious or compromised system masquerading as a trusted system, and to intercept and alter information during transmission. Recommendations:

- Segment networks
- Implement strong firewall rules
- Secure connections across security zones
- Implement intrusion detection
- Implement secure access to network devices

Next Steps

The security of SCADA systems used in critical energy infrastructure installations throughout the United States relies on a cooperative effort between SCADA product vendors and the owners of critical infrastructure assets. These recommendations can be used by SCADA vendors to deliver and support systems that are able to survive attack without compromising critical functionality, by SCADA integrators to configure their systems securely before they are put into production, and by SCADA owners to perform due diligence in procuring, configuring, securing, and protecting these energy delivery control systems.

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 energy sector control systems. This document was made possible by the exceptional technical expertise and sincere dedication of May Chaffin.

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 System
SDL	Security Development Lifecycle
SDLC	Software Development Life Cycle
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
TASE.2	Tele-control 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
XSS	Cross-site Scripting

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 Supervisory Control and Data Acquisition/Energy Management Systems (SCADA/EMS, hereafter referred to as SCADA) 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).

> Throughout the developed world, governments, defense industries, and companies in finance, power, and telecommunications are increasingly targeted by overlapping surges of cyber attacks from criminals and nation-states seeking economic or military advantage.[1]

The DOE-OE NSTB program, established in 2003, supports national laboratory, academia, and industry collaborative research and development activities that strengthen our Nation's energy sector cybersecurity posture, and reduce the risk of cyber-attack against the computers and networks that control energy delivery. A key part of the program is SCADA system vulnerability analysis that identifies and provides mitigation approaches for vulnerabilities that could put these systems at risk of a cyber attack. A cybersecurity 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 actor or is present in conjunction with particular events or circumstances. These weaknesses are not usually a problem unless exploited by a cyber attack.

The NSTB has seen a significant improvement in operating system (OS) and network security since 2003. Some improvement has been observed in reducing host exposure by reducing the number of available ports and services on SCADA hosts and in vulnerability remediation and secure development of new products. However, vulnerabilities caused by less secure coding practices can be found in new and old products alike, and the introduction of Web applications into SCADAs has created more, as well as new, types of vulnerabilities.

This report presents vulnerabilities at a high level to provide awareness of the common SCADA security vulnerability areas without divulging product-specific information. Vulnerabilities that could be used as part of an attack against an SCADA are consolidated into generic common SCADA vulnerabilities. Even though SCADA functions, designs, and configurations vary among vendors, versions, and installations, their vulnerabilities and defensive recommendations are quite similar at a high level. The 10 most significant cybersecurity risks identified during NSTB software and production SCADA assessments. These are listed in Table 1.

Table 1. Ten common vulnerabilities identified in NSTB assessments

Common vulnerability	Reason for concern
Unpatched published vulnerabilities	Most likely attack vector
Web Human-Machine Interface (HMI) vulnerabilities	Supervisory control access
Use of vulnerable remote display protocols	Supervisory control access
Improper access control (authorization)	SCADA functionality access

Common vulnerability	Reason for concern
Improper authentication	SCADA applications access
Buffer overflows in SCADA services	SCADA host access
SCADA data and command message manipulation and injection	Supervisory control access
SQL injection	Data historian access
Use of standard IT protocols with clear-text authentication	SCADA host access
Unprotected transport of application credentials	SCADA credentials gathering

The NSTB SCADA vulnerability assessment process

The NSTB assessment process is highly flexible and may be tailored to the mutual interests of the industry partner and the NSTB program. SCADA 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 SCADA to do harm. From a cybersecurity perspective, this means that they must create an attack path from their attack computer to the SCADA. An attack could potentially start from any point between the Internet and the physical equipment that the SCADA 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 SCADA network. The number of possible paths to the target is the system's exposure. SCADAs 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 SCADA assessments (i.e., onsite assessments) concentrate on the aspects of the SCADA 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 SCADA 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 SCADA 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 SCADA 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 SCADA security assessments identified general security problems, such as the use of insecure 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 insecure 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 SCADA vulnerabilities. SCADA vendors and asset owners can refer to the CWE for additional guidance in identifying, mitigating, and preventing weaknesses that cause vulnerabilities.[9]

The common weaknesses in this report are similar security weaknesses found on two or more unique SCADA 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 SCADA components they were found in.

More information about the NSTB assessment methodology is provided in Appendix A, “NSTB Assessment Methodology.” The typical process includes the following sequence:

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

Responsible sharing of non-attributable information

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.

Characterize risk using the Common Vulnerability Scoring System Version 2

DOE-OE supported and participated in development of the *Roadmap to Secure Control Systems in the Energy Sector*, published in January 2006 and updated in the 2011 *Roadmap to Achieve Energy Delivery System Cybersecurity*. The roadmap development effort was a collaboration of asset owners, vendors, government, national laboratories, and members from various energy sector organizations. One of the four main goals outlined in the roadmap is to "Measure and Assess Security Posture." The goal is that "Companies should thoroughly understand their current security posture to determine system vulnerabilities and the actions required to address them." One of the mid-term milestones (2–5 years) is "Common metrics available for benchmarking security posture." The INL NSTB program previously published common vulnerability reports in 2006 and 2008. Vulnerabilities common to the control systems evaluated by the INL NSTB program were detailed in those reports with mitigation information provided to help utilities and vendors develop and implement cybersecurity strategies and implementations.

In the present report, we introduce a standard metrics-based approach to evaluate the relative risk associated with common SCADA vulnerabilities. We used the Common Vulnerability Scoring System Version 2 (CVSS) and Common Weakness Enumeration methodologies to apply standard terminology and metrics-based scores to common SCADA vulnerabilities.

While it is not possible to quantify an absolute measurement of risk, it can be useful to apply a formal, structured process that characterizes relative risk associated with a particular vulnerability taking into account the environment within which that vulnerability exists. The CVSS metrics evaluate vulnerabilities so that higher risk vulnerabilities can be mitigated first, using mitigations that reduce CVSS scores, and the risk these scores reflect, the most.

In appendix C, we have scored each of the ten common vulnerabilities identified by SANS, The Top Cyber Security Risks[1], according to the CVSS process. We used input values to the CVSS metrics that we found to be the most common values observed in NSTB assessments. This creates a generic score for each of the ten most common vulnerabilities.

The generic CVSS scores found in Appendix C do not describe the risk to a particular SCADA configuration because each system installation is unique and the same vulnerability in different environments poses different risks. Technical mitigations, operating procedures and cybersecurity policies in place in different operational environments will result in different, in some cases very different, CVSS scores.

This report recommends mitigations that reduce risk of cyber-attack and will consequently lower CVSS scores in SCADA installations where these mitigations are implemented. These tools allow a qualitative analysis of the vulnerabilities found in control systems by INL researchers and are applied in this report to help vendors and asset owners evaluate their relative risk profile associated with vulnerabilities that apply to their systems and reference additional security information on reducing this risk.

SCADA system security priorities and obstacles are different from those of common IT computers and networks. SCADA systems cannot be evaluated using common IT security assessment techniques or protected using common IT security solutions. The main goal of SCADA cybersecurity is to prevent unauthorized manipulation of the physical 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 SCADA system. There is no single solution that can completely secure the SCADA system from cyber attack, so this defense-in-depth approach is needed.

SCADA system cybersecurity measures protect system data and functionality from unauthorized access, use, disclosure, disruption, modification, or destruction by preventing actions that violate security goals for data confidentiality, integrity, and availability of cyber assets. In this respect, SCADA systems and common IT computers and networks share cybersecurity objectives,

- 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).

However, SCADA systems prioritize cybersecurity objectives differently depending on the physical system under control and the functionality provided by the individual SCADA component.

This version of this document differs from a released draft version in that it clearly highlights actions that can be performed by SCADA vendors and owners to reduce the risk to the most significant weakness areas identified through NSTB assessments. The goal of NSTB assessments is to identify the most significant security vulnerabilities in SCADA products and production installations and provide guidance in remediating these risks. NSTB assessment reports provide specific and detailed recommendations to the SCADA vendor or owner for addressing the security vulnerabilities found in their systems. This report addresses common vulnerabilities and security weaknesses that are routinely identified during NSTB SCADA security assessments. The goal of this document is to share this information with the rest of the industry so that all energy stakeholders can use this information to identify and mitigate vulnerabilities in their systems.

In the next section of this document, section two, we describe how CVSS metrics evaluate risk associated with common SCADA vulnerabilities and guide risk reduction efforts. Next, section three discusses several ways to categorize vulnerabilities found in NSTB assessments, and presents the relative frequency of vulnerabilities using each of these categorizations. Section four presents NSTB detailed vulnerability assessment findings and recommended mitigations with references to further related security information. Finally, section five summarizes recommendations for SCADA developers and administrators to help mitigate the risk posed by common vulnerabilities found in NSTB SCADA security assessments.

2. PRIORITIZE COMMON SCADA VULNERABILITIES

The most significant vulnerabilities identified in SCADA systems are those that allow unauthorized control of the physical system. Compromise of the SCADA system's availability and ability to function correctly may also have significant consequences to the control systems and physical infrastructure it manages. Security efforts are made most effective by addressing the high consequence and high threat vector vulnerabilities first.

This report uses The Common Vulnerability Scoring System, version 2 (CVSS) vulnerability scoring methodology prioritizes common SCADA system vulnerabilities. The goal of CVSS metrics is to provide "an open framework for communicating the characteristics and impacts of IT vulnerabilities."[2] This standardized scoring system accounts for the consequence of compromise of a vulnerable SCADA component(s), and also the level of risk due to available exploit techniques and mitigations. CVSS metrics provide a standard way to prioritize vulnerabilities according to the relative risk they pose to the SCADA system.

Section 2.1 describes CVSS and its application to SCADA. The following section 2.2 shows how CVSS scores appropriately tailored to the unique configuration of a particular SCADA system are used to evaluate and mitigate vulnerabilities to reduce risk.

2.1 SCADA and the Common Vulnerability Scoring System

> 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 Tables 2, 3, and 4 present the CVSS Base, Temporal and Environmental-metrics respectively. Base-metrics describe an inherent characteristic of the vulnerability and are not unique to SCADA systems. Security vulnerabilities in individual software programs, network protocols, or hardware have intrinsic characteristics described by the Base-metrics.

These same vulnerabilities also have characteristics that may change with time and depend on the surrounding environment. These characteristics include varying potential impacts and ease of exploitability that depend on the type of system the vulnerability resides on and the protective measures that have been applied to that system. CVSS attempts to capture these important factors through Temporal and Environmental metrics and to provide a standard method for individual system owners to assess their unique risk to a particular vulnerability.

For instance, CVSS Base-metrics used to evaluate the potential consequences of a compromise are Confidentiality Impact, Integrity Impact, and Availability Impact. These three Base-metrics characterize the direct effect that the vulnerability, if exploited, will have on an IT asset and are defined respectively as the degree of loss of confidentiality, integrity, and availability.

However, the influence of a particular environment changes the risk represented by these Base-metrics. To describe impact for a particular environment, the Base-metrics are weighted and the result is represented by the Security Requirement Environmental-metrics. Likewise, Collateral Damage Potential is the Environmental-metric that includes the potential physical and monetary impacts in the risk equation.

The Base-metric scores that describe exploitability evaluate the probability of compromise by assessing the exposure, or difficulty in accessing the vulnerability. These are Access Vector, Access Complexity, and Authentication. Temporal-metrics further refine this probability estimation based on the current offensive and defensive tools and techniques available as characterized by the Exploitability Temporal-metric. Finally, the Target Distribution Environmental-metric customizes the probability of exploitation on an individual SCADA system. As discussed further in section 4.2 on the mitigation of known vulnerabilities,

> Initially, real-world exploitation may only be theoretical. Publication of proof of concept code, functional exploit code, or sufficient technical details necessary to exploit the vulnerability may follow. Furthermore, the exploit code available may progress from a proof-of-concept demonstration to exploit code that is successful in exploiting the vulnerability consistently. In severe cases, it may be delivered as the payload of a network-based worm or virus.[2]

Table 2 summarizes the CVSS Base-metrics. Secure software development and system administration practices can reduce the number of vulnerabilities on a SCADA system, and also reduce the severity assessed by the Base-metrics. Vendors can affect Base-metrics by following the "principle of least privilege" when designing and implementing SCADA products. SCADA computer and network administrators may do the same on their installed systems, but can be limited by vendor support restrictions and/or the potential risk to system availability.

Table 2. CVSS Base-metrics.

Base Metric	Assessment Criteria	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
Integrity Impact	Impact to integrity of a successfully exploited vulnerability	Increased integrity 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 CVSS Temporal-metrics. Temporal-metrics, like Base-metrics, are directly tied to the vulnerability, but unlike Base-metrics change over time as exploit techniques, as well as mitigation techniques, are developed and automated. SCADA owners can affect Temporal-metric scores by using available mitigations that may include patches, updates, and temporary fixes from the software vendor, as well as unofficial patches or workarounds developed by users. SCADA owners can apply, and in some cases create, workaround techniques to mitigate vulnerabilities on their systems that do not have a patch available or for which the official patch cannot be applied. A security obstacle for SCADA owners is the unavailability of patches for vulnerabilities in their SCADA software, or the inability to apply patches to third-party software. Temporal metrics provide scores for workaround mitigations to vulnerabilities that do not have patches available. The effectiveness of the available work-around mitigations is used to adjust the Temporal score of the remaining vulnerabilities.

For instance, the Remediation Level Temporal-metric may not be valid for an installed system if the available patch or temporary fix cannot be applied without compromising SCADA functionality. If patch testing results indicate that a patch cannot be applied or the old version cannot be replaced with a secure version, then altering the SCADA product to accommodate the patch or otherwise mitigate the problem may become a top priority for the SCADA vendor (depending on the risk to which this vulnerability exposes the system). The scoring system provides the selection - “Unavailable: There is either no solution available or it is impossible to apply”[2] – to allow for this situation.

Table 3. CVSS Temporal-metrics.

Temporal Metric	Assessment Criteria	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

Table 4 summarizes the CVSS Environmental-metric group that captures the characteristics of a vulnerability associated with a specific environment. The same vulnerability in different environments poses different risks to an organization and its stakeholders. Environmental-metrics capture the risk that a vulnerability poses to the unique SCADA installation. These metrics reflect the criticality of the affected component in its environment. As an example, denial of service (DoS) vulnerabilities in SCADA components that require higher availability in a particular environment will receive higher criticality scores. Security Requirements Environmental-metrics enable SCADA owners to customize the CVSS score depending on the importance of the affected component to their own organization, measured in terms of confidentiality, integrity, and availability.

SCADA vendors and owners can use these metrics to separate high-consequence vulnerabilities from those that are less significant in their SCADA environment, or to change their environments to reduce the potential impact that could result were these vulnerabilities to be exploited.

Table 4. CVSS Environmental-metrics.

Environmental Metric	Assessment Criteria	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

The CVSS metrics were applied "generically" to evaluate the most significant vulnerabilities identified during NSTB assessments. These "generic" scores were derived from the most representative, prevalent, and persistent values, and are presented in Appendix C, "Top 10 Most Critical SCADA Vulnerabilities."[2] It is important to note that actual CVSS scores are unique to individual environments and must be customized to the particular vendor software or installed system. In this way, CVSS metrics can be used by SCADA vendors and owners to help prioritize security risks for mitigation in their unique, individual environment.

SCADA vendors and owners can use these scores as a starting point for vulnerability identification and evaluation on their own systems. They can conduct internal and third-party assessments of their systems to identify vulnerabilities then use CVSS metrics to:

1. Evaluate risk
2. Prioritize remediation and mitigation efforts
3. Evaluate risk reduction due to remediation and mitigation efforts.

The following section explores how SCADA vendors and owners can evaluate and reduce risk in the context of CVSS metrics tailored to their unique, individual environment. Additional information on CVSS criticality scoring is in Appendix B, "Vulnerability Scoring," and at the CVSS Web site.[2] Owners and developers can review generic CVSS scores for ten of the most common and critical SCADA security vulnerabilities in Appendix C, and consider the mitigations discussed there as options for lowering risk.

2.2 SCADA cybersecurity risk reduction

The ultimate goal of security activities is to reduce risk. Business risk is a function of threat, impact/consequence, and vulnerability[3], that is, the consequence that would result if a threat successfully exploited a vulnerability, and the probability that this could take place. It can be thought of as "the relative impact that an exploited vulnerability would have to a user's environment."[2] Understanding exposure to attack, attacker awareness of the vulnerability, and exploitation knowledge helps to assess the probability of a successful attack. To reduce risk, reduce the probability that a threat can exploit a vulnerability, and reduce the consequences that would follow if this did occur.

CVSS provides a qualitative analysis tool for vulnerability analysts, vendors and control system users to characterize their cybersecurity risk profile. Tables 5 and 6 list examples of how SCADA vendors and owners, respectively, can reduce the CVSS scores for their products.

Table 5. Actions for SCADA vendors to reduce vulnerability scores

CVSS metric	Action that can reduce CVSS score	Section of this report where this is discussed
Base	Secure development practices Apply the concept of least privileges to compartmentalize and limit the privileges of all users and services/applications Test software for weaknesses vulnerable to common exploit techniques (and remediate)	Section 4.1.1 Design and implement secure code Section C-3 Supervisory Control Access: Use of vulnerable remote display Protocols Section 4.1.1.3 Design and implement secure code
Temporal: Remediation Level	Rapid patch testing for third-party patches Provide publicly-accessible contacts for reporting security issues within products and services (*e.g.*, a "/security" URL directly and prominently accessible from the vendor main Web page with contact information and disclosure policies) Rapid patch development Support security products and techniques Use and support strong authentication and encryption mechanisms Provide detailed documentation of necessary communications between SCADA components to enable development of specific firewall and IDS rules.	Section 4.2.2 Implement effective patch management Section 4.2.2 Implement effective patch management 4.3.2.4 SCADA data and command message communication protocols 4.1.2.2 Minimize ports and services

Table 6. Actions for SCADA owners to reduce vulnerability scores

CVSS Metric	Action that can reduce CVSS score	Section of this report where this is discussed
Temporal: Remediation Level	Establish security function to monitor user groups and vendor announcements for release of security and vulnerability notifications Rapid patch testing and deployment Identify and apply work-around mitigations for vulnerabilities that cannot be patched Restrict SCADA user privileges to only those required Use and support strong authentication and encryption mechanisms Protect critical functions Create and deploy specific firewall and IDS rules Closely monitor critical functions as well as security and operational logs for signs of abnormal/exceptional activity.	Section 4.2.2 Implement effective patch management Section 4.2.2 Implement effective patch management (unpatched vulnerabilities) Section 5.2 Secure SCADA Installation and Maintenance for the Owners/Operators

3. CATEGORIZE COMMON SCADA VULNERABILITIES

Common SCADA vulnerabilities can be categorized in different ways to support different analytic objectives. The first step in the NSTB vulnerability assessment process is to identify the Assessment Targets (AT) and involves a collaborative effort between vendor representatives and INL researchers. ATs are points of entry, processes, protocols, and/or pieces of equipment that if compromised could result in a severe impact on the control system. A typical vulnerability assessment project on a SCADA control system includes approximately 900-1000 hours of cyber researcher effort to evaluate the selected ATs. The team selects ATs reflecting what is considered "low hanging fruit" or more vulnerable targets.

The scope of vulnerabilities identified in this report is limited to the selected ATs, and to the control systems selected for NSTB assessments. It is important to note that the relative frequencies presented here do not necessarily reflect those that would be observed if all ATs, on all control systems operating in the "real world" were represented.

SCADA systems are comprised of components including process equipment, process control hardware, network equipment, and computers. SCADA component products include software, hardware, firmware, and network equipment built to support supervisory control and data acquisition.

SCADA component function provides operational functionality of the system. Physical equipment can be monitored and controlled through connections to basic control devices. These devices are computer hardware running a minimalistic operating system referred to as firmware, created by a SCADA vendor.

Supervisory control software is installed on traditional computer hardware and operating systems. The system that software runs on is referred to as its host. Supervisory control software typically comprises multiple applications and can be installed on multiple computer hosts. The supervisory control host computers, control hardware, and equipment under control are all connected using networking equipment, which may include common IT computer network protocols and devices, as well as computer network protocols and devices created specifically for SCADA systems.

SCADA systems can be categorized into operational zones such as shown in Figure 5, ISA SCADA architecture functional-level reference model. This model separates the overall system architecture into levels starting at the monitored and controlled physical device to the I/O (input/output) Network (level 0) through several zones to the Corporate Network (level 4). The main focus of the NSTB vulnerability assessment project has been in the level 2 Supervisory Control LAN with some work in the Level 1 and 3 zones.

This section presents the relative frequency of SCADA vulnerabilities observed in NSTB assessments, categorized by:

- Section 3.1: Vulnerability type, with a discussion of associated attack paths
- Section 3.2: SCADA component
- Section 3.3: SCADA component function
- Section 3.4: SCADA architecture functional-level

3.1 Relative frequency of NSTB observed vulnerabilities by type

Figure 1 shows the relative frequency of vulnerabilities found in NSTB assessments, categorized by vulnerability type.

An understanding of the way in which we counted vulnerabilities is needed for meaningful interpretation of the relative frequency distributions shown here. For instance, there are many more buffer overflow (and similar) type vulnerabilities than the other kinds of vulnerabilities shown. Consider that fifty buffer overflows found in an application represent fifty attack vectors, and are counted as one vulnerability. A single way to bypass authentication is also counted as one vulnerability. Improper authentication or no authentication found for an application is counted as one vulnerability (maybe more if there are multiple ways to authenticate or steps taken in the authentication process); however, there can be many buffer overflow or other input validation vulnerabilities in that application.

In the SCADA environment a single vulnerability can be much more critical or far-reaching than others. For instance, a communication channel vulnerability in a SCADA protocol affects the whole system, but is counted as a single vulnerability. CVSS environmental scoring takes this into consideration.

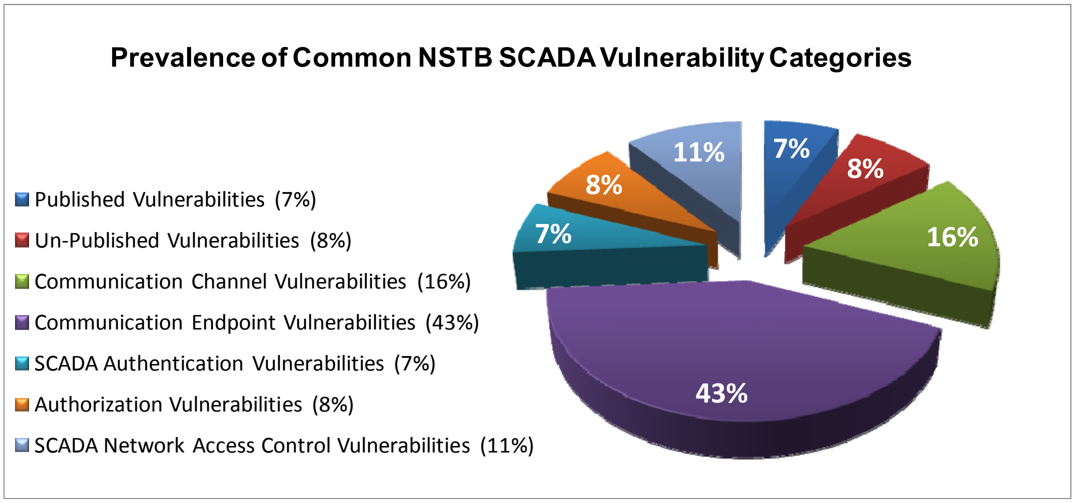


Figure 1. Percentage of NSTB assessment findings by vulnerability type

3.1.1 Assessment targets associated with each vulnerability type

NSTB assessments prioritize assessment targets based on the likelihood and impact of compromise as determined through the combined experience of the assessment team and the SCADA vendor and/or owner. Table 7 shows SCADA vulnerability types and associated assessment targets that could allow access to core SCADA functionality.

Table 7. Types of vulnerabilities, and associated assessment targets that could allow access to core SCADA functionality

Vulnerability Type	Assessment Target Category	Source of Vulnerability
Known Vulnerabilities	Most Likely Attack Paths	Unpatched Or Old Versions Of Third-Party Applications Incorporated Into SCADA Products
		Unpatched OS on SCADA Hosts
Un-Published Vulnerabilities	Potential 0-day and Unpatched Vulnerabilities	Excessive SCADA Host Exposure Through Unnecessary Services
		Improper SCADA Code Quality
Communication Channel Vulnerabilities	Unauthorized Access to SCADA Functionality through Vulnerable Communication Channels	Remote Access Protocols Vulnerable to Spoofing and MitM Attacks
		SCADA Protocols Vulnerable to Spoofing and MitM Attacks
Communication Endpoint Vulnerabilities	Unauthorized Access to or DoS of SCADA Hosts and Applications	Vulnerable Server Applications for SCADA Communication and Data Transfer Protocols
		Database Vulnerabilities
		Web Vulnerabilities
SCADA Application Authentication Vulnerabilities	Access to SCADA Applications by Exploiting Authentication Mechanisms	Authentication Bypass Issues
		Credentials Management
SCADA Host Authorization Vulnerabilities	Ability to Cause Harm from an SCADA Account	Failure to Secure Host Environment
SCADA Network Vulnerabilities	Access to SCADA Hosts and Functionality through Available Network Paths	Improper Network Design
		Weak Firewall Rules
		Failure to Secure Network Devices
		Improper Network Monitoring

3.2 Relative frequency of NSTB observed vulnerabilities by SCADA component

Figure 2 shows the relative frequency of vulnerabilities found in NSTB assessments, categorized by SCADA component. Vulnerabilities in SCADA component products could allow an attacker to gather information about, disrupt, or manipulate SCADA operations. Figure 2 shows that NSTB assessments have focused on SCADA component products to characterize the vulnerabilities they are most affected by, and evaluate ways in which their design and operational requirements affect host and network security.

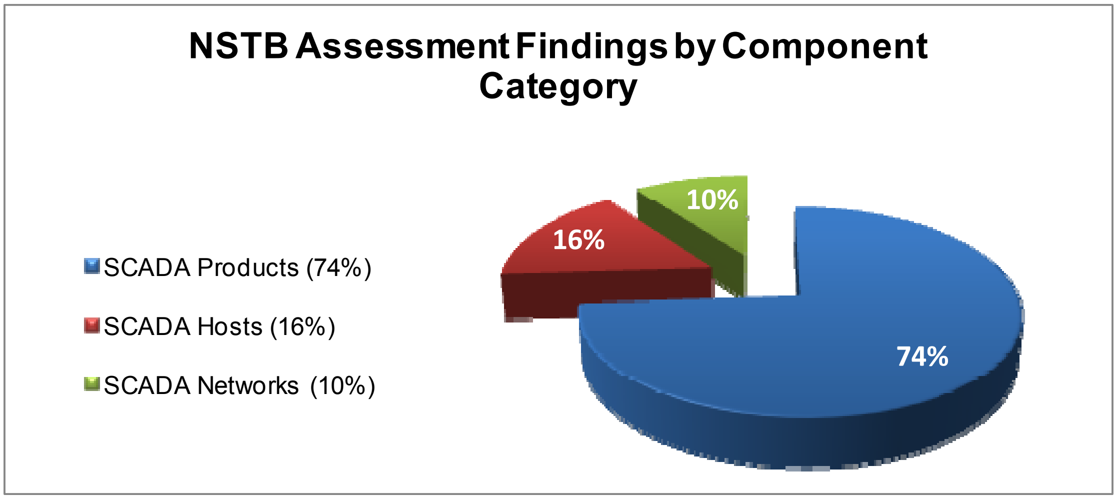


Figure 2. Percentage of NSTB assessment findings by SCADA component category

3.3 Relative frequency of observed vulnerabilities by SCADA component function

Figure 3 shows the relative frequency of vulnerabilities found in NSTB assessments, categorized by SCADA component function. This provides insight into the system consequence were the function to be compromised, and illustrates the high percentage of vulnerabilities that NSTB assessments found in SCADA server applications (services).

The distribution is skewed by the SCADA products that were selected for evaluation. Supervisory control protocols were available for assessment on almost all NSTB assessments. In this report the representation of SCADA protocols that are used for external communications is skewed based on their availability for assessment. The Inter-Control Center Communications Protocol (ICCP) was selected for an in-depth assessment[5] while "basic" or "local" control protocols like Distributed Network Protocol Version 3 (DNP3) were not configured on every assessed system.

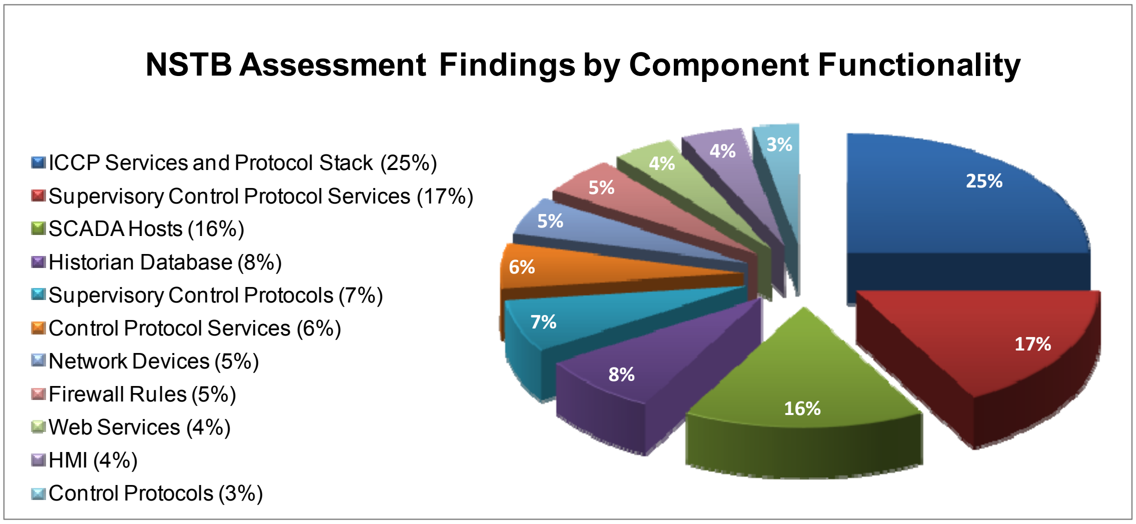


Figure 3. Percentage of NSTB assessment findings by component function

3.4 Relative frequency of observed vulnerabilities by SCADA architecture functional-level

Figure 4 shows the relative frequency of vulnerabilities found in NSTB assessments, categorized by SCADA architecture functional-level using the ISA99 reference model, which is shown in Figure 5.[4] This model describes a SCADA system as a series of logical levels based on functionality. It is useful to categorize vulnerabilities by this established frame of reference.

The NSTB focus on core SCADA functionality is evident in Figure 4. Individual SCADA components can be evaluated by the risk they contribute to the overall security of the system based on their SCADA functionality. The largest portion of products and functionalities that NSTB tested belong in the supervisory control and operations management categories.

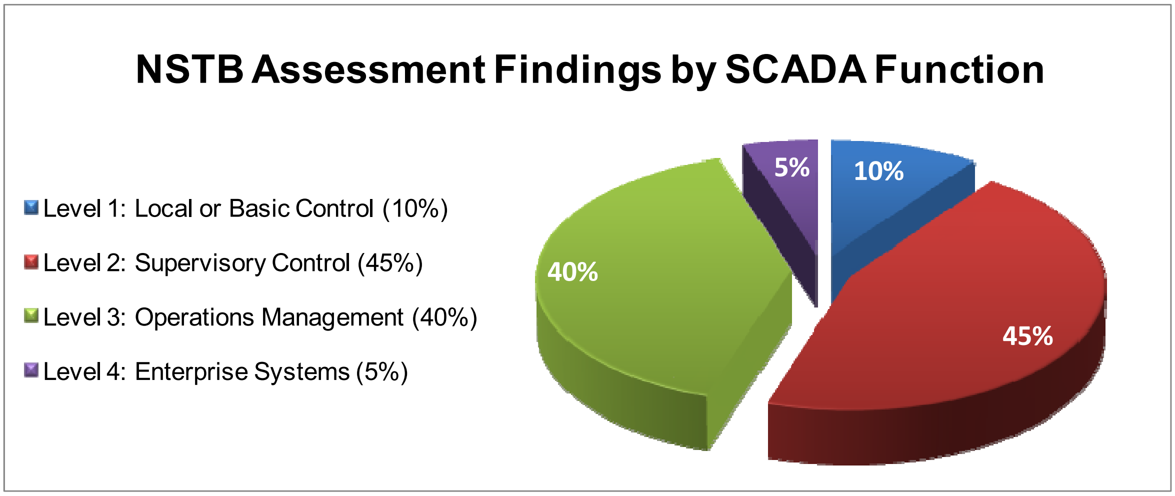


Figure 4. Percentage of NSTB assessment findings by SCADA architecture functional-level

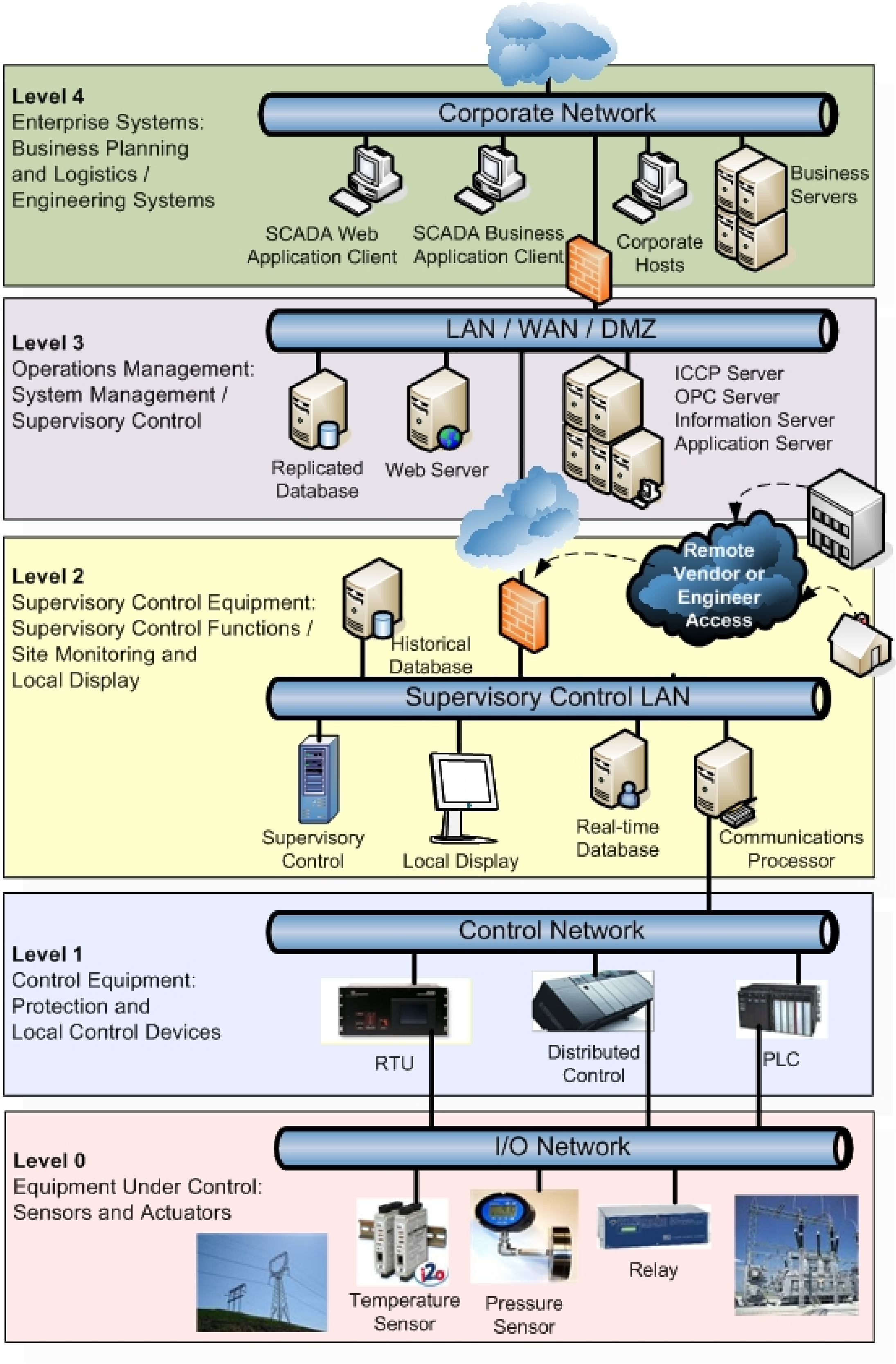


Figure 5. ISA SCADA architecture by functional-level reference model

4. NSTB ASSESSMENT RESULTS

NSTB assessments identify and analyze SCADA system vulnerabilities that could allow unauthorized access to SCADA hosts, applications, and data, or unauthorized manipulations that affect operations, that spoof or manipulate SCADA data and commands or that impose a DoS that could impede communications and jeopardize SCADA functionality. This section presents common vulnerabilities found in NSTB assessments, and recommends mitigations. First, it discusses ways to secure the SCADA cyber-attack surface through secure code and removal of unneeded ports and services. Then it describes:

- Known vulnerabilities mitigated through removal of all unneeded applications and services and effective patch management.
- Communication channel vulnerabilities mitigated through protected transmission of authentication credentials, secure control of local and remote access and SCADA data integrity checks.
- Communication endpoint vulnerabilities mitigated through secure coding practices that enforce rigorous input data validation in SCADA and ICCP services, database applications and web services.
- Authentication vulnerabilities mitigated through authentication at both the server and the client, and effective management of authentication credentials.
- Authorization vulnerabilities mitigated through least-privileges access control, removal of unneeded functionality and secure configuration of SCADA components.
- Network access vulnerabilities mitigated through network segmentation, strong firewall rules, secure connections across security zones and intrusion detection.

