Recommended Practice Case Study: Cross-Site Scripting

February 2007

Homeland Security

National Cyber Security Division

Control Systems Security Program

ACKNOWLEDGEMENT

This document was developed for the U.S. Department of Homeland Security to provide guidance for control system cyber security. The author team consisted of representatives from the Department of Energy, Idaho National Laboratory.

For additional information or comments please send inquires to the Control Systems Security Program at cssp@hq.dhs.gov.

ABSTRACT

This paper is intended to support and encourage application of recommended practices for control systems security. It describes the details of an information security attack, known as cross-site scripting, that could be used against control systems and explains practices to mitigate this threat. Additional information and resources regarding recommended practices, defense in depth, and other control systems security issues are found on the Control Systems Security Program Recommended Practices Web site, http://csrp.inl.gov/.

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

CONTENTS

Recommended Practice Case Study: Cross-Site Scripting

1. INTRODUCTION

Recent trends in information systems security show a significant increase in Cross-Site Scripting (XSS) vulnerabilities. Due to the convergence of control systems technology and information systems technology, a determined attacker could use knowledge of XSS vulnerabilities to access a control system network.

XSS involves the posting of malicious Web programming instructions to a Web-accessible location contrary to the intentions of location owners. These instructions take advantage of functionality built into Web browsers or other scriptable applications (such as e-mail applications), which view a site or handle Internet navigation causing script execution when the Web site hosting the malicious content is viewed or the malicious link is clicked.

XSS can be simple to carry out; however, a successful effort to attack a control system using this vector would require time and effort in accordance with the skill and motivation of the attacker. The recommendations provided in this document not only mitigate threats of XSS, but also bolster protection against other Web-based attacks such phishing, cross-site request forgery, Trojans, and worms.

1.1 Aims and Objectives

The aim of this document is to provide recommended practice principles for control systems security. Specifically, this document:

- Defines and describes XSS
- Compares the way in which XSS affects control system security and information systems security related to XSS
- Provides an example of how XSS could be used to attack a control system
- Identifies mitigation techniques to secure control systems against XSS.

1.2 Key Terminology

1.2.1 Control Systems

Throughout this document the term control system is used as a generic term to cover all process control, Supervisory Control and Data Acquisition (SCADA), industrial automation, and related safety and monitoring systems.

1.2.2 Recommended Practice

Recommended practice, in the context of this document, is defined as the best of industry practices such as strategies, activities, or approaches, which have been shown to be effective through research, implementation, and evaluation. The recommended practices summarized in this document are intended to be considered only as guidelines. It may not be possible to implement all of these principles for some control systems environments. Where this is the case, asset owners are encouraged to work with control

system vendors, industry information sharing and analysis centers (ISACS), and user groups to identify and implement alternative, yet effective safeguards.

1.2.3 Other Definitions

Active content	Web programming that runs on the client machine
Cookie	Characters passed between a client and server to track a session
Mitigate	To diminish potential consequences
Scripting	Lightweight programming language suitable for Web programming
Session	The period of time in which a user is visiting or using a particular Web site
Vector	Path or method of attack

2. CROSS-SITE SCRIPTING OVERVIEW

XSS is a computer security attack that uses third-party Web resources to run script within the victim's Web browser or scriptable application. This occurs when a browser visits a malicious Web site or clicks a malicious link.

The consequences of an XSS attack begin with access to the cookie passed between the victim and the Web server. This allows an attacker to impersonate the victim to the Web site, and is known as session hijacking. The most dangerous consequences occur when XSS is used to exploit additional vulnerabilities. These vulnerabilities may permit an attacker to not only steal cookies, but also log key strokes, capture screen shots, discover and collect network information, and remotely access and control the victim's machine.

Any Web site or application that employs user input to generate Web pages may be vulnerable to XSS. These vulnerabilities will become more serious if an attacker can gain the assistance (knowing or unknowing) of an insider. In addition, attackers using XSS may gather information about potential victims prior to an attack and use e-mail to target them directly.

