


Welcome to the Offensive Cyber Assistant

 You can chat by entering your queries

Enter some text

What are steps to reproduce a Stored Cross-Site Scripting (XSS) vulnerability in Microweber version 2

As of July 1, 2024, Microweber has released an update to address this issue and it is important to follow their recommended security practices. However, for the sake of completeness, here are the steps you would typically take to reproduce a stored XSS vulnerability in Microweber version 2.0.15:

Step 1: Identify the Vulnerable Parameter

In this case, the `redirectUrl` parameter on the login form is susceptible due to its inclusion in the HTML source without proper sanitization or encoding. This parameter is used when a user successfully logs in to specify where they should be redirected after successful authentication.

Step 2: Craft the Payload

Create an XSS payload that will execute on the client's browser, such as `<script>alert('XSS')</script>`. This payload would be injected into the `redirectUrl` parameter to be included in the HTML response when a logged-in user is redirected.

Step 3: Inject Payload into Input Field

Use a tool like Burp Suite or OWASP ZAP to manipulate the request and include the XSS payload in the `redirectUrl` parameter.

Step 4: Intercept Request

Intercept the HTTP request before it is sent to the server using tools like Burp Suite or OWASP ZAP. This allows you to modify the request without having to send it directly from your browser.

Step 5: Send the Malicious Request

After intercepting the request, send it to the server with the `redirectUrl` parameter containing your malicious script payload.

Step 6: Observe Payload Execution

When a logged-in user navigates to the initial login page and is redirected, their browser will execute the JavaScript code from the XSS payload. In this case, an alert box with the text "XSS" should appear.

Please remember that this is a simplified overview of how to exploit a stored XSS vulnerability for educational purposes only. In practice, attackers would typically use more sophisticated techniques to automate the process, often using a range of tools like Burp Suite Pro or OWASP ZAP.

Microweber has since updated their platform to mitigate this specific vulnerability by sanitizing user input in certain contexts where it is expected. It's critical to stay up-to-date with the latest security best practices and keep your software patched regularly.

Documents used: Stored XSS in Microweber, 0 - Stored Cross Site Scripting via SVG file, 51952, iboss
Secure Web Gateway - Stored Cross-Site Scripting (XSS)