4.1 Secure the cyber-attack surface

The attack surface comprises all possible avenues of attacking a system. "A system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage. Hence the smaller the attack surface, the more secure the system."[8] All open ports, installed services and applications that can potentially be exploited create the attack surface. This section first discusses the design and implementation of secure code to decrease the number of vulnerabilities that could be exploited. Then it addresses the need to minimize the number of open ports, installed services and applications. This, in turn, will minimize the number of both known and existing but unknown vulnerabilities and further decrease the attack surface.

4.1.1 Design and implement secure code

Secure coding practices minimize vulnerabilities in SCADA applications and services. Insecure coding practices can introduce bugs that could be exploited for malicious purposes and could also make SCADA systems fragile, so that administrators may hesitate to implement changes after the initial configuration.

Many SCADA protocols and core applications were built before the rise of the Internet and before the concepts of secure design and coding practices became well established. These legacy SCADA systems were developed as stand-alone systems with reliability and efficiency the primary design requirements. Many SCADA protocols and core applications still in use today were developed before Internet connection became common practice. Although security is an added requirement for many new SCADA applications, NSTB assessments have found that today's systems share many of the same vulnerabilities in the design and coding principles as legacy SCADA software.

SCADA code review and reverse engineering exercises suggest that SCADA software may not always be designed or implemented using secure software development concepts. SCADA software vulnerabilities observed in NSTB assessments often result from insecure coding practices and inadequate testing. NSTB assessments revealed a common need to increase secure coding practices. The three most common observations were the need to strengthen input validation, authentication, and access controls. Most of the SCADA vulnerabilities identified during NSTB assessments that allow remote code execution result from the use of potentially dangerous functions. Adherence to secure programming standards and guidelines can strengthen code against these vulnerabilities during software development and automated source code analysis tools can identify existing vulnerabilities for remediation.

The NSTB code audits and fuzz testing revealed numerous potential vulnerabilities. It was infeasible to investigate every unsafe function call and the cause of every crash to determine which among these could be exploited for malicious remote code execution. The NSTB assessment demonstrates the existence and impacts of at least one instance of unsafe coding then recommends that unsafe function calls be remediated. As a result, these vulnerabilities may be under-represented in this report.

SCADA software vendors can 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 SCADA product can greatly enhance quality and security, or confirm the security of the product. In this way, SCADA vendors can thoroughly test all SCADA features to validate SCADA stability and security levels before release, and SCADA customers can require products to be tested by a third party and vulnerabilities remediated before acceptance of a SCADA product.

SCADA software can have large, complex, and legacy codebases, SCADA 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 SCADA customer. With the new focus and requirements for SCADA security, including SCADA product vulnerabilities starting to be publicly announced, vendors may find code audits and associated code changes to be very cost effective in comparison to fixing individual vulnerabilities as they are publicly announced.

Secure coding resources are available for all application types and languages. The CWE list[9] provides information about all types of software weaknesses including the most common SCADA programming errors listed in Table 8.

Table 8. Common vulnerabilities associated with insecure SCADA code design and implementation.

Weakness Classification	**Common Vulnerability**
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

4.1.1.1 Replace potentially dangerous functions with safe counterparts

Well-known unsafe C/C++ functions - in particular unsafe string and memory functions - can be replaced by safe counterparts and secured by proper input validation. For example, the strcpy function in C is a potentially dangerous function, and has a safe counterpart strncpy. Maliciously malformed input to strcpy could create a buffer overflow and allow an attacker to execute remote code.

The SCADA software developer is responsible for writing secure code that properly validates function input, and for replacing dangerous functions with their safe counterparts wherever possible. However, the prevalence of publicly announced buffer overflow vulnerabilities and other vulnerabilities exposed through malformed input, suggest that reliance on the software developer to secure potentially dangerous functions can introduce high risk.

4.1.1.2 Validate input data

Input data 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. All of the weaknesses in the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[10] can be associated with improper input validation. Like other software products, the most significant security weakness in SCADA is inadequate input validation. Many different kinds of input validation errors were identified in NSTB assessments. Only the most significant ones are specifically addressed here in relation to the SCADA system components most affected by them.

NSTB assessments found that SCADA applications frequently suffer from coding practices that could allow an adversary to use unexpected input data to modify program execution. A function that does not check the validity of input data can cause the application to crash or execute malicious commands provided as input by an adversary. Software that does not properly check the size of user input, does not 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 XSS code to gain control over vulnerable machines. An attacker may be able to impose a DoS, bypass authentication, access unintended functionality, execute remote code, steal data and escalate privileges. While some input validation vulnerabilities may not allow exploitation for remote access, they might still be exploited to cause a crash or a DoS attack.

Input validation vulnerabilities were found in server applications written to process SCADA protocol traffic. Most result in unauthorized access to the host on which the server was running. Sanity checks of incoming messages can ensure that the lengths and counts seem reasonable, that the data in the message is valid, and that the message is valid given the state of the connection. 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 it is recommended that least-privileges be implemented so that service privileges are minimized as much as possible to reduce risk.

Weak or missing security features in SCADA software leave the system components vulnerable to manipulation by any threats to which they are exposed. Giving each component of the SCADA its own protection mechanisms can be a good defensive measure. As another layer of defense, compiler protection options can be used when compiling C/C++ code to increase the difficulty for an attacker to execute exploit code. This decreases the impact of 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 SCADA to a DoS-type attack.

Protect against buffer overflow

Secure coding practices protect SCADA system software against buffer overflow vulnerabilities, which are the most common type of input validation weaknesses reported on NSTB assessments. Buffer overflows can be caused by 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. Exploit code can use the buffer overflow to create an interactive session and send malicious commands with the privileges of the exploited program.

For instance, secure coding practices advise against using the input value to determine the buffer size. Even if values are never input directly by a user, sometimes data are not 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. Consequently, network protocol implementations that do not validate input values can be vulnerable to buffer overflow attacks, so it is best to implement network data bounds and integrity checking as well.

An application that calls a potentially dangerous function must properly validate input to that function or risk exposure of a buffer overflow vulnerability through an unsafe function call. For example, the developer might assume that no one would ever create a username longer than 1,024 characters. If he then reserves a 1024-byte memory buffer for the username and does not validate input, an attacker could try a few usernames to discover that more than 1,024 characters creates a buffer overflow and provides an avenue for attack. The developer can mitigate this vulnerability by validating that the size of the input does not exceed the size allocated for the memory buffer that stores the input. Safe functions, such as strncpy require that the memory buffer size be specified and eliminate this risk. As another example, the developer might assume that input such as array index values will always be bounded within an expected range. However, if negative or exceedingly large numbers can be input for array index values this could result in unexpected behavior such as causing essential services to crash.

Five of the *2010 SANS/CWE Top 25 Most Dangerous Programming Errors*[10] are types of buffer overflows. Table 9 lists CWE entries related to buffer overflows.[9]

Table 9. Five of the 2010 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

Protect against Structured Query Language (SQL) injection

Secure coding practices protect the SCADA historian database against SQL injection vulnerabilities, which result from incorrect or inadequate filtering of user input that does not assure the integrity of special elements used in an SQL query statement. For instance, if an attacker inserts literal escape characters into a database query, the attacker may gain arbitrary read or write access to the database. The SQL statements used to communicate with the database could be modified to steal, corrupt, or otherwise change data in the database. Also, if SQL queries are used for security controls such as authentication, attackers could alter the logic of those queries to bypass security.

SQL injection vulnerabilities are accessible through client (often Web) applications. SQL injection exploits the database by forwarding SQL commands to the database, where they are executed. If database-backed applications get and receive data from a server on a secure network, it is a target for SQL injection. For example, if a client application connects to a database on a secure network through a Web server that is isolated in a DMZ (as shown in Figure 6), SQL injection attacks the SQL server within the secure network. A successful compromise could give the attacker control of the SQL server within the secure network, even if the firewall blocks all other connections to that host.

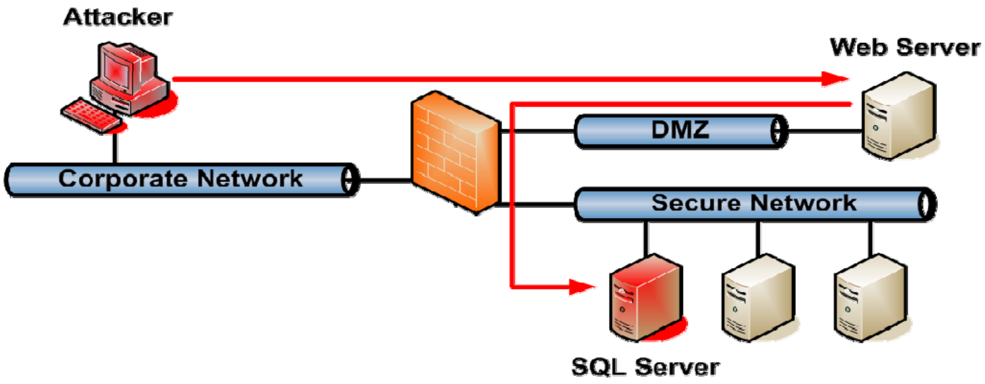


Figure 6. Example of an SQL injection attack via Web applications

Protect against cross-site scripting

Secure coding practices protect SCADA system web applications and services against cross-site scripting vulnerabilities (XSS). The root cause of a XSS vulnerability is the same as that of an SQL injection, lack of input data validation. However, a XSS attack is unique in the sense that the Web application itself unwittingly sends the malicious code to the user. According to the 2010 *CWE/SANS Top 25 Most Dangerous Programming Errors*[10] report, XSS is the most widespread and critical programming error.[5] 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.

Cross-site scripting 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 that is browsing the site within the context of a client-server session.

Many XSS attacks rely on user interaction and typically take the form of a malicious link sent by the attacker. Users may be deceived into clicking on a link that appears to be associated with a known, respected and trusted entity. In many cases, the attack can be launched without the victim even being aware of it. 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 even to careful users.

It is possible for an attacker to inject malicious script into a link and have a Web site return the malicious script to the victim as though it was legitimate. The victim's Web browser will then run the malicious script, trusting it because it came from the server and potentially compromise the victim's computer by using one of many browser exploits. There are many such scenarios that allow for this behavior, all of which are caused by a lack of input data validation.

Once the malicious script is injected, the attacker can perform a variety of malicious activities. The most common attack performed with XSS involves the disclosure of information stored in user cookies that may include session information. In this way, the attacker could transfer private information from the victim's machine to the attacker. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. XSS vulnerabilities can be exploited to: manipulate or steal cookies; create requests that can be mistaken for those of a valid user; compromise confidential information; execute malicious code on the end user systems; redirect the Web page to a malicious location; hijack the client-server session; exploit a vulnerability in the Web browser itself, possibly taking over the authorized SCADA Web client host; engage in network reconnaissance; install Trojan horse programs; plant backdoor programs; run "Active X" controls (under Microsoft Internet Explorer) from sites the user perceives as trustworthy; and modify content viewed by the user.

Protect against directory traversal

Secure coding practices protect SCADA system applications and services against directory traversal vulnerabilities, which occur when file paths are not validated. A directory traversal vulnerability can occur when the software uses external input to construct a pathname that is intended to identify a file or directory located underneath a restricted parent directory but the software does not properly neutralize special elements within the pathname so that the pathname resolves to a location outside of the restricted directory.[33.] Directory traversals are commonly associated with Web applications, but all types of applications can have this class of vulnerability.

The attacker may be able to read, overwrite or create critical files such as programs, libraries, or important data by exploiting a directory traversal vulnerability. This may allow an attacker to: execute unauthorized code or commands; read or modify files or directories; and crash, exit, or restart critical files or programs, potentially causing a DoS.

The damage that a directory traversal vulnerability can cause is related to the permission of the vulnerable application. If the vulnerable application has limited read/write permissions, the attacker may not be able to do anything of importance. However, if the application has system or root privileges then the damages can be extensive.

4.1.1.3 Recommendations and resources

To find and remediate cyber-attack surface vulnerabilities, such as insecure code, input validation, buffer overflow, SQL injection, XSS, and directory traversal, it is recommended that:

- All stakeholders follow secure development standards when developing new products
- All stakeholders redesign or patch current products to remediate vulnerabilities

- All stakeholders redesign SCADA-specific protocols to include strong authentication and integrity checks
- All stakeholders who write custom applications train developers in secure coding practices
- Vendors follow secure coding best practices, such as
  - Validate input and output data
  - Use safe string and buffer functions
  - Use robust integer operations and data types for memory operations
- Vendors carefully test and evaluate internally developed and third-party application software:
  - In all of their SCADA components that handle data from other components, starting with services that are exposed to less-trusted networks and working inward
  - During SCADA software development, including thorough code reviews via manual and automated processes
  - Using automated static analysis tools
  - Using dynamic tools and techniques such as fuzz testing, robustness testing, and fault injection
  - Using a qualified third part that performs additional security testing
- Owners work with vendors to explicitly address the security of SCADA products during the procurement and acceptance processes
  - Conduct a security audit of a SCADA product
  - Determine appropriate mitigations to meet specified security levels
  - Require and validate that products are delivered with secure configurations
- Owners verify that third-party application software vendors have conducted detailed security testing of their products
- Owners verify that IT products deployed on the SCADA network pass a security review

To help prevent exploitation or limit damage from cyber-attack surface vulnerabilities, it is recommended that additional precautions be implemented, such as:

- Vendors, and possibly owners, thoroughly assess, secure and test SCADA products, starting with the most exposed and powerful functions
- Vendors use compiler options to detect some types of buffer overflows; however, an attack could still cause a DoS, since the typical mitigation is to exit the application
- Vendors and owners if possible, can use as part of a defense in depth solutions, a CPU and OS that offers protection against buffer overflow attacks
- Owners identify critical components and develop corresponding risk analysis and mitigation strategies for both operations and security

SCADA vendors and owners can use this report for high-level information on common types of software vulnerabilities identified in SCADA systems. More detailed information is needed to implement the software security recommendations. SCADA developers can reference the CWE site for additional secure development information and references on security weaknesses that lead to vulnerabilities.[9] SCADA administrators can use this information to better understand

SCADA software vulnerabilities. They can then work with their vendors to mitigate the associated risks as much as possible in existing systems, and create procurement requirements that enforce security standards.

All weaknesses listed on the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors*[10] have been found during SCADA assessments. SCADA developers can refer to this list of more detailed weaknesses and the associated *2010 CWE/SANS Top 25: Monster Mitigations*[11] 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*[12]
- SAFECode Secure Coding Standards[13,14]
- *20 Critical Security Controls: Critical Control 7: Application Software Security*[15]

4.1.2 Minimize ports and services

Services or applications running on a system may open network ports to communicate with the outside world. An attacker can only gain access to and receive information from the SCADA through an open port. Each open port provides a possible access path for an attacker that may be used to send exploits and receive data. If an attacker can remotely connect to services listening on accessible network ports, she then has a foothold onto the protected network, and can target all services listening on the local network hosts because she has breached the perimeter controls. A vulnerable network application may be exploited by an attacker and used to send malicious code as well as receive unauthorized data.

The more services listening at open ports on the SCADA hosts, the more exposed the SCADA system is to attack. It is consequently of the utmost importance that only required services and applications run on SCADA systems so that no ports are opened unnecessarily. However, SCADA vendors may not always provide enough documentation of required component communications.

4.1.2.1 Identify necessary ports and services

NSTB vulnerability assessments evaluated open ports and running services on vendor-configured SCADA hosts and found that, in some cases, vendor- published ports and services did not match those ports and services actually observed. Because the lists of required ports and services have been found to disagree with delivered systems, SCADA owners can validate the necessity of services installed on new systems before they are deployed.

SCADA owners can monitor their own system traffic and create rules that describe their system's behavior. This must be done with care to avoid overly restrictive rules that create the risk of DoS for legitimate control traffic. In some cases communications cannot be blocked and sometimes communications require access between security zones. The design of the communication protocol determines the degree to which access to these services can be restricted. Another concern is that some SCADA protocols require an excessive number of ports to be opened. For example, NSTB assessments found one service that required over 21,000 ports to be opened.

Traffic monitoring between system components during all phases of acceptance testing can be used to help identify required communications. Ideally this testing and verification of required ports and services can be determined by the vendor during system design, then tested and verified by the owner/operator before operation. This technique may eliminate communications not required by a particular SCADA configuration. This can create a more secure system, if owners exercise all potential functionality to ensure that it will be available when needed.

This same process can be used with great care to minimize risk on an operational SCADA system. The service can first be moved onto a backup or development system to insulate the primary system from potential damage. Before stopping any services or programs on an operational system, SCADA administrators can 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. In addition, IDS logs and system logs may also inform administrators when requested services are not available. Finally, it is necessary to employ monitoring systems that ensure removed or disabled services are not re-installed or re-enabled.

4.1.2.2 Recommendations and resources

To minimize the cyber-attack surface, such as possible access paths in unnecessary ports and services, it is recommended that:

- Vendors create a secure configuration
    - Restrict the ports, installed services and applications used to support their systems to the minimum necessary
    - Identify and disable all OS or third-party application services not explicitly needed for the SCADA to operate
- Vendors identify and document
    - All OS or third-party application dynamic port services and respective port ranges
    - The services needed by each SCADA component and the port ranges that each needed service uses, and also explicitly identify each device that is allowed to initiate a connection with one of these ports
- Vendors include ports and services documentation and secure configuration as part of the product deliverable
- Owners create a secure network architecture
    - Use vendors' documentation and secure configuration
    - Validate the necessity of services installed before they are deployed
    - Remove unneeded applications and services with great care
    - Owners document the way SCADA system components use the network so that effective firewall and IDS rules can be created

4.2 Known vulnerabilities

Known vulnerabilities in common IT products installed on SCADA hosts have a high probability of being attacked and may provide an attack path into the system using publically available exploit code. While the NSTB has taken care to inform only the responsible SCADA vendors of vulnerabilities identified in their products, other security researchers are starting to announce SCADA vulnerabilities in more open forums. "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]

New vulnerabilities in computer applications and services are found every day. Some are published shortly after their discovery. Others are kept a close secret by those who discover them. These remain unpatched and can be exploited at will by their discoverers. They may not result in direct control of the SCADA, but may help provide an attack path into the system for external attackers as well as malicious insiders. The NSTB SCADA assessments use publically available tools to identify and develop mitigations for publically known vulnerabilities. This section discusses mitigation of known vulnerabilities through removal of all unneeded applications and services, and effective patch management.

4.2.1 Minimize applications and services

The risk of successful exploitation due to existing vulnerabilities in available services increases with each additional service. Reducing the number of installed applications and services decreases the likelihood of an attacker finding a vulnerability.

Before stopping any services or programs, the vendor can confirm the service is not needed for system functionality. 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 could disable the entire system.

As discussed further in the next section, adequate resources need to be allocated to ensure that all services and applications are completely patched and up-to-date. SCADA owners/operators must rely on their SCADA vendor for validation of patch compatibility before testing and applying patches to their operational system themselves. One way to reduce this resource-intensive activity is to reduce the number of applications that need to be patched. Therefore, removing all unneeded applications and services is recommended.

4.2.2 Implement effective patch management

NSTB assessment findings make clear the importance of effective patch management for operating system (OS) and non-OS applications, services and libraries, as well as third-party software. SCADA product vendors can provide mitigation to vulnerabilities in their products by deploying patches to their users. Identifying and patching vulnerabilities rapidly minimizes the risk of public discovery before mitigation is implemented. SCADA vendors can also test patches to third-party products and incorporate them into their base product. This is especially important for upgrades to third-party products, which may require changes to the SCADA code that interfaces with them.

Operating system patches repair vulnerabilities in the operating system that could allow an attacker to exploit the computer. OS patch management support has improved to the point where OS patching is common for most situations, and in particular has been improving in new product installations and on production systems. Many SCADA vendors now provide timely OS patch test results for their newer releases.

While operating system (OS) patch management support has improved and many SCADA vendors now provide timely OS patch test results, application patching is difficult and often delayed. 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]

NSTB assessments found that applications, services, and libraries not included as part of the OS tend not to get patched on many SCADA systems. Non-OS services and libraries are inconspicuous and consequently often neglected. Many developers and administrators may not even be aware of them. Statically linked libraries can be independently kept up-to-date if they are different from the libraries associated with the operating system.

SCADA software generally uses third-party applications such as common Web servers, database servers, remote access services, and encryption services. NSTB assessments have discovered out-of-date, unpatched and vulnerable versions of third-party software applications and services incorporated into new SCADA software, on production SCADA. This indicates that new SCADA systems may be installed with vulnerable software and that SCADA system vendors may not be supporting third-party patch management for their software. This becomes a significant problem if these products cannot be patched by the SCADA owners.

Patching SCADA machines can present unique challenges. Application patching becomes increasingly difficult if security fixes change the way that the SCADA software interfaces with other applications. Often new versions of third-party applications, services and libraries have changed the Application Programming Interface (API) so that the third-party product cannot be upgraded without changing the vendor’s SCADA code that it interfaces with, and SCADA owners may be unable to make the needed modifications to the SCADA vendor’s code.

OS and application patching is resource-intensive for SCADA users since patches need to be thoroughly tested in a development environment prior to use in production systems. Thorough testing by both the SCADA vendor and owner is needed to ensure the patch does not jeopardize SCADA functionality, security, and timeliness.

The patching process requires close collaboration with vendor support to ensure SCADA application integrity is maintained. Any patch process test should be first performed on a backup or development system before being implemented on an operational system, to isolate the primary system from potential damage. The testing process can exercise all functionality to reveal any undesired behavior that may result from the patch. Functionality tests used during development, factory acceptance, and site acceptance testing can be used for patch testing on the vendor’s code base and the owners’ test systems.

Asset owners/operators can identify and deploy security workarounds, defense-in-depth strategies, and use monitoring methods and tools to mitigate risk introduced by the presence of unpatched vulnerabilities until patches can be properly tested and deployed. For example, properly configured firewalls, IDS, and antivirus solutions can be deployed to mitigate risk

Figure 7 shows the percentage of NSTB assessments that found un-patched software in SCADA system, categorized by the SCADA software product.

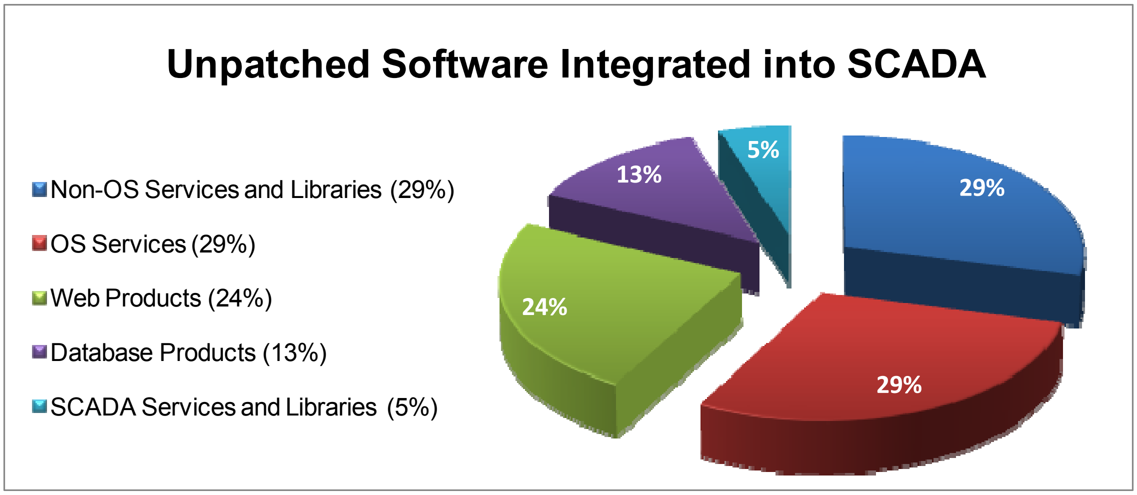


Figure 7. Percentage of unpatched components NSTB assessments found integrated into SCADA systems.

4.2.3 Recommendations and resources

To mitigate known vulnerabilities through effective patch management, it is recommended that:

- Vendors create a procedure and have personnel assigned to integrate updates into their products
  - Upgrade and patch products and services used by the SCADA system to current version and patch levels prior to deployment
  - Test operating system patches for compatibility with their SCADA system and provide the testing results to users as soon as possible after the patch release, to limit the length of time the customer's system remains vulnerable.
  - Test and approve OS, non-OS and third-party software patches
  - Statically linked libraries are independently kept up-to-date if they are different from the libraries associated with the operating system.
- Vendors strengthen methods of notifying customers about potential security problems and patches
  - Create and maintain security mailing lists with readily-accessible contact information of researchers and customers
  - Practice the procedures used to notify customers about security problems
- Vendors design SCADA products that have third-party services and applications incorporated into their functionality to be updated or replaced as easily as possible
- Owners test patches before deployment, even if the vendor has approved them and particularly if vendor support is unavailable

More detailed guidance on production SCADA patch management can be found in the DHS *Recommended Practice for Patch Management of Control Systems*[6] and the National Institute of Standards and Technology (NIST) *Guide to Industrial Control Systems (ICS) Security.*[7]

4.3 Communication channel vulnerabilities

As SCADA systems become increasingly connected to company intranets and to the external Internet, they can also become more exposed to cyber attack. Communication channels are a focus of NSTB assessments because they often connect different network security zones and may have access rights and functionality to manipulate the SCADA system. NSTB assessments looked for communication channel vulnerabilities that allow:

- SCADA authentication credentials gathering
- SCADA data and command spoofing and manipulation
- DoS of SCADA communications that jeopardizes functionality.

This section first discusses common IT communication protocols used for common IT functionality in SCADA systems then discusses SCADA communication protocols used to transmit SCADA data and command messages. The discussion covers the shared need for rigorous protection of authentication credentials during transmission in both cases. Then it describes the need to secure remote access using common IT protocols, and the opportunity to detect intrusion by performing SCADA data integrity checks. Finally, this section explores the benefits and challenges of data encryption for SCADA systems.

4.3.1 Common IT protocols in SCADA systems

SCADA systems use common IT protocols for common IT functionality, such as network device management, remote logins, or file transfers. Although more secure alternatives are available for most of these services, active unused or obsolete services may still be present in many SCADA systems and may be vulnerable to spoofing and MitM attacks. SANS.org defines spoofing[34] as "Attempt by an unauthorized entity to gain access to a system by posing as an authorized user" in the Glossary of Security Terms, SearchSecurity.com defines Man in the Middle[35] attack as "…an exploit in which an intruder intercepts and alters communications between two parties."

Because they are not used for real-time functionality, they can be replaced with their secure counterparts in most cases. As an example, SSH can replace all file transfer and remote login protocols such as FTP, telnet, and rlogin with encrypted versions. Also, any communication protocol can be "tunneled" through SSH, and HTTP can be sent over the Secure Socket Layer (HTTPS).

The use of insecure IT protocols in SCADA systems can be high risk because attackers are aware of their known vulnerabilities, and may have access to automated tools created to exploit these vulnerabilities. For example, network protocol analysis tools can intercept and decode most common protocols.[36] Likewise, password cracking tools can decode messages and passwords that have been encrypted with weak or improperly implemented encryption schemes.[37]

Insecure protocols and services connected to the SCADA system hosts can create a high-risk access path into the system. This introduces a significant vulnerability because it can provide remote access to SCADA hosts along with access to the functionality allowed by the privilege-level of the remote user whose authentication credentials were stolen.

4.3.1.1 Protect IT authentication credentials during transmission

Clear-text authentication credentials can be captured (sniffed) during transmission. If an attacker is able to capture a username and password, then the attacker can log onto the system with that user's privileges. Therefore, plain-text remote login services can be replaced with encrypted services such as SSH, as discussed above. This can be particularly important in multipoint networks such as Ethernet. Table 10 lists examples of common vulnerabilities associated with clear-text authentication protocols used in SCADA systems.

Table 10. Common vulnerabilities associated with insecure common IT communications protocols

Common Vulnerability	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	

4.3.1.2 *Implement secure remote access*

Remote display protocols and applications provide the ability to logon and remotely control another machine using a graphical display. The use of common remote display software for remote access to supervisory control functions could be the most significant vulnerability on a SCADA system. Easily exploited known vulnerabilities in common remote display software could allow unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user. Vulnerability exposure depends on how and where the connection is initiated.

NSTB assessments have found common remote display protocols used by SCADA systems that accept connections from anywhere and transport credentials in clear text or by a weak 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 SCADA host's display and all of the client's functionality including the encrypted channel. Table 11 lists examples of common vulnerabilities found in remote display protocols. In the appendix, Table C-10 "Summary of SCADA Web application security characteristics" lists common vulnerabilities in remote display software used by SCADA for remote access to supervisory control functions.

Table 11. Common vulnerabilities found in software used by SCADA for remote access to supervisory control functions.

Common Vulnerability	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 weak or improperly implemented cryptographic algorithm	Attacker is able to steal remote display credentials by "sniffing" and decrypting network traffic while a remote display connection is established

4.3.1.3 Recommendations and resources

To find and remediate communication channel vulnerabilities in common IT protocols and common remote display protocols, it is recommended that:

- All stakeholders follow IT security practices and use the current secure versions of common protocols
- All stakeholders where possible replace insecure versions of common IT services with their secure versions
    - Replace plain-text remote login services with encrypted services such as SSH.
    - When replacement is not feasible, minimize access to the services, and limit unencrypted communication to within the SCADA whenever possible.
- Vendors remove insecure protocols in their products
- Vendors provide secure options for remote access of SCADA hosts and provide secure default configurations that include access restrictions and secure authentication, including configuration of the host to validate the security of the remote access client to protect against unauthorized access through a trusted compromised client host.
- Vendors notify owners when more secure remote access and file transfer solutions are available
- Owners configure remote access protocols and services on SCADA hosts to limit access, require strong authentication, use a trusted path, and be kept up to date

4.3.2 SCADA data and command message communication protocols

SCADA network protocols, including those used to send control commands and status data, can be vulnerable to MitM attacks, altered, replayed, or spoofed if they lack sufficient access control and integrity checking mechanisms. This vulnerability can be exploited with minimal skill to intercept or create the network messages. An attacker's ability to intelligently interpret and manipulate process status depends on how much of the SCADA protocol and physical process can be discovered and reverse engineered.

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

Most SCADA network protocols were designed with the original SCADA code base to be fast and are not designed to provide robust authentication and integrity checks. As a result, many protocol designs contain common security pitfalls. When possible, network protocols, and the service applications that implement them, can be re/designed to improve security by avoiding common security pitfalls such as designs that make implementation difficult, and by including secure authentication and encryption methods.

Several characteristics of secure protocol are relevant to this discussion. Secure protocols are simple, minimize duplicate data, are streamlined, have secure authentication methods and options for encryption or data integrity and do not rely on security by obscurity.

Protocols that are secure are also simple. More complex protocols are more likely to have vulnerabilities within the implementation.

Protocols that are secure 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 that are secure are streamlined. They contain enough functionality to get the job done and nothing more. Protocols with many optional fields and features are less secure because no two implementations will agree on what is optional, which invites incorrect assumptions. Also, if protocols contain seldom used or never used components, then those components tend to contain more vulnerabilities than well-used components because they will be tested to a lesser degree.

Protocols that are secure have secure authentication methods and options for encryption or data integrity.

Protocols that are secure do not rely on security by obscurity. Insider knowledge or reverse engineering can be used to recreate valid network packets. Some SCADA protocol analyzers have already been developed and, given the increasing interest in SCADA security, one can expect to see more.

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

4.3.2.1 Protect SCADA authentication credentials during transmission

Authentication can protect against unauthorized access to SCADA applications. If a SCADA system allows clear-text authentication credentials (username and password), they can be sniffed during transmission and used by an attacker to authenticate to the SCADA application. The attacker can then log into the SCADA application with that user's privileges. Table 12 lists examples of unprotected SCADA application credentials sent over the network.

Table 12. Common vulnerabilities associated with transmission of unprotected SCADA authentication credentials

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

4.3.2.2 Implement secure access control and check data integrity

SCADA communication protocols may not have been designed and implemented to adequately verify the identity of actors at both ends of a communication channel, or ensure the integrity of the channel, in a way that prevents the channel from being accessed or influenced by an actor that is not a legitimate endpoint. Weak authentication can prevent the protocol from detecting that the message is from an unauthorized sender. However, some SCADA protocols may rely on weak authentication, such as hostname or IP address, which can be easily spoofed. In a spoofing attack, false information could be sent to the operator's console. Likewise, inadequate data integrity checks can prevent a protocol from detecting bad or corrupted data. If integrity check values or "checksums" are omitted from a protocol, there is no way of determining whether data have been corrupted in transmission. If integrity check values are easily reverse engineered and duplicated, data manipulation in transmission can remain undiscovered despite security inspection.

4.3.2.3 SCADA data encryption considerations

A long-term mitigation to secure SCADA communication channels could be to replace SCADA protocols with an encrypted version of the protocol. However, there is no drop-in replacement currently available—with the exception of secure ICCP. Even if there were, it would be an infeasible task to rewrite, test and validate all of the software involved.

One near-term mitigation option is to use IPSec with only the authentication header (AH). IPsec[38] is a network layer security control implemented through a framework of open standards for ensuring private communications over public networks. AH mode does not encrypt the data, but it provides 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 near-term mitigation option is to transmit the insecure protocol through an encrypted protocol tunnel, such as IPSec, SSL, etc. Encryption solutions, such as IPSec and VPN tunneling, can be used for confidentiality, integrity, authenticity, and/or replay protection.

This would require that the components at the communication endpoints have the software installed, configured, and tested to ensure the setup is correct and secure, which could be a substantial 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, this approach introduces the need for dedicating labor resources to key management.

Another difficulty of this option is that all of the communications between the systems can be blocked if the end points are susceptible to ARP cache poisoning. In this case, depending on the particular IPSec configuration communications would either stop, or continue in an unencrypted mode.

A recent trend in the SCADA industry has been to encrypt core SCADA communications with IPSec which can be configured not to jeopardize critical communications. SCADA hosts that require high availability can be configured with an IPSec "request" policy instead of "require". This means that encryption is requested, but communications will continue in clear text if the IPSec connection cannot be established. On the other hand, if the IPSec policy is set to "require" IPSec for communications this failure can cause DoS. 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 is based on whether the highest priority for the communication between the IPSec partners is encryption (confidentiality, integrity, authenticity, or replay protection) or is high availability.

Encryption solutions that tunnel SCADA 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.

A final near-term mitigation 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 challenges. However, the addition of encryption capabilities to SCADA products may allow the communication channels to be secured.

SCADA vendors may find secure design and vulnerability remediation activities to be impractical due to time, cost, and/or backward compatibility issues and look toward encryption of SCADA communications as mitigation to all vulnerabilities on the SCADA. While adding encryption can limit exposure, this does not prevent access through the encrypted channel if an attacker has compromised an encryption endpoint. Consequently, these encryption solutions cannot be used as a replacement for fixing vulnerabilities because a VPN connection extends the attack surface of the system to the VPN client's computer. These encryption solutions do not prevent an attacker from compromising a VPN endpoint computer and using the VPN tunnel as an encrypted pathway to exploit vulnerabilities in the other endpoint host.

A remote end-point joins the trusted domain when it is allowed to remotely connect to the SCADA network. If VPN endpoints (hosts) are compromised, an attacker may be able to utilize the VPN connection when it is established so secure these hosts 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 is not recommended as the only defense measure. VPN access can be granted to the minimum set of hosts and users when necessary and those VPN connections can be restricted to allow access only to the necessary components.

The difficulty of implementation and the introduction of an impediment to viewing network communications when needed for trouble-shooting can prevent the use of encryption for operational IDS. Encryption can pose a risk to network throughput, bandwidth, availability, and IDS capabilities, as well as CPU availability on endpoint systems. Encryption can also make system monitoring difficult, and can add undesired latency to communications. NSTB experience and feedback have shown that encryption of SCADA communications is difficult and is rarely accomplished successfully. NSTB assessments have found insecure encryption configurations. Encryption might not be implemented because it could not be accomplished without disabling SCADA communications, or because the communications partner did not support it.

However, encryption can be implemented as a layer of defense, where confidentiality or integrity is a higher priority than availability (i.e., external and non-critical connections), to provide confidentiality and prevent MitM attacks between encryption endpoints. Encryption can be used to support strong authentication and authorization, and protect from unauthorized access to data in files or in transit, but it is not recommended as a primary mitigation for other types of vulnerabilities, such as input validation weaknesses. SCADA designers and administrators can carefully consider the priorities of each communication channel when implementing encryption. An appropriate encryption solution can be selected for each SCADA communication channel that can handle the associated risk, support encryption, and be configured securely and safely to support the SCADA's priorities. Administrators can securely manage and protect cryptographic keys that are strong and are not hard-coded, default, published, or discoverable. SCADA designers and customers can refer to current and specific documentation on network cryptography options and implementation instructions from their providers and reference standards and guidance material from the NIST Cryptographic Toolkit[39] project. Encryption may not fix the vulnerabilities in a SCADA system, but it can be used as a layer of defense.

4.3.2.4 Recommendations and resources

To protect SCADA data and command messages from manipulation and ensure they have not been altered in transit, it is recommended that:

Stakeholders:

- Use secure authentication and data integrity checks
- Require physical access for controller configuration and firmware update
- Vigorously protect user credentials and make inaccessible to an attacker
- Secure as much as possible communications between security zones
- For long-term mitigations, replace insecure SCADA protocols with protocols that provide strong authentications and integrity checks
- Vendors:
- Ensure their system design implements strong authentication into SCADA communication protocols
- Validate that their products securely encrypt or hash passwords before storing or transmitting them
- Integrate the latest ICCP protocol stack into their products
- Owners:
- Take defensive actions to reduce exposure including secure network access, anomaly detection, and content filtering rules
- Use IDS monitoring to detect the attacker's presence on the network and MitM activities
- Configure network equipment to prevent MitM attacks, possible defensive options include:
    - Hard-code the Media Access Control (MAC) addresses of the communication endpoints in each system's ARP tables.
    - Employ all of the features of the installed networking equipment, such as port security on switches and 1-to-1 rules on firewalls.
    - Employ IDS solutions that detect ARP MitM activities
- When using Web applications with Secure Sockets Layer (SSL), SSL is used for the entire session from login to logout and not just for the initial login page
- Secure VPN end-point hosts to the maximum extent possible
- May be able to secure SCADA application connections using third-party encryption solutions
- Validate that their ICCP implementations use the latest versions or patches

More information about communication channel vulnerabilities can be found in the related security weaknesses from the CWE[9]:

- 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.

Domain Name System (DNS) spoofing can be protected against by using the available DNS security measures.[16]

The DHS *Control Systems Communications Encryption Primer*[17] provides SCADA specific information on encryption. More detailed information on recommended configurations is available in the NIST Special Publication 800-113.[18]

4.3.3 Protect SCADA data and command messages against man-in-the-middle

Network traffic can be exposed to man-in-the-middle (MitM) attacks that gather unauthorized information, alter messages, or drop messages. A MitM attack is possible if the communication protocol does not authenticate the identity of each communication partner or ensure the integrity of the message. If an attacker can pose as a legitimate communication partner 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 communications of a target system. The ARP protocol is used to determine which hardware addresses coincide with the Internet Protocol (IP) addresses on the network. The MitM attack can be effective against multipoint networks because the hosts send their data to the attacker's (compromised) computer thinking it is the intended recipient of their data. This attack exploits the network ARP caches of hosts on the LAN and generally requires that the attacker compromise a host on the victim computer's LAN. The MitM attack sends deceptive ARP commands that direct the two hosts to communicate with the attacker computer which poses as the intended recipient. In a successful MitM attack, the hosts on each side of the attack are unaware that their network data has been redirected through the attacker's computer, which must forward all packets to the intended host so the connection stays synchronized and does not time out. Multiple free, publicly available tools exist to perform ARP poisoning and MitM attacks, as well as brute-force and dictionary attacks (attacks where a password can be obtained by systematically entering every word in a dictionary as a password) against Terminal Services to local or remote access applications.

If the attacker has gained access to a host that is allowed to send control messages, then the attacker need not invest effort in implementing a MitM attack. In that case, even if the control protocol is encrypted the attacker can still send control messages by gaining access to the host that encrypts the packet.

SCADA communication protocols may be vulnerable to spoofing and MitM attacks because they may not have been designed and implemented to prevent these attacks. Strategically manipulating the communications on a control network requires an in-depth understanding of the protocol and process being manipulated. The NSTB assessment team is generally able to gather enough information about a network protocol to perform a network layer attack against the system. With a full ARP MitM attack in place, an attacker can intercept and manipulate commands and messages that control SCADA devices, control field equipment and/or modify data flowing back to the operator's console to misrepresent the system state. This tampering could allow an attacker to manipulate the system and keep the operator unaware of the attacker's interference so that the operator is deceived into performing dangerous actions or not taking the appropriate preventative actions.

The following attack scenario was demonstrated in multiple NSTB assessments, and could be accomplished on any network link between the client and server: Using a freely available tool called Cain, the NSTB assessment 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.

One defensive measure against MitM attacks is to hard-code the Media Access Control (MAC) addresses of the communication endpoints in each system's ARP tables. The systems will then ignore the spoofed ARPs sent during an ARP MitM attack, which makes the attack ineffective. This mitigation approach comes with several challenges. Some systems do not provide a way to hard-code the ARP tables. Some systems provide a temporary method, but it must be repeated 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. And finally, this only protects against one MitM attack method, ARP spoofing. Another defensive measure that is also relatively inexpensive is to employ all of the features of the installed networking equipment, such as port security on switches and 1-to-1 rules, static addressing, on firewalls. However, this may not be a viable option because firewalls are not commonly used in the basic control communications path. A third option is to employ IDS solutions that detect ARP MitM activities. LAN MitM techniques include ARP spoofing, DNS spoofing, IP address spoofing, port stealing, and STP (Spanning-Tree Protocol) mangling.[40] Techniques that can be used to create a MitM from a local network to a remote network through a gateway include ARP poisoning, DNS spoofing, DHCP spoofing, Gateway spoofing, ICMP redirection, and route mangling. Remote MitM attacks can be performed using DNS poisoning, route mangling and traffic tunneling.

4.4 Communication endpoint vulnerabilities

Network services at communication end-points listen for messages to accept, and can be exposed to attacks that exploit input and output validation vulnerabilities to create a buffer overflow or perform an SQL injection. These services are vulnerable to remote compromise if they do not properly check the size of user input, filter out unneeded and potentially malicious character sequences, and initialize and clear variables properly, as discussed above in section 4.3.

NSTB assessments look for SCADA programming errors in functions that parse network code without proper validation or "sanity check" of input values. Vulnerabilities in services that parse network traffic can allow unauthorized access to their host. NSTB assessments also focus on authentication systems because these systems can allow authentication bypass if they are not implemented correctly.

This section discusses vulnerabilities in SCADA and ICCP services that could be exploited to create a buffer overflow. Then it presents vulnerabilities found in SCADA historian database that could be exploited by an SQL injection. Finally, it describes vulnerabilities found in SCADA web applications and services that could be exploited to allow directory traversal or cross-site scripting. Each of these cyber-attacks – buffer overflow, SQL injection, directory traversal and cross-site scripting – were discussed above in section 4.3 and are mitigated through secure coding practices that enforce validation of input data integrity.

4.4.1 SCADA and ICCP services

NSTB assessments have found input validation vulnerabilities in custom server applications written to process SCADA protocol messages and other SCADA network traffic. These applications are:

- Control protocol services

- Supervisory control protocol services
- ICCP services.

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. 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.

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 communication 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. 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 an NSTB assessment. Figure 8 shows how these entities could be connected via ICCP.

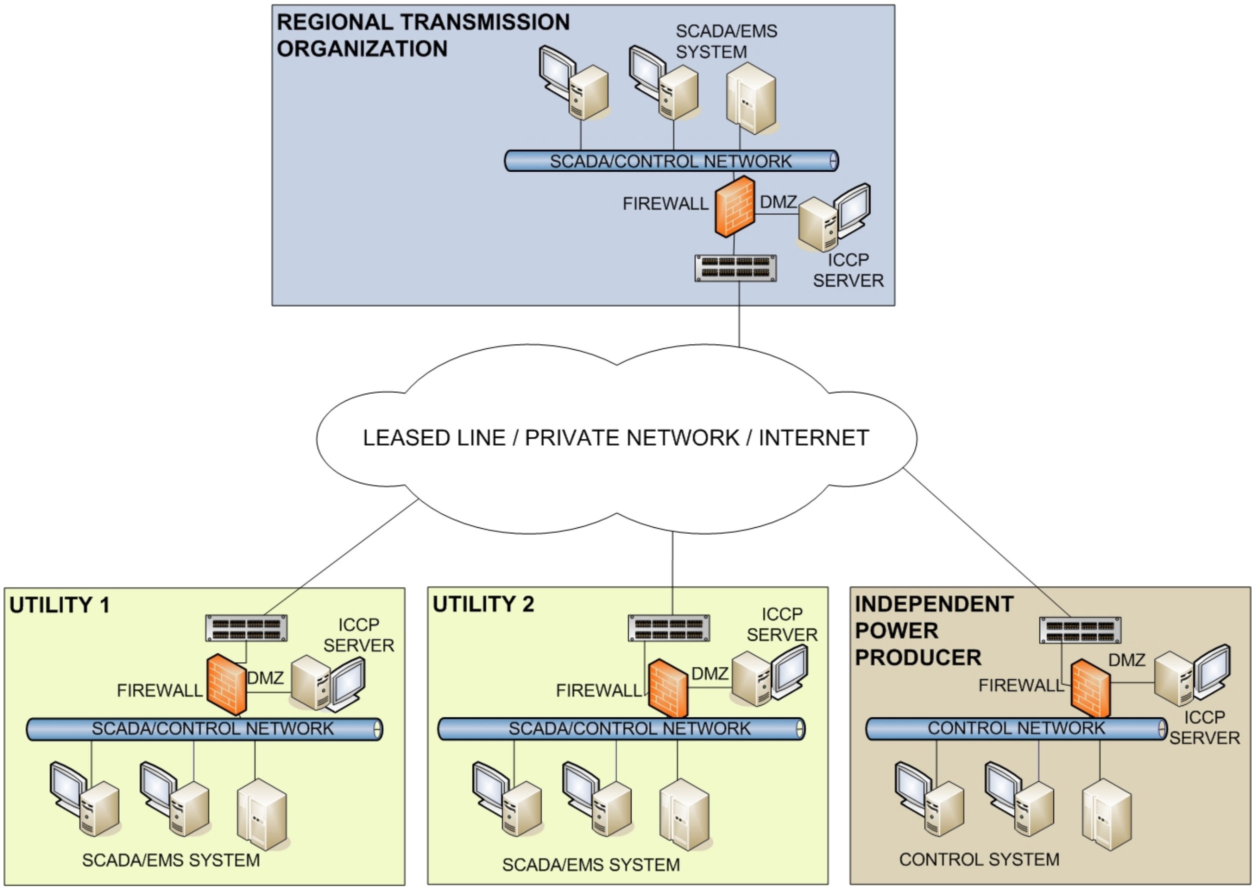


Figure 8. Sample ICCP network

The majority of NSTB findings in the ICCP assessment were vulnerabilities that, if exploited, could cause buffer overflow and DoS. Buffer overflow vulnerabilities can potentially allow an attacker to take control of the ICCP server, providing a possible path to the SCADA network. DoS events are less severe, but can be used to cause an outage of the ICCP service. The NSTB also found that the complexity of the ICCP protocol contributed to the number of vulnerabilities found. Even if all functionality in the ICCP protocol is not used, the vulnerabilities in the unused layers can still be available for attackers. In addition, some new ICCP implementations were found to use older versions of the third-party protocol stack. These older versions contained known vulnerabilities, including multiple vulnerabilities that could be exploited to create DoS and vulnerabilities that could lead to remote code execution. SCADA vendors can integrate the latest ICCP protocol stack into their products, and owners validate that their ICCP implementations use the latest versions or patches. The results from these ICCP assessments have been consolidated into a public report, available from the NSTB.[5]

4.4.1.1 Protect SCADA services against buffer overflow

Most buffer overflows identified in NSTB assessments were in the server applications that process SCADA protocol traffic. In most cases, values input from network traffic were intercepted and altered in transit. Therefore, network data bounds and integrity checking can be implemented to reduce risk.

Every network protocol has an associated program that builds packets or processes communications traffic off the network. These applications are written by the SCADA vendor for their propriety protocols as well as for common SCADA protocols, such as Object Linking and Embedding (OLE) for Process Control (OPC), ICCP, and DNP3. If these applications are vulnerable to invalid input, then an attacker who is able to gain network access may exploit this vulnerability to create a buffer overflow. In this case exploitation by anyone who is able to gain access to the SCADA host and port is possible. Remotely exploitable vulnerabilities can be accessed over the network (i.e., an attacker does not require local network access or local host access). The more remote an attacker can be to attack a host, the more likely it is that the host will be exploited.[2]

Vulnerabilities in services that are exposed to less-trusted networks may 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. SCADA design requires that SCADA protocols be allowed through firewalls to support external data collection and sharing. These protocols and services should have top priority for vulnerability remediation activities.

Data integrity checks can be designed and implemented into SCADA communication protocols. The lack of, or weak, data integrity checks prevent a protocol from detecting bad data. An attacker can take advantage of the improper 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 SCADA protocol has improper integrity checks.

4.4.1.2 Recommendations and resources

To protect against buffer overflow cyber attack, it is recommended that SCADA software vendors:

- Provide validation of all input data, not just those proven to cause buffer overflows.

- Ensure programmers are trained in secure coding practices
- Review and test all code for input functions that could be susceptible to buffer overflow attacks.
- Use automated static analysis tools to detect buffer overflow vulnerabilities in their code.
- Use dynamic tools and techniques such as fuzz testing, robustness testing, and fault injection.
- Integrate the latest ICCP protocol stack into their products

Owners are recommended to validate that their ICCP implementations use the latest versions or patches.

When allocating and managing an application's memory, the MITRE Common Weakness Enumeration report recommendations include:

- Double check that your buffer is as large as you specify.
- When using functions that accept a number of bytes to copy, such as strncpy(), be aware that if the destination buffer size is equal to the source buffer size, it may not NULL-terminate the string.
- Check buffer boundaries if accessing the buffer in a loop and make sure you are not in danger of writing past the allocated space.
- If necessary, truncate all input strings to a reasonable length before passing them to the copy and concatenation functions.[9]

Vendors, and possibly owners, can take additional precautions to help prevent exploitation or limit damage from buffer overflow vulnerabilities. Languages, libraries, and frameworks that do not allow for some types of buffer overflows may be used in software development. C-based programs are known 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.

Developers can also use compiler options to detect some types of buffer overflows; however, an attack could still cause a DoS, since the typical mitigation is to exit the application.

Some SCADAs only support one CPU and/or OS. SCADA vendors and owners if possible, can use a CPU and OS that offers protection against buffer overflow attacks. Again, this is only a defense in depth solution because they will not prevent all attacks and could still cause DoS because the typical mitigation is to exit the application.

4.4.2 Database applications and services

An historian server archives data, performs analyses and is integral to most SCADA systems. It is usually located in a DMZ or on the corporate network. Consequences of an exploited vulnerability for the historian include compromise of the historian host and data corruption. SCADA historians typically use a common SQL server as the backend, and historical data are often made available for viewing via a custom Web interface or application that uses SQL queries to retrieve information.

The historian client applications can be high-risk components because they are often accessible from the corporate environment and can provide an attacker with a point of entry to the SCADA network if not properly secured. Additionally, an attacker may gain access to unauthorized information which, in some cases, can be used to cause economic damage.

NSTB assessments have found unsafe function calls in code written to parse historian data messages (see Table 13). Rigorous input validation is required to protect against a DoS or unauthorized access to the associated host. One potential attack pathway is SQL injection, another is weak authentication that may be defeated to gain database access.

Table 13. Common vulnerabilities associated with the historian database

Common Vulnerability	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	
Insecure 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 Protect SCADA historian against Structured Query Language (SQL) injection

SQL injection vulnerabilities in applications that access data from the SCADA historian database allow an attacker to execute arbitrary SQL queries and/or commands on the database. A successful SQL injection exploit of a SCADA historian may allow the attacker to:

- bypass authentication to the historian-backed application,
- read historical data from the database (and potentially user names and passwords),
- modify historical data,
- execute administration operations on the database (such as shutdown the historian database),
- recover the content of a files present on the historian's host file system, and
- in some cases, issue commands to the historian's host operating system.

4.4.2.2 Recommendations and resources

To find and remediate database application and services vulnerabilities, such as SQL injection and weak authentication, it is recommended that:

- All stakeholders use the principle of least privilege

- Vendors use vetted libraries or frameworks that prevent SQL injection vulnerabilities or provide constructs that make these vulnerabilities easier to avoid
- Vendors use secure coding practices when constructing SQL queries
- Owners replicate databases out to the DMZ
- Owners with Web servers use an application firewall to detect common Web attacks

Information regarding SQL injection that is specific to SCADA systems is available from the ICS-CERT portal.[19] For further information on SQL injection:

- CWE-89: Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection')[9]
- *SQL Injection Attacks by Example*[20]

The *Open Web Application Security Project (OWASP) Top Ten for 2010*[21] 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)
    a. Bind variables allow the interpreter to distinguish between code and data
3. Encode all user input before passing it to the interpreter
    a. Always perform “white list” input validation on all user-supplied input
    b. Always minimize database privileges to reduce the impact of a flaw
4. Follow the guidance from the OWASP *SQL Injection Cheat Sheet.*[22]

4.4.3 Web applications and services

Many SCADA systems have recently incorporated Web applications and services to allow remote supervisory control, monitoring or corporate analysis of SCADA data. NSTB assessments have found unauthorized directory traversal and authentication vulnerabilities within SCADA Web implementations. Many of the code quality and input validation vulnerability findings in this report refer to proprietary Web applications.

Common vulnerabilities found in SCADA Web services along with associated potential impacts are listed below in Table 14. 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.

The SANS 2009 report indicates that Web vulnerabilities are 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]

NSTB assessment indicate that lack of good programming practices can result in SCADA Web services being vulnerable to the most popular attack techniques such as Structured Query Language (SQL) injection, cross-site scripting (XSS), directory traversal and authentication bypass. SCADA Web applications tend to be more exposed to attack than most SCADA components, and may provide the capability to alter SCADA data or state. Table 14 shows common vulnerabilities NSTB assessments found associated with Web services.

Table 14. Common vulnerabilities associated with Web services

Common Vulnerability	Potential Impact
No authentication between corporate clients and Web server on DMZ	Unauthorized access from corporate network to DMZ
HTTP Port 80 had no default page. Displayed directory structure.	Unauthorized access to files and directories on the Web server Compromise of Web server
Arbitrary files can be read on Web server by adding ../../ or ..\..\ in front of file name.	
Multiple cross-site scripting Vulnerabilities	Compromise of Web client
Persistent cross-site scripting Vulnerability	
Cross-site scripting on Login and History Analysis Pages	
Browser plug-in exploit allowed control of workstation	

4.4.3.1 Control access to web services

NSTB assessments have found authentication, session tracking, structured SQL injection and XSS vulnerabilities that can allow unauthorized access to Web servers and applications. Web services developed for SCADA systems can be vulnerable to attacks that exploit the SCADA Web server to gain unauthorized access. System architectures often use network DMZs to protect critical systems and limit exposure of network components. Vulnerabilities in SCADA DMZ Web servers may provide the first step in the attack path by allowing access within the SCADA exterior boundary. Vulnerabilities in lower-level component's Web servers can provide the next steps in the attack path.

4.4.3.2 Protect SCADA web services against cross-site scripting

Cross-site scripting presents an entry point for attackers to access and manipulate SCADA networks. 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.

4.4.4 Recommendations and resources

To find and remediate web application and services vulnerabilities, such as SQL, injection, directory traversal, XSS, authentication, and session tracking, it is recommended that:

- All stakeholders that perform security checks on the client side duplicate them on the server side.

- All stakeholders decode and convert inputs to the application's current internal representation before being validated. When the set of acceptable objects, such as filenames or URLs, is limited or known, developers can create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs.
- Vendors design their systems to perform security checks on the server side.
- Vendors perform input validation using a white list of acceptable inputs that strictly conform to specifications, reject any input that does not strictly conform to specifications, or transform the input so that it does conform to specification
- Owners minimize and validate that the access controls are configured to restrict all unwanted users and available functionality of:
  - Web servers and clients
  - SCADA
  - Owners place web servers on a DMZ

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

1. SCADA Internet access policy
2. SCADA user awareness and training
3. Coordination of security efforts between corporate IT network and SCADA network
4. Firewall between the SCADA network and the information technology network
5. Up-to-date patches
6. Web browser and e-mail security
7. Secure code.[23,24]

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*[21] ranks the most critical Web application security flaws and 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.[22] The CWE can also be referenced for information about Web security weaknesses. CWE categories, CWE-287: Improper Authentication and CWE-442: Web Problems*,* contain related authentication and Web programming information as well.[9] Table 15 lists related OWASP and CWE resources.

Table 15. 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

4.5 Authentication vulnerabilities

High risk or common SCADA authentication vulnerabilities are discussed further below, and include:

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

Many of the input and output validation vulnerabilities described in Section 4.4, Communications end point vulnerabilities, have the potential to bypass authentication. Authentication is used to enforce access controls. Weak authentication can allow access controls to be subverted. NSTB SCADA security assessments have shown that access to process data and control functionality can be achieved because authentication is not required, or authentication can be circumvented.

Strong authentication and encryption mechanisms can be implemented and strenuously tested to mitigate authentication vulnerabilities.

Applications that process network traffic or accept network connections can use strong authentication to prevent unauthorized access and messages. Without sufficient protection, weak authentication in network protocols may allow information disclosure, or replay or spoof attacks that send unauthorized messages. Improper authentication may also allow unauthorized users or computers to connect to a device or application. The lack of authentication in most SCADA-specific network protocols may allow for manipulation of time synchronization and process alarms, commands, and data updates. Improper authentication in protocol server applications can allow unauthorized access to SCADA components, including SCADA hardware. Proven authentication services can be used when available.

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

A well-vetted encryption algorithm considered to be strong by experts in the field and well-tested implementations can be used. It is best if software is designed so that one cryptographic algorithm can be replaced with another, improving upgrade capability to stronger algorithms.

SCADA vendors can periodically examine their systems to ensure that the current encryption methods used have not been broken because many old algorithms and implementations have become obsolete or have been discovered to be flawed.

Protecting SCADA applications, like the operator's user interface, from unauthorized access is important because they possess the functionalities and permissions to affect the physical process. The operator interface, or Human Machine Interface (HMI), provides graphical monitor and control of the physical system. Table 16 lists common vulnerabilities that allow HMI authentication to be bypassed.

Table 16. Common vulnerabilities associated with bypass of HMI authentication

Common Vulnerability	Potential Impact
Login information remembered	Authentication without credentials
Kerberos authentication always succeeds	
Client-side user and password validation	
No limit on authentication attempts	Password guessing or cracking

This section first presents the recommendation that authentication take place at both the server and the client. It then discusses several ways to manage authentication credentials, including changing default passwords, the use of strong passwords, implementation of an effective password policy, and rigorous protection of authentication credentials.

4.5.1 Perform authentication at both the server and the client

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 hacker can extract that information or modify the client to not require authentication. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values can be submitted to the server.

SCADA developers can implement robust authentication by the server or component that is granting access.

4.5.2 Manage authentication credentials

Passwords can be 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. SCADA systems have many levels of passwords that could be 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 can ensure that both the local accounts and the domain accounts have strong passwords.

The reality of working on an SCADA 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 and the corresponding emphasis on availability. 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 SCADA DMZ and/or the SCADA LAN and may also prevent authorized users from performing actions that cannot easily be attributed to them. Likewise, effective passwords that meet minimum security requirements and are frequently changed are key components of secure authentication to these systems. Typical continual manning of operating consoles can provide additional physical security that reduces the need for distinct operator user IDs and passwords.

Application-level passwords can be managed as well. This includes Web applications, SCADA applications, etc.

Passwords that protect backend services, such as SQL services, can be overlooked 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 may 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.

4.5.2.1 Change default passwords

NSTB assessments have identified many cases of third-party products delivered with the SCADA system that had no passwords or had default passwords. These accounts can remain un-configured, sometimes because SCADA owners are unaware that they exist.

Some assessments discovered applications that had been configured without passwords, so that access to these applications guarantees the ability to authenticate and interact with them. Default database accounts were often found without passwords. Another common finding during NSTB assessments was that even though secure authentication applications were used, installations and configurations were not correct. This may be due to oversight during test system configuration rather than a problem with the default settings on a newly deployed system.

Hosts can be exposed to attack by anyone able to connect and authenticate using the default accounts and passwords. Default passwords can give an attacker easy access to the equipment that controls the process. Exploiting a system with default accounts is possible with access to the documentation, or access to a sample system that allows an attacker can discover the accounts for themselves. In many cases default passwords are globally available on the Internet.

Unless disallowed by SCADA software requirements, SCADA owners can change default passwords from the manufacturers of SCADA and networking equipment to a robust, unpublished password. In the case that the software uses hardcoded passwords, SCADA owners may be able to 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.

Configuration procedures can be put in place to ensure secure and consistent default configurations.

Default accounts can be removed, with each installation using different and strong passwords. Implementing different accounts and passwords on different systems prevents an attacker from translating knowledge learned on one system to another system.

Documentation about the default accounts can be distributed to all users so they know that the accounts exist and can take the initiative to remove or change their passwords.

Strong passwords can be required and deployed on networking, client, and server equipment and that passwords are implemented on SCADA components to prevent unauthorized access.

Administrators can change the default manufacturer passwords on SCADA and networking equipment. Default passwords may give an attacker easy access to the equipment that controls the process. Owners/Operators should change default passwords to robust, unpublished passwords. In the case that the software uses hardcoded passwords, SCADA Owners/Operators can work with the vendor to address this vulnerability. They can then implement a password policy that enforces strong passwords which impedes password cracking and guessing.

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 can not only help ensure that accounts have passwords, but also help detect compromised systems.

In some SCADA 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 staffing of operating consoles can provide 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 can ensure all users have separate accounts for all other account types in the SCADA to help increase security and accountability. These actions can prevent an attacker from using a user ID and password obtained from the business LAN to gain access to the SCADA DMZ and/or the SCADA LAN and also prevent authorized users from performing actions that cannot easily be attributed to them.

4.5.2.2 Use strong passwords

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

A password strength policy can contain the following attributes: (1) minimum and maximum length; (2) require mixed character sets (alpha, numeric, special, mixed case); (3) do not contain user name; (4) expiration; and (5) no password reuse. Authentication mechanisms require sufficiently complex passwords and require that they be periodically changed.[5] Table 17 lists common vulnerabilities associated with lack of SCADA application authentication.

Table 17. Common vulnerabilities associated with lack of SCADA application authentication

Common Vulnerability	Potential Impact
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

4.5.2.3 Implement an effective password policy

Passwords exist at multiple locations, many of which don't have automated policies that can be applied. If a product does not use strong passwords, it is 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."[5]

The length, strength, and complexity of passwords are balanced with security and operational ease of access within the capabilities of the software and underlying operating system. If a password is too complicated and difficult to remember, or it changes too often, users may undermine their security to remember them. Passwords have been found in control rooms on small pieces of paper on the bottom of the keyboard, in a drawer, etc. Complex passwords do protect against some of the advanced password cracking attacks, but they can create a physical and social engineering vulnerability that could be exploited by an attacker. Therefore, passwords can be created from passphrases or other memorable means rather than auto-generated.

Password policies can be implemented to define when passwords are needed, how strong they must be and how they should be maintained. Without a password policy, systems might not have appropriate password controls, making unauthorized access 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 SCADA application was designed to support more stringent passwords. Table 18 shows general weak password findings.

Table 18. Common vulnerabilities associated with weak password requirements.

Common Vulnerability	Potential Impact
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	

4.5.2.4 Protect user authentication credentials

User authentication credentials can be vigorously protected and made inaccessible to an attacker. Whenever credentials are passed in clear text, they are susceptible to being captured. If stored password hashes are not properly protected, they may be accessed by an attacker and cracked.

Password files can be secured by making hashed passwords more difficult to acquire (e.g., restrict access by using a shadow password file or equivalent on UNIX systems). Services can be replaced or modified so that all user credentials are passed through an encrypted channel.

LAN Manager (LM) password hashes can be cracked within seconds using freely available tools. 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 can be disabled on all Windows hosts and domain controllers.26 Client security policies can be configured so that only the Windows NT (NTLM) response is given.

If LM authentication is required, the configuration settings can be updated so that only the new 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 19 shows examples of weak protection of user credentials.

Table 19. Common vulnerabilities associated with weak protection of user authentication credentials

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

4.5.3 Recommendations and resources

To find and remediate authentication vulnerabilities, in outward facing services, default accounts, and support services, it is recommended that:

- All stakeholders follow well-vetted strong authentication and cryptographic practices
  - In applications that process network traffic or accept network connections
  - When available, use proven authentication services
  - When creating authentication and encryption mechanisms, involve experienced personnel
  - Before deploying solutions, rigorously test authentication and encryption systems for correctness
  - After solution implementation, strenuously test authentication and encryption mechanisms for correctness on the server and client
- Vendors periodically examine their systems for broken encryption
- Vendors distribute documentation about any default accounts to all owners
- Vendors implement support for strong passwords and protect authentication credentials in the software
- Owners develop, implement, and enforce password policies as part of an overall SCADA security program, including:
  - Strong password mandates
  - On networking, client, and server equipment
  - On both the local and domain accounts of Windows computers
  - For all cyber assets inside the electronic perimeter with a reasonable lifespan limit
  - Take into account the capabilities of the SCADA system to handle the most complex passwords as possible
  - Are created from passphrases or other memorable means rather than auto-generated
  - Password mandates on SCADA components
  - Discouragement of the use of common passwords, especially common administrative passwords

- Owners develop and implement authentication procedures to ensure secure and consistent default configurations
  - System integrators and administrators configure the systems to require and protect strong passwords
  - Remove default accounts, use different passwords for each installation, and make each password strong
  - Change the default manufacturer passwords for SCADA and networking equipment
  - Follow and test instructions for secure installation and proper configuration for each application
  - Implement authentication mechanisms that require sufficiently complex passwords and require that they be periodically changed
  - Vigorously protect and make inaccessible user authentication credentials
  - SCADA administrators ensure that all users on any system are documented and have secure passwords
  - Regularly poll all usernames to help ensure that all accounts have passwords, and also help detect compromised systems
  - Users create and protect their own authentication credentials
  - If user-level authentication is not an option for operators, integrators or administrators ensure all users have separate accounts for all other account types in the SCADA

The SANS Institute's sample password policies provide guidance on creating, protecting, and changing passwords.[24,25] Tips for creating strong passwords are widely available. A few examples are listed below:

- CWE-521: Weak Password Requirements, http://cwe.mitre.org/data/definitions/521.html
- *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.[24,25]

4.6 Authorization vulnerabilities

Access controls restrict access to entities in a network, host, or software system. Access control mechanisms check whether an entity is authorized to have access to a given object or perform a given action. This act of validating access rights is called authorization.

Access controls can be incorporated into SCADA components to help prevent and contain compromise. The SCADA 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.

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

SCADA developers and owners can reduce the attack surface by restricting access to functionality, hosts and networks. Restrict SCADA information and functionality to those who require access to them. Enforce authorization for applications that perform SCADA functions and the computers that host them. Implement mechanisms to control access to SCADA hosts and functionality. All applications, hosts and networks can be locked down as much as possible to limit the consequences of compromise. Once an attacker has gained access to a host, compartmentalization and access controls can be used to contain them.

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 which 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 can 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. Protect keys used for encryption and decryption.

This section discusses the need to implement least-privilege access control and the associated recommendation to limit functionality only to that required, which enables the ability to implement least-privilege access control. Finally, this section discusses authorization vulnerabilities that may arise through configuration of SCADA components.

4.6.1 Restrict privileges, permissions and access to the least needed

Privileges, permissions and access can be restricted to the least needed to perform the work at hand. This includes access privileges for user accounts, file permissions, Web server and database, as well as process execution privileges for services.

SCADA vendors can implement or enhance access controls and authentication in their products to limit within them the severity of vulnerabilities. Running SCADA services and applications with full privileges can increase the potential consequences of compromise. Applications that require high privileges usually require that the user have elevated privileges as well. Limiting the access privileges of users and programs to the required privileges and functionality can limit the ability of an attacker to successfully exploit associated vulnerabilities by creating less-accessible attack vectors and increasing the attack complexity. The potential impacts to the system can be limited to the privileges and available functionality of the exploited program.

Defensive measures can minimize vulnerability exposure and the opportunity for malicious actions. Restricting access and permissions of processes and users can minimize 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, such as compartmentalization to enable privilege separation, can be designed into all critical infrastructure SCADA systems.

In the case that users operate a computer system (consoles, servers, etc.) with more permissions than required, an attacker may be able to redirect execution; exploit code will run with those same privileges giving the attacker full access to that device.

Carefully evaluate user accounts to determine the lowest set of permissions necessary. File access can then be restricted to those who require access. If network access to a file is necessary, access can be restricted as much as possible and require strong authentication.

Many SCADA user accounts are given administrator or root privileges which gives the authenticated user full access over the host. User accounts used for interactive logon can be carefully evaluated for the proper set of permissions. The OS access control capabilities can be configured with Access Control Lists (ACLs) using a "default deny" policy. SCADA vendors can specify the minimum set of access controls necessary for users of their system to perform the required operations. Otherwise, SCADA owners may not make changes due to concerns that vendor support may be unavailable.

Likewise, file shares can be restricted to only those users who require access and to the access level they require. Files shared by SCADA vendors and owners can be restricted to only the computers and accounts that require them. To implement this:

- Restrict the read and write permissions of 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.

For Web server access control for Web applications, SCADA developers can make sure that the access control mechanism is enforced correctly at the server side on every page. Another way to reduce exposure is to prevent users from accessing unauthorized functionality or information through a simple request for direct access to that page. One way to do this is for SCADA developers 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.

For database access control, SCADA vendors can ensure that their database functionalities follow the principle of least privilege. Database access controls may need to be more specific than the generic roles-based rules. Access control checks can be based on the SCADA's functionality and business logic. For example, database access should be based on the record being accessed, not just by database user.

Process execution with unnecessary privileges is a high-level root cause of many vulnerabilities. By default, some SCADA installations start services as the root user and root group. Many services may not need to be started with this privilege level, and doing so may expose system resources to preventable risks. A common problem with applications and services is that they are run with system or root-level privileges. Software products may run with these super user permissions by default even though their functions do not require them. Therefore, permission levels of applications and services can be lowered to that necessary for their required functions. Applications and services that execute with higher than necessary privileges may 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.

Exploitation of a service could allow an attacker a foothold on the SCADA network with the exploited service's permissions. Services are restricted to the user rights granted through the user account associated with the service. 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 could allow full access to the exploited host. By restricting necessary privileges during SCADA design and implementation, SCADA vendors can significantly reduce the window of exposure and criticality of impact in the event that a vulnerability is found in that service.

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

4.6.2 Remove unnecessary functionality

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

Incorporation of the principle of least privilege may require a redesign of some SCADA components. Removing unnecessary functionality on SCADA hosts includes removing all services and applications that are not necessary for each individual host's role. This requires documentation on required services, communication partners, and direction of communication.

4.6.3 Secure SCADA component configuration

NSTB assessments have encountered quality control issues related to configuration errors. Host configurations are inconsistently deployed by SCADA 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 may not be sufficiently detailed or followed.

SCADA vendors can create a methodical and documented procedure or automated process for configuring SCADA components. Procedures can be customized for specific SCADA components and functionality. SCADA integrators and administrators can request security documentation, procedures, and/or tools and use them to secure their systems. SCADA owners can specifically request and enforce this support either during system procurement or when negotiating security contracts.

SCADA customers can conduct security audits of SCADA products and determine appropriate mitigations to meet specified security levels as part of the procurement process. This allows the SCADA customers to identify security risks of the products and determine whether they are acceptable and/or able to be mitigated. SCADA owners/operators can also conduct external security audits on their existing systems to identify risks that need to be mitigated.

Complete documentation and/or automated setup of security features can be provided to allow for straightforward and more consistent implementation of SCADA components and security features. Security features that are difficult to configure and implement are typically not used or are used incorrectly in the field installations of SCADA. Security features that are inconsistently implemented or provide inconsistent results may be a risk to reliability and availability of the SCADA in an operational environment.

4.6.4 Recommendations and resources

To find and remediate compromise in authorization vulnerabilities, it is recommended that:

- All stakeholders share files only to the computers and accounts that require them
- All stakeholders implement protection schemes that deny access by default and can identify conditions under which access is permitted
- Vendors provide complete documentation of security features
    - Specify the minimum set of access controls necessary for users of their system to perform the required operations
    - Specify necessary applications, services, user accounts, as well as the necessary privileges and communications for each
- Vendors provide where possible automated setup of security features, including processes for configuring SCADA components
- Owners work with Vendors to restrict user accounts, applications, and services to the lowest set of privileges, permissions, and access necessary for system functionality
- Owners individually restrict the privileges granted to user accounts, applications and services

Improper Access Control (Authorization) is fifth on the *2010 CWE/SANS Top 25 Most Dangerous Programming Errors* list.[10] The third recommendation in the SANS/CWE software "Monster Mitigation" list addresses unnecessary privileges.[11]

4.7 Network access control vulnerabilities

Attackers can take advantage of the tendency for network devices to become less securely configured over time as system users demand exceptions for specific and temporary business needs, the exceptions are deployed and are often not removed when the business need is no longer applicable. In some cases, the security risk of the exception is not properly analyzed, nor is this risk measured against the associated business need.

Attackers can 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 can gain access to sensitive data, alter important information, or use one compromised machine to pose as another trusted system on the network.

Complete information on how their system operates can help SCADA customers develop effective network isolation architectures and configurations to mitigate some of the identified vulnerabilities and others that may arise. To this end, SCADA vendors can identify and delineate all required ports and services necessary to support their products. This will better equip the end user with the tools needed to achieve effective network isolation for their implementation of the SCADA product.

This section discusses the need to implement secure access to network devices, and to configure a secure network architecture through segmentation, strong firewall rules and secure connections across security zones. Finally, it explores the use of intrusion detection techniques in the SCADA system.

4.7.1 Secure network device access

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

Given the static nature of SCADA environments, port security may be used to ensure MAC addresses do not change and new devices are not introduced to the network. Actions such as limiting known MAC addresses to specific interfaces and disabling unused interfaces can be implemented to assist in network security.

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. A lack of port security on network equipment was a common finding in NSTB assessments. A malicious user who has physical access to a port on a network switch behind the firewall can circumvent its incoming filtering protection. Table 20 shows network device configuration common vulnerabilities.

Table 20. Common vulnerabilities associated with network device configuration

Common Vulnerability	Risk
Network device configured for management over insecure 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

4.7.2 Secure network architecture

Firewall rules implement the network design. They determine which network packets are allowed in and out of a network. Without outbound restrictions the system can be vulnerable to indirect attack on connections that originated from the SCADA. 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 restrict traffic flow as much as possible. They enforce network access permissions and allowed message types and content.

4.7.2.1 Use network segmentation

Network segmentation creates security zones that separate systems with different security and access requirements to provide access control. Components on the same network segment are effectively given the same level of trust. Each security zone includes components that need to communicate and can be allowed the same trust levels.

NSTB assessments have revealed that the security level of production SCADAs can be largely dependent on the effectiveness of the SCADA network design to prevent unauthorized access. Networks can be separated into layers and zones based on security levels and functionality with specific access rules to restrict all communication to only that necessary for system functionality. Although not all cyber attacks will have a catastrophic impact, these attacks can be prevented, detected, or stopped before they have the opportunity to affect critical SCADA functions. If the network has been designed and implemented correctly, an attacker is limited to finding vulnerabilities in the network equipment, authorized users/systems, protocols, or associated applications/servers allowed into each network segment only, and must do so without being detected.

Networks with 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 the SCADA for attackers to exploit and take full control of the system.

SCADA components with connections located on the business network are subject to the same exposure as any other device with a 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).

Even with good network design that uses security zones, SCADA vulnerabilities can be exposed to less-trusted networks that provide remote monitoring, data sharing, historical, and other remote access functions. Because access to SCADA software vulnerabilities cannot be prevented entirely, vulnerability remediation remains necessary.

To provide defense-in-depth, firewalls can be used to separate different layers of the SCADA network (i.e., the HMI level LAN from the SCADA DMZ from the corporate network). These layers can be further segregated into security zones to protect systems from attack through compromised systems on that layer. Consider creation of multiple DMZs, or security zones, 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 SCADA LAN is considered part of the perimeter. Often these perimeters may not be well documented and some connections may be neglected. All entry points into the SCADA LAN can be determined and strictly managed by a security policy. Ensuring all connections are routed to the SCADA LAN through the firewall will enforce proper configuration management policy and monitoring. Maintaining an accurate network diagram of the SCADA LAN and its connections to other protected subnets, DMZs, the corporate network, and the outside is essential for network administrators. Table 21shows common vulnerabilities found within SCADA network designs.

Table 21. Common vulnerabilities associated with insecure network design

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

Training

In some cases, the individuals in charge of securing the SCADA network may not have adequate security training. Training can provide 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 SCADA network and its perimeter defenses with the most current security techniques may be essential.

4.7.2.2 Use strong 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 port number, IP address, 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.

Well-configured firewalls are critical to SCADA security. Communications can be restricted to those necessary for system functionality. SCADA traffic can be monitored, and rules 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.

The top priority of most SCADA installations is availability. The risk to availability of any security feature must be weighed against the expected added security benefit (lowered risk). SCADA network administrators may not want to risk the chance of impacting SCADA functionality by redesigning the network or updating rules as components are added or removed. 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 and 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.

Implement firewall rules on production SCADA carefully, slowly working toward a rule set that excludes all traffic, with exceptions for including needed communication. Once the necessary outbound traffic has been determined, a more secure configuration can then be created that blocks all traffic with exceptions for necessary communication.

SCADA owners can determine necessary communication by monitoring network traffic, implementing 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. Monitor firewall logs for indications that legitimate system traffic is being blocked.

Port numbers

The attacker may remotely connect to services listening on ports allowed through a firewall. Open ports and services that are not necessary may provide a potential foothold or path for an attacker. 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 can prevent connections to unauthorized services through the firewall.

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 a 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 SCADA 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. With access to the host, an attacker could download his own server and configure it to listen on one of these open ports.

Inbound and outbound directions

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

A common oversight in firewall configuration is not restricting outbound traffic. Firewall rules considering both directions through the firewall are recommended. An exploit that cannot connect back to the attacker is limited to blind attacks. To be successful, an attacker needs to obtain information from and send files and commands to the SCADA network. To remotely control exploit code running on a SCADA computer, a return connection must be established from the SCADA 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 available to add expensive logic for the attacker to use 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.

In contrast, insufficient outbound restrictions can make the system vulnerable to indirect attack on connections that originated from the SCADA. The lack of outgoing access restrictions can allow 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.

Another common NSTB assessment finding was that firewall rules allowed access to unused IP addresses traceable to legacy configuration of the firewall. This can create an attack path that allows an attacker to use this IP address to gain access through the firewall.

The remaining specific NSTB assessment finding associated with this vulnerability involved access to specific ports granted for an entire address space or unrestricted by an IP address at all. Firewall rules that restrict access to specific ports, but not IP addresses, provide little protection. Assessment findings that fall under this vulnerability are firewall rules that are based on address groups that include a wider range than may be necessary.

Network defenses that utilize specific firewall and Intrusion Detection System (IDS) rules can help control and monitor access, even if an attacker has gained access inside the SCADA perimeter. The better legitimate traffic can be defined, the more likely unauthorized network traffic can be blocked or detected. 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 SCADA).

Specific firewall and IDS rules can block or detect abnormal activity. An IDS does not add 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.

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 monitoring IDS logs during this period and immediate investigation of unexpected communication is recommended.

Not all assessment findings were related to system functionality. Many findings from production SCADA assessments related to firewall rules that allowed IP addresses to initiate connections between networks even though they did not require this access for SCADA functionality. Firewall rules that apply to functional groups can use defined finite groups that are restricted to required IP addresses. Firewall rules that are no longer needed can be removed as part of a change management procedure or periodic system review or audit. Access control lists can be used to limit management access of network equipment to only those who need it. Table 22 lists specific NSTB assessment findings associated with overly permissive firewall rules.

Table 22. Common vulnerabilities associated with unnecessary exposure from firewall rules

Common Vulnerability	Potential 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	

Business applications sometimes require connections from the corporate network into the SCADA. These connections can create potential attack paths from the Internet onto the corporate network and then into the SCADA networks. For instance, some business applications 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 SCADA functions can extend the exposure of associated vulnerabilities to the corporate network (or to wherever the connection is initiated).

SCADA owners can reduce the risk of business application connections by minimizing exposure to the business network and closely monitoring the necessary communication paths. Different DMZs can 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 9 shows this separation into multiple DMZs.

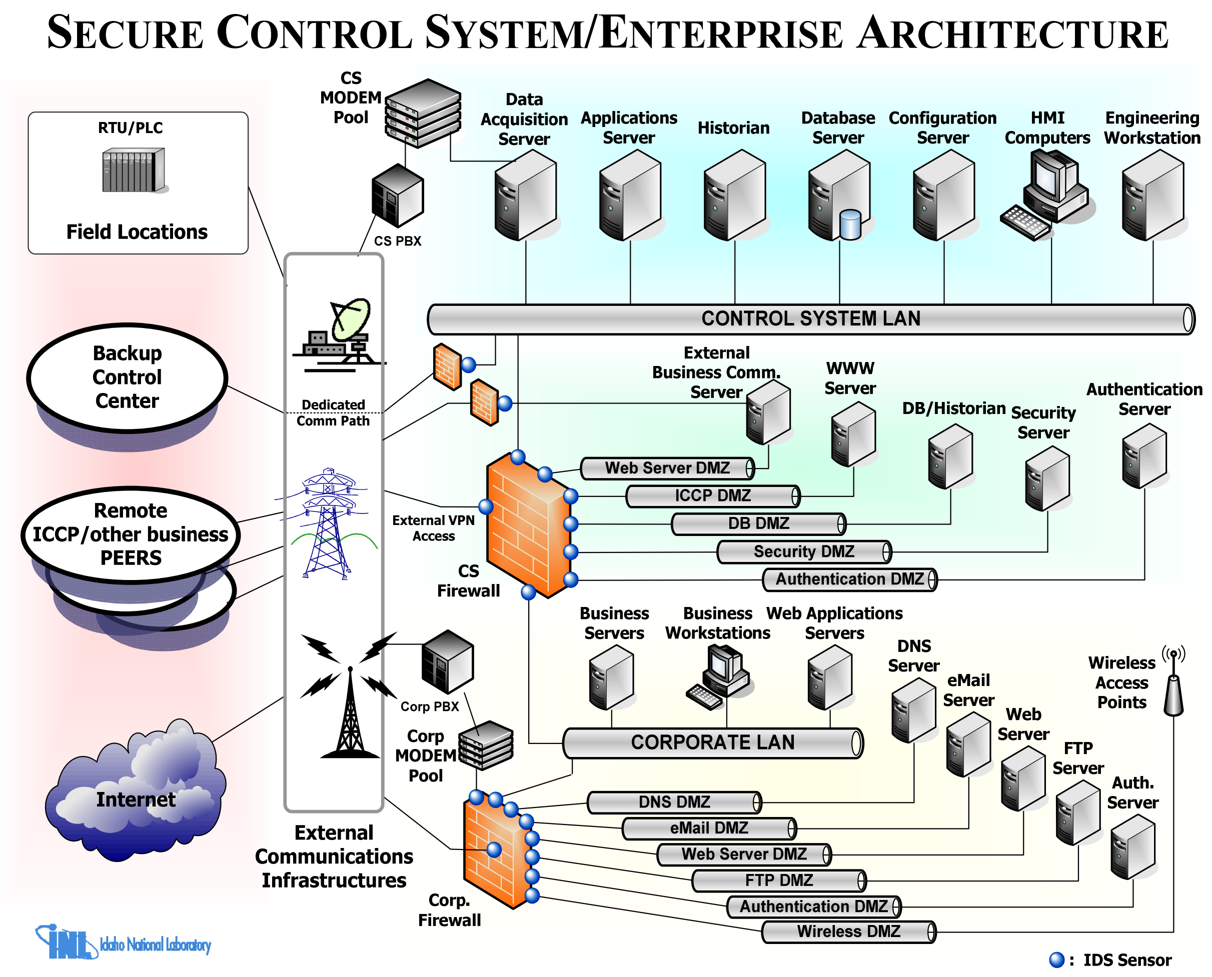


Figure 9. Recommended defense-in-depth SCADA architecture

Perimeter protection techniques minimize the necessary connections, and corresponding exposure, to the business network, but vulnerabilities in the protocols and services used for business functions can be exploited to gain access inside the SCADA perimeter. The design of SCADA protocols can force sub-optimal network designs and implementations. For example, the use of protocols that require access to wide port ranges limits the ability to prevent unauthorized system access with firewall rules.

Data sharing protocols such as ICCP and OPC are used to send and receive data from remote sites and peer utilities. These connections are un-trusted if the remote site or intermediate pathway networks are unknown. Vulnerabilities in services that must be allowed to accept connections from less-trusted networks are exposed to possible exploitation from these networks.

Table 23 lists typical types of SCADA services that must be exposed to possible attack from external networks.

Table 23. Common vulnerabilities associated with external communications.

Common Vulnerability	Potential Impact
Business applications require holes through the firewall from the corporate network into the SCADA 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 SCADA owner has no control over	Increases exposure to unknown external networks
SCADA vendor and administrator VPN connections	

One way to prevent direct access to the SCADA LAN from the corporate clients is by using a replicated data server in a DMZ as shown in Figure 10. In this architecture, a Web server is located in a DMZ between the SCADA and corporate networks. Replication of data from the SCADA is accomplished by the data application running on the SCADA 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 SCADA data.

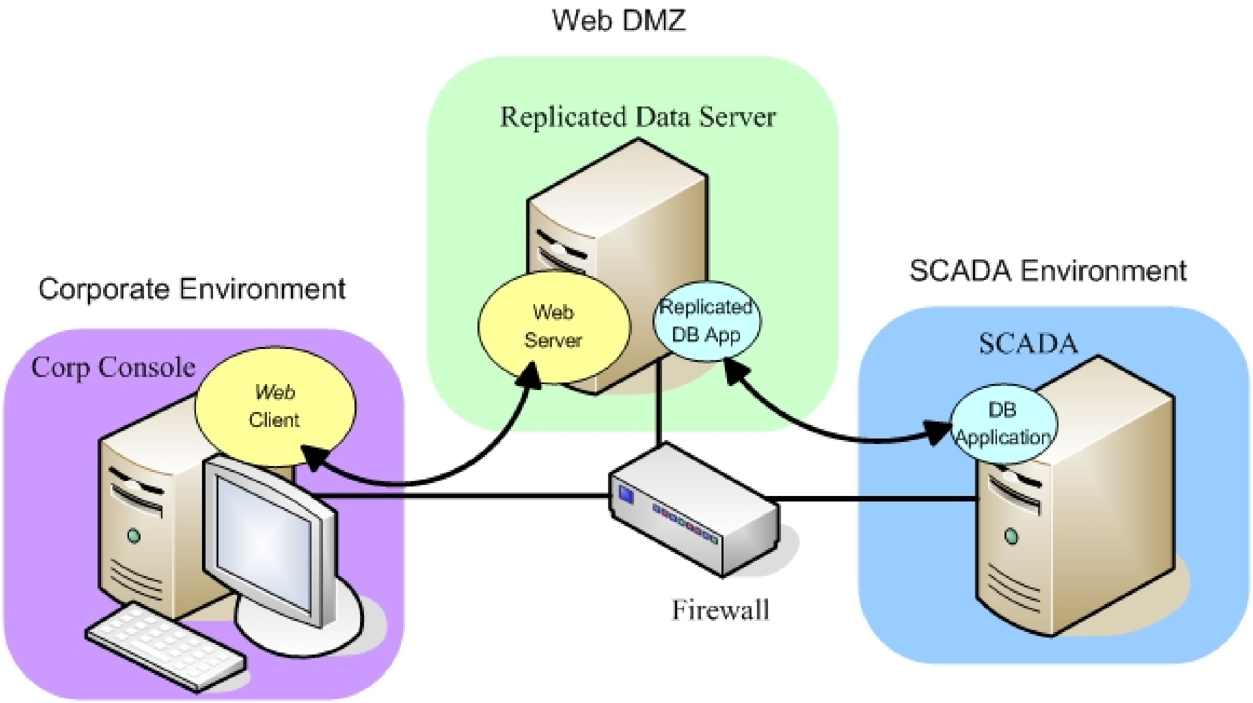


Figure 10. Replicated data server

4.7.3 Monitor network to detect intrusion

An intrusion detection system (IDS) looks for actions that attempt to compromise the confidentiality, integrity, or availability of a resource. 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.

IDSs don't determine the nature of the probable intrusion or take action to prevent the intrusion. Intrusion prevention systems are not generally recommended along paths of critical functionality.

Intrusion detection can be manual or automatic. Manual intrusion detection is done by examining log files or other evidence for signs of intrusions, such as abnormal network traffic. Automated approaches monitor system logs, network traffic flow, and packets, then send an alert when a "probable intrusion" is identified. As an example, SCADA networks have predictable normal activity so IDS rules can look for abnormal behavior such as a protocol that is not normally used between two computers.

Although intrusion detection was not a focus for NSTB assessments, in many cases it was noted whether the system gave any indication of abnormal conditions, such as alerts on the operator screen. NSTB laboratory assessments found that SCADA systems lacked adequate indicators of abnormal conditions. A common observation during onsite assessments was that the installation, monitoring, validation and updating of IDS tools deployed on SCADA networks also offered opportunities for improvement.

Event logging (applications, events, login activities, security attributes, etc.) can be turned on and monitored to identify security issues. Logs and other security sensors monitored on a real-time basis allow security incidents to be rapidly detected and countered. Train assigned individuals and give them the responsibility of monitoring system data logs and keeping the various tool configurations current.

4.7.3.1 Customize IDS Rules for the SCADA and Closely Monitor Logs

The configuration and deployment of IDSs for SCADA is not as straightforward as it is for typical IT computer networks. Traditionally, specific payloads and port numbers for the unique communications protocols used in SCADA systems - such as Modbus or DNP3 - have not been included as signatures in contemporary IDSs. Although IDS signatures are available to detect a wide range of attacks, few SCADA-specific signatures are available. Consequently, modern IDSs deployed on SCADA networks may be blind to the types of attacks that a SCADA system could experience.

NSTB assessments showed the advantage of developing security signatures and rules in a cooperative relationship with the SCADA vendor. Many security vendors, including those specializing in SCADA security, have created signatures for the IDS that are deployed in control architectures.

An IDS deployed in a SCADA network benefits from the ability to add unique SCADA-specific signatures. Likewise, the need to remove some default signatures and response capability is commonplace, as they may have no relevance to an SCADA network. Rules sets and signatures unique to the SCADA domain are imperative for deploying effective IDSs on SCADA networks. Analysis that ensures the inherent capability of the IDS is leveraged, with some of the capability refined and augmented, is useful.

IDS logs can also be used to identify normal communication patterns between each of the SCADA components. Investigate all unexpected traffic and either include it on the required communication list or blocked it by firewalls. A one-to-one mapping of firewall rules and IDS signatures allows the IDS sensor to alert if a firewall rule is not successfully applied, and allows administrators to take corrective action on the firewall.

The external IDS sensor provides notification of malicious attempts on the firewall and monitors egress rules from the SCADA out to the DMZ or corporate networks. The internal IDS sensor and the DMZ IDS sensor closely monitor the exceptions in the firewall for malicious activity.

4.7.4 Recommendations and resources

To find and remediate access control vulnerabilities in network devices, architectures, and inbound and outbound communications, it is recommended that:

- Vendors redesign their systems for security, giving protocols and services that connect to less-trusted networks top priority in SCADA software vulnerability remediation activities
- Vendors provide documentation on how the SCADA system components use the network so that effective firewall and IDS rules can be created
    - If SCADA network requirements and protocol specifications are not available, owners can monitor network traffic to identify normal system behavior. Validate network communications to the extent possible, to avoid base-lining malicious activity. Vendors can document their system requirements using this method as well.
- Owners make network access rules as restrictive as possible
    - Use access control lists to limit management access of network equipment to only those who need it
    - Restrict host and user network permissions and access rights as much as possible
    - Restrict access to the required port numbers and IP addresses
    - Use directional rules to prevent activities such as database connections initiated from the corporate network
    - Set as specific as possible, any exceptions created in the firewall rule set, including host, protocol, and port information
- Owners configure a secure network architecture
    - Set up network devices to only allow access using secure protocols
    - Consider the direction of network packets in rule configuration
    - Limit connectivity of ports to hardware interfaces.
    - Segment networks into security zones
    - Use firewalls to create DMZs
    - Firewalls restrict communications to only what is necessary for system functionality
    - Remove as part of a change management procedure or periodic system review or audit any firewall rules that are no longer needed
    - Route all connections to the SCADA LAN through the firewall and eliminate hardwired connections that circumvent the firewall

    - Place Web servers, Historian Databases, and other servers required for business functions on DMZs that have been segmented into security zones

- Owners use change management procedures that include
    - Updates to network diagrams and other documentation whenever changes are made to the SCADA networks
    - Access for network administrators to an accurate network diagram of their SCADA LAN and its connections to the other protected subnets, DMZs, corporate network, and external networks
- Owners enable logging, monitor system traffic, and implement rules that allow only necessary access
- Owners initiate connections from trusted networks rather than less-trusted networks and filter outbound connections

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

- *21 Steps to Improve Cyber Security of SCADA Networks*[27]
- NIST SP 800-82, *Guide to Industrial Control Systems (ICS) Security*,[7] pages 5-1 to 5-19
- *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies*[28]
- *Control Systems Cyber Security: Defense in Depth Strategies*[29]
- *Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks*[30]

General network access control recommendations can be found in the *Twenty Critical Security Controls for Effective Cyber Defense*[15]

- 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 SCADA Vulnerabilities Summary

NSTB assessments look for access paths to SCADA resources and functionality.

Weak or missing security features in SCADA software can leave the system components vulnerable to manipulation by any threats to which they are exposed. Protection mechanisms for each component of the SCADA can help reduce risk.

Vendors have different standard processes for building, testing, and installing a SCADA system. 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 SCADA 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 SCADA installation is dependent on those responsible for installing and configuring the operating systems, SCADA applications, and third-party applications.

Each SCADA installation provides a unique combination of components and functionality offered by a SCADA product vendor to meet specific customer specifications. Large scale

SCADA systems are generally major purchases, requiring much time and money. Few systems from each SCADA product line are delivered before features are added and a new version is released. The large investment of financial and personnel resources needed for SCADA upgrades can contribute to insufficient standard procedures for securely configuring each SCADA product.

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

Vulnerabilities may exist because of the way that a SCADA has been implemented, or in how it is used. Vulnerable third-party products can be replaced, but in some cases SCADA vendor assistance or code changes may be necessary. SCADA vendors can remove unnecessary functionality from SCADA applications and default application and OS configurations. SCADA owners can work with their vendors to secure system configurations. Production SCADA installations can implement secure practices where possible and apply computer and network security techniques to remove or mitigate remaining vulnerabilities.

Consider risk and consequence in implementing security mitigations and prioritizing efforts for enhancing security. Weigh 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, SCADA 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.

Table 24 summarizes the security vulnerabilities that cannot be completely remediated through perimeter defenses.

Table 24. Common vulnerabilities not mitigated by perimeter defenses

Common Vulnerability	Limitations of Perimeter Defenses
Unsecure coding practices	Necessary protocols cannot be blocked Encryption does not fix vulnerabilities
Unpatched OS, third-party products, and third-party libraries	The SCADA may not be compatible with the newer, or patched, versions
Least privileges violations	Privileges are required by the SCADA 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
Network layout effectiveness limited by SCADA protocol requirements	Protocol designs limit the effectiveness of network security mechanisms (i.e., large port ranges)
Insecure protocols	Necessary protocols cannot be blocked

5. ADDITIONAL SCADA SECURITY RECOMMENDATIONS

The classifications of vulnerabilities identified in this report can be used during self-assessment activities to assist in identifying potential problem areas that can aid in identification and mitigation of vulnerabilities in SCADA networks, components, services, and code. SCADA system installations have varying performance and reliability requirements and use operating systems and applications that may be considered unconventional compared to typical IT systems. In addition, for SCADA systems, the goals of safety and efficiency may conflict with security in the design and operation of SCADA. Security solutions (i.e., requiring password authentication and authorization) must not interfere with emergency actions for SCADA or compromise critical operational functionality. Therefore, all security functions integrated into the SCADA must be tested in a safe mode (such as offline on a comparable SCADA system configuration) to verify that they do not compromise normal operational functionality and safety. It is recommended that SCADA vendors and Owners/Operators use qualified security and SCADA experts to verify that the proposed mitigation will be effective and ensure that the actions will not impair the system's reliability and operational requirements.

There are multiple ways the SCADA system and the physical process it controls can be threatened. Each component of the SCADA system can have its own protection mechanisms to protect both the information and physical systems. The building-up, layering and overlapping of security measures is called defense in depth. With this defense in depth strategy, should one defensive measure fail there are other defensive measures in place that continue to provide protection.

Remediation and mitigation of cyber risk starts with a defense-in-depth strategy to securing SCADA systems. It is essential that energy asset Owners/Operators and SCADA vendors work together to implement the defensive measures necessary to support an acceptable risk posture without compromising system functionality. For the vendor this includes identification and remediation of existing vulnerabilities in current products, development of secure new products, and support for patching and secure configurations of system components. Energy asset Owners/Operators can determine business risk associated with critical operational nodes, then install, maintain and monitor secure operating system, software, SCADA systems, and network configurations. Additional SCADA-specific security resources can be found on the NSTB, United States Computer Emergency Readiness Team (US-CERT), and other Web sites:

- http://www.controlsystemsroadmap.net/documents.shtml
- http://www.oe.energy.gov/controlsecurity.htm
- http://www.us-cert.gov/control_systems/csdocuments.html

In addition to the specific recommendations identified in Section 4, there are additional actions the vendors and owner/operators can take to make the control system more secure from cyber attack. The following sections address these SCADA specific issues that should also be considered.

5.1 Recommendations for SCADA vendors to improve product security

Vendors can incorporate cybersecurity into every phase of the product development life cycle and can use both manual and automated means to ensure proper bounds checking. Once products are deployed, vendors can establish a process to manage and mitigate product security defects.

NSTB assessment results and discussions with the vendor indicate that the top SCADA vendor recommendations are:

- Create and maintain a culture that emphasizes security
- Educate/train developers to use secure coding practices
- Expeditiously test security patches
- Create the necessary communication paths to quickly notify customers/users of security problems, and create the methods needed to provide patches in an effective way
- Implement and strenuously test strong authentication and encryption mechanisms
- Increase the robustness of network parsing code
- Document how the systems use the network so that effective firewall and IDS rules can be created
- Have third-party security source code audits performed and fix problems identified during the audit
- Redesign network protocols to avoid common problems and enhance security
- Enhance test suites to perform more testing for failure with emphasis on testing for potential vulnerabilities
- Create custom protocol parsers for common IDSs so they can be more effective.

5.1.1 Create a Security Culture

SCADA vendors can educate/train developers in secure coding and help create a culture that emphasizes security. Hardware, operating system, and software application vendors have experienced the cost and negative publicity that accompany public announcement of security flaws to force quicker patch response time. The security development lifecycle (SDL), created by Microsoft in 2002 as a response to heightened awareness of cybersecurity threats, is a high-visibility example of a security culture change. This process was developed to catch security flaws during the product development lifecycle, not just after the product is released. For example, Microsoft created a culture that promotes safe code development by forcing all new code to pass a series of tests before incorporation into the main product. All developers are 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.