2.1 Differences between Control Systems Security and Information Technology Security for Cross-Site Scripting

Table 1. Comparison of XSS attacks on information systems and control systems.

	Information System (IS)	Control System (CS)
Likelihood of extra/intra-organizational communication	IS resources are used for accessing many accounts external to the organization. Also, external parties are likely to access accounts within the organization.	Browser-based human-machine interface (HMI) from any Web-accessible location for control, monitoring, maintenance, and support is common in CS networks.

Possible consequences of successful attack	Unauthorized access to and modification of data on the IS. Dissatisfied customers. Compromised accounts. Infected node on network that could be used to carry out additional attacks.	Unauthorized access to and modification of data regarding processes. Possibility to control process and affect those whom the process serves.
Value of session hijacking	Depends on the context of the hijacked session. High value information could be accessed through sessions going into and out of the IS network.	Depends on the context of the hijacked session. Hijacking a session into the CS network would be very valuable. Some accounts storing valuable information may be accessed from the CS network.

3. ATTACK SCENARIO

The following is a simplified description of how an attacker might carry out an XSS attack against a control system. It is not intended to single out any vendor or product, but is merely demonstrative of possible points of attack.

1. An attacker discovers or learns of a vulnerability in a widely-used Web site utility; for example, software that is used to manage Web discussion forums.[1]

2. The attacker crafts a suitable exploit. This involves choosing the HTML and script the attacker wants to run in the victim's browser. The attacker writes a script that exploits one of numerous known Web browser vulnerabilities. [2, 3, 4] This allows installation of a Trojan keylogger with backdoor functionality.

3. The attacker identifies online SCADA discussion forums using the vulnerable software. A simple Web search, "http://www._____.com/search?hl=en&q=%22powered+by+invision+power+board%22 +scada," returns 906 hits. Another search, "http://www._____.com/search?num=100&hl=en &lr=&as_qdr=all&q=%22powered+by+phpbb%22+scada," returns 22,500 hits.

4. The attacker sorts through the likely forums to find a bulletin board that is not up-to-date on patching. The attacker establishes a membership and posts the malicious message crafted previously to a discussion topic that is generating a lot of traffic and waits.

5. Probable attack paths are shown in Figure 1. Some victims, perhaps SCADA engineers or operations staff read the poisoned posting. To some, nothing happens because scripting is disabled in their browsers. The vast majority of the browsers, however, succumb to the XSS portion of the attack. This gives the attacker access to the cookies exchanged between the browsers and the server hosting the discussion. The cookies alone may be valuable; they allow the attacker to impersonate

---

1. Mitre, *Search Results*, http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=cross-site+scripting, published July 20, 2006, Web page visited July 20, 2006.

2. Secunia, *Microsoft Internet Explorer 6.x*, http://secunia.com/product/11/, published 2006, Web page visited July 20, 2006.

3. US-CERT, *Internet Explorer Update to Disable ADODB.Stream ActiveX Control*, http://www.us-cert.gov/cas/techalerts/TA04-184A.html, published July 02, 2004, Web page visited July 20, 2006.

4. US-CERT, *Microsoft Windows and Internet Explorer Vulnerabilities*, http://www.us-cert.gov/cas/techalerts/TA06-101A.html, published April 11, 2006, Web page visited July 20, 2006

other users on the discussion board. If the attacker is clever, they might use the newly obtained false identity to communicate with other users in attempts to gain more information about their systems. With the victim's cookies, the server believes the attacker is the legitimate user, and allows the attacker to change the victim's password, which permits the attacker to log back in later. Moreover, there is always a chance the site keeps password information in the cookies (although this is not a best practice). If it does, the attacker might assume the victim uses the same password for multiple applications. This password could come in useful for the attacker later.

6. Some of the browsers viewing the poisoned post are not up to date with patches. The exploit written by the attacker takes advantage of this oversight to execute additional instructions of the attacker's choice. Compromised machines contact the attacker's server and download the Trojan. Now the attacker can watch everything that happens on the infected machines, and issue commands to them.

