LEARNING

XSS

#xss

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: xss

It is an unofficial and free xss ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official xss.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with xss

Remarks

Overview Cross-Site Scripting, commonly referred to as XSS, is a type of web application injection attack in which malicious scripts are injected into trusted websites.

XSS attacks occur when an attacker takes advantage of, or "exploits," a flaw in a web application to send the attacker's payload to the client's browser. These flaws are typically encountered when a web application sends user-input to the browser without validating or encoding it beforehand.

An XSS payload executes within the domain of the "trusted site" and has the potential of accessing that website's cookies, modifying the page's DOM and even abusing the client's browser or extensions.

XSS Types

Though the end result is the same for all XSS attacks (an attacker controlled payload in the server's response), there are three different types of XSS vulnerabilities.

- **Stored XSS** is an attack where the XSS payload is permanently *stored* on the target website, such as in a database. When a client (e.g. victim) loads a page such as a forum board or comment section that loads the payload, it will execute in their browser.
- **Reflected XSS** is an attack where the XSS payload is sent with the request to the server and *reflected* back in the response. These attacks can be triggered from clicking a crafted link, submitting a form, or many other delivery mechanisms.
- **Client Side XSS**, also referred to as **DOM Based XSS**, is an attack that takes place solely in the client's browser (i.e. it's not sent by the response from the server) by manipulating the DOM's environment to force existing trusted scripts on the page to execute the XSS payload.

Examples

Search Results Example

Let's assume we have a search results page that displays a user's search query back to them. The code below is an example of how this could be done in PHP:

```
Results for "<?php echo $_GET['query'] ?>"
```

For this to work, you would access the page with a URL like:

```
https://yoursite.test/search?query=stackoverflow
```

In the response, we get:

```
Results for "stackoverflow"
```

Now we will attempt to inject our payload into the response:

```
https://yoursite.test/search?query=<script>alert(1)</script>
```

And our new response:

```
Results for "<script>alert(1)</script>"
```

We have successfully injected our XSS payload.

Read Getting started with xss online: https://riptutorial.com/xss/topic/10229/getting-started-with-xss

Credits

S. No	Chapters	Contributors
1	Getting started with xss	Community, newfurniturey