SCADA products have gained considerable attention in recent years as cybersecurity threats have been identified and publicized. Public announcements of SCADA vulnerabilities are starting to appear and SCADA protocol dissectors are becoming available. Those companies willing to embrace a security culture change will benefit from fewer security patches for deployed systems and greater customer confidence and loyalty.

SCADA vendors may need to adapt to changing customer needs for security in the products used to control physical systems where compromise can have catastrophic consequences. It is difficult to attach security onto a product that has already matured and it is practically 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 earlier in the life cycle that security is integrated into the product, the better chance it has of competing in a market where SCADA products are required to survive cyber attack without compromising critical functionality.

SCADA vendors can 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 can be consolidated, integrated, and centralized into a security process that supports the defined strategy for creating the most secure product possible. Numerous resources are available for information and training on building a security culture and software security best practices. SCADA 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 for the end user
- Have a continual product improvement process that identifies vulnerabilities in SCADA component designs and then redesign the identified components or develop secure mitigations for the legacy components
- Require secure source coding handling to protect against malicious vulnerabilities
- Perform thorough security testing
- Provide security documentation

5.1.2 Enhance SCADA Test Suites

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

SCADA product design and code logic can be implemented to properly handle invalid or unwanted cases, even if they never occur. The connection of SCADA to other networks has created the threat of cyber attacks that can cause errors that would never occur naturally or by accident. The possibility of malicious input requires protective logic that handles every possible error condition.

Unconventional scenarios, that test various input values and abnormal conditions, should be included in SCADA test suites. This may require 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 SCADA 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 successful in increasing awareness of the unconventional attacks the SCADA 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.[31]

5.2 Secure SCADA Installation and Maintenance for Owners/Operators

A successful method for securing an SCADA is to gather industry-recommended practices and engage in a proactive, collaborative effort among management, the controls engineer and operator, and the IT organization. This team can draw upon the wealth of information available from ongoing federal government, industry groups, vendor, and standards organizational activities. SCADA owners/operators can 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 can be implemented for prioritization of the tasks necessary to enhance SCADA 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 SCADA must be reliable and perform its required mission. The recommended 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 can be weighed and the appropriate amount of security measures added for the specific situation.

Asset Owners/Operators may use enforceable procurement specifications to ensure that security development life-cycle requirements are met by the vendor. For example, asset Owners/Operators may require that the SCADA products be reviewed by an independent security assessment team and that all findings be remediated prior to purchase. Vulnerability and patch management programs and policies can be established and enforced.

An effective cybersecurity program for SCADA may apply a strategy known as defense-in-depth[28, 29], layering security mechanisms to minimize the impact of a failure in any one mechanism. Implementing security controls, such as intrusion detection software, antivirus software, and file integrity checking software, where technically feasible, may prevent, deter, detect, and mitigate the introduction, exposure, and propagation of malicious software to, within, and from the SCADA. Good defense in-depth perimeter protections can be used to help prevent access to vulnerable components and communication on SCADA 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.

Owners/Operators can increase the security of their systems by implementing the following recommendations:

- Redesign network layouts to implement a network topology for the SCADA that has multiple layers, with the most critical communications occurring in the most secure and reliable layer.
- Restrict physical access to the SCADA network and devices
- Expeditiously deploy security patches after testing all patches under field conditions on a test system before installation on the SCADA
- Work with vendors to test and apply patches for all operating systems and software on the SCADA networks
- Customize IDSs for the SCADA hosts and networks (network security administrators should be able to write custom IDS signatures tailored to their environment)
- Restrict SCADA 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
- Conduct continued product cybersecurity improvement through internal and external audits

5.2.1 Restrict Physical Access to SCADA Network and Devices

Access to SCADA system equipment should only be allowed for authorized personnel. Continual manning of operating consoles can provide physical security that reduces the need for distinct operator user IDs and passwords.

Require physical access for controller configuration and firmware updates. Requiring physical access to implement modifications and updates helps prevent possible exploitation over the network. Authentication and data integrity checks also protect against unauthorized physical access and manipulation of software and firmware files.

6. CONCLUSION

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

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

NSTB assessments found large SCADA attack surfaces created by excessive open ports allowed through firewalls and insecure and excessive services listening on them. Well-known insecure coding practices account for most of the SCADA software vulnerabilities, which result in system access vulnerability or Denial of Service (DoS). However, improper patch management provides more likely attack targets because the vulnerabilities are public and attack tools are available for them. Once SCADA network access is obtained, status data and control commands can be manipulated as they are communicated by insecure SCADA protocols.

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

The most significant vulnerabilities associated with the assessment target research evaluated by the NSTB assessment program can be mitigated by patch management, elimination of unnecessary and unsafe services, implementation of strong authentication and integrity checks to network protocols, and securing applications that accept external input, particularly network traffic. Secure configurations and network layers of defense can then be used to protect these critical assets.

SCADA vendors can establish a culture where software security best practices are adopted throughout the product development life cycles. Security practices should be consolidated, integrated, and centralized into a security process that supports the defined strategy for creating the most secure product possible. Numerous resources are available for information and training on building a security culture and software security best practices. The following is a summary of recommendations for SCADA vendors:

- Create a security culture
- Enhance SCADA test suites
- Create and test patches
- Redesign network protocols for security
- Increase robustness of network parsing code
- Create custom protocol parsers for common IDSs
- Document necessary services and communication channels

- Implement and test strong authentication and encryption mechanisms
- Improve security through external software security assessments

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

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

SCADA vendors can improve the security of the SCADAs used in critical energy infrastructure installations by identifying and remediating existing vulnerabilities in current SCADA products, developing secure new products, and supporting patching and secure configurations of SCADA components. SCADA owners can then more effectively secure their systems by installing, maintaining, and monitoring secure OS, software, SCADA, and network configurations.

SCADA owners can work with vendors to better understand known system vulnerabilities and do what they can to implement as many defensive protective measures as possible without compromising system functionality. Owners/operators are recommended to increase the security of their systems by completing the following recommendations:

- Restrict SCADA user privileges to only those required
- Change all default passwords and require strong passwords
- Test and apply patches
- Protect critical functions with network security zones and layers
- Customize IDS rules for the SCADA and closely monitor logs
- Force security through external software security assessments

The security of SCADAs used in critical energy infrastructure installations throughout the United States relies on a cooperative effort between SCADA product vendors and the owners of critical infrastructure assets. SCADA product vendors can deliver and support systems that are able to survive attack without compromising critical functionality. SCADA integrators can configure systems securely before they are put into production. SCADA owners can ensure that the physical systems they operate do not put lives, economy, or environment at risk by performing due diligence in procuring, configuring, securing, and protecting the SCADA for critical infrastructure.

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. CPNI Report, "Good Practice Guide, Process Control and SCADA Security - Guide 1. Understand the Business Risk": Centre for the Protection of National Infrastructure, PA Consulting Group, 2008, http://www.cpni.gov.uk/documents/publications/2008/2008031-gpg_scada_security_good_practice.pdf, webpage accessed March 2011.
4. 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.
5. Lee, Kathy, et al., *NSTB ICCP Security Assessment*, U.S. Department of Energy Office of Electricity Delivery and Energy Reliability, February 2010.
6. DHS, *Recommended Practice for Patch Management of Control Systems*, Department of Homeland Security, http://csrp.inl.gov/Documents/PatchManagementRecommendedPractice_Final.pdf, December 2008.
7. NIST, NIST SP 800-82, Guide to Industrial Control Systems (ICS) Security, Final Public Draft, National Institute of Standards and Technology, September 29, 2008.
8. Pratyusa Manadhata and Jeannette Wing, *An Attack Surface Metric*, IEEE Transactions on Software Engineering, 2010, http://www.cs.cmu.edu/~pratyus/tse10.pdf.
9. MITRE*, CWE (Common Weaknesses Enumeration),* Department of Homeland Security, January 11, 2009, http://cwe.mitre.org/, Web page accessed May 2010.
10. 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.
11. 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.
12. Seacord, Robert*, CERT Secure Coding Standards*, Carnegie Mellon University, June 7, 2010, https://www.securecoding.cert.org, Web page accessed May 2010.
13. 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.
14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

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

21. 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.

22. 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.

23. 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.

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

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. *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.

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

32. Clapper, R. James, *Statement for the Record on the Worldwide Threat Assessment of the U.S. Intelligence Community for the House Permanent Select Committee on Intelligence*, Office of the Director of National Intelligence, February 10, 2011, www.dni.gov/testimonies/20110210_testimony_clapper.pdf, Web page accessed July 2011.

33. MITRE, *CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'),* http://cwe.mitre.org/data/definitions/22.html, Web page accessed July 2011.

34. SANS.org, *Glossary of Security Terms*, http://www.sans.org/security-resources/glossary-of-terms/s, Web page accessed July 2011.

35. SearchSecurity.com, http://searchsecurity.techtarget.com/search/query?start=0&filter=1&q=mitm, Web page accessed July 2011.

36. Wireshark, http://www.wireshark.org/, Web page accessed July 2011.

37. Kismet, http://www.kismetwireless.net/, Web page accessed July 2011.

38. NIST, NIST SP 800-77, *Guide to IPsec VPNs*, December 2005. http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf

39. NIST, Cryptographic Toolkit, http://csrc.nist.gov/groups/ST/toolkit/index.html, Web page accessed July 2011.

40. IT Toolkit, *Man-in-the-Middle Attack*, http://it.toolbox.com/wiki/index.php/Man-in-the-Middle_Attack, Web page accessed July 2011.

Appendix A

NSTB Assessment Methodology

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

NSTB product assessments focus on the core components of new SCADA 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. Since supervisory control software usually only supports one or two operating systems, operating systems can also be thought of as integrated into the SCADA.

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 SCADA components or manipulate the system. Widely known vulnerabilities and configuration errors represent the most likely attack paths of a SCADA because the information and tools for discovering and exploiting them are publicly available.

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

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

Programming errors are identified in SCADA 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.

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

SCADA and OS user accounts, services, and applications are evaluated for unnecessary privileges to files and SCADA and OS commands. SCADA functionalities and user accounts that are not compartmentalized can make it hard to contain an attacker who has gained access to a system component.

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

The most common and significant SCADA vulnerability types are described in this report. The information is presented at a high level to facilitate reporting and understanding of the major SCADA security issues without disclosing system-specific details. Vulnerabilities are derived from NSTB SCADA 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 A-1 shows high level, generic SCADA security assessment targets.

Table A-1. Generic SCADA security assessment targets.

Assessment Target	Targeted Component	Methods
Identify Known Vulnerabilities and Listening Services	Unauthorized access to SCADA 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 SCADA 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

SCADA 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 SCADA to do harm. From a cybersecurity perspective, this means that they must create an attack path from their attack computer to the SCADA. An attack could potentially start from any point between the Internet and the physical equipment that the SCADA 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 SCADA network. The number of possible paths to the target is the system's exposure. SCADAs 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 SCADA assessments (i.e., onsite assessments) concentrate on the aspects of the SCADA 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 SCADA 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 SCADA 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 SCADA 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 SCADA security assessments identified general security problems, such as the use of insecure 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 insecure 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 SCADA vulnerabilities. SCADA vendors and asset owners can refer to the CWE for additional guidance in identifying, mitigating, and preventing weaknesses that cause vulnerabilities.[9]

The common weaknesses in this report are similar security weaknesses found on two or more unique SCADA 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 SCADA components they were found in.

Appendix B

Vulnerability Scoring

The most significant vulnerabilities identified in SCADA are those that allow unauthorized control of the physical system. Compromise of the SCADA'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 SCADA 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 B-1.

Table B-1. 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.

Table B-1. (continued).

Base Metrics	Metric Value	Metric Description
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 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 v2 Temporal scoring metrics are summarized in Table B-2.

Table B-2. 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.

Table B-2. (continued).

Temporal Metrics	Metric Value	Metric Description
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.
	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

Assessments have shown that different environments can have a 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 SCADA security requirements are used to score generic SCADA vulnerabilities.

Security requirements metrics enable SCADA 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 SCADA 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 B-3.

Table B-3. 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.

Table B-3. (continued).

Environmental Metrics	Metric Value	Metric Description
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.
	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.

Appendix C

Top 10 Most Critical SCADA Vulnerabilities

CONTENTS

Top 10 Most Critical SCADA Vulnerabilities

The most significant SCADA vulnerabilities are listed in Table C-1 using CVSS v2 metrics applied generically to the vulnerabilities identified during NSTB assessments. These scores can be adjusted for specific vulnerabilities on individual SCADA installations.

This list does not represent the state of all SCADA products and systems. It represents the most significant common vulnerabilities identified on SCADA products and systems that have been evaluated by the INL DOE-OE NSTB program from 2003 through 2011. This list is intended as awareness of common security weaknesses in SCADA that can have serious consequences if exploited. SCADA vendors and owners can assess their systems for these common SCADA vulnerabilities and remediate or mitigate them to the extent possible.[a]

Table C-1. Top 10 most critical SCADA vulnerabilities.

Vulnerability	SCADA Impact
Unpatched Published Vulnerabilities	Most Likely Access Vector
Web Human-machine Interface (HMI) Vulnerabilities	Supervisory Control Access
Use of Vulnerable Remote Display Protocols	Supervisory Control Access
Improper Access Control (Authorization)	Access to SCADA Functionality
Improper Authentication	Access to SCADA Applications
Buffer Overflows in SCADA Services	SCADA Host Access
SCADA Data and Command Message Manipulation and Injection	Supervisory Control Access
SQL Injection	Data Historian Access
Use of Standard IT Protocols with Clear-text Authentication	SCADA Credentials Gathering
Unprotected Transport of SCADA Application Credentials	SCADA Credentials Gathering

[a] CVSS metrics are intended for specific vulnerabilities. This section attempts to use the concepts of the CVSS generically to help explain the risks associated with vulnerabilities commonly found in SCADAs. In an attempt to illustrate how the severity of vulnerabilities depends on how and where the affected SCADA components are used, some of the examples may deviate from the intended use which is to focus on a particular vulnerability.

Patterned after the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors,[10] the most critical SCADA vulnerability types are summarized in Sections C1–C10 using the attributes in Table C-2, when applicable. Attributes assigned to weaknesses that appeared on the SANS/CWE Most Dangerous Programming Errors list were used for associated vulnerabilities.

Table C-2. Most critical SCADA vulnerability attributes.

Vulnerability Attribute	Attribute Description
Possible Consequences	When this weakness occurs in software to form a vulnerability, what are the typical consequences of exploiting it?[10]
SCADA Impact	SCADA specific consequences.
Vulnerable Components	SCADA components that may have this vulnerability.
Ease of Detection	How easy it is for an attacker to find this weakness.[10]
Attacker Awareness	The likelihood that an attacker is going to be aware of this particular weakness, methods for detection, and methods for exploitation.[10]
Internet Attack Frequency	How often the weakness occurs in vulnerabilities that are exploited by an attacker.[10]
Remediation Cost	The amount of effort required to fix the weakness.[10]
Weakness Prevalence	How often the issue is encountered in software.[10]
SCADA Prevalence	How often the weakness is encountered during assessments.

The10 vulnerability types were selected because they are commonly seen on SCADA assessments and can pose a significant risk to a SCADA. CVSS scores are designed to support vulnerability mitigation prioritization. Scores for common vulnerabilities found on SCADAs have been generated using the most common attributes for the individual SCADA installations.

These vulnerabilities generally have a higher probability of being exploited and can potentially have a higher impact on an SCADA. High risk factors (probability and potential impact scores) and the widespread nature of these security weaknesses put these vulnerabilities at the top of the list for remediation in SCADAs as a whole. SCADA vendors and owners can assess their systems for these types of vulnerabilities, prioritize them using CVSS scores tailored to each vulnerability and the unique SCADA environment, and remediate them to the extent possible.

Complete remediation is not always feasible. Mitigation techniques can be guided by the CVSS metric values. Actions that alter vulnerability characteristics to a lower metric value lower the actual risk to the SCADA. For example, removing the vulnerable application from hosts that do not require it will reduce the probability of exploit, measured by the Target Distribution metric. Table C-3 lists generic guidance on lowering the probability of a vulnerability being exploited (aka exploitability).

Table C-3. Lowering risk due to the probability of a vulnerability being exploited.

Recommendation	Potential Methods
Increase Access Complexity	Restrict physical and electronic access to hosts, services, applications and other devices to the extent possible using method such as: • Detailed host and network firewall rules (directional, by IP, port, etc.) • Restrict physical access, including port security on network switches
Increase Level of Authentication Needed	Increase access complexity by enforcing access controls with strong authentication. • Configure hosts, services, applications and other devices to require credentials • Utilize multi-factor authentication where feasible and prudent (i.e.. VPN connections)
Decrease Target Distribution	Uninstall vulnerable service or application wherever possible
Increase Level of Remediation	Vendor: Develop and distribute a fix Owner: Develop and share a workaround mitigation for unsupported components

Table C-4 lists generic guidance on lowering the potential impact of a vulnerability being exploited.

Table C-4. Lowering risk due to the potential impact of vulnerability exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Lower the privileges that users, applications, and services are granted
Reduce Environmental Impact	Remove unnecessary functionality from SCADA hosts, applications, and services

Base metrics are intrinsic qualities of a specific vulnerability, but vendors and owners may be able to remove or change their vulnerabilities. Base characteristics about how a vulnerability can be accessed and its potential impacts cannot be directly changed, but may be indirectly changed with mitigating techniques that change how the vulnerability can be accessed, by whom, and the number of credentials required. Configuration changes may also be made that lessen the accessibility and potential impacts.

Assessments have shown that vulnerabilities in SCADA products can be introduced by their design, code, or integration of third-party products. SCADA vendors may be able to remediate vulnerabilities by redesign, code changes, or changing which third-party products they integrate with. They also have the ability to prevent vulnerabilities in new products.

Some attributes are configuration dependent and some vulnerabilities are caused by insecure configurations. SCADA vendors and/or owners can affect these vulnerabilities.

Environmental metrics adjust the Base metrics for the effect the vulnerability has on the unique organization’s environment. For example, an SCADA environment is more vulnerable to availability impacts than most other types of computer systems. This helps SCADA vendors and owners evaluate the risk individual vulnerabilities bring to their particular environments.

Sections C1–C10 summarize each of the 10 most critical common SCADA vulnerabilities. Attributes used in determining their criticality are listed, as well as the potential CVSS metric values. Specific information on each of the CVSS metrics is given to aid in understanding and scoring these types of vulnerabilities. This information may help SCADA vendors and owners identify ways they can lower the probability of exploitation or potential impact due to these common SCADA vulnerability types. Some of these risk factors cannot be influenced for a vulnerability type. Environmental attributes specific to the common vulnerability are addressed. Common SCADA environmental security requirements are discussed in Section C11. Section C12 summarizes ways of lowering the risk associated with the highest risk common cyber vulnerabilities in SCADA systems.

C-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 C-5 summarizes the security relevant attributes of unpatched software and their potential risk to SCADAs.

Table C-5. Summary of unpatched published vulnerabilities' security characteristics.

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

C-1.1 Generic CVSS v2 Score

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.[b] 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 published vulnerabilities are higher priority.

The example scores in Table C-6 represent the most dangerous known vulnerabilities identified on SCADA systems for commonly unpatched components.

In general, OS services are network accessible and do not require authentication to exploit. 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, they 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

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

SCADA 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.

Potential CVSS v2 metric values are summarized in Table C-6.

Table C-6. Generic CVSS v2 score for published vulnerabilities.

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	High
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	10.0
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	10.0
Total Score	10.0

C-1.2 Scoring and Reducing the CVSS V2 Risk Metrics

C-1.2.1 Base Metrics

Base metric values are fundamental characteristics of each individual vulnerability. All Base metric values are possible for unpatched published vulnerabilities. Most published vulnerabilities along with Base metric values are available from the National Vulnerability Database.[c]

C-1.2.2 Environmental Metrics

All Environmental metric values are possible and depend on the SCADA and its vulnerable components.

C-1.2.2.1 Environmental Security Requirements

Security Requirements are characteristics of the individual SCADA components that host the vulnerable service or application. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the vulnerable SCADA hosts or devices. See Section C-11, "SCADA Environmental Security Requirements," below.

C-1.2.2.2 Collateral Damage Potential

A remote attacker will most likely gain access to the SCADA by exploiting a known vulnerability. Unpatched published vulnerabilities are the most likely access vector because exploits for published vulnerabilities do not require knowledge of the SCADA. The potential impact to the SCADA depends on the individual vulnerability.

Collateral damage potential may be reduced by removing the vulnerable application from systems where comprise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

C-1.2.2.3 Target Distribution

Target Distribution is the percentage of SCADA hosts and devices that have the vulnerable software installed on them.

Target Distribution can be reduced by removing the vulnerable application from as many systems as possible. SCADA vendors can help by using up to date third-party applications that do not contain published vulnerabilities.

SCADA owners can lower the Target Distribution metric by removing the vulnerable application from as many systems as possible.

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

C-1.2.3 Temporal Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They may have the power to change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-1.2.3.1 Remediation Level

SCADA vendors can reduce the risk due to published vulnerabilities by delivering new systems without known vulnerabilities and test patches for third-party products as they are released. In some cases changes to the SCADA code is necessary for interaction with new versions of third-party products.

SCADA owners can test patches and updates themselves if SCADA vendor support is not available. SCADA owners may need to choose between applying patches and vendor support, if the SCADA vendor will not support the application of third-party patches or updates to their system. Table C-7 shows potential Remediation Level scenarios.

Table C-7. CVSS v2 Remediation Level metric values.

Metric Value	Metric Description
Official Fix	A complete vendor solution is available. The application vendor has issued an official patch or an upgrade is available, and the SCADA vendor has approved it.
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. SCADA owners have created a patch of their own or provided steps to work around or otherwise mitigate the vulnerability.
Unavailable	There is either no solution available or it is impossible to apply. For example, an upgrade is available, but it will void SCADA support or it has been proven to be incompatible with the SCADA.

C-1.2.4 Unpatched Published Vulnerabilities Recommendations

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They also may not be able to change the criticality of each system component. However, they can change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

Table C-8 lists specific recommendations for lowering the probability of a known vulnerability being exploited (aka exploitability).

Table C-8. Lowering risk due to the probability of a known vulnerability being exploited.

Recommendation	Potential Methods
Increase Access Complexity	Restrict access to vulnerable applications and services
Decrease Target Distribution	Uninstall the vulnerable service or application wherever possible
Increase Level of Remediation	Vendor: Deliver new systems without known vulnerabilities and test patches for third-party products as they are released Owner: Develop and share a workaround mitigation for unsupported components

Table C-9 lists recommendations on lowering the potential impact of a known vulnerability being exploited.

Table C-9. Lowering risk due to the potential impact of known vulnerability exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Minimize the privileges of users, applications and services that are allowed to access vulnerable applications, services, etc.
Reduce Environmental Impact	Remove unnecessary data and functionality from SCADA components that host vulnerable applications

See Section 4.1, "Published Vulnerabilities," for more information about this vulnerability.

C-1.2.5 SCADA Vendor Recommendations

SCADA vendors can reduce this score by changing the Remediation Level metric. The most important things that SCADA vendors can do to reduce the risk due to published vulnerabilities is to deliver new systems without known vulnerabilities and test patches for third-party products as they are released.

C-1.2.6 SCADA Owner Recommendations

SCADA owners may be able to reduce the risk from a known vulnerability on their system by creating a patch of their own or providing steps to work around or otherwise mitigate the vulnerability. If vendor support is not available for testing patches, owners can test patches and mitigations on backup or test systems. If SCADA code is available, owners can alter the code to interface with new third-party Application Programming Interfaces (APIs). Firewall rules or IDS signatures may be able to prevent or detect attempts to exploit the vulnerability.

C-2. Supervisory Control Access: Web HMI Vulnerabilities

Assessments have found that web services developed for the SCADA tend to be vulnerable to attacks that can exploit the SCADA 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 SCADA DMZ Web servers may provide the first step in the attack path by allowing access within the SCADA exterior boundary. Vulnerabilities in lower level components' Web servers can provide more steps in the attack path.

SCADA assessments have also found improper authentication, improper session tracking, SQL injection, and XSS vulnerabilities that can allow unauthorized access to Web servers and applications. Improper authentication can allow an attacker to impersonate another user's identity.

The use of vulnerable Web applications or servers for supervisory control functions can pose 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 C-10 summarizes the relevant security attributes of SCADA Web application vulnerabilities and their potential risks to the SCADA.

Common Web vulnerability details are given in Section 4.4.3, "Web applications and services."

Table C-10. Summary of SCADA Web application security characteristics.

SCADA 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
SCADA 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	SCADA Web applications and servers and/or SCADA Web clients' and servers' hosts
Ease of Detection	Medium to High
Attacker Awareness	High
Remediation Cost	Low
SCADA Prevalence	High

C-2.1 Generic CVSS v2 Score

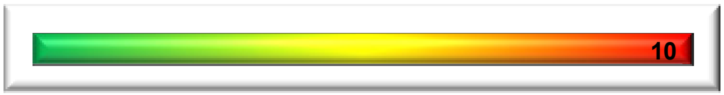
Since an attacker can remotely exploit Web applications over the network, the Access Vector is "Network". All other Base metrics are possible.

Web attack techniques are well known, even if a particular SCADA's Web vulnerabilities are unknown. Automated tools for identifying and exploiting Web vulnerabilities are available, so Exploitability is set to "High."

Collateral Damage Potential is high because a successful compromise of the Web HMI application or server may result in supervisory control access. CVSS v2 metrics for the use of remote display protocols on SCADAs are summarized in Table C-11 using the most common or critical values seen on SCADA.

Table C-11. Generic CVSS v2 score for SCADA 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	High

Metric	Values
Remediation Level	Not Defined
Report Confidence	Not Defined
Temporal Score	10.0
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	10.0
Overall CVSS Score	10.0
	

C-2.2 Scoring and Reducing the Web HMI CVSS Risk Metrics

C-2.2.1 Base Exploitability Metrics

C-2.2.1.1 Related Exploit Range (Access Vector)

Web HMI applications are bound to the network stack. Since an attacker can remotely exploit Web applications over the network, the Access Vector is Network. This is the nature of Web applications, and cannot be changed.

SCADA vendors should have all code evaluated for security. Network accessible applications should be first priority because they are most exposed to attack. Web applications that provide SCADA control functionality should be at the top of the list for vulnerability identification and remediation.

C-2.2.1.2 Access Complexity

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. Table C-12 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-12. Web HMI CVSS v2 Access Complexity scenarios.

Metric Value	Web HMI Access Complexity Scenarios
High	Specialized access conditions exist. For example, if access to the Web HMI is filtered, the attacking party is limited to the group of systems or users that have been given authorization. This does include, however, an attacker who is able to spoof or gain access to an authorized system or user account (i.e., access to the Web HMI is limited to a group of systems and users).
Medium	The access conditions are somewhat specialized; the following are examples: • The attacking party is limited to a group of systems or users at some level of authorization,

	possibly untrusted (i.e., access to the Web HMI server is granted to the business LAN). • The attack requires a small amount of social engineering that might occasionally fool cautious users. Many Web vulnerabilities require the victim to perform an action, such as phishing attacks that require the user to click on a link or download a file.
Low	Specialized access conditions or extenuating circumstances do not exist. The following are examples: • Access to the Web HMI is not filtered. • The vulnerable configuration is default or ubiquitous (i.e., the default Web HMI configuration does not use authentication or encryption, or allows control of the physical process).

SCADA owners may want to make access to a Web HMI as difficult as possible. The group of systems and users that have access to the Web HMI can be restricted as much as possible. Web HMI monitor and control functionality can be as limited as possible, also. Permissions can be customized to the individual clients and users that must be given access.

SCADA vendors can provide the functionality to restrict access and provide a secure communication path to the SCADA Web server. The default configuration should have all security features enabled.

C-2.2.1.3 Level of Authentication Needed

The Authentication metric measures the number of times an attacker must authenticate before exploiting the Web HMI vulnerability. The fewer authentication instances that are required, the higher the vulnerability score. The more different passwords and different types of authentication required before one is able to gain access to a Web HMI, the less likely it will be that an attacker will gain the access required to exploit the vulnerability.

A vulnerability can be scored as None, Single or Multiple, depending on how many instances of authentication must occur before exploiting a vulnerability. Table C-13 gives example scenarios that fit each of the Authentication metric values in an SCADA environment.

Table C-13. Web HMI CVSS v2 Authentication scenarios.

Metric Value	Web HMI Authentication Scenarios
Multiple	A user must provide multiple credentials to gain the access required to exploit the Web HMI.
Single	To exploit the Web HMI, attacker must provide credentials once. For example, credentials are required to access the Web HMI, but after authenticating, the user can escalate from read-only access and gain unauthorized SCADA control functionality.
None	Unauthorized access to the Web HMI can be gained without providing credentials. For example, access to the Web HMI is only filtered by IP address. If the attacker is able to spoof an authorized IP address, he can gain access to the Web HMI without providing credentials.

SCADA vendors can make sure that the Web HMI supports secure authentication and it is recommended that they use proven authentication products.

SCADA owners should require a password to log in to the Web HMI that is different from the password required to log onto the client machine. This prevents an attacker from automatically gaining Web HMI access to an SCADA host after compromising an authorized client.

C-2.2.2 Impact Metrics

Potential consequences of SCADA Web application vulnerabilities include user account compromise and hijacking of user Web sessions. Exploitation of SCADA Web applications may lead to exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or allowing access to the Web client's or server's host.

Confidentiality, integrity and availability impacts depend on the individual vulnerability. The Web client's host, the Web server, and the Web HMI application are possible targets of attack. Complete compromise of any of these components is possible.

C-2.2.2.1 Environmental Security Requirements

Security Requirements are characteristics of the individual SCADA components. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the affected SCADA hosts or devices. This may include the Web Server, Database server, and Web HMI client hosts.

Security requirements for the Web client's host and the Web server depend on the supervisory control functionalities they possess and are responsible for, the data they contain, and the physical process they are connected to.

Confidentiality is generally not as important as integrity or availability for an HMI. Integrity of SCADA data is generally the highest priority in relation to confidentiality and availability. Availability is known for being the highest priority for an SCADA, but improper integrity of SCADA process or system data can be dangerous.

Confidentiality requirements of Web clients and servers depend on the organization's confidentiality requirements for the SCADA data processed by the Web HMI.

The impact on integrity of unauthorized access to the Web HMI depends on the functionality available in the Web HMI. If the Web HMI provides supervisory control functionality, unauthorized access to, or exploitation of, the Web HMI client or server could allow an attacker supervisory control access. Table C-14 gives example scenarios for the Integrity Requirement metric values in an SCADA environment.

Table C-14. Web HMI CVSS v2 Integrity Requirement scenarios.

Metric Value	Web HMI Integrity Requirement Scenarios
Low	Loss of data integrity on the Web HMI is likely to have limited effect on the operation of the SCADA or its organization's business interests. For example, the Web HMI is used for status information and the process is not affected by data going to and from the Web HMI.
Medium	Manipulation of data on the SCADA host could have a serious adverse effect on the business or safety. For example, access to a Web HMI could give an attacker the ability to control the SCADA.
High	Depending on the system or process under control and the individual component, Web HMI functionality could have a catastrophic adverse effect on the business or safety. Alteration of system data or malicious operation of the physical system may result in economic, environmental or safety catastrophes.

The Availability Requirement metric measures the potential for loss of revenue or life due to loss of availability of the Web HMI. Table C-15 gives example scenarios for the Availability Requirement metric values in an SCADA environment.

Table C-15. Web HMI CVSS v2 Availability Requirement scenarios.

Metric Value	Web HMI Availability Requirement Scenarios
Low	Loss of availability of the Web HMI is likely to have limited effect on the operation of the SCADA or its organization's business interests. For example, Web HMI functionality is not required for SCADA operations.
Medium	Loss of the Web HMI availability could have a serious adverse effect on the business or safety. For example, monitor and control of the system or process is only available through Web HMI access.
High	Loss of the Web HMI is likely to have catastrophic adverse effect on the business or safety. For example, SCADA monitor and control capability is only available through the Web HMI and loss of this capability could result in catastrophic consequences.

C-2.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

All environmental metric values are possible and depend on the SCADA and its vulnerable components. SCADA-specific impacts include unauthorized access to the Web HMI, Web server or other Web applications and functionalities. This can possibly lead to 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.

C-2.2.2.3 Reducing Potential Impact

Collateral damage potential can be reduced by removing functionality from the Web HMI where compromise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

SCADA vendors and owners can lower their actual Confidentiality Impact from unauthorized Web HMI access by limiting the information that can be accessed through the Web HMI.

SCADA vendors can lower the Integrity Impact from unauthorized Web HMI access by reducing the Web HMI's functionality. Web HMI access may have no impact on system integrity if only monitor functionality is built in.

C-2.2.3 Percentage of Vulnerable Systems (Target Distribution)

The Target Distribution is the percentage of SCADA components hosting the vulnerable SCADA Web applications or servers. The Web HMI server(s) generally represent a small portion of the SCADA environment. If the vulnerability is in the Web HMI (client) software, the target distribution depends on the percentage of hosts in the SCADA environment that have the Web HMI software installed. All hosts that have the software installed should be counted.

Target Distribution can be reduced by removing the Web HMI software from as many systems as possible.

C-2.2.4 Temporal Metrics

These metrics describe elements about the vulnerability that change over time.

C-2.2.4.1 Availability of Exploit (Exploitability)

Vendors and owners cannot change the availability of exploits, but they should prioritize the mitigation of vulnerabilities that have exploit code or tools published.

Even if there is not a specific exploit developed for a vulnerability in a proprietary SCADA application, common security weaknesses and associated exploit techniques are well known for Web applications. The SysAdmin, Audit, Network, Security (SANS) *2009 Top Cyber Security Risks* report states: “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 improper programming practices, SCADA Web services are vulnerable to the most popular attack techniques, such as Structured Query Language (SQL) injection, XSS, directory traversal, and authentication bypass. SCADA Web applications are also more exposed to attack than most SCADA components, and may provide the capability to alter SCADA data or state.

Vendors and owners can conduct in-house and third-party security assessments of their products and remediate identified vulnerabilities. Table C-16 shows potential scenarios.

Table C-16. Web HMI vulnerability CVSS v2 Exploitability scenarios.

Metric Value	Web HMI Vulnerability Exploitability Scenarios
Unproven	No exploit code is available, or an exploit is entirely theoretical. This should not apply to most Web vulnerabilities.
Proof-of-Concept	Web attack techniques are well known, even if a particular SCADA’s Web vulnerabilities are unknown. This applies to most Web vulnerabilities that do not have specific exploit code available and cannot be exploited via automated tools.
Functional	Functional exploit code is available. The code works in most situations where the vulnerability exists.
High	According to the SANS top cybersecurity risks report, if the Web application is vulnerable, there are automated tools available to discover and exploit them. “Automated tools, designed to target custom Web application vulnerabilities, make it easy to discover and infect several thousand Web sites.”[1]

C-2.2.4.2 Type of Fix Available (Remediation Level)

SCADA vendors have the power to remediate vulnerabilities in their own products. Table C-17 shows potential scenarios.

Table C-17. Web HMI vulnerability CVSS v2 Remediation Level scenarios.

Metric Value	Web HMI Vulnerability Remediation Level Scenarios
Official Fix	The SCADA vendor has issued an official patch, or an upgrade is available.
Temporary Fix	The SCADA vendor has issued a temporary hotfix, tool, or workaround.
Workaround	There is an unofficial, non-vendor solution available. For example, SCADA users have created a patch of their own or provided steps to work around or otherwise mitigate the vulnerability.
Unavailable	There is either no solution available or it is impossible to apply.

SCADA 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 SCADA Web server.

If Web HMI functionality is not intended to provide control of the SCADA, this functionality should not be built into the Web HMI application. This removes the possibility of unauthorized control through the Web HMI.

C-2.2.4.3 Level of Verification that Vulnerability Exists (Report Confidence)

NSTB assessments have found vulnerabilities in Web HMI server and client applications. These vulnerabilities are reported to the SCADA vendors with supporting information to aid in remediation. Table C-18 lists potential SCADA Web vulnerability reporting scenarios.

Table C-18. Web HMI vulnerability CVSS v2 Report Confidence scenarios.

Metric Value	Web HMI Report Confidence Scenarios
Unconfirmed	Web HMI vulnerabilities may be announced by independent security companies or research organizations, but the SCADA vendor has not confirmed the vulnerability and no known exploits exist.
Uncorroborated	Web HMI vulnerabilities may be announced by independent security companies or research organizations, but there are conflicting technical details or other lingering ambiguity.
Confirmed	Known vulnerabilities may exist in the Web server or even in the SCADA vendor’s code. The SCADA or Web server vendor has confirmed the vulnerability or exploit code has been published.

C-2.2.5 Web HMI Recommendation Summary

Table C-19 lists specific recommendations for lowering the probability of a Web application or server being exploited.

Table C-19. Lowering risk due to the probability of a Web application or server being exploited.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors and owners can restrict access to SCADA Web servers and authorized clients: • SCADA vendors can implement and test proven access control mechanisms, making sure to correctly enforce access controls on the server side. • SCADA owners can configure Web servers to restrict access to the minimum IP range, ports, and users necessary.
Decrease Target Distribution	SCADA owners can minimize the distribution of SCADA Web applications and the number of authorized clients.
Increase Level of Remediation	SCADA vendors can assess Web applications for vulnerabilities and remediate the applications using secure Web programming resources such as the Open Web Application Security Project (OWASP).[d] SCADA owners can use an application firewall that can detect typical Web attacks. Secure hosts and browser settings to the extent possible. They can also educate employees on how to detect and avoid attacks aimed at users, such as phishing attacks. Users can mitigate some vulnerabilities by using HTTPS for the entire Web session.

Table C-20 lists recommendations on lowering the potential impact of a Web application or server being exploited.

Table C-20. Lowering risk due to the potential impact of Web application or server exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Potential impacts can be reduced by minimizing the privileges required to run Web applications and the privileges of Web clients on the Web server. SCADA vendors can design and implement Web applications to run using the lowest privileges that are required to accomplish the necessary tasks. SCADA owners can configure user accounts and run services and applications with the least privileges necessary.
Reduce Environmental Impact	SCADA vendors can remove unnecessary functionality from the SCADA Web server and client applications. SCADA owners can ensure that unwanted (unnecessary) functionality is not available in SCADA Web applications. Disable unnecessary functionality and validate that it cannot be accessed.

C-2.2.5.1 SCADA Vendor Recommendations

SCADA 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 SCADA Web server.

The *OWASP Top Ten*[21] 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.[22]

[d] http://www.owasp.org/index.php/Main_Page

CWE categories, *CWE-287: Improper Authentication* and *CWE-442: Web Problems*, contain related authentication and Web programming information as well.[9]

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

1. SCADA Internet access policy
2. SCADA user awareness and training
3. Coordination of security efforts between corporate IT network and SCADA network
4. Firewall between the SCADA network and the information technology network
5. Up-to-date patches
6. Web browser and e-mail security
7. Secure code.[23]

C-2.2.5.2 SCADA Owner Recommendations

SCADA owners should minimize access and available functionality of Web servers and clients. They should validate that the access controls are configured to restrict all unwanted users and SCADA functionality. Web servers should be placed on a DMZ and use a replicated database on the DMZ as well.

C-3. 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 SCADA to administer SCADA hosts remotely or access operator screens and other SCADA applications. Remote display protocols used by SCADA 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 SCADA host's display.

The use of remote display software for remote access to supervisory control functions could be the most significant vulnerability on an SCADA because it allows unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user. (Remote display vulnerabilities are well known.) Table C-21 summarizes the relevant security attributes of remote display protocols and their potential risk to SCADAs.

See Section 4.3.1.2, "Implement secure remote access " for more information.

Table C-21. 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.
SCADA Impact	Unauthorized access to SCADA components: Possible unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user.

Possible Consequences	May allow DoS, code execution, data loss, or security bypass.
Vulnerable Components	SCADA 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
SCADA Prevalence	High

C-3.1 Generic CVSS v2 Score

From an SCADA 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 C-1 and C-2 illustrate these different scenarios. Figure 1 illustrates the scenario where the operator screens are displayed using a remote display protocol from an HMI, or operator screen, server on the local SCADA 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.

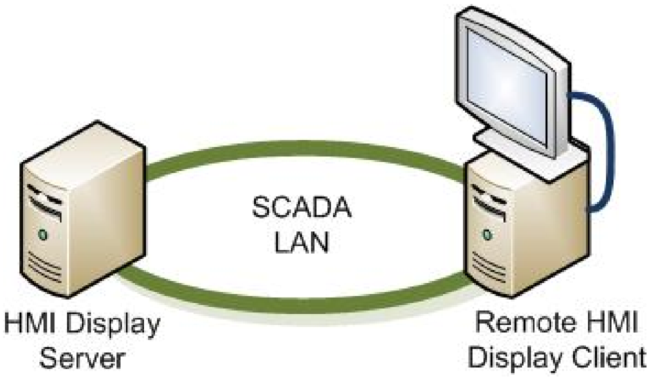


Figure C-1. Operator screens are remotely displayed from HMI LAN.

Figure C-2 illustrates how remote display connections from outside the SCADA 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 Figure C-1, but the attacker does not need to gain access to the SCADA LAN in the second scenario in Figure C-2.

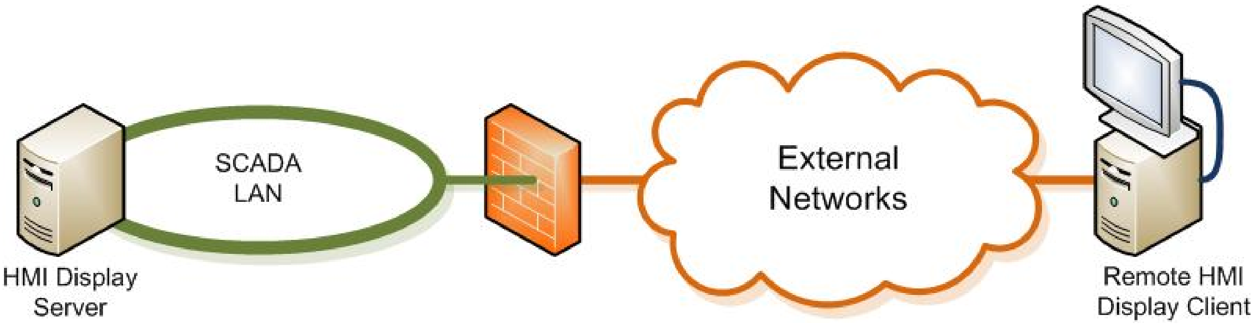


Figure C-2. Operator screens are remotely displayed from a remote network.

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.

CVSS v2 metrics for the use of remote display protocols on SCADAs are summarized in Table C-22 using the most common or critical values seen on SCADA.

Table C-22. Generic CVSS v2 score for the use of remote display protocols on SCADAs.

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
Environmental Metrics		
Collateral Damage Potential	High	High
Target Distribution	Not Defined	Not Defined
Availability Requirement	High	High
Integrity Requirement	High	High
Confidentiality Requirement	Medium	Medium
Environmental Score	9.8	9.0

Metric	Remote Connection	Same-LAN Connection
Overall CVSS Score	**9.8**	**9.0**

C-3.2 Scoring and Reducing the CVSS Risk Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They also may not be able to change the criticality of each system component. They can change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-3.2.1 Base Exploitability Metrics

C-3.2.1.1 Related Exploit Range (Access Vector)

In strict terms, remote display protocols are bound to the network stack, but whether the attacker requires local network access or local access to the client depends on how the server is configured and how it is accessed. SCADA vendors and owners can affect the Access Vector metric by the way they configure and use remote display protocols. Table C-23 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-23. SCADA remote display vulnerabilities CVSS v2 Access Vector scenarios.

Metric Value	Remote Display Access Vector Scenarios
Adjacent Network	A vulnerability exploitable with adjacent network access requires the attacker to have access to either the broadcast or collision domain of the vulnerable software. In general, MitM attacks require local network access. If remote display protocols are unencrypted or use weak encryption, they are vulnerable to MitM attack by anyone who is able to access the network traffic along its path between the SCADA host and the remote display client. If the remote display client and server are on the same local network, an attacker would have to gain access to the SCADA local network to be able to intercept it.
Network	A vulnerability exploitable with network access means the vulnerable software is bound to the network stack and the attacker does not require local network access or local access. If the remote display protocol server is configured on the SCADA host to allow access to any client, an attacker can connect to it from a remote network. Even if the SCADA host is configured to only allow remote display connections to a set of IP addresses, an attacker may be able to spoof an authorized IP address and gain access.

C-3.2.1.2 Access Complexity

The configuration of the remote display access controls determines the access complexity. The default configuration or the most common configuration is generally used for scoring a vulnerability, but SCADA vendors can apply the criteria to their default configurations and owners can apply the appropriate level for how they have configured access to their SCADA displays. Table C-24 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-24. SCADA remote display vulnerabilities CVSS v2 Access Complexity scenarios.

Metric Value	Remote Display Access Complexity Scenarios
Medium-High	If access to the remote display protocol is filtered, the attacking party is limited to the group of systems or users that have been given authorization. This does include, however, an attacker who is able to spoof or gain access to an authorized system or user account.
Low	If a remote display protocol is configured to accept connections from anywhere, the access complexity is low. Once an attacker has gained access to the remote display port on the SCADA host, he can view its display. Access complexity is low if remote display traffic is unencrypted. It is also low if the encryption used by the protocol can be broken by script kiddie tools.

C-3.2.1.3 Level of Authentication Needed

This metric measures the number of times an attacker must authenticate before they are able to gain remote display access to an SCADA server. The fewer authentication instances that are required, the higher the vulnerability score. The more different passwords and different types of authentication required before one is able to gain access to an SCADA host's display, the harder it will be for anyone to gain unauthorized access.

SCADA vendors can make sure that the remote access functionality that they provide supports secure authentication. They can also be sure that the remote access protocols and encryption protocols integrated into their products are up to date and do not have unpatched vulnerabilities.

SCADA owners can require a password to establish a remote display connection that is different from the password required to log onto the client machine. This prevents an attacker from automatically gaining remote display access to an SCADA host after compromising an authorized client. Make sure the password is sent securely and any encryption keys are securely stored.

Table C-25 gives example scenarios that fit each of the Authentication metric values in an SCADA environment.

Table C-25. SCADA remote display vulnerabilities CVSS v2 Authentication scenarios.

Metric Value	Remote Display Authentication Scenarios
Multiple	A user must provide multiple credentials to remotely access an SCADA host's display. For example, after authenticating as a user on an authorized client, one must then use another password to establish a remote display connection to an SCADA host. Strong host authentication is used so that the authorized client cannot be spoofed and the remote display password cannot be sniffed or decrypted. Another password should be required for access to SCADA applications.
Single	To access an SCADA display remotely, the attacker must be logged into the system. This means that they must provide credentials once.
None	If remote display protocols are configured to allow access to anyone, or even a set of IPs, credentials are not required to establish a connection.

C-3.2.2 Impact Metrics

Possible consequences from the use of vulnerable remote display protocols include DoS, code execution, data loss, or security bypass. This depends on the privileges granted to the account that was given remote access.

Compromise of accounts with root-level access can lead to a complete loss of confidentiality, integrity, and availability, while compromise of accounts with user-level access should only lead to a partial loss of confidentiality, integrity, and availability.

C-3.2.2.1 Environmental Security Requirements

Security Requirements are characteristics of the individual SCADA components that host the vulnerable remote display services. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices that can be accessed using the vulnerable remote display protocol. See Section C-11, "SCADA Environmental Security Requirements," below.

C-3.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

Assessments have found that exploitation of remote display protocols can lead to unauthorized access to SCADA components; possible unauthorized remote access to graphical supervisory control software, as well as any other functionality allowed to the remote user.

From an SCADA perspective, the most severe impact that can result from the use of vulnerable remote display protocols is unauthorized supervisory control access. In CVSS terms, this translates to process integrity and availability.

C-3.2.2.3 Reducing Potential Impact

Collateral Damage Potential can be reduced by removing remote access capability from systems where comprise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

SCADA vendors can lower the impact from unauthorized remote display access by reducing the privileges needed to run applications that utilize remote display protocols, such as the HMI.

SCADA owners can lower the impact from unauthorized remote display access by limiting the privileges of accounts that are allowed to be remote accessed. They can also limit the data that resides on servers that allow remote access.

SCADA vendors and owners can also limit the functionality and data that resides on servers that allow remote access. If remote access is only needed for viewing system status, the remote user should not have privileges to change system information. The safest route is to not build control functionality into applications that are meant only for viewing purposes. This ensures that they cannot be exploited to gain control access.

SCADA vendors and owners can lower the Integrity requirement by designing their systems so that they do not rely on data from servers that allow remote access. For example, system data can be duplicated (push operation) onto the remote access server. Placing a copy of the system database on a DMZ SCADA server used for remote access guarantees that compromise of the database by the remote user does not compromise the integrity of the main system data.

C-3.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution is the percentage of SCADA hosts that allow remote display connections. Target Distribution can be reduced by removing or securing vulnerable remote access applications and services on as many systems as possible. SCADA vendors can help by using up to date third-party applications that do not contain published vulnerabilities.

SCADA owners can lower the Target Distribution metric by removing remote access capability from as many systems as possible.

C-3.2.4 Temporal Score Metrics

These metrics describe elements about the vulnerability that change over time. SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They may have the power to change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-3.2.4.1 Availability of Exploit (Exploitability)

Vendors and owners cannot change the availability of exploits, but they can prioritize the mitigation of vulnerabilities that have exploit code or tools published. Vendors and owners can search for vulnerabilities and exploits available for the services they support and use. Table C-26 shows the potential scenarios.

Table C-26. SCADA remote display vulnerabilities CVSS v2 Exploitability scenarios.

Metric Value	Remote Display Exploit Availability Scenarios
Unproven	No exploit code is available, or an exploit is entirely theoretical. Many Secure Shell (SSH) vulnerabilities are theoretical or have no exploit code available. If SSH is used to tunnel the remote display connection, it could fall into this category. Other encryption solutions or remote display protocols may fit as well.
Proof-of-Concept	Proof-of-concept exploit code or an attack demonstration that is not practical for most systems is available.
Functional	Functional exploit code is available. The code works in most situations where the vulnerability exists. Graphic User Interface (GUI) MitM tools exist and some have options that target specific remote display protocols.
High	The CVSS definition does not apply to the exploitation of remote display protocols: “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).”

C-3.2.4.2 Type of Fix Available (Remediation Level)

MitM and access control vulnerabilities can be fixed with access control rules, secure authentication, and strong encryption. There are many encryption products that can be used to secure the channel between the SCADA server and remote client. There is still risk associated with allowing remote access due to the level of trust that is given to the client. Although the channel can be secured using encryption, an attacker may be able to gain access by compromising the client.

Possible mitigations include:

- **Risk Remediation:** Disable or remove remote display protocols.
- **Access Vector Risk Reduction:**
  - Only allow connections from hosts that you control and maintain a high level of security.
  - Only allow connections from clients in the same security zone or local network.
- **Access Complexity:**
  - Configure remote display protocol servers to restrict access to minimal set of IPs and users.

  - Place a server in a DMZ and duplicate applications for remote access or to proxy access to SCADA hosts.
  - Encrypt remote display traffic using strong encryption (e.g., SSH tunnel or IPSec).
- **Authentication:**
  - Require a password to establish a remote display connection that is different from the password required to log onto the client machine. This prevents an attacker from automatically gaining remote display access to an SCADA host after compromising an authorized client. Make sure the password is sent securely and any encryption keys are securely stored.
- **Impact Reduction:**
  - Allow the least privileges necessary to remote access accounts.
  - Install the least functionality necessary on remote access servers.

C-3.2.4.3 Level of Verification that Vulnerability Exists (Report Confidence)

Remote access protocols are widely used and vulnerabilities are well known. Vendors and owners can search for vulnerabilities associated with the protocols they use and alternative products.

C-3.2.5 Remote Access Recommendation Summary

Vendors can apply remote access recommendations to default configurations. Owners can apply them to their systems.

Table C-27 lists specific recommendations for lowering the probability of a remote display service or connection being exploited (aka exploitability).

Table C-27. Lowering risk due to the probability of a remote display protocol being exploited.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors and owners can restrict access to remote display services.
Decrease Target Distribution	SCADA vendors and owners can uninstall or disable remote display services on SCADA hosts that do not require remote access
Increase Level of Remediation	SCADA vendors can design HMI clients to securely connect to HMI servers; support and recommend secure remote access options; and provide secure configuration documentation that includes required services for each SCADA host, application, or device along with required communication partners. SCADA owners can tunnel required insecure connections using secure shell, VPN technology, or other encryption products. Restrict access and validate the security of authorized remote access clients.

Table C-28 lists recommendations on lowering the potential impact of a remote display service or connection being exploited.

Table C-28. Lowering risk due to the potential impact of remote display protocol exploitation.

Recommendation	Potential Methods
Reduce Base Impact	SCADA vendors and owners can minimize the privileges of users that are allowed to remotely access SCADA hosts.

Recommendation	Potential Methods
Reduce Environmental Impact	SCADA vendors and owners can remove unnecessary data and functionality from SCADA hosts that allow remote access. They can add a DMZ host to proxy remote requests or provide SCADA status.

C-3.2.5.1 SCADA Vendor Recommendations

Access Vector, Access Complexity, and Authentication are affected by the way the remote access protocols are configured and utilized. SCADA vendors can reduce this score for their systems by providing secure options for remotely accessing the HMI and providing secure default configurations. Secure configurations include access restrictions and secure authentication.

Confidentiality, Integrity, and Availability impact depends on the permissions of the users allowed to remotely connect. SCADA vendors can reduce these metrics by reducing the permissions required to run SCADA applications.

C-3.2.5.2 SCADA Owner Recommendations

SCADA owners can reduce the risk from vulnerable remote access protocols by configuring remote access protocols on SCADA hosts to limit access, require secure authentication, and use a trusted path. When configuring SCADA 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 client host.

Access Vector, Access Complexity, and Authentication scores can be lowered by minimizing remote access and requiring that they be accessed through secure channels. The level of authentication needed can reduce risk by requiring single or multi-factor authentication.

Confidentiality, Integrity, and Availability impact depends on the permissions of the users allowed to remotely connect. Owners can reduce these metrics by limiting the permissions of the accounts that are allowed to connect.

C-4. Access to SCADA Functionality: Improper Access Control (Authorization)

Access control mechanisms determine which network, host, and SCADA 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 SCADA installations start services as the root user and root group. Many services may not need to be started with this privilege level, and doing so exposes system resources to preventable risks. By restricting necessary privileges during SCADA 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 SCADA 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 SCADA protocols, services, and applications may increase the impact from compromise as well.

This is a significant vulnerability because it allows unauthorized access to SCADA networks, hosts, and functionality. Table C-29 summarizes the security relevant attributes of improper access control.

Table C-29. Summary of Improper Access Control security characteristics.

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

C-4.1 Generic CVSS v2 Score

Many SCADA services are allowed unnecessary privileges. Assuming that some access was intentionally granted, “Authentication” is the number of credentials that were required. The worst-case option is “None.”

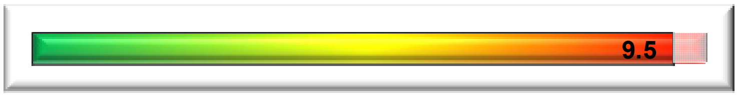
The Access Complexity is “Low” because no additional access or specialized circumstances need to exist for the exploit to be successful.

Each of the Impact metrics is set to “Complete.” The actual impact depends on the application.

CVSS v2 metrics for least user privileges violations on SCADAs are summarized in Table C-30 using the most common or critical values seen on SCADA. See Section 4.6, “Authorization ,” for additional information.

Table C-30. Generic CVSS v2 score for least user privileges violations.

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	Proof-of-Concept
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	9.0

Metric	Value
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	High
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.5
Overall CVSS Score	**9.5**
	

C-4.2 Scoring and Reducing the CVSS Risk Metrics

Access controls are security mechanisms used to limit access to SCADA components. SCADA vendors build functionality into their applications and services. These applications and services must be granted permission to perform the functions and access the files necessary. If SCADA vendors design their systems to execute programs with unnecessary privileges, SCADA users may require unnecessary privileges.

SCADA vendors can reduce the probability and impact of compromise by removing unnecessary functionality from SCADA services and applications, and minimizing the system privileges they require.

SCADA owners can reduce the probability and impact of compromise by installing and running only the necessary services and applications, and creating user accounts with the least privileges necessary to perform their functions.

C-4.2.1 Base Exploitability Metrics

C-4.2.1.1 Related Exploit Range (Access Vector)

The access vector for exploiting SCADA access control vulnerabilities depends on how the vulnerable access control mechanism is designed. Access control mechanisms built for users logged into the associated host fit under the "Local" category. Access control mechanisms built to allow access over the network can either be vulnerable to attackers that have gained access to the local network "Adjacent Network," or remote attackers "Network" depending on how they are implemented.

Table C-31 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-31. Improper SCADA access control CVSS v2 Access Vector scenarios.

Metric Value	Improper Access Control: Access Vector Scenarios
Local	Local access controls apply to the user accounts on a host.
Adjacent Network	Adjacent network access controls apply to permissions that are granted at the local network level, such as local network such as a TCP/IP subnet or a wireless or Bluetooth network.

Metric Value	Improper Access Control: Access Vector Scenarios
Network	Improper network access controls apply to firewall and router access control lists that allow more than the necessary IP addresses and ports in and out of the SCADA's network security zones.

C-4.2.1.2 Access Complexity

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. An access control vulnerability is lower risk if it can only be exploited when specific conditions exist, in rare configurations, or by easily detected social engineering methods.

Exploitation of authorization requires that the attacker have access to the target, albeit with an account that is less privileged than could be appropriate for the targeted resources. The target must have incorrectly configured their access control mechanisms such that sensitive information, which should only be accessible to more trusted users, remains accessible to less trusted users. Table C-32 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

SCADA vendors can increase the access complexity by deploying their systems with minimal privileges by default. SCADA owners would have to change the default configuration to grant users unnecessary privileges.

SCADA vendors can reduce the attack surface when designing applications by carefully mapping roles with data and functionality. Developers can then use role-based access controls to enforce the roles at the appropriate boundaries.

Table C-32. Improper SCADA access control CVSS v2 Access Complexity scenarios.

Metric Value	Improper Access Control: Access Complexity Scenarios
High	The SCADA default configuration grants minimal privileges and SCADA owners would have to grant users unnecessary privileges.
Medium	The access conditions are somewhat specialized, for example either: • The affected configuration is non-default, and is not commonly configured • The access is granted to a limited group of systems or users.
Low	Specialized access conditions or extenuating circumstances do not exist, for example either: • The SCADA design requires overly permissive access controls • The SCADA software does not properly constrain access to resources or functionality.

C-4.2.1.3 Level of Authentication Needed

This metric measures the number of times an attacker must authenticate before they are given the unnecessary access. It is important to note that the Authentication metric is different from Access Vector. Here, authentication requirements are considered once the system has already been accessed. Specifically, for locally exploitable vulnerabilities, this metric should only be set to “single” or “multiple” if authentication is needed beyond what is required to log into the system.

Table C-33 gives example scenarios that fit each of the Authentication metric values in an SCADA environment.

Table C-33. Improper SCADA access control CVSS v2 Authentication scenarios.

Metric Value	Improper Access Control Authentication Scenarios
Multiple	A user must provide multiple credentials to authenticate to the SCADA application, but additional authorization checks are not made for SCADA functionalities that should not be available to all users.
Multiple - Single	A user must provide multiple credentials to authenticate to the SCADA application, but additional authorization checks are not made for SCADA functionalities that should not be available to all users.
None	No credentials are necessary to access the SCADA application.

C-4.2.2 Impact Metrics

Access control mechanisms determine which network, host, and SCADA 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 SCADA 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 SCADA design and implementation, the window of exposure and criticality of impact may be 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 SCADA 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.

Base impact metrics measure the impact on confidentiality, integrity, and availability due to granting or not restricting unnecessary access to an SCADA network, host, or application. The possible consequences of weak access controls include information leaks.

C-4.2.2.1 Environmental Security Requirements

Security Requirements are characteristics of the individual SCADA components. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices that lack sufficient access controls. See Section C-11, "SCADA Environmental Security Requirements," below.

C-4.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

Unauthorized access to information or functionality provided by SCADA hosts and applications can result is unauthorized supervisory control access. Unnecessary functionality in SCADA protocols, services, and applications increases the impact from compromise as well.

C-4.2.2.3 Reducing Potential Impact

Collateral damage potential can be reduced by adding access controls to the software, hosts, and networks where comprise could lead to loss of revenue, safety, or productivity (if possible).

C-4.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution depends on the percentage of hosts the unnecessary access is granted.

SCADA vendors and owners can remove unnecessary applications and users from SCADA hosts individually, and restrict access to SCADA hosts, applications and networks as much as possible.

C-4.2.4 Temporal Score Metrics

C-4.2.4.1 Availability of Exploit (Exploitability)

Exploitability of individual vulnerabilities can range between “Unproven” and “High.” Attack methods are well known, so the exploitability of improper SCADA access control mechanisms is “Proof-of-concept.”

According to the Common Attack Pattern Enumeration and Classification (CAPEC), *CAPEC-1: Accessing Functionality Not Properly Constrained by ACLs* requires a low skill or knowledge level. “In order to discover unrestricted resources, the attacker does not need special tools or skills. The attacker only has to observe the resources or access mechanisms invoked as each action is performed and then try and access those access mechanisms directly.”[e]

C-4.2.4.2 Type of Fix Available (Remediation Level)

Some access control weaknesses exist because firewalls, network devices, hosts, applications, SCADA devices, etc., were not configured as securely as they could be. They allow access that is not required for SCADA operation. This can be remediated by removing unnecessary applications and individually restricting the privileges granted to user accounts, applications, and services. 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.

Other access control weaknesses exist because SCADA owners do not know what access must be allowed for SCADA operation. SCADA vendors can remediate this problem by providing documentation on required services, permissions, and communication channels. For each SCADA 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. SCADA owners can work with vendors to identify system access requirements.

SCADA owners can deduce this information themselves, if necessary. Applications can be removed and permissions can be lowered on a test or backup system. 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.

A third scenario is that the SCADA design prevents the application of sufficient network and host access controls. For example:

- Some SCADA applications require users to log in with administrative privileges
- Some SCADA services require administrative privileges

[e] http://capec.mitre.org/data/definitions/1.html

- Some SCADA protocols require large port ranges to be opened
- Some SCADA protocols require access between network security zones.

SCADA vendors can redesign their products using the principle of least privileges, but until an update is released the remediation level is “Unavailable.”

SCADA software authorization vulnerabilities are bugs in code written to perform authorization checks by the application. Generally, the SCADA vendor must release a patch to remediate this type of vulnerability.

Table C-34 lists possible SCADA access control weakness remediation scenarios.

Table C-34. Improper SCADA access control CVSS v2 Remediation Level scenarios.

Metric Value	Improper Access Control Remediation Level Scenarios
Official Fix	There is a patch available that fixes a vulnerability in an application’s access control mechanism.
Temporary Fix	The SCADA vendors document known requirements as they work toward identifying and reducing all requirements.
Workaround	SCADA owners can lock down test system hosts to safely identify necessary applications and the least privileges necessary. SCADA owners can monitor their systems and create detailed firewall rules.
Unavailable	The vendor has not provided a solution and no workaround mitigation can be applied. For example, SCADA design prevents the application of sufficient network and host access controls.

C-4.2.4.3 Level of Verification that Vulnerability Exists (Report Confidence)

All Report Confidence values are possible. SCADA owners can verify the lack of application level access control mechanisms with the SCADA vendor. They can also verify that overly permissive network, host, and application access controls on their systems are required for SCADA operation.

C-4.2.5 Improper Access Control Recommendation Summary

Access controls are the basis of protecting a system from attack. They offer the best protection for unknown vulnerabilities in particular. Preventing an attacker from reaching the vulnerability addresses the risk of exploitation. Potential impact is reduced as well if less functionality and network access is available to an attacker who has gained access to an SCADA component. Therefore, the recommended solution for access control weaknesses are mechanisms that allow more control over access privileges and the most restrictive use of these mechanisms.

SCADA vendors can build strong access control mechanisms into their products and provide documentation that allows customers to remove unnecessary applications and minimize privileges on their systems.

Table C-35 lists specific recommendations for lowering the probability of unauthorized access to an SCADA application. See Section 4.6, “Authorization ,” for additional information.

Table C-35. Lowering risk due to the chance of authorization mechanism exploitation.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors and owners can lock down the SCADA environment as much as possible: • SCADA vendors can reduce the attack surface when designing applications by carefully mapping roles with data and functionality. Developers can then use role-based access controls to enforce the roles at the appropriate boundaries. • SCADA vendors can deploy their systems with minimal privileges by default. • SCADA owners can restrict access to SCADA hosts and networks. They may be able to configure user accounts in SCADA applications and restrict access to network applications to the necessary minimum IP range, ports, and users.
Increase Authentication	SCADA vendors can implement strong authentication methods to enforce access controls. SCADA owners can to enforce access controls by implementing strict password policies in applications, hosts, and devices that support them.
Decrease Target Distribution	SCADA vendors and owners can remove unnecessary applications and users from SCADA hosts individually, and restrict access to SCADA hosts, applications and networks as much as possible.
Increase Level of Remediation	SCADA vendors can provide more access control capabilities in their products. They can also provide secure configuration documentation that identifies required applications and services. SCADA owners may be able to test and apply access control capabilities available in host and network equipment to control access as much as possible and provide defense in depth.

Table C-36 lists recommendations on lowering the potential impact of SCADA access controls being exploited.

Table C-36. Lowering risk due to the potential impact of SCADA authorization exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Potential impacts can be reduced by minimizing the privileges SCADA applications run with and the privileges of SCADA application users. • SCADA vendors can design and implement SCADA applications to run using the lowest privileges that are required to accomplish the necessary tasks. • SCADA owners can configure user accounts and run services and applications with the least privileges necessary.
Reduce Environmental Impact	SCADA vendors can compartmentalize critical SCADA functionality so that it can only be given to those who need it, when they need it. They can also remove unnecessary functionality from SCADA applications. SCADA owners may be able to disable unnecessary functionality from SCADA applications and only give critical functionality to those who require it.

C-4.2.5.1 SCADA Vendor Recommendations

SCADA vendors can increase the security of their products by following the principle of least privileges during design and implementation.

Automated analysis may detect many or all possible interfaces that do not require authorization, but manual analysis is required to determine the correctness of custom authorization mechanisms and whether the lack of authorization violates business logic. This can be accomplished with penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. SCADA vendors can specifically request that security assessments of their products test for authorization weaknesses.

Complete documentation and/or automated setup of security features can be provided to allow for quicker, easier, and more consistent implementation of SCADA 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 SCADA. Security features that are inconsistently implemented or provide inconsistent results are considered a risk to reliability and availability of the SCADA in an operational environment.

C-4.2.5.2 SCADA Owner Recommendations

Lock down the host environments as much as possible by individually restricting the privileges granted to user accounts, applications, and services. SCADA integrators and administrators can use the access control capabilities built into the host operating systems and network devices to define customized access control lists. Use a “default deny” policy when defining these ACLs. This means that, by default, access is denied and the protection scheme identifies conditions under which access is permitted.

Apply the principle of least privilege when assigning access rights to entities in a software system. Communication, user and application permission levels may be restricted to levels that are necessary to maintain function.

C-5. Access to SCADA Applications: Improper Authentication

Authentication is used to enforce access controls. Weak authentication allows access controls to be subverted. SCADA 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 SCADA 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 SCADA functionality. Table C-37 summarizes the security relevant attributes of improper authentication.

Table C-37. Summary of improper authentication to SCADA applications security characteristics.

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

C-5.1 Generic CVSS v2 Score

Many SCADA applications are "Network" accessible. Access to other applications requires an attacker to be logged into the system (such as at a command line or via a desktop session or Web interface). For these applications, "Authentication" is "Single" or "Multiple."

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." The actual impact depends on the application. CVSS v2 metrics for improper authentication to SCADA applications are summarized in Table C-38 using the most common or critical values seen on SCADA. See Section 4.5, "Authentication vulnerabilities," for additional information.

Table C-38. Generic CVSS v2 score for improper authentication to SCADA applications.

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	Proof-of-Concept
Remediation Level	Not Defined
Report Confidence	Not Defined
Temporal Score	9.0
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined

Metric	Value
Base Metric	
Availability Requirement	High
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.5
Overall CVSS Score	**9.5**

C-5.2 Scoring and Reducing the CVSS Risk Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They also may not be able to change the criticality of each system component; however, they can change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-5.2.1 Base Exploitability Metrics

C-5.2.1.1 Related Exploit Range (Access Vector)

The access vector for exploiting SCADA authentication vulnerabilities depends on how the vulnerable authentication mechanism is designed. Authentication mechanisms built for users logged into the associated host fit under the "Local" category. Authentication mechanisms built to allow access over the network can either be vulnerable to attackers that have gained access to the local network "Adjacent Network," or remote attackers "Network," depending on how they are implemented.