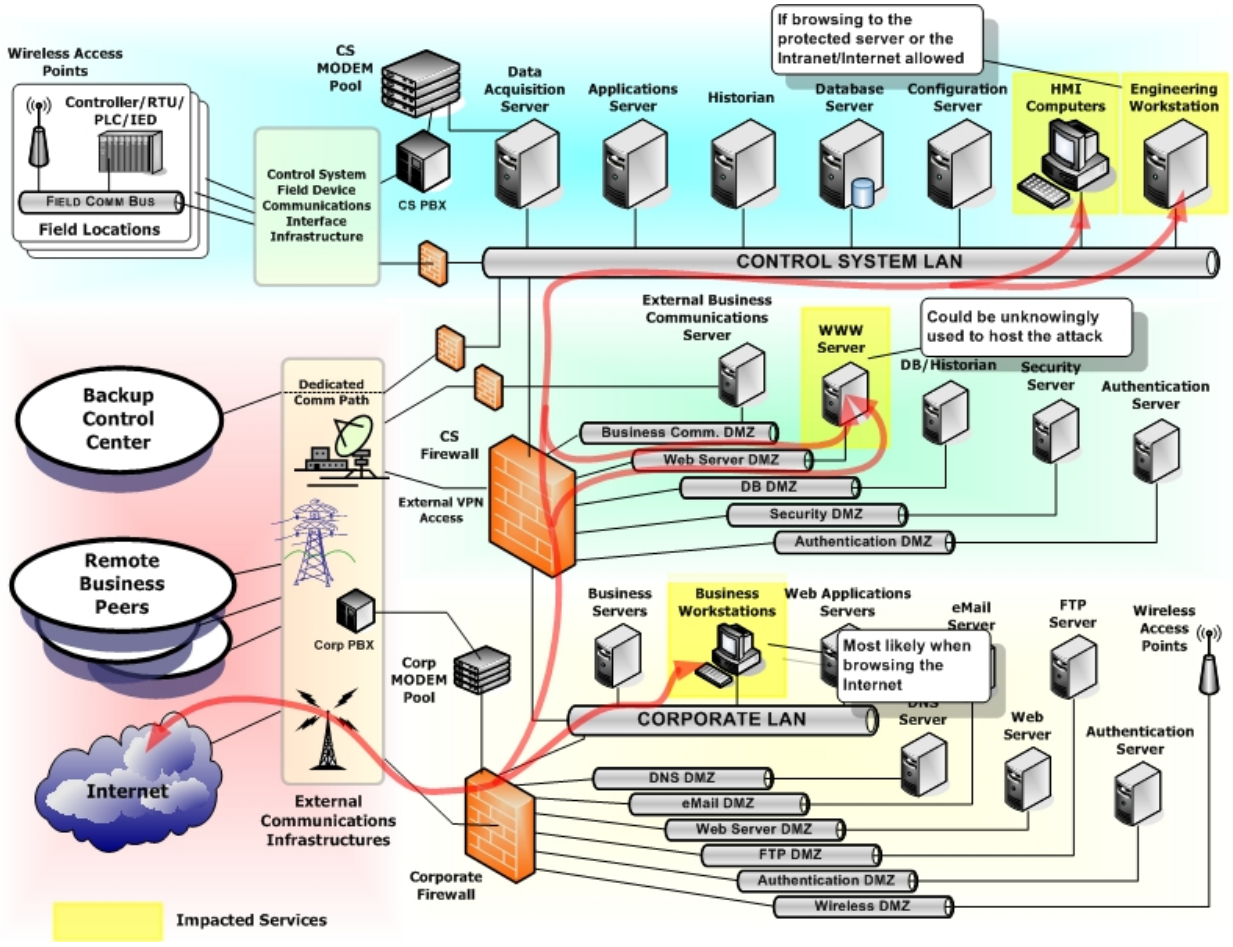

Figure 1: Communications pathways.

