



Cross Site Scripting

Enrique Alexis Peinado Rodriguez
Julio Cesar Silva Leon

Gerson André Garrido Mantillo

June 28, 2017

Seguridad Informática - Ciencia de la Computación



Overview

- 1 Introduction
- 2 Antecedentes
- 3 ¿Cómo clasificarlas?
- 4 ¿Soy vulnerable?
- 5 ¿Cómo Protegerse?



Introduction

XSS es un tipo de vulnerabilidad en seguridad informática que se encuentran típicamente en aplicaciones web benignos y de confianza .



Introduction

XSS es un tipo de vulnerabilidad en seguridad informática que se encuentran típicamente en aplicaciones web benignos y de confianza .
Cross-Site Scripting (XSS) se producen cuando:



Introduction

XSS es un tipo de vulnerabilidad en seguridad informática que se encuentran típicamente en aplicaciones web benignos y de confianza . Cross-Site Scripting (XSS) se producen cuando:

- 1 Los datos que se ingresan a una aplicación Web a través de una fuente no fiable, la más frecuente es una solicitud web.
- 2 Los datos se incluyen en el contenido dinámico que se envía a un usuario web sin ser validado para contenido malintencionado.



Introduction

XSS es un tipo de vulnerabilidad en seguridad informática que se encuentran típicamente en aplicaciones web benignos y de confianza . Cross-Site Scripting (XSS) se producen cuando:

- 1 Los datos que se ingresan a una aplicación Web a través de una fuente no fiable, la más frecuente es una solicitud web.
- 2 Los datos se incluyen en el contenido dinámico que se envía a un usuario web sin ser validado para contenido malintencionado.



Antecedentes

Las vulnerabilidades de XSS se han reportado y explotado desde los años 1990. Sitios prominentes afectadas en el pasado incluyen los sitios de redes sociales de Twitter, Facebook, MySpace, YouTube y Orkut.



Facebook





¿Cómo clasificarlas?

¿Cómo clasificarlas?

Categorías de vulnerabilidades

- XSS almacenado (AKA persistente o tipo I)
- XSS reflejado (AKA no persistente o tipo II)
- DOM basado en XSS (AKA tipo-0)



¿Cómo clasificarlas?

¿Cómo clasificarlas?

Categorías de vulnerabilidades

- XSS almacenado (AKA persistente o tipo I)
- XSS reflejado (AKA no persistente o tipo II)
- DOM basado en XSS (AKA tipo-0)

Tipos de scripts entre sitios

- Server XSS
- Client XSS



¿Cómo clasificarlas?

Formas de tipos de ataques

Where untrusted data is used			
Data Persistence	XSS	Server	Client
	Stored	Stored Server XSS	Stored Client XSS
	Reflected	Reflected Server XSS	Reflected Client XSS

- ☐ DOM Based XSS is a subset of Client XSS (where the data source is from the DOM only)
- ☐ Stored vs. Reflected only affects the likelihood of successful attack, not the nature of vulnerability or the most effective defense

Figure: Formas de ataques



¿Soy vulnerable?

¿Como saber si soy vulnerable?

La mejor manera de encontrar fallas es realizar una revisión de seguridad del código.

- buscar todos los lugares donde exista una entrada de una solicitud HTTP pues posiblemente podría hacer su camino en la salida HTML.

Debemos tener en cuenta que una variedad de etiquetas HTML diferentes se pueden utilizar para transmitir un JavaScript malicioso.



¿Soy vulnerable?

¿Qué hacer?

Existen herramientas disponibles que pueden ayudar a escanear un sitio web para estos defectos.

- 1 Scan My Server
- 2 SUCURI
- 3 Qualys SSL Labs, Qualys FreeScan
- 4 Quttera
- 5 Detectify
- 6 SiteGuarding
- 7 Web Inspector
- 8 Acunetix



¿Cómo Protegerse?

Principales defensas en el OWASP

2 XSS Prevention Rules

- 2.1 RULE #0 - Never Insert Untrusted Data Except in Allowed Locations
- 2.2 RULE #1 - HTML Escape Before Inserting Untrusted Data into HTML Element Content
- 2.3 RULE #2 - Attribute Escape Before Inserting Untrusted Data into HTML Common Attributes
- 2.4 RULE #3 - JavaScript Escape Before Inserting Untrusted Data into JavaScript Data Values
 - 2.4.1 RULE #3.1 - HTML escape JSON values in an HTML context and read the data with `JSON.parse`
 - 2.4.1.1 JSON entity encoding
 - 2.4.1.2 HTML entity encoding
- 2.5 RULE #4 - CSS Escape And Strictly Validate Before Inserting Untrusted Data into HTML Style Property Values
- 2.6 RULE #5 - URL Escape Before Inserting Untrusted Data into HTML URL Parameter Values
- 2.7 RULE #6 - Sanitize HTML Markup with a Library Designed for the Job
- 2.8 RULE #7 - Prevent DOM-based XSS
- 2.9 Bonus Rule #1: Use HTTPOnly cookie flag
- 2.10 Bonus Rule #2: Implement Content Security Policy
- 2.11 Bonus Rule #3: Use an Auto-Escaping Template System
- 2.12 Bonus Rule #4: Use the X-XSS-Protection Response Header

Figure:

[https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Chea](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)



Primeras acciones

Desactivar el soporte de HTTP TRACE en todos los servidores web. Un atacante puede robar datos de cookies a través de Javascript, incluso cuando document.cookie está deshabilitado o no es compatible con el cliente.

```
TRACE /<script>foo()</script> HTTP/1.1
Host: test.lab
X-Wing: <script>bar()</script>
\r\n
```

Request

```
HTTP/1.1 200 OK
Date: Tue, 11 May 2010 18:58:16 GMT
Server: MPS
Transfer-Encoding: chunked
Content-Type: message/http
\r\n
5a
TRACE /<script>foo()</script> HTTP/1.1
Host: test.lab
X-Wing: <script>bar()</script>
\r\n
\r\n
0
\r\n
```

Response



Defensas Recomendadas

Servidor XSS

El servidor XSS se debe a la inclusión de datos no confiables en una respuesta HTML. La defensa más fácil y más fuerte contra el servidor XSS en la mayoría de los casos es:

- Codificación de salida del lado del servidor sensible al contexto



Defensas Recomendadas

Servidor XSS

El servidor XSS se debe a la inclusión de datos no confiables en una respuesta HTML. La defensa más fácil y más fuerte contra el servidor XSS en la mayoría de los casos es:

- Codificación de salida del lado del servidor sensible al contexto

Cliente XSS

El cliente XSS se produce cuando se utilizan datos no fiables para actualizar el DOM con una llamada insegura de JavaScript. La defensa más fácil y más fuerte contra el Cliente XSS es:

- Uso de API JavaScript seguras



Credits

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_)
- <http://blog.segu-info.com.ar/2016/02/cross-site-scripting-xss-en.html>
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet