Chapter 4
Cross-Site Scripting:
An Overview

Almudena Alcaide Raya
University Carlos III of Madrid, Spain

Jorge Blasco Alis
University Carlos III of Madrid, Spain

Eduardo Galán Herrero
University Carlos III of Madrid, Spain

Agustín Orfila Diaz-Pabón
University Carlos III of Madrid, Spain

ABSTRACT

This chapter is a comprehensive survey on a currently relevant security threat to Web applications: cross-site scripting (XSS). The rise of reported XSS vulnerabilities has made this family of attacks an interesting area for computer security researchers. XSS consists of the injection of code in Web pages. As injected code is client side scripts, it is executed at the user's Web browser. Injected script can perform unauthorized accesses, identity theft, or even cause financial loss to the attack's victim. Main reason for the existence of this kind of vulnerabilities is the incorrect or insufficient handling of the input performed by Web applications. In this chapter, guidelines on proper input treatment for Web developers are offered. Additionally, existing proposals for XSS mitigations are exposed and future lines of research are indicated to interested researchers and developers.

As any other computer program, Web applications are susceptible of including vulnerabilities that may not only disrupt the provided service, but also facilitate private and personal information to an attacker. As these applications are usually public or even publicized, attacks are expected to be more and more frequent, making it necessary to supply the means to provide an adequate level of security in the utilization of Web applications.

DOI: 10.4018/978-1-60960-765-4.ch004

INTRODUCTION

As electronic commerce is becoming a consolidated channel for businesses and costumers to perform their purchases and sales, involved Web applications get to handle an increasing volume of sensitive data concerning customer's personal information and their associated transaction records. This implies that attempts to steal and manipulate that information are expected to be more and more frequent, making it necessary to supply the means to provide an adequate level of security in the utilisation of Web applications.

Among the many threats that affect e-commerce and e-banking websites, cross-site scripting (XSS) is one of the attacks most frequently reported in well-known vulnerability lists (Stock, Williams, & Wichers, 2007) (SecurityFocus, 2009). The high number of occurrences of XSS vulnerabilities makes the problem worthy of a deep and exhaustive study in order to understand what it is, how it works and why it has become such an important issue.

The purpose of a cross-site scripting attack is the injection of arbitrary code into a Web application by an attacker. The injected code is a script or a reference to a script elaborated by the attacker. That script is intended to be executed at the Web browser of the user. The execution of those commands represents a critical security breach of a system as it could allow the execution of commands which would not be executed under normal circumstances. The danger of executing malicious injected code relies on the extremely high damage it can cause. Damage caused by code injection attacks range from quite inoffensive Web defacements to privilege escalation or even exposure, theft or corruption of sensitive information.

Globally, code injection attacks are successful because attacked applications do not validate properly all the input they receive (Su & Wassermann, 2006). This causes the system to accept injected code as correct input which will eventually be executed, as if it was a legitimate code fragment of the Web system. XSS attacks take advantage of vulnerabilities on input validation of a Web application.

There exists another variant of injection attack that also constitutes a serious threat to the security of Web applications: SQL injection. SQL (Standard Query Language) is the language used to retrieve content from a database, as well as modifying contents and structures. The purpose of an SQL Injection Attack is to execute unauthorized SQL queries through the Web application code. In this way, an attacker may be able to manipulate, steal or delete any information stored in the Web application database (from product information, to user accounts or personal data). Unlike XSS, SQL does not need a client executing the injected code, as it is executed by the Web-application itself. SQL attacks will not be subject of study in this chapter as we will be focusing on XSS attacks.

In the vast majority of the reported XSS attacks, the injected script is written in JavaScript. JavaScript is a script language designed to be embedded on HTML documents or be referenced from HTML documents. It executes in the client's side, at the Web browser and it is used to provide static Web pages with some amount of dynamism, improving interfaces and generally enhancing the navigation experience of the user.

Being the most extended mechanism for client-side Web page enriching, it is not a surprising fact that the wide majority of injected scripts are written in JavaScript. In order to prevent potential issues of code injection, JavaScript defines the same-origin policy, which prevents a document or script loaded from one origin from getting or setting properties of a document from another origin. Web Browsers which follow this policy will check that executable code that accesses or manipulates contents of a site does not come from a domain different from the site's domain. Nevertheless, XSS attacks inject JavaScript code directly into the Web application, being invulnerable to the same-origin policy, as once the code