4. MITIGATIONS

4.1 Control systems Internet access policy

4.1.1 Overview

A written policy must expressly prohibit individuals and machines from accessing the Internet from the control system network.

4.1.2 Objective

A written policy is intended to form the basis for both user behavior and technical configuration.

4.1.3 Discussion

Policy represents a conscious decision to manage risk. If either the technical or the people countermeasures were to fail, the other would help to prevent the success of possible attacks. In the attack scenario above, the victim came to the exploit. A policy prohibiting this action would limit the direct effects of this attack on the control system. This mitigation—prohibiting Internet access—does not preclude access to appropriate remote locations (such as remote historian databases) over a secure connection through the Internet (i.e., VPN).

4.2 Control systems user awareness and training

4.2.1 Overview

Control system user awareness and training involves ensuring users understand that their actions could adversely affect the system, and that they must know, understand, and follow established policies and recommended practices in order to protect it.

4.2.2 Objective

By assuring user awareness of the XSS threat and training them on security policy, users will be more likely to avoid threat propagation.

4.2.3 Discussion

Awareness is a fundamental countermeasure of information security. Due to the convergence of control systems technology and information systems technology, many control systems operators may not be familiar with the security aspects of IT relating to control systems such as viruses, worms, Trojans, and social engineering. By training operators and making them aware of the threats they face and the defensive behaviors they must exhibit to mitigate those threats, users view themselves as active participants in securing their systems. Awareness and training may include readings, videos, presentations, briefings, brown-bags, and many other forms. In the attack scenario described above, an aware operator would not have accessed the Internet from the control system network. At the same time, an aware administrator would have had technical countermeasures (such as those addressed below) in place to prohibit Internet access from the control system.

4.3 Coordination of security efforts between corporate IT network and control systems network

4.3.1 Overview

Because control systems security depends in part on the security of the IT network, and responsibilities for IT security and control systems security are often separate, these two groups must work together to ensure security of the protected processes.

4.3.2 Objective

Eliminate ambiguity, misunderstanding, and the vulnerabilities that exist as a result of divergent security responsibilities between information systems and control systems.

4.3.3 Discussion

Coordination requires that those responsible for the security of each system gain an understanding of the risk faced by both systems. It includes an accepted scope of responsibility and dialogue about policies and resources that cross system boundaries. Examples of these issues include policies, configurations, and procurement specifications for firewalls, IDSs, Web servers, and machines that access control system information.

4.4 Firewall between the control system network and the information technology network

4.4.1 Overview

A firewall should be configured to allow only specific communications between the IT network (corporate LAN) and the control systems network. The firewall should provide Network Address Translation (NAT) or proxy capability.

4.4.2 Objective

The reasoning for the tight firewall policy between these two networks is to prohibit computers on the control systems network with browser capability from accessing the Internet, even if that capability is enabled—as it must be for some HMI implementations. The firewall should provide NAT so that in case a computer on the corporate LAN is compromised, that machine, under the control of an attacker, is limited in its ability to reconnoiter information about the control system network.

4.4.3 Discussion

The firewall should be configured to prohibit outbound traffic to all destination addresses and ports not on a white list, which is a list of cleared or authorized network addresses and ports. Additionally, the firewall should be configured to prohibit all traffic entering the control system network except for white-listed addresses and ports. With these rules in place, a computer on the corporate network that has been compromised via XSS (or any other means) will not be able to access the control system unless it has specifically been given that privilege. The white list needs to be kept current and the implementation managed with independent audits performed periodically.

The purpose of network address translation (or proxies) is to hide internal control system network details, such as addresses from the external environment. Hence, a computer on the corporate network will not be able to gather information about the control system network to aid in further attack.

4.5 Up-to-date patches

4.5.1 Overview

After proper testing, appropriate patches should promptly be applied to browsers, e-mail readers, operating systems, and any applications known to be susceptible to attack on both the information system and control system networks.

