Cross-Site Scripting (XSS)

Introduction

This repository focuses on common web application vulnerabilities, including parameter tampering, Cross-Site Scripting (XSS), and SQL Injection. This section provides an overview of XSS, examples of how it can be exploited, and guidance on how to prevent it.

Cross-Site Scripting (XSS)

Explanation

Cross-Site Scripting (XSS) is a type of web application attack where an attacker injects malicious scripts into content from otherwise trusted websites. These scripts can then be executed in the context of the user's browser, leading to various security issues.

Common Techniques

- Reflected XSS
- Stored XSS
- DOM-based XSS

Example Scenarios

1. **Reflected XSS**
   - Original URL: `http://example.com/search?q=<script>alert('XSS')</script>`
   - Impact: An attacker can inject a script that executes when the user clicks on the link, potentially stealing cookies or other sensitive information.
2. **Stored XSS**
   - Original Comment: `Nice post!`
   - Tampered Comment: `<script>alert('XSS')</script>`
   - Impact: An attacker can inject a script that is stored in the database and executed whenever a user views the comment, potentially leading to data theft or account compromise.

Code Snippets

1. **Reflected XSS**

```
<!-- Original URL -->
<a href="http://example.com/search?q=<script>alert('XSS')</script>">Click here</a>
```

2. **Stored XSS**

```
<!-- Original Comment -->
<div>Nice post!</div>
```

```
<!-- Tampered Comment -->
<div><script>alert('XSS')</script></div>
```

Prevention Techniques

- Validate and sanitize all user inputs on the server side.
- Use Content Security Policy (CSP) to restrict the sources from which scripts can be loaded.
- Encode output to prevent the execution of injected scripts.
- Implement secure coding practices, such as using frameworks that automatically escape user inputs.

Conclusion

Cross-Site Scripting (XSS) is a serious security threat that can lead to unauthorized access and data manipulation. By understanding common techniques and implementing prevention measures, developers can protect their applications from such attacks.

For further reading, refer to the following resources: - OWASP: Cross-Site Scripting (XSS) - OWASP: Content Security Policy (CSP)