Table C-39 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-39. Improper SCADA authentication CVSS v2 Access Vector scenarios.

Metric Value	Improper Authentication Access Vector Scenarios
Local	An SCADA application's authentication vulnerability is only exploitable after an attacker has gained local access to the SCADA host. This requires the attacker to have either physical access to the SCADA host it runs on or a local (shell) account. This means that a user cannot login to the SCADA application over the network. For example, the HMI application does not require a password, or does not support complex passwords. If unauthorized personnel (or an attacker over the Internet) is able to login to the operator's workstation, he can then exploit the authentication vulnerability to gain access to the HMI application.
Adjacent Network	The SCADA application's authentication vulnerability is only exploitable after an attacker has gained local network access.
Network	The SCADA application's authentication vulnerability is remotely exploitable over the network. The SCADA application is designed to allow access over the network, but the authentication mechanism is improperly implemented. For example, • the SCADA application authenticates authorized clients by IP address or hostname

	(which can be spoofed), or • authentication is built into the client component of the SCADA application.

C-5.2.1.2 Access Complexity

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. An authentication vulnerability is lower risk if it can only be exploited when specific conditions exist, in rare configurations, or by easily detected social engineering methods. Table C-40 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-40. Improper SCADA authentication CVSS v2 Access Complexity scenarios.

Metric Value	Improper Authentication Access Complexity Scenarios
High	The authentication method can only be bypassed when specialized conditions exist, for example the SCADA owner must change the default configuration to an insecure method, and this is rarely done in practice.
Medium	The access conditions are somewhat specialized, for example either: • The affected configuration is non-default, and is not commonly configured • Client-side authentication is used, but the attacker must obtain a copy of the client application (or knowledge of the server) before it can be exploited.
Low	Specialized access conditions or extenuating circumstances do not exist, for example either: • The weak configuration is default or used by most SCADA customers • Password gathering can be performed by well-known methods, such as sniffing plain text passwords or replaying hashes off the network.

C-5.2.1.3 Level of Authentication Needed

This metric measures the number of times an attacker must authenticate before they are able to circumvent the SCADA application's authentication. The CVSS scoring guide states, "If the vulnerability exists in an authentication scheme itself (e.g., PAM, Kerberos) or an anonymous service (e.g., public FTP server), the metric should be scored as 'None' because the attacker can exploit the vulnerability without supplying valid credentials."[f]

C-5.2.2 Impact Metrics

Improper authentication can allow unauthorized access to information or functionality provided by SCADA applications. Authentication vulnerabilities in SCADA applications typically have partial confidentiality, integrity, or availability impacts to the SCADA hosts they are installed on. Vulnerabilities that give root-level access to SCADA hosts or devices should be scored with complete loss of confidentiality, integrity, and availability.

C-5.2.2.1 Environmental Security Requirements

Security Requirements are characteristics of the individual SCADA components that host the vulnerable SCADA application. They measure the potential for loss of revenue or life due to loss of

[f] http://www.first.org/cvss/cvss-guide.html

confidentiality, integrity, or availability of the SCADA hosts or devices the vulnerable application is installed on. See Section C-11, “SCADA Environmental Security Requirements,” below.

C-5.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

Improper authentication can allow unauthorized access to SCADA applications. From an SCADA perspective, the most severe impact that can result is unauthorized supervisory control access. In CVSS terms, process confidentiality, integrity and availability may be compromised.

C-5.2.2.3 Reducing Potential Impact

Collateral damage potential can be reduced by removing the vulnerable application from systems where comprise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

C-5.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution is the percentage of hosts in the SCADA environment that have the vulnerable application installed.

C-5.2.4 Temporal Score Metrics

C-5.2.4.1 Availability of Exploit (Exploitability)

Vendors and owners cannot change the availability of exploits, but they can prioritize the mitigation of vulnerabilities that have exploit code or tools published. Vendors and owners should have their systems assessed for vulnerabilities. Authentication mechanisms should be evaluated by the assessment teams. Table C-41 shows potential authentication exploit scenarios.

Table C-41. Improper SCADA authentication CVSS v2 Exploitability scenarios.

Metric Value	Improper Authentication Exploit Availability Scenarios
Unproven	No exploit code is available, or an exploit is entirely theoretical. For example, the code written to validate identity in the SCADA application can be bypassed, but no exploit code has been published for it.
Proof-of-Concept	Proof-of-concept exploit code or an attack demonstration that is not practical for most systems has been released.
Functional	Functional exploit code is available. The code works in most situations where the vulnerability exists. For example, GUI MitM tools exist and can be used to capture unprotected transmission of passwords.
High	No exploit is required. For example, no authentication is implemented or configured.

C-5.2.4.2 Type of Fix Available (Remediation Level)

The remediation level depends on whether the vendor has released a patch or if the vulnerability can be mitigated with available security products. Table C-42 shows potential scenarios.

Table C-42. Improper SCADA authentication CVSS v2 Remediation Level scenarios.

Metric Value	Improper Authentication Remediation Level Scenarios
Official Fix	There is a patch available that fixes the vulnerability in the application's authentication mechanism.
Temporary Fix	The SCADA vendor has released or recommended a temporary mitigation while the official fix is being developed.
Workaround	Potential workaround mitigations include: • The SCADA owner is able to use an encryption product to tunnel authentication traffic or securely store application credentials. • The SCADA owner is able to prevent authentication via the vulnerable method.
Unavailable	The vendor has not provided a solution and no workaround mitigation can be applied. For example, potential mitigations such as encryption or firewall rules cannot be applied without breaking the system functionality or are not effective when configured for system requirements.

C-5.2.4.3 Level of Verification that Vulnerability Exists (Report Confidence)

Report confidence is "Unconfirmed" or "Uncorroborated" until the vendor acknowledges the vulnerability or exploit code is released. The lack of authentication or ability to create complex passwords can be easily validated by the application owners. Storage and transmission of plain text passwords can be easily verified as well.

C-5.2.5 Improper Authentication Recommendation Summary

Vendors can audit their applications for strong authentication methods and remediate any weak or vulnerable implementations. Owners can securely configure their authentication settings and regularly audit their passwords and system settings.

Table C-43 lists specific recommendations for lowering the probability of unauthorized access to an SCADA application. See Section 4.5.3, "Recommendations and resources" for additional information.

Table C-43. Lowering risk due to the chance of SCADA authentication mechanism exploitation.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors and owners can restrict access to SCADA applications: • SCADA vendors can implement and test proven identity proofing mechanisms, making sure to correctly enforce access controls on the server side. • SCADA owners can restrict access to SCADA hosts and networks. They may be able to configure user accounts in SCADA applications and restrict access to network applications to the minimum IP range, ports, and users necessary.
Increase Authentication	SCADA vendors can implement strong authentication methods. SCADA owners can require the use of passwords in applications that support them.
Decrease Target Distribution	SCADA owners can uninstall unnecessary applications from each SCADA host.
Increase Level of Remediation	SCADA vendors can assess authentication mechanisms in SCADA applications and remediate weaknesses.

	SCADA owners may be able to change default passwords, enforce strong passwords, and securely store credentials.

Table C-44 lists recommendations on lowering the potential impact of SCADA application identity proofing being exploited.

Table C-44. Lowering risk due to the potential impact of SCADA authentication exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Potential impacts can be reduced by minimizing the privileges SCADA applications run with and the privileges of SCADA application users. • SCADA vendors can design and implement SCADA applications to run using the lowest privileges that are required to accomplish the necessary tasks. • SCADA owners can configure user accounts and run services and applications with the least privileges necessary.
Reduce Environmental Impact	SCADA vendors can remove unnecessary functionality from SCADA applications. SCADA owners may be able to disable unnecessary functionality from SCADA applications.

C-5.2.5.1 SCADA Vendor Recommendations

SCADA vendors can reduce this score for their applications by implementing and supporting strong authentication methods. Verify that any security checks that are performed on the client side are duplicated on the server side. Avoid implementing custom authentication routines; use authentication capabilities provided by the surrounding framework, operating system, or environment if possible. Authentication vulnerabilities are easier to avoid when using a vetted library or framework. For example, consider using libraries with authentication capabilities such as OpenSSL.

C-5.2.5.2 SCADA Owner Recommendations

SCADA owners can reduce the risk of unauthorized access to their SCADA applications by utilizing available authentication methods and preventing access to vulnerable methods. For example, owners can enforce strong password policies for applications that support them. Owners can configure applications to not allow insecure features and prevent access through host and network firewalls.

C-6. SCADA Host Access: Buffer Overflows in SCADA Services

Buffer overflow vulnerabilities are the most common type of input validation weaknesses reported on SCADA 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. SCADA 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 C-45 summarizes the security relevant attributes of buffer overflow vulnerabilities and their potential risk to SCADAs.

Table C-45. Summary of buffer overflow characteristics.

Buffer Overflows in SCADA Services	
Possible Consequences	Compromise of SCADA hosts and applications. May allow DoS, code execution, data loss, or security bypass.
SCADA Impact	Unauthorized access to SCADA 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
SCADA Prevalence	Widespread

C-6.1 Generic CVSS v2 Score

In general, the server application vulnerabilities are network accessible, or remotely exploitable. SCADA protocols generally do not require authentication, therefore the Access Vector is “Network” and “Authentication” is “None.”

The Access Complexity is “Low” because no additional access or specialized circumstances need to exist for the exploit to be successful. The CVSS v2 guide states, “This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. For example, consider a buffer overflow in an Internet service: once the target system is located, the attacker can launch an exploit at will.”[g]

Successful exploitation of buffer overflow vulnerabilities in network applications may allow the attacker to execute arbitrary code with the privileges of the exploited application. SCADA network applications are often executed with administrative (system) privileges. Each of the Impact metrics is set to “Complete” because of the possibility of a complete system compromise.

Known exploits do not currently exist for SCADA service vulnerabilities, so Exploitability is “Unproven.” Some SCADA 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 may still exist in SCADA 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 SCADA vendors review and provide feedback before assessment reports are finalized.

Security requirements are dependent on the host functionality and the nature of the SCADA. Full compromise of any SCADA 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

[g] http://www.first.org/cvss/cvss-guide.html

requirements are therefore rated as "Medium," but will range between "Low" and "High" for individual systems and hosts.

A successful compromise of an SCADA 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 SCADA environment are running custom SCADA network applications. If they are exploitable, most of the SCADA is at risk.

The CVSS v2 values for this generic vulnerability are listed in Table C-46.

Table C-46. CVSS v2 score for buffer overflow vulnerabilities in SCADA protocol server applications.

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	Proof-of-Concept	Unproven
Remediation Level	Not Defined	Not Defined
Report Confidence	Not Defined	Not Defined
Temporal Score	9.0	7.0
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	9.5	8.5
Total Score	**9.5**	**8.5**

C-6.2 Scoring and Reducing the CVSS Risk Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They also may not be able to change the criticality of each system component. They can change the available mitigations as well as the host permissions that can be acquired via compromise of the SCADA service vulnerabilities in their environment.

C-6.2.1 Base Exploitability Metrics

C-6.2.1.1 Related Exploit Range (Access Vector)

Service vulnerabilities are remotely exploitable; therefore, the Access Vector is “Network.”

C-6.2.1.2 Access Complexity

This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. Access complexity of a buffer overflow in a network service depends on whether additional steps are required after the target system is located before the attacker can launch an exploit. Table C-47 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-47. SCADA service buffer overflow CVSS v2 Access Complexity scenarios.

Metric Value	Buffer Overflow Vulnerabilities in SCADA Services Access Complexity Scenarios
Medium	The access conditions are somewhat specialized. SCADA services often employ weak access control methods such as IP address or host name which can be easily spoofed.
Low	Specialized access conditions are not required for some SCADA services. Once an SCADA host is located, the attacker can launch an exploit at will.

C-6.2.1.3 Level of Authentication Needed

An attacker does not need to authenticate before launching an exploit against an SCADA service. Many SCADA vendors and owners are protecting SCADA communications with IPSec encryption. If IPSec is configured to only allow connections from authenticated IPSec partners, an attacker would have to authenticate to (or compromise) an IPSec partner before launching an exploit against the SCADA service. Table C-48 gives example scenarios that fit each of the Authentication metric values in an SCADA environment.

Table C-48. SCADA service buffer overflow CVSS v2 Authentication scenarios.

Metric Value	SCADA Service Buffer Overflow Authentication Scenarios
Multiple	A user must provide multiple credentials before launching an exploit against an SCADA service. For example, IPSec is configured to only allow connections from authenticated IPSec partners, and two-factor authentication is required to authenticate as a user on all hosts authorized as IPSec partners.
Single	A user must provide credentials before launching an exploit against an SCADA service. For example, IPSec is configured to only allow connections from authenticated IPSec partners, and a password is required to authenticate as a user on all hosts authorized as IPSec partners.
None	An attacker can send network packets to a target service on an SCADA host. The host will process the malicious packet without requiring credentials, which is the typical scenario.

C-6.2.2 Impact Metrics

Successful exploitation of many SCADA service vulnerabilities results in a buffer overflow condition allowing the attacker to execute arbitrary code with the privileges the service is running as. Many times, SCADA services run with administrative (system) privileges.

The impact of a buffer overflow exploit is access to the portion of memory used by the operation for executing programs. Instructions for running the vulnerable service can be altered to include malicious code sent by the attacker. Buffer overflow vulnerabilities are most often used to run services that connect back to the attacker or allow the attacker to connect to the host. The attacker gains the privileges of the service that was exploited on the SCADA host. Other vulnerabilities on the host may then be exploited to escalate privileges.

Each of the Impact metrics is set to “Complete” for the generic common vulnerability because of the possibility of a complete system compromise. If the service is not running with system privileges, the impact values will be lower.

C-6.2.2.1 Confidentiality and Integrity Impacts

The Confidentiality and Integrity impacts depend on the read and write privileges given to the vulnerable service. Many times, SCADA services are given administrative (system) privileges.

Table C-49 gives example scenarios that fit each of the Confidentiality Impact metric values in an SCADA environment. Table C-50 gives these scenarios with respect to the impact on confidentiality.

Table C-49. SCADA service buffer overflow CVSS v2 Confidentiality Impact scenarios.

Metric Value	SCADA Service Buffer Overflow Confidentiality Impact Scenarios
None	The buffer overflow can only be exploited to cause the service to crash; remote code cannot be executed and the attacker is not able to gain access to information on the host. Protections have been built into many new processors, operating systems, and compilers to help protect against buffer overflow attacks. These protections can prevent code execution aimed at gaining access to the host.
Partial	The SCADA service is running with limited permissions. Code executed by overflowing the buffer will run with the permissions of the SCADA service. Information available to the SCADA service could be disclosed to the attacker.
Complete	Vulnerable SCADA services running with root or administrator privileges may be exploited to gain full control of the host. The attacker is able to read all of the system’s data (memory, files, etc.)

Table C-50. SCADA service buffer overflow CVSS v2 Integrity Impact scenarios.

Metric Value	SCADA Service Buffer Overflow Integrity Impact Scenarios
None	The buffer overflow can only be exploited to cause the service to crash; remote code cannot be executed and the attacker is not able to alter information on the host. Protections have been built into many new processors, operating systems, and compilers to help protect against buffer overflow attacks. These protections can prevent code execution aimed at gaining access to the host.
Partial	The SCADA service is running with limited permissions. Code executed by overflowing the buffer will run with the permissions of the SCADA service. Information available to the SCADA service could be disclosed to the attacker.
Complete	Vulnerable SCADA services running with root or administrator privileges may be exploited to

	gain full control of the host. The attacker is able to read all of the system's data (memory, files, etc.)

The impact to confidentiality and integrity can be eliminated by preventing the ability to overwrite memory and execute the inserted commands. Protections have been built into processors, operating systems, and compilers. Attackers have found methods for circumventing many of these controls, but these protections can increase the difficulty of exploiting vulnerabilities caused by insecure coding practices or limit their impact. The potential impacts are configuration dependent. The security of an SCADA application will vary depending on the compiler options used when the particular instance was compiled and the hardware and operating system it is installed on.

SCADA vendors should audit their source code for the usage of functions that do not truncate input to the size of the buffer. These unsafe function calls should be replaced with their safe counterparts, which take the buffer size as a parameter to prevent buffer overruns. All code that processes input data should be assessed and modified to validate all input data. This includes numeric input data used to determine the size of the input or position in an array. Secure coding resources are widely available and should be used and enforced by SCADA vendors. Developers should take advantage of their compilers' security options and resolve all warnings.

C-6.2.2.2 Availability Impact

Buffer overflow vulnerabilities always have the ability to impact availability. Overflowing a buffer results in overwriting of other program data or instructions. This situation typically leads to a crash of the program (SCADA service), unless the data fed into the buffer was specially designed to overwrite memory with valid instructions. Typical vulnerability discovery and exploit development methods feed invalid input to a service or application, using a crash as the indicator of vulnerability.

The final exploit code may not crash the service to reduce the chance of being detected, but a crash is still possible.

New "safe" C functions have been designed to crash a process when a buffer overflow is attempted. This prevents unauthorized code execution with the assumption that availability is low priority. SCADA developers should implement input validation to services that require high availability.

Table C-51 gives example scenarios that fit each of the Availability metric values in an SCADA environment. The CVSS definition refers to the availability of the information resources on a system.

Table C-51. SCADA service buffer overflow CVSS v2 Availability Impact scenarios.

Metric Value	SCADA Service Buffer Overflow Availability Impact Scenarios
Partial	There is reduced performance or interruptions in SCADA service availability.
Complete	The service can be rendered unavailable. The attacker is able to crash the service as quickly as it can be restarted. For example, the service may require a reboot to restart or stop restarting after a specified number of crashes.

C-6.2.2.3 SCADA Environmental Security Requirements

SCADA security requirements measure the importance of the affected component to the SCADA product or installed system, measured in terms of confidentiality, integrity, and availability. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices that run servers for SCADA protocols.

SCADA have higher integrity and availability requirements than confidentiality requirements for status and command data. See Section C-11, “SCADA Environmental Security Requirements,” below.

C-6.2.2.4 Organization Specific Potential for Loss (Collateral Damage Potential)

Exploitation of buffer overflow vulnerabilities in SCADA services can lead to unauthorized access to SCADA components, many times from a different security zone. Buffer overflow vulnerabilities can be exploited to crash the SCADA service as well. This can make critical SCADA communications unavailable until the services are restarted, which requires a system reboot in some cases. An attacker may also repeatedly crash a service until the attack can be blocked.

C-6.2.2.5 Reducing Potential Impact

Collateral damage potential can be reduced by reducing the privileges SCADA services run as and the available information and functionality on the hosts on which they run.

C-6.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution can be reduced by removing the vulnerable application from as many systems as possible. SCADA vendors can re-evaluate the need for SCADA services on each component. SCADA owners can make sure that they are only running the services needed for their environment.

C-6.2.4 Temporal Score Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They may be able to change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-6.2.4.1 Availability of Exploit (Exploitability)

Vendors and owners cannot change the availability of exploits, but they can prioritize the mitigation of vulnerabilities that have exploit code or tools published. Vendors and owners should search for vulnerabilities and exploits available for the services they support and use. Table C-52 shows the potential scenarios.

Known exploits specifically for SCADA service buffer overflow vulnerabilities are not currently available. However, buffer overflow exploit techniques are well known.

Table C-52. SCADA service buffer overflow CVSS v2 Exploitability metric value.

Metric Value	SCADA Service Buffer Overflow Exploit Availability Scenarios
Proof-of-Concept	Instructions on writing exploit code have been widely available for many years. Fuzzers and instructions for other detection methods are widely available. Shell code that can be used to gain access to the host is available for all processors and can be inserted into code for a specific buffer overflow exploit.

C-6.2.4.2 Type of Fix Available (Remediation Level)

If the vendor has patched the vulnerability, the Remediation Level is “Official-Fix” and the Report Confidence is “Confirmed” for that vulnerability. If security assessments identify buffer overflows in

SCADA services, the Remediation Level is “Unavailable” until a patch is released. If the assessment was ordered by an SCADA owner, or announced by another non-official source, a Workaround may be developed that prevents the vulnerability from being exploited. Table C-53 shows the potential scenarios.

Table C-53. SCADA service buffer overflow CVSS v2 Remediation Level scenarios.

Metric Value	SCADA Service Buffer Overflow Remediation Level Scenarios
Official Fix	There is a patch available from the SCADA vendor.
Temporary Fix	There is an official but temporary fix available. Many SCADA vendors have addressed service vulnerabilities with encryption. Encryption can reduce the exposure to attack, but does not fix the vulnerabilities in the protocol server applications.
Workaround	Many SCADA users are employing as many defense in depth measures as possible to reduce their risk of compromise through vulnerabilities in the SCADA products they use. The following are examples of workaround mitigations to reduce the probability of SCADA service exploitation: • Network security zones • Limiting access with firewall rules • Network traffic monitoring • Operating system memory protection features • Encryption.
Unavailable	There is either no solution available or it is impossible to apply.

C-6.2.4.3 Level of Verification that Vulnerability Exists (Report Confidence)

Confidence is “Unconfirmed” or “Uncorroborated” until the vendor acknowledges the vulnerability or exploit code is released. NSTB assessments have identified and reported buffer overflow and other vulnerabilities in SCADA services to the responsible vendors. Some buffer overflow vulnerabilities in SCADA services have been reported by other security researchers and can be found in public vulnerability databases.

C-6.2.5 SCADA Services Recommendation Summary

SCADA design requires that some protocols be allowed through firewalls to support external data collection and sharing. 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. These protocols and services should have top priority for vulnerability remediation activities.

Section 4.1.1, “Design and implement secure code ,” contains additional guidance and references for avoiding and remediating this type of programming errors. Table C-54 lists specific recommendations for lowering the probability of a buffer overflow being exploited.

Table C-54. Lowering risk due to the probability of a buffer overflow being exploited.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors can document system communication channels to aid owners in locking down, restricting access, and monitoring their systems. SCADA vendors can provide a list of required services for each SCADA component, including: • Required or potential communication partners • Direction of communication initiation • Port ranges • Valid message formats. SCADA owners can configure host and network firewall and IDS rules to filter SCADA network traffic as much as possible. Owners can monitor their own traffic to identify normal traffic patterns and create IDS rules that alert on atypical traffic (without affecting SCADA operations).
Increase Level of Remediation	SCADA vendors can assess network code for vulnerabilities and remediate them using basic secure programming principles. SCADA vendors and owners can use options built into compilers and operating systems for mitigating some types of buffer overflow vulnerabilities.
Decrease Exploitability	Compiler and operating system level buffer protection technologies generally do not remediate the risk, but increase the difficulty of exploit.

Table C-55 lists recommendations on lowering the potential impact of a buffer overflow being exploited.

Table C-55. Lowering risk due to the potential impact of buffer overflow exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Potential impacts can be reduced by compiler and operating system level buffer protections. Many times these features prevent code execution by exiting the application, reducing the potential impact only to DoS (availability). SCADA vendors can design and implement SCADA services to run using the lowest privileges that are required to accomplish the necessary tasks. SCADA owners can validate that SCADA services are running with the least privileges necessary.
Reduce Environmental Impact	SCADA vendors can separate functionality, or remove unnecessary functionality from SCADA services. SCADA owners may potentially be able to remove vulnerable services from critical components.

C-6.2.5.1 SCADA Vendor Recommendations

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

When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to protocol specifications. Programmers should use a whitelist of acceptable inputs that strictly conform to specifications and reject any input that does not strictly conform to specifications, or transform it into something that does conform.

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 SCADA 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.

SCADA vendors need to perform code reviews of all SCADA 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.

C-6.2.5.2 SCADA Owner Recommendations

SCADA owners can use defense in depth methods to reduce the risk from service vulnerabilities.

SCADA owners can reduce the risk from vulnerable services by limiting access to and monitoring them. They can also monitor vulnerability databases and vendor alerts for identified vulnerabilities in software and other components of the installed SCADA. For this type of vulnerability, search for vulnerabilities in services installed on the SCADA, such as ICCP, DNP, and vendor-proprietary services.

C-7. Supervisory Control Access: SCADA Data and Command Message Manipulation and Injection

SCADA 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. SCADA and network programming skills are needed for this attack. The SCADA 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.

SCADA 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 C-56 summarizes the security relevant attributes of SCADA network protocol channel vulnerabilities and their potential risks to the SCADA.

Table C-56. Summary of vulnerable SCADA network protocols' security characteristics.

SCADA 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.
SCADA Impact	Unauthorized access to network level supervisory control functionalities

SCADA Data and Command Message Manipulation and Injection	
Vulnerable Components	SCADA communication channels, potentially between security zones
Ease of Detection	Medium to High
Attacker Awareness	High
Remediation Cost	High
SCADA Prevalence	Widespread

C-7.1 Generic CVSS v2 Score

These attacks bypass authentication, therefore “Authentication” is “None.”The Access Complexity is “Medium”.

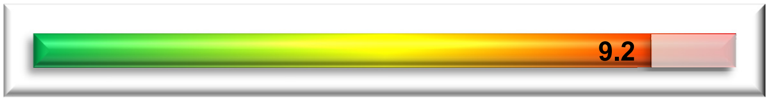
MitM attack tools exist, and protocol analyzers are available for some control protocols, so Exploitability is “proof-of-concept.” Network traffic can be encrypted in some cases, so the Remediation Level metric varies between “Unavailable” and “Temporary-Fix.” The Remediation Level metric for this generic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

Security requirements are dependent on the protocol functionality and the nature of the SCADA. Interception of SCADA 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 SCADA environment are communicating using SCADA network protocols. If they are vulnerable to MitM attack or spoofing, the SCADA 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 C-57. See Section 4.3.2, “SCADA data and command message communication protocols ,” for more information on this vulnerability.

Table C-57. Generic CVSS v2 score for SCADA protocol vulnerabilities.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	Medium
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	9.3
Temporal Metric	
Exploitability	Proof-of-concept
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	8.4
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	High
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	9.2
Overall CVSS Score	**9.2**
	

C-7.2 Scoring and Reducing the CVSS Risk Metrics

C-7.2.1 Base Exploitability Metrics

C-7.2.1.1 Related Exploit Range (Access Vector)

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. In general, MitM attacks require local network access, or “Adjacent Network.” Table C-58 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-58. SCADA protocol traffic manipulation and injection CVSS v2 Access Vector scenarios.

Metric Value	SCADA Data and Command Message Manipulation and Injection Access Vector Scenarios
Adjacent Network	MitM altering of SCADA network traffic requires access to the SCADA network traffic. MitM attacks require access to one of the networks that the traffic passes through.
Network	Spoofing of SCADA status and command messages can be executed by a remote attacker.

C-7.2.1.2 Access Complexity

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The Access Complexity is “Low” because no additional access or specialized circumstances need to exist for the exploit to be successful. SCADA protocols generally do not require strong authentication or integrity checks. Access complexity is generally rated “Low” for network sniffing attacks. Table C-59 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-59. SCADA protocol traffic manipulation and injection CVSS v2 Access Complexity.

Metric Value	SCADA Data and Command Message Manipulation and Injection Access Vector Scenarios
Medium - High	The access conditions are specialized (High) or somewhat specialized (Medium). For example either: • An SCADA network may only be connected to other networks for short periods of time, when necessary. • The associated service or network traffic is not always available. An attacker must wait for the service to be running to intercept or spoof messages to it. Information gathering is required for intelligently altering or initiating command messages to the physical system. Even if the attacker understands the SCADA protocol, knowledge of the physical system is required to interpret and intelligently set individual values.
Low	SCADAs are commonly configured to regularly report status data. They are designed for high availability and generally accept command messages at any time. Special conditions are not normally required to sniff, alter, or spoof SCADA traffic.

C-7.2.1.3 Level of Authentication Needed

An attacker does not need to authenticate before launching an exploit against an SCADA service. Many SCADA vendors and owners are protecting SCADA communications with IPSec encryption. If IPSec is configured to only allow connections from authenticated IPSec partners, an attacker would have to authenticate to (or compromise) an IPSec partner before spoofing the SCADA protocol. Table C-60 gives example scenarios that fit each of the Authentication metric values in an SCADA environment.

Table C-60. SCADA protocol traffic manipulation and injection CVSS v2 Authentication scenarios.

Metric Value	SCADA Data and Command Message Manipulation and Injection Access Vector Scenarios
Multiple	A user must provide multiple credentials before spoofing SCADA messages. For example, IPSec is configured to only allow connections from authenticated IPSec partners, and two-factor authentication is required to authenticate as a user on all hosts authorized as IPSec partners.
Single	A user must provide credentials before spoofing SCADA messages. For example, IPSec is configured to only allow connections from authenticated IPSec partners, and a password is required to authenticate as a user on all hosts authorized as IPSec partners. SCADA services often employ weak authentication methods such as IP address or host name, which can be easily spoofed.
None	An attacker can intercept/manipulate/spoof network packets to a target service on an SCADA host. The host will process the altered packet without requiring authentication, which is the typical scenario.

C-7.2.2 Base Impact Metrics

MitM attacks allow viewing, altering, and dropping of messages. Table C-61 shows the Impact metric values for SCADA network protocol channel vulnerabilities.

Table C-61. SCADA protocol traffic manipulation and injection CVSS v2 Impact metric values.

Metric	Metric Value	Rational
Confidentiality	**Partial**	Clear-text transmission of SCADA messages allows disclosure of SCADA data.
Integrity	**Partial**	Altering of SCADA messages compromises SCADA data integrity.
Availability	**Partial**	MitM techniques can make SCADA data unavailable.

SCADA security requirements depend on the importance of the affected component to the SCADA product or installed system. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices that use vulnerable SCADA communication protocols. In general, SCADA have higher integrity and availability requirements than confidentiality requirements for status and command data.

C-7.2.2.1 Organization Specific Potential for Loss (Collateral Damage Potential)

From a SCADA perspective, the most severe impact that can result is unauthorized supervisory control access. In CVSS terms, process confidentiality, integrity and availability may be compromised.

Collateral damage potential can be reduced by removing these services from systems where compromise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

C-7.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution is the percentage of SCADA hosts and devices that use the vulnerable protocol.

C-7.2.4 Temporal Score Metrics

C-7.2.4.1 Exploitability and Report Confidence

Vendors and owners cannot change the availability of exploits. MitM attack tools exist, and protocol analyzers are available for some control protocols, but at least some reverse engineering of the captured SCADA traffic is required, so Exploitability is “Proof-of-concept.”

Report Confidence is scored as “Confirmed,” in general, because all SCADA vendors review and provide feedback before assessment reports are finalized. All protocols tested by the INL NSTB assessment teams have been vulnerable to manipulation and injection attacks due to improper identity proofing and integrity checks, but other SCADA protocols may be scored differently. Note that the SCADA protocols are vulnerable to manipulation and injection attacks; however, some SCADA vendors and owners have used encryption to mitigate this vulnerability.

Table C-62 shows the Exploit Availability and Report Confidence values for SCADA protocol manipulation and injection vulnerabilities.

Table C-62. SCADA protocol traffic manipulation and injection CVSS v2 Exploitability scenarios.

Metric	Metric Value	Rational
Exploit Availability	**Proof of Concept**	At least some reverse engineering of the captured SCADA traffic is required to understand or manipulate SCADA data.
Level of Verification that Vulnerability Exists	**Unconfirmed - Confirmed**	SCADA vendors and owners can easily monitor their own network traffic for clear-text messages and test MitM attacks with available tools.

C-7.2.4.2 Type of Fix Available (Remediation Level)

Some SCADA vendors and owners have used encryption to mitigate this vulnerability. This is classified as a Workaround or Temporary Fix because it does not fix the SCADA protocol. SCADA integrators and administrators must be able to implement encryption of these protocols on their systems while maintaining system functionality. SCADA vendors have not been able to implement encryption on test systems sent for assessment. Similar to situations in the real world, encryption was disabled to restore functionality. SCADA administrators may disable encryption for trouble-shooting purposes and not restore it again. Encryption prevents many trouble shooting and intrusion detection techniques such as from monitoring traffic. Due to high availability requirements, encryption is not an ideal mitigation, but rather a temporary fix. SCADA protocols need to be redesigned for better security and reliability.

Table C-63 shows potential Remediation Level scenarios for SCADA protocols.

Table C-63. SCADA protocol traffic manipulation and injection CVSS v2 Remediation Level.

Metric Value	Remediation Level Scenarios
Official Fix	The SCADA protocol is redesigned to incorporate strong authentication and integrity checks by default.
Temporary Fix	The SCADA vendor has released or recommended a temporary mitigation while the protocol is being redesigned. For example, the vendor has provided instructions for encrypting SCADA traffic.
Workaround	SCADA owners have implemented their own mitigation. For example, an SCADA administrator added identity proofing and integrity checks on top of SCADA traffic using a third-party product, such as IPSec. Instructions are shared with other users at users' group meetings.
Unavailable	The vendor has not provided a solution and no workaround mitigation can be applied. For example, potential mitigations such as encryption or firewall rules cannot be applied without breaking the system functionality or are not effective when configured for system requirements.

C-7.2.5 SCADA Protocol Recommendation Summary

Table C-64 lists specific recommendations for lowering the probability of SCADA protocols being exploited.

Table C-64. Lowering risk of SCADA protocols being exploited.

Recommendation	Potential Methods
Increase Level of	SCADA vendors can redesign SCADA communication protocols to implement strong

Remediation	authentication and integrity checks. SCADA owners can monitor their network for MitM attacks and unusual messages.

Table C-65 lists recommendations on lowering the potential impact of SCADA protocols being exploited.

Table C-65. Lowering potential impact of SCADA protocol exploitation.

Recommendation	Potential Methods
Reduce Environmental Impact	SCADA vendors can compartmentalize functionality, or remove unnecessary functionality from SCADA services.

C-7.2.5.1 SCADA Vendor Recommendations

The system design needs to implement strong authentication into SCADA 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 SCADA process devices.

Physical access to the controller should be required for configuration and firmware updates. Ensuring that updates occur in this environment will help prevent possible exploitation over the network. Authentication and data integrity checks should also be used to protect against unauthorized physical access and manipulation of firmware files.

SCADA vendors can change the Remediation Level metric by providing a temporary fix while developing a complete solution. Some SCADA vendors recommend using a third-party encryption product as a temporary fix. In some cases this qualifies as unavailable because it affects system functionality or is inadequately supported by the vendor.

C-7.2.5.2 SCADA Owner Recommendations

Defenses that reduce exposure to this vulnerability are network access and content filtering rules. IDS monitoring can detect 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 may still send control messages if he has gained access to the host that encrypts the packet.

