Cross-Site Scripting (XSS)

`HTML Injection` vulnerabilities can often be utilized to also perform Cross-Site Scripting (XSS) attacks by injecting `JavaScript` code to be executed on the client-side. Once we can execute code on the victim's machine, we can potentially gain access to the victim's account or even their machine. `XSS` is very similar to `HTML Injection` in practice. However, `XSS` involves the injection of `JavaScript` code to perform more advanced attacks on the client-side, instead of merely injecting HTML code. There are three main types of `XSS`:

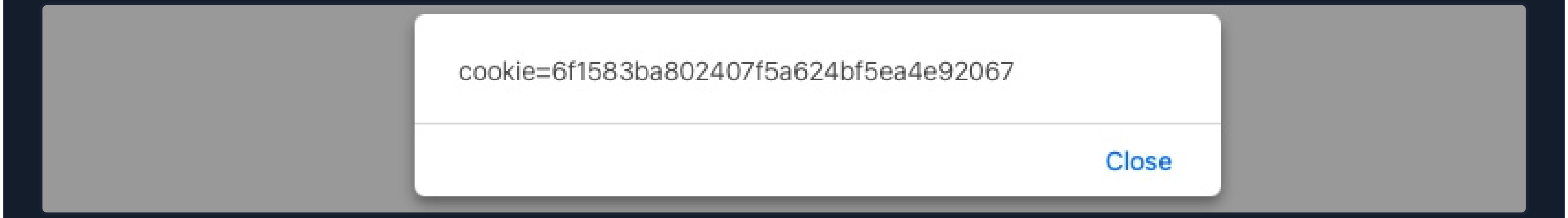
Type	Description
`Reflected XSS`	Occurs when user input is displayed on the page after processing (e.g., search result or error message).
`Stored XSS`	Occurs when user input is stored in the back end database and then displayed upon retrieval (e.g., posts or comments).
`DOM XSS`	Occurs when user input is directly shown in the browser and is written to an `HTML` DOM object (e.g., vulnerable username or page title).

In the example we saw for `HTML Injection`, there was no input sanitization whatsoever. Therefore, it may be possible for the same page to be vulnerable to `XSS` attacks. We can try to inject the following `DOM XSS` `JavaScript` code as a payload, which should show us the cookie value for the current user:

Code: javascript

```javascript
#"><img src=/ onerror=alert(document.cookie)>
```

Once we input our payload and hit `ok`, we see that an alert window pops up with the cookie value in it:



cookie=6f1583ba802407f5a624bf5ea4e92067

This payload is accessing the `HTML` document tree and retrieving the `cookie` object's value. When the browser processes our input, it will be considered a new `DOM`, and our `JavaScript` will be executed, displaying the cookie value back to us in a popup.

An attacker can leverage this to steal cookie sessions and send them to themselves and attempt to use the cookie value to authenticate to the victim's account. The same attack can be used to perform various types of other attacks against a web application's users. `XSS` is a vast topic that will be covered in-depth in later modules.

Start Instance

1 / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: Click here to spawn the target system!

+ 1 Try to use XSS to get the cookie value in the above page

Submit your answer here...

Submit

Hint

Previous Next

Go to Questions

Table of Contents

Introduction to Web Applications

Front End Components

Front End Vulnerabilities

Back End Components

Web Servers

Databases

Development Frameworks & APIs

Back End Vulnerabilities

Common Web Vulnerabilities

Public Vulnerabilities

Next Steps

Next Steps

My Workstation

OFFLINE

▶ Start Instance

1 / 1 spawns left