Chapter 14
Countering Cross-Site Scripting in Web-Based Applications

Loye Lynn Ray
University of Maryland University College, USA

ABSTRACT

Today's dynamic web-based applications have become a normal and critical asset to an organizations business. They come with an increase in the number of web vulnerabilities and attacks. These weaknesses allow hackers to focus their attention on attacking this important information source. The most common vulnerability is cross-site scripting (XSS) and one of the Open Web Application Security project (OWASP) top ten web-threats. XSS occurs when a Web-based application allows untrusted information be accepted and sent back to a browser. Also they can execute scripts within a browser that can deface web sites, redirect users to malicious content and hijack browsers. One reason for this problem was the lack of developers understanding the causes of XSS. In this paper, the authors address the causes of XSS and countermeasures to defense against these threats.

INTRODUCTION

Web-based applications play a critical part in our lives and a way to accessing information globally. They provide a vital communications channel between service providers and Internet users (Jovanovic, Kregel & Kirda, 2010). This emerging strategy and change in our lives has set a new paradigm in the way we treat information. These applications provide access to information for millions of people around the world. Improvement to Web-based technology with such things as mobile apps has opened up new avenues and software techniques for developers. Using AJAX, Java Script and other languages help make this possible.

However, inclusion of effective security mechanisms to these applications is a growing concern (Jovanovic, Kregel & Kirda, 2010; Garcia-Alfaro & Navarro-Arribas, 2009). The increase in value to these applications offer an organization reliable security mechanisms to protect user data. Unfortunately, this increase in technology has opened the door to new vulnerabilities and threats. These vulnerabilities result into increasing compromises of sensitive information leading to damages of an organization or

DOI: 10.4018/978-1-5225-3422-8.ch014

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

personal nature (Selvamani, Duraisamy & Kannan, 2010). These applications are frequently targeted by attackers and lead to compromises in sensitive corporate information. It can also cause lost of a company's reputation, financial loss and lawsuits. Thus, web application security plays a big part of its normal operations (Steinke, Tunrea & Kelly, 2011).

Hackers now have focused their attention to attacking this important information source. This method is called cross-site scripting (XSS) and can contaminate a Web site or any user's browser accessing the information on it. It has become one of the leading threats to today's applications (Selvamani, Duraisamy & Kannan, 2010). This type of attack happens when a web-based application is infected with a malicious code such as JavaScript that can be called from a user's browser (Rao, Tejaswini & Preethi, 2012; Bisht & Venkatakrishnan, 2008). It can also use code injection from other languages such as Java, Flash, HTML and VBScript. This can complicate the process of detecting and preventing XSS because of the different way browser's interpret the code (Saha, 2009). Thus, an XSS attack can be a serious threat to any organization doing business on the Web. XSS can cause users to lose respect for a Web site. It has become one of the biggest problems for developers of web applications. Fixing these vulnerabilities in a large system can be challenging and may not be accomplished (Bates, Barth & Jackson, 2010). Even worst, businesses could close down due to loss of revenue. Combating this threat is necessary to preserve the freedom of using the Web and Internet. This paper is divided up into two sections to describe XSS and ways to overcome this threat. The first section explains what XSS is, how it works and different categories. The last section provides detailed countermeasures one can deploy to combat the threat. The result is an understanding what is needed to protect Web sites from this malicious attack.

OVERVIEW OF CROSS-SITE SCRIPTING

Before countering XSS, one needs to understand how they work in detail. Understanding the weaknesses of Web applications and what methods attackers use will be important to combating these threats.

What Is It?

According to OWASP (2013) and Harris (2010), cross-site scripting is an attack that exploits a vulnerability that allows malicious code to be placed in a Web application. Attackers send a malicious code based on a browser script to an unsuspecting user. The data is added to dynamic content and sent to the user without being checked for malicious code or validity. The user's browser activates the malicious code by clicking on a Uniform Resource Locator (URL) or other content being displayed. The attacker can use this attack to create phishing sites and capture Web session for stealing sensitive data (Wurzinger, Platzer, Ludl, Kirda & Kruegel, 2009). This malicious code can also hijack user sessions or get cookies, tokens and other information about users (Steinke, Tundrea & Kelly, 2011; Garcia-Alfaro & Navarro-Arribas, 2009). OWASP (2011) also mentions XSS attackers can use scripts to change Web site content in a Hyper Text Transfer Language (HTML) page.

Cross-site scripting vulnerabilities are considered design and coding errors because of a failure to handle inputs to the Web-based application (Steinke, Tundrea & Kelly, 2011). They can also use the developer's inability to check input and output code to attack a Web application on the server. This incomplete or incorrect input sanitization technique makes web-based application vulnerable to XSS (Wang, Mao & Lee, 2010; Saha, 2009). Through this hole they can inject scripts written in JavaScript to steal a victims

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/countering-cross-site-scripting-in-web-based-applications/188214

Related Content

Simple System Dynamics and Control System Project Models
A. S. White (2014). *Systems and Software Development, Modeling, and Analysis: New Perspectives and Methodologies (pp. 113-133).*
www.irma-international.org/chapter/simple-system-dynamics-and-control-system-project-models/108812

Development of Nonlinear Filtering Algorithms of Digital Half-Tone Images
E. P. Petrov, I. S. Trubin, E. V. Medvedevaand S. M. Smolskiy (2013). *Integrated Models for Information Communication Systems and Networks: Design and Development (pp. 278-304).*
www.irma-international.org/chapter/development-of-nonlinear-filtering-algorithms-of-digital-half-tone-images/79669

Closing Service Quality Gaps Using Dynamic Service Level Agreements
Carlos Mendesand Miguel Mira da Silva (2016). *International Journal of Information System Modeling and Design (pp. 48-71).*
www.irma-international.org/article/closing-service-quality-gaps-using-dynamic-service-level-agreements/162696

Enhancing Martial Arts Safety Through 6G-Enabled Risk Assessment and Alert Systems: A Model for Enterprise Engineering and Semantic Integrity
WenYan Liuand Zhen Wang (2024). *International Journal of Information System Modeling and Design (pp. 1-19).*
www.irma-international.org/article/enhancing-martial-arts-safety-through-6g-enabled-risk-assessment-and-alert-systems/353158

Cellular Automata Based Model for E-Healthcare Data Analysis
Hakam Singhand Yugal Kumar (2019). *International Journal of Information System Modeling and Design (pp. 1-18).*
www.irma-international.org/article/cellular-automata-based-model-for-e-healthcare-data-analysis/234768