C-8. Data Historian Access: SQL Injection

A Historian server is used for data archiving and analysis and is typically an integral part of an SCADA. 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. SCADA 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 SCADA 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*[10] report, SQL injection is the second-most widespread and critical programming error. Table C-66 summarizes the security relevant attributes of SQL injection vulnerabilities and their potential risk to SCADAs.

Table C-66. Summary of SQL injection characteristics.

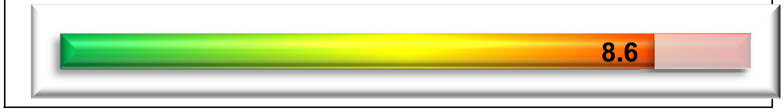
SQL Injection	
Possible Consequences	Data loss: Unauthorized read or write access to the database Security bypass: DoS of the database service or unauthorized access to the associated host
SCADA Impact	Historical data exposure, loss, or manipulation Attack path into the SCADA network
Vulnerable Components	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
SCADA Prevalence	Common

C-8.1 Generic CVSS v2 Score

If the Historian and other SCADA 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 SCADA database may result in a significant loss of revenue or productivity. CVSS v2 metrics for Data Historian SQL injection are summarized in Table C-67 using the most common or critical values seen on an SCADA. See Section 4.4.2, “Database applications and services,” for general database security recommendations and references.

Table C-67. Generic CVSS v2 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	Proof-of-Concept
Remediation Level	Not Defined
Report Confidence	Not Defined
Temporal Score	6.8
Environmental Metrics	
Collateral Damage Potential	Medium-High
Target Distribution	Not Defined
Availability Requirement	High
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	8.6
Overall CVSS Score	**8.6**
	

C-8.2 Scoring and Reducing the CVSS Risk Metrics

SCADA vendors and owners cannot change the publicity of vulnerabilities and associated attack techniques. They also may not be able to change the criticality of each system component. They may be able to change the available mitigations as well as the host, network, and SCADA access that can be acquired via compromise of the vulnerability in their environment.

C-8.2.1 Base Exploitability Metrics

C-8.2.1.1 Related Exploit Range (Access Vector)

The access vector metric reflects how the vulnerable database-backed application is accessed. This measures the difficulty of the attack vector. Table C-68 gives example scenarios that fit potential SQL injection Access Vector metric values.

Table C-68. SCADA SQL injection CVSS v2 Access Vector scenarios.

Metric Value	SQL Injection Access Vector Scenarios
Local	The vulnerable database application requires local access to the database server.
Network	The vulnerable database application is available over the network.

C-8.2.1.2 Access Complexity

Access complexity depends on the number of systems and users that have access to the vulnerable database application. Table C-69 gives example scenarios that fit each of the Access Complexity metric values in an SCADA environment.

Table C-69. SCADA SQL injection CVSS v2 Access Complexity scenarios.

Metric Value	SQL Injection Access Complexity Scenarios
High	The vulnerable database application is only accessible to a few trusted users.
Medium	The vulnerable database application is only accessible to an authorized group.
Low	The vulnerable database application is accessible to a wide range of systems and users.

C-8.2.1.3 Level of Authentication Needed

This metric measures the number of times an attacker must authenticate after gaining access to the system defined by the access vector. This metric should be set to the number of times an attacker must provide credentials to a database application before exploitation. All values are possible.

C-8.2.2 Impact Metrics

Vulnerabilities that give root-level access should be scored with complete loss of confidentiality, integrity, and availability, while vulnerabilities that give user-level access should be scored with only partial loss of confidentiality, integrity, and availability.

Successful injection can cause information disclosure as well as the ability to add or modify data in the database. Impact metric values for SQL injection are “Partial” or “Complete,” depending on the access gained by the attacker. When malicious SQL content is executed by the database, it can lead to arbitrary queries being executed, causing disclosure of information, unauthorized access, privilege escalation, and possibly system compromise.

C-8.2.2.1 Environmental Security Requirements

These metrics enable SCADA vendors and owners to customize the CVSS score depending on the importance of the affected database server to the SCADA product or installed system, measured in terms of confidentiality, integrity, and availability. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the affected system.

The Historian database may have a high integrity requirement. The SCADA real-time database typically has high integrity and availability requirements. Other SCADA databases, such as an operator database, may also have high integrity and availability requirements.

C-8.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

A Historian server is used for data archiving and analysis and is typically an integral part of an SCADA. Compromise of the Historian may provide an attacker with a point of entry into the SCADA network.

If the Historian and other SCADA 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.

Collateral damage potential can be reduced by providing a replicated database for business applications.

C-8.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution is generally low for SQL injection because database servers make up less than 25% of a typical SCADA environment.

C-8.2.4 Temporal Score Metrics

C-8.2.4.1 Availability of Exploit (Exploitability)

Exploitability of individual vulnerabilities can range between “Unproven” and “High.” Attack methods are well known, so the exploitability of SQL injection vulnerabilities is “Proof-of-concept.”

SQL injection is a standard attack pattern that requires a low skill or knowledge level. “It is fairly simple for someone with basic SQL knowledge to perform SQL injection, in general. In certain instances, however, specific knowledge of the database employed may be required.”[h]

C-8.2.4.2 Type of Fix Available (Remediation Level)

SCADA vendors can lower the probability of an SQL injection attack by assessing database backed applications and remediating vulnerabilities with strong input validation. Table C-70 shows potential scenarios.

Table C-70. SCADA SQL injection CVSS v2 Remediation Level scenarios.

Metric Value	Remediation Level Scenarios
Official Fix	A complete vendor solution is available. The SCADA vendor has made a patch available.
Unavailable	There is either no solution available or it is impossible to apply.

C-8.2.5 SQL Injection Recommendation Summary

Table C-71 lists specific recommendations for lowering the probability of SQL injection.

Table C-71. Lowering risk due to the chance of SQL injection exploitation in SCADA applications.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors and owners can restrict access to SCADA database application.
Increase Authentication	SCADA vendors can implement strong authentication methods to enforce access controls. SCADA owners can to enforce access controls by implementing strict password policies in applications that support them.
Decrease Target Distribution	SCADA vendors and owners can remove database applications SCADA hosts that do not require them.
Increase Level of Remediation	SCADA vendors can validate input to database applications.

[h] http://capec.mitre.org/data/definitions/66.html

Table C-72 lists recommendations on lowering the potential impact of SQL injection being exploited.

Table C-72. Lowering risk due to the potential impact of SQL injection exploitation.

Recommendation	Potential Methods
Reduce Base Impact	Developers can decrease the potential impact by following the principle of least privilege.
Reduce Environmental Impact	SCADA owners can reduce the potential impact by replicating databases on a DMZ.

C-8.2.5.1 SCADA Vendor Recommendations

SCADA vendors can reduce the number of SQL injection vulnerabilities by using vetted libraries or frameworks that do not allow them 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.

Developers can decrease the potential impact by following the principle of least privilege. For example, use the strictest permissions possible on all database objects, such as execute-only for stored procedures.

C-8.2.5.2 SCADA Owner Recommendations

SCADA owners can reduce the potential impact by replicating databases on a DMZ. If an attacker finds and exploits an SQL injection, he will simply own another server in the DMZ rather than jumping directly into the SCADA network.

Administrators of SCADAs with Web servers can 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 SCADA historian and Web servers to attack from the Web client's network.

C-9. SCADA Host Access: Use of Standard IT Protocols with Clear-text Authentication

Insecure services developed for IT systems have been adopted for use in SCADA for common IT functionality. Although more secure alternatives exist for most of these services, active unused or obsolete services still exist in many SCADAs. 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 insecure protocols and services to connect to the SCADA hosts creates a high-risk access path into the system. This is a significant vulnerability because it allows unauthorized remote access to SCADA hosts and the functionality allowed to the compromised account. Table C-73 summarizes the security relevant attributes of the use of clear-text authentication protocols and their potential risk to SCADAs. See Section 4.3.1, "Common IT protocols in SCADA systems," for more information.

Table C-73. Summary of clear-text authentication protocols' security characteristics.

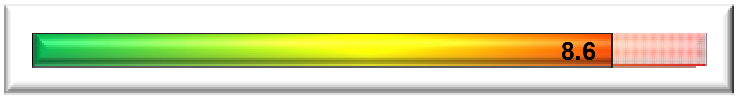
Use of Standard IT Protocols with Clear-text Authentication	
Possible Consequences	Lack of identity proofing
SCADA Impact	Unauthorized access to SCADA components: Possible unauthorized remote access to hosts with privileges to any functionality granted to the compromised remote user.
Vulnerable Components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

C-9.1 Generic CVSS v2 Score

CVSS v2 metrics for the use of clear-text authentication protocols on SCADAs are summarized in Table C-74 using the most common or critical values seen on SCADA. The actual impact depends on the privileges of the account whose credentials were stolen.

Table C-74. Generic CVSS v2 score for the use of clear-text authentication protocols on SCADAs.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	High
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	7.6
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	7.2
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	Medium
Integrity Requirement	High
Confidentiality Requirement	High
Environmental Score	8.6
Overall CVSS Score	**8.6**

Metric	Value
	

C-9.2 Scoring and Reducing the CVSS Risk Metrics

Some SCADA applications still utilize insecure IT protocols between SCADA hosts. Some SCADA operators and administrators still use these protocols to remotely access SCADA hosts.

C-9.2.1 Base Exploitability Metrics

C-9.2.1.1 Related Exploit Range (Access Vector)

Protocols that transmit passwords in clear text allow remote attackers to obtain authentication credentials by sniffing the network. Table C-75 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-75. Clear-text IT authentication CVSS v2 Access Vector scenarios.

Metric Value	Use of Standard IT Protocols with Clear-text Authentication Access Vector Scenarios
Adjacent Network	If these protocols are only used between hosts on the same local network (i.e., an SCADA security zone), an attacker would require access to that network to capture usernames and passwords as they are transmitted in plain text.
Network	Connections using clear-text authentication protocols in and out of SCADA networks can be sniffed by remote attackers to gather credentials that can be used to log into SCADA hosts.

C-9.2.1.2 Access Complexity and Level of Authentication Needed

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The Access Complexity is “High” because this vulnerability is not exploitable at the attacker's whim. There is an additional layer of complexity because the attacker must wait for a user to login using a clear-text authentication protocol.

This vulnerability allows the capture of SCADA host account user names and passwords. An attacker does not need to provide credentials to sniff plain-text network traffic.

Table C-76 shows the Access Complexity and Authentication metric values.

Table C-76. Clear-text IT authentication CVSS v2 Access Complexity and Authentication values.

Metric	Metric Value	Rational
Access Complexity	High	The access conditions are somewhat specialized: • The attacker must wait for a user to login • The network must be vulnerable to MitM attacks (commonly the case)
Authentication Needed	None	An attacker does not need authenticate to a system to sniff plain-text network traffic.

C-9.2.2 Impact Metrics

Clear-text messages can be viewed by anyone able to gain access to the network traffic. This confidentiality vulnerability discloses authentication information. An attacker can then use these credentials to login. The potential impact depends on the privileges of the compromised account. If an administrative user's credentials are captured, then a full system compromise is possible. Since the most harmful case must be considered, each of the three Impact metrics is set to "Complete." Owners can adjust these values to "Partial" if administrative users are not allowed to login remotely using clear-text protocols. Table C-77 shows the Impact metric values for the use of standard it protocols with clear-text authentication.

Table C-77 Clear-text IT authentication CVSS v2 Impact metric values.

Metric	Metric Value	Use of Standard IT Protocols with Clear-text Authentication Impacts
Confidentiality	Partial - Complete	The use of plain-text authentication protocols allows disclosure of SCADA host account credentials (as well as other data sent). The attacker can then log in and view all data visible to the compromised account.
Integrity	Partial - Complete	The attacker can log in using the stolen credentials and alter any data the compromised account has permission to edit.
Availability	Partial - Complete	The attacker can delete any data or stop any services and applications the compromised account is able to.

C-9.2.2.1 SCADA Environmental Security Requirements

These metrics enable SCADA vendors and owners to customize the CVSS score depending on the importance of the vulnerable components to the SCADA product or installed system, measured in terms of confidentiality, integrity, and availability. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices that run servers for insecure protocols. See Section C-11, "SCADA Environmental Security Requirements," below.

C-9.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

The use of insecure protocols and services to connect to the SCADA hosts creates a high-risk access path into the system. This is a significant vulnerability because it allows unauthorized remote access to SCADA hosts and the functionality allowed to the compromised account.

Collateral damage potential can be reduced by removing these services from systems where comprise could lead to unacceptable loss of revenue, safety, or productivity (if possible).

C-9.2.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution is the percentage of SCADA hosts that have servers for plain-text authentication protocols installed.

Target Distribution can be reduced by removing services for clear-text authentication protocols.

C-9.2.3 Temporal Score Metrics

C-9.2.3.1 Exploitability and Report Confidence

Vendors and owners cannot change the availability of exploits. GUI MitM tools exist and can be used to capture unprotected transmission of passwords. The availability of plain-text protocol exploits is "Functional." Table C-78 shows the CVSS v2 Exploitability metric values.

Table C-78. Clear-text IT authentication CVSS v2 Exploitability metric values.

Metric	Metric Value	Description
Exploit Availability	**Functional**	Functional exploit code is available. The code works in most situations where the vulnerability exists. For example, GUI MitM tools exist and can be used to capture unprotected transmission of passwords.
Report Confidence	**Confirmed**	Clear-text transmission of credentials is a well known vulnerability. SCADA vendors and owners can assess their hosts for these services and monitor their own network traffic for their use.

C-9.2.3.2 Type of Fix Available (Remediation Level)

The remediation level depends on whether the vulnerable protocols can be replaced or removed from the SCADA. In some cases, the SCADA owner cannot remove the dependence on insecure protocols. The SCADA vendor must alter the SCADA code that calls these services.

Some SCADA devices do not support secure protocols for remote administration. SCADA vendors should include secure remote login services on their devices. SCADA owners can implement workaround mitigations for their vulnerable devices by directly connecting them to a server capable of secure authentication. Remote device administration can then be proxied through the server, and the clear-text credentials are only sent on the direct connection between the proxy and SCADA device.

Table C-79 lists potential scenarios for each of the CVSS v2 Remediation Level metric values.

Table C-79. Clear-text IT authentication CVSS v2 Remediation Level scenarios.

Metric Value	Use of Standard IT Protocols with Clear-text Authentication Remediation Level Scenarios
Official Fix	There is a patch available that removes the dependence on insecure protocols.
Temporary Fix	The SCADA vendor has released or recommended a temporary mitigation while the official fix is being developed.
Workaround	Potential workaround mitigations include: • The SCADA owner is able to use an encryption product to tunnel authentication traffic. • The SCADA owner is able to isolate the clear-text traffic to a single network and limit access to the extent possible.
Unavailable	The vendor has not provided a solution and no workaround mitigation can be applied. For example, potential mitigations such as encryption or firewall rules cannot be applied without breaking the system functionality or are not effective when configured for system requirements.

C-9.2.4 Clear-text Protocol Recommendation Summary

Users of clear-text authentication protocols should be aware of more secure remote access and file transfer solutions that are available.

Insecure versions of common IT services should be replaced where possible by their secure versions. SCADAs 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, in most cases they can be replaced with their secure counterparts. 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).

Table C-80 lists specific recommendations for lowering the probability of a clear-text authentication protocol being exploited.

Table C-80. Lowering risk due to the probability of a clear-text authentication protocol being exploited.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors can minimize the use of insecure IT protocols in their products. SCADA owners can minimize the use of insecure IT protocols. These protocols should only be necessary on devices that must be remotely accessed, but do not support secure protocols.
Decrease Target Distribution	SCADA vendors and owners can uninstall or disable these protocols on SCADA hosts that do not require them for SCADA functionality.
Increase Level of Remediation	SCADA vendors can assess their products for the use of IT protocols that transmit credentials in clear text (or use other weak authentication mechanisms) and replace them with secure alternatives. They should securely configure them by default and provide secure configuration documentation to their customers. SCADA owners can monitor their network for the use of insecure protocols and test the removal or replacement of the associated services on backup or test systems. Owners can find their own work-around solutions. For example, they can proxy remote administration of SCADA devices through a server that supports encryption and is directly connected to the device.

Table C-81 lists recommendations on lowering the potential impact of a clear-text authentication protocol being exploited.

Table C-81. Lowering risk due to the potential impact of clear-text authentication protocols.

Recommendation	Potential Methods
Reduce Base Impact	SCADA vendors and owners can minimize the privileges of users that are allowed to use these protocols.
Reduce Environmental Impact	SCADA vendors and owners can separate functionality, or remove unnecessary functionality from SCADA hosts and user accounts.

C-9.2.4.1 SCADA Vendor Recommendations

SCADA vendors and customers should follow IT security practices and use the current secure versions of common protocols. In some cases, the SCADA owner cannot remove the dependence on insecure protocols. The SCADA vendor must alter the SCADA code that calls these services.

Some SCADA devices do not support secure protocols for remote administration. SCADA vendors should include secure remote login services on their devices.

C-9.2.4.2 SCADA Owner Recommendations

SCADA owners 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 SCADA whenever possible. Communications between security zones should be secured as much as possible.

SCADA owners can implement workaround mitigations for their vulnerable devices by directly connecting them to a server capable of secure authentication. Remote device administration can then be proxied through the server, and the clear-text credentials are only sent on the direct connection between the proxy and SCADA device.

C-10. SCADA Credentials Gathering: Unprotected Transport of SCADA Application Credentials

The difference between this vulnerability and use of clear-text authentication protocols in Section C-9, "SCADA Host Access: 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 attacker is able to capture SCADA application credentials, he can then log into the SCADA application and gain access to the associated SCADA functionality. This may include controlling the physical process, altering data, or reconfiguring SCADA devices.

This is a significant vulnerability because it allows unauthorized remote access to SCADA functionality, possibly the HMI application (control functionality). Table C-82 summarizes the relevant security attributes of transmitting SCADA application credentials across the network in clear text.

Table C-82. Unprotected transport of SCADA application credentials summary.

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

C-10.1 Generic CVSS v2 Score

These attacks bypass authentication; therefore, “Authentication” is “None.”

The Access Complexity is “High” because the attacker must wait for a user to login to the SCADA application.

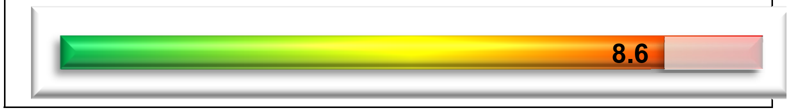
Each of the Impact metrics is set to “Complete.” The actual impact depends on the application and privileges of the account whose credentials were stolen.

User credentials require high confidentiality. SCADA system data and functionality require integrity and availability.

A successful compromise of an SCADA host 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 unprotected transport of SCADA application credentials are summarized in Table C-83 using the most common or critical values seen on SCADA. See Section 4.3.2.1, “33Protect SCADA authentication credentials during transmission” for more information.

Table C-83. Generic CVSS v2 score for unprotected transport of SCADA application credentials.

Metric	Value
Base Metric	
Access Vector	Network
Access Complexity	High
Authentication	None
Confidentiality Impact	Complete
Integrity Impact	Complete
Availability Impact	Complete
Base Score	7.6
Temporal Metric	
Exploitability	Functional Exploit Exists
Remediation Level	Not Defined
Report Confidence	Confirmed
Temporal Score	7.2
Environmental Metrics	
Collateral Damage Potential	High
Target Distribution	Not Defined
Availability Requirement	High
Integrity Requirement	High
Confidentiality Requirement	Medium
Environmental Score	8.6
Overall CVSS Score	**8.6**
	

C-10.2 Scoring and Reducing the CVSS Risk Metrics

C-10.2.1 Base Exploitability Metrics

C-10.2.1.1 Related Exploit Range (Access Vector)

Clear-text transmission of SCADA credentials allows remote attackers to access SCADA applications after gathering the required credentials by sniffing them off the network.

Table C-84 gives example scenarios that fit each of the Access Vector metric values in an SCADA environment.

Table C-84. Unprotected transport of SCADA credentials CVSS v2 Access Vector scenarios.

Metric Value	Clear-text Transmission of SCADA Application Credentials Access Vector Scenarios
Adjacent Network	If the application is accessed from the same local network (i.e., an SCADA security zone), an attacker would require access to that network to capture usernames and passwords as they are transmitted in plain text.
Network	Authentication traffic to SCADA applications from an external network can be sniffed by remote attackers to gather credentials that can be used to log into the applications themselves.

C-10.2.1.2 Access Complexity and Level of Authentication Needed

Access Complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The Access Complexity is “High” because this vulnerability is not exploitable at the attacker’s whim. The attacker must wait for a user to login to the SCADA application. He must also obtain a copy of the client application to use the stolen credentials.

This vulnerability may allow the capture of SCADA application credentials. An attacker does not need to authenticate to any additional system to sniff plain-text network traffic.

Table C-85 shows the Access Complexity and Authentication metric values in an SCADA environment.

Table C-85. Unprotected transport of credentials Access Complexity and Authentication values.

Metric	Metric Value	Rational
Access Complexity	**High**	The access conditions are somewhat specialized: • The attacker must wait for a user to login • The network must be vulnerable to MitM attacks (commonly the case).
Authentication Needed	**None**	An attacker does not need to provide credentials to sniff plain-text network traffic.

C-10.2.2 Impact Metrics

Clear-text transmission of SCADA credentials can allow unauthorized access to information or functionality provided by SCADA applications. If the attacker is able to capture SCADA application credentials, he can then log into the SCADA application and gain access to the associated SCADA functionality. This may include controlling the physical process, altering data, or reconfiguring SCADA devices.

From an SCADA perspective, the most severe impact that can result is unauthorized supervisory control access. In CVSS terms, process confidentiality, integrity, and availability may be compromised.

Table C-86 shows the Impact metric values for SCADA applications that use clear-text authentication.

Table C-86. Unprotected transport of SCADA credentials CVSS v2 Impact metric values.

Metric	Metric Value	Rational
Confidentiality	**Partial - Complete**	Clear-text transmission of SCADA credentials allows disclosure of SCADA application account credentials. The attacker can then log in to the SCADA application and view all data available to the compromised account.
Integrity	**Partial - Complete**	The attacker can log in using the stolen credentials and perform any SCADA functions for which that user has access.
Availability	**Partial - Complete**	The attacker can impact host or SCADA availability if that functionality is provided to the compromised account by the SCADA application.

C-10.2.2.1 Environmental Security Requirements

These metrics enable SCADA vendors and owners to customize the CVSS score depending on the importance of the vulnerable components to the SCADA product or installed system, measured in terms of confidentiality, integrity, and availability. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA hosts or devices have the vulnerable application installed on them. See Section C-11, “SCADA Environmental Security Requirements,” below.

C-10.2.2.2 Organization Specific Potential for Loss (Collateral Damage Potential)

If the attacker is able to capture SCADA application credentials, he can then log into the SCADA application and gain access to the associated SCADA functionality. This may include controlling the physical process, altering data, or reconfiguring SCADA devices.

This is a significant vulnerability because it allows unauthorized remote access to SCADA functionality, possibly the HMI application (control functionality).

C-10.2.3 Percentage of Vulnerable Systems (Target Distribution)

Target Distribution depends on the percentage of SCADA components on which the vulnerable applications are installed.

C-10.2.4 Temporal Score Metrics

These metrics describe elements about the vulnerability that change over time.

C-10.2.4.1 Exploitability and Report Confidence

Vendors and owners cannot change the availability of exploits. GUI MitM tools exist and can be used to capture unprotected transmission of passwords. The availability of plain-text protocol exploits is “Functional.” Table C-87 shows the CVSS v2 Exploitability and Report Confidence metric values for the unprotected transport of credentials.

Table C-87. Unprotected transport of credentials CVSS v2 Exploitability and Report Confidence values.

Metric	Metric Value	Rational
Exploit Availability	**Functional**	Functional exploit code is available. The code works in most situations where the vulnerability exists. For example, GUI MitM tools exist and can be used to capture unprotected transmission of passwords.
Level of Verification that Vulnerability Exists	**Unconfirmed - Confirmed**	Clear-text transmission of credentials can be identified or validated by capturing the network traffic as a user authenticates to the application over the network. SCADA vendors and owners can easily confirm whether passwords are transmitted in clear text.

C-10.2.4.2 Type of Fix Available (Remediation Level)

SCADA vendors can encrypt or hash application credentials in a way that prevents them from being captured and replayed. SCADA owners may be able to secure SCADA application connections using third-party encryption solutions. Table C-88 lists example scenarios that fit the potential CVSS Remediation Level metric values.

Table C-88. Unprotected transport of SCADA credentials CVSS v2 Remediation Level scenarios.

Metric Value	Clear-text Transmission of SCADA Application Credentials Remediation Level Scenarios
Official Fix	There is a patch available that securely transmits application credentials.
Temporary Fix	The SCADA vendor has released or recommended a temporary mitigation while the official fix is being developed.
Workaround	Potential workaround mitigations include: • The SCADA owner is able to use an encryption product to tunnel authentication traffic. • The SCADA owner is able to isolate the clear-text traffic to a single network and limit access to the extent possible.
Unavailable	The vendor has not provided a solution and no workaround mitigation can be applied. For example, potential mitigations such as encryption or firewall rules cannot be applied without breaking the system functionality or are not effective when configured for system requirements.

C-10.2.5 Clear-text Transmission of SCADA Application Credentials Recommendation Summary

Table C-89 lists specific recommendations for lowering the probability of the unprotected transport of SCADA application credentials being exploited.

Table C-89. Lowering risk of unprotected transport of SCADA application credentials being exploited.

Recommendation	Potential Methods
Increase Access Complexity	SCADA vendors can minimize the unprotected transportation of credentials in their products. SCADA owners can minimize the use of SCADA applications that do not transmit credentials securely.
Decrease Target Distribution	SCADA owners can uninstall vulnerable applications from SCADA hosts that do not require them.

Recommendation	Potential Methods
Increase Level of Remediation	SCADA vendors can assess their server applications and use vetted mechanisms for securely transmitting credentials over the network. SCADA owners can monitor their network for clear-text transmission of credentials. SCADA owners may be able to secure SCADA application connections using third-party encryption solutions.

Table C-90 lists recommendations on lowering the potential impact of unprotected transport of SCADA application credentials being exploited.

Table C-90. Lowering potential impact of the unprotected transport of SCADA application credentials.

Recommendation	Potential Methods
Reduce Base Impact	SCADA vendors can compartmentalize application account types and functionality. SCADA owners can grant the minimum privileges necessary for each user.
Reduce Environmental Impact	SCADA vendors can separate functionality, or remove unnecessary functionality from SCADA applications.

C-10.2.5.1 SCADA Vendor Recommendations

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 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.

C-10.2.5.2 SCADA Owner Recommendations

SCADA owners may be able to secure SCADA application connections using third-party encryption solutions.

C-11. SCADA Environmental Security Requirements

Environmental impact metrics enable SCADA vendors and owners to customize the CVSS score depending on the importance of the affected component to the SCADA product or installed system, measured in terms of confidentiality, integrity, and availability. They measure the potential for loss of revenue or life due to loss of confidentiality, integrity, or availability of the SCADA host or device. This is based on the host’s security objectives due to its role as part of the SCADA.

The security requirements of the data and applications installed on an SCADA host or device determine that component’s security requirements. In general, SCADA have higher integrity and availability requirements than confidentiality requirements for status data and command functions.

C-11.1 SCADA Component Confidentiality Requirements

Confidentiality is generally not as important as integrity or availability for an SCADA, but it may be for individual SCADA components. Table C-91 gives example scenarios that fit each of the Confidentiality Requirement metric values in an SCADA environment.

SCADA vendors and owners can lower the confidentiality requirement by minimizing the confidential information on SCADA hosts.

Table C-91. SCADA scenarios for each of the CVSS v2 Confidentiality Requirement metric values.

Metric Value	SCADA Component Confidentiality Requirement Scenarios
Low	Loss of confidentiality of the information on an SCADA host is likely to have limited effect on the operation of the SCADA or its organization's business interests.
Medium	Disclosure of information on an SCADA host could have a serious adverse effect on the business or safety. For example, data on SCADA hosts could give an attacker the system-specific information needed to intelligently control the SCADA.
High	Depending on the system or process under control and the individual component, disclosure of information on the SCADA host could have a catastrophic adverse effect on the business or safety.

C-11.2 SCADA Component Integrity Requirements

Integrity of SCADA data is generally the highest priority in relation to confidentiality and availability, but it may not be for individual SCADA components. Table C-92 gives example scenarios for the Integrity Requirement metric values in an SCADA environment.

The integrity of data on SCADA hosts is high, but the potential impact is dependent on the nature of the individual system or process.

Table C-92. SCADA scenarios that fit each of the CVSS v2 Integrity Requirement metric values.

Metric Value	SCADA Component Integrity Requirement Scenarios
Low	Loss of data integrity on an SCADA host is likely to have limited effect on the operation of the SCADA or its organization's business interests. For example, the SCADA is not the primary method used to monitor and control the process.
Medium	Manipulation of data on the SCADA host could have a serious adverse effect on the business or safety. For example, inaccurate process data or unauthorized commands may require the plant to throw out a batch of product, but could not endanger lives.
High	Depending on the system or process under control and the individual component, data manipulation on the SCADA host could have a catastrophic adverse effect on the business or safety. Alteration of system data or malicious operation of the physical system may result in economic, environmental, or safety catastrophes.

C-11.3 SCADA Component Availability Requirements

This metric measures the potential for loss of revenue or life due to loss of availability of the SCADA host. This is unique to the host and its security objectives based on its role as part of the SCADA. Table C-93 gives example scenarios that fit each of the Availability Requirement metric values in an SCADA environment.

This depends on the functionality of the SCADA service and host.

Table C-93. SCADA scenarios that fit each of the CVSS v2 Availability Requirement metric values.

Metric Value	SCADA Component Availability Requirement Scenarios
Low	Loss of the data, services, or applications on the affected SCADA component is likely to have limited effect on the operation of the SCADA or its organization's business interests. The service and host are not critical SCADA components that require high availability.
Medium	Loss of the data, services, or applications on the affected SCADA component could have a serious adverse effect on the business or safety. For example, the service or the SCADA components it runs on require high availability, and the loss of either will result in serious consequences.
High	Loss of the data, services, or applications on the affected SCADA component is likely to have catastrophic adverse effect on the business or safety. For example, the service or the SCADA components it runs on are critical SCADA components that require high availability, and the loss of either will result in catastrophic consequences.

C-12. Summary of SCADA Common Vulnerability Evaluation

Common vulnerabilities identified during NSTB vulnerability assessments were evaluated and discussed in this appendix to aid SCADA vendors and owners in assessing, prioritizing, and mitigating vulnerabilities in the systems in which they are responsible. This prioritizes common vulnerability types to help guide vulnerability identification and mitigation activities. Mitigation of risk associated with these 10 vulnerability types should lead to the greatest reduction in the total risk to an SCADA from cybersecurity events. In summary, the following 10 recommendations can be used as a starting point for mitigating the highest risk vulnerabilities commonly identified on SCADAs:

1. **Routinely assess all SCADA components, including operating systems, applications, services, network devices, etc., for published vulnerabilities.**

   Vendors: SCADA vendors can reduce the risk due to published vulnerabilities by delivering new systems without known vulnerabilities and testing patches for third-party products as they are released.

   Owners: Apply patches as quickly as possible. Work with vendor to test and apply patches for all operating systems and software on the SCADA networks. If patches are not available or cannot be applied, SCADA owners can restrict access to and closely monitor vulnerable systems. They can also disable or remove the vulnerable application wherever possible.

2. **Web servers and clients should be assessed and secured, especially those that allow access to the physical system.**

   Vendors: SCADA 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 SCADA Web server. Vendors should prioritize and remediate vulnerabilities as quickly as possible.

   Owners: Minimize access and available functionality of Web servers and clients. Validate that the access controls are configured to restrict all unwanted users and SCADA functionality.

   Administrators of SCADA 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 SCADA historian and Web servers to attack from the Web client's network.

3. **Minimize usage, exposure, and available functionality of remote display protocols.**

   Vendors: Provide secure options for remotely accessing the HMI and secure default configurations. Secure configurations include access restrictions and secure authentication.

   Owners: Configure remote access protocols on SCADA hosts to limit access, require secure authentication, and use a trusted path.

4. **Lock down all applications, hosts, and networks 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. SCADA 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.**

   Vendors: All SCADA system ports and services necessary to support system operation needs to identified and delineated. Document how SCADA system components use the network so that effective firewall and IDS rules can be created. For each SCADA 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.

   Owners: Use good defense in-depth perimeter protections to help prevent access to vulnerable components and communication on SCADA networks. Redesign network layouts to take full advantage of firewalls, VPNs, etc. Create security zones using multiple layers, with the most critical communications occurring in the most secure and reliable layer. Customize IDSs for the SCADA hosts and networks. Restrict SCADA 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).

   Replace insecure versions of common IT services where possible with their secure versions. SCADA 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 SCADA whenever possible. Communications between security zones should be secured as much as possible.

5. **Use proven authentication services when available. Strong authentication and encryption mechanisms should be implemented and strenuously tested.**

   Vendors: Vendors can audit their applications for strong authentication methods and remediate any weak or vulnerable implementations.

   Owners: Owners can securely configure their authentication settings and regularly audit their passwords and system settings.

6. **Remediate vulnerabilities in SCADA services.**

   Vendors: Code can 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.

   Owners: SCADA owners can reduce the risk from vulnerable SCADA services by limiting and monitoring their access.

7. **Redesign SCADA network protocols and the service applications that implement them for security.**

Vendors: Document how the systems use the network so that effective firewall and IDS rules can be created. Create custom protocol parsers for common IDSs so that they can be more effective.

8. **Protect SCADA databases.**

   Vendors: Developers should use vetted libraries or frameworks that do not allow SQL injection and XSS weaknesses to occur or provide constructs that make this weakness easier to avoid. 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.

   Owners: 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.

9. **Use secure protocols to access SCADA components.**

   Vendors: Replace the use of insecure protocols in SCADA code with secure alternatives. This may require a redesign of data sharing between SCADA components.

   Provide secure services and methods for connections to, and remote administration of, SCADA devices.

   Deliver SCADA hosts and devices securely configured by default and provide secure configuration documentation.

   Owners: Uninstall or disable insecure services where possible. Correctly configure services to protect credentials and provide secure authentication. Use secure protocols to connect to SCADA components.

10. **Protect user credentials and make them inaccessible to an attacker. Passwords should be securely encrypted or hashed before being stored or transmitted.**

    Vendors: Implement secure authentication in SCADA applications. Validate that all SCADA application credentials are securely stored and transferred. When using Web applications with SSL, use SSL for the entire session from login to logout, not just for the initial login page.

    Owners: Correctly configure all applications to securely store and transfer credentials.