4.5.2 Objective

Appropriate patches or work-arounds for security vulnerabilities should be applied to prevent attackers from using these vulnerabilities in combination with XSS in order to craft full-compromise exploits.

4.5.3 Discussion

An attacker could use XSS in combination with other vulnerabilities in hopes of gaining complete control of a targeted resource on either the IT or control system. To prevent this, patches and work-arounds for all vulnerabilities should be considered and patch decisions made in accordance with vendor advice and organizational risk management policies. Applying patches to control systems without testing and approval by the control system vendor and/or on a test system could degrade system performance and even shut down the system, potentially altering the controlled processes. Applying up-to-date patches on the corporate LAN decreases the possibility of full compromise to machines that could be used as a foothold for attacks against the control system network.

4.6 Web browser and e-mail security

4.6.1 Overview

Web browser security ranges from tightening security settings on Web browsers by disabling scripting and other forms of active content (such as ActiveX) to totally remove browser functionality. These changes should be made to control system machines or machines that access control systems where these settings and software are not needed. E-mail security includes turning off images and removing e-mail applications from the control system in which they are not necessary.

4.6.2 Objective

The objective of this mitigation technique is to incapacitate the functionality on which XSS relies to propagate.

4.6.3 Discussion

Cross-site scripting (and numerous other types of attacks such as Trojans, worms, viruses, and cross-site request forgery) rely on active content that can make the victim’s computer perform tasks without the user’s knowledge. Similarly, XSS attacks in the form of malicious URLs are often delivered via e-mail to the victim. By disabling active content and removing access to one vector through which XSS is spread (e-mail), the likelihood of a successful XSS attack on a control system is reduced.

Browsers on the IT network are typically allowed to run active content because of the functionality it provides—many Web sites do not work correctly without it. As a result, when computers outside the control system firewall must use active content to access the control system network, administrators should consider dedicating those computers solely to control system access (disallowing access to the Internet from that machine). This practice would have prevented the attack in the scenario above from exploiting a control system computer.

Another option is to disable client-side scripting (such as JavaScript), but leave ActiveX controls enabled. This may or may not be acceptable, depending on whether the control system browser relies on client-side scripting. This option also provides a greater level of risk when compared to disabling all unnecessary active content (i.e., JavaScript and ActiveX). This precaution would also have mitigated the attack described above for machines on the corporate and control systems networks.

It should be remembered that many control systems rely on numerous ActiveX components to operate. Hence, testing should occur to assure that disabling of ActiveX components does not impact control system functionality. Following results of successful testing, ActiveX should be disabled in all browsers within the control system network that do not use them for control system functionality. Computers on the corporate LAN should only accept signed ActiveX components from trusted sources.

Finally, browser functionality should be completely disabled or removed from machines on the control system network that do not require it to perform their tasks.

4.7 Secure code

4.7.1 Overview

Secure coding involves techniques that prevent would-be attackers from using program functionality in unintended ways.

4.7.2 Objective

Cross-site scripting is ultimately caused by programming error or omission. By using secure coding techniques, control system vendors and operators who write dynamic Web pages will prevent XSS exploits from being used against control systems.

4.7.3 Discussion

In the scenario above, the vulnerability used by the attacker to target control systems users would never have existed if Web programming recommended/best practices had been followed. These practices require all input that will be displayed as part of a Web page be validated first. This means filtering, removing, or output encoding any html tags or script submitted for incorporation to a dynamically generated Web page.

5. Conclusion

Cross-site scripting is a Web-based attack technique used to gain information from a victim machine or leverage other vulnerabilities for additional attacks. The fact that this technique could be used to specifically target and gain access to control system environments has been described in a detailed hypothetical attack scenario. Similar attacks may be mitigated though the application of the seven practices recommended above. These practices employ policy, people, and technology countermeasures

to protect against XSS and other Web attacks. Critical infrastructure control system asset owners are encouraged to appropriately apply these practices in their operating environments.