**Reflected XSS in Zixem**

Reflected cross-site scripting (XSS) occurs when an application receives data in an HTTP request and includes it in the immediate response in an unsafe manner.

For example, consider a website with a search function that receives the search term via a URL parameter:

**https://insecure-website.com/search?term=gift**

The application displays the search term in the response:

**<p>You searched for: gift</p>**

If the application does not safely process the data, an attacker can craft a malicious URL:

**https://insecure-website.com/search?term=<script>/\*+Bad+stuff+here...+\*/</script>**

This URL would result in the following response:

**<p>You searched for: <script>/\* Bad stuff here... \*/</script></p>**

When another user visits the attacker's URL, the malicious script executes in their browser within the context of their session with the application.

**Impact of Reflected XSS Attacks**

Reflected XSS allows attackers to fully compromise a user's session by executing malicious scripts in their browser. This enables attackers to:

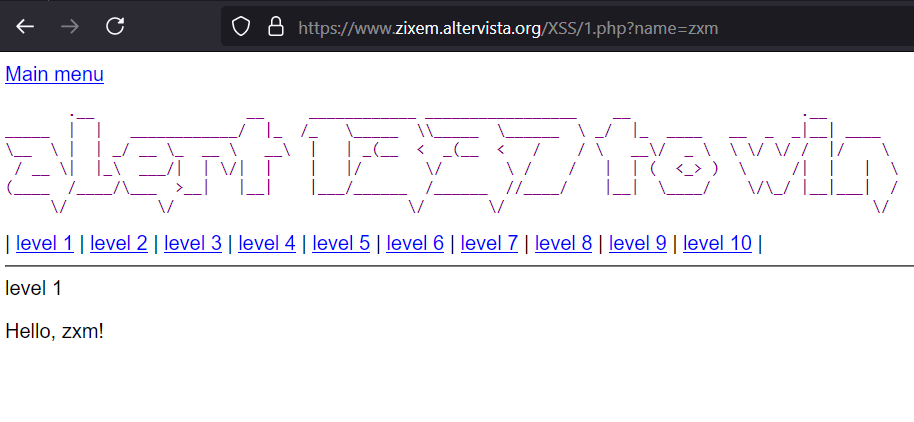
1. Perform actions the user can perform.
2. Access and modify the user's data.
3. Interact with other users, potentially launching further attacks.

Attackers can deliver these attacks via links on websites, emails, or social media. Reflected XSS requires external delivery, making it generally less severe than stored XSS, which can affect multiple users directly.

**Walkthrough: Reflected XSS in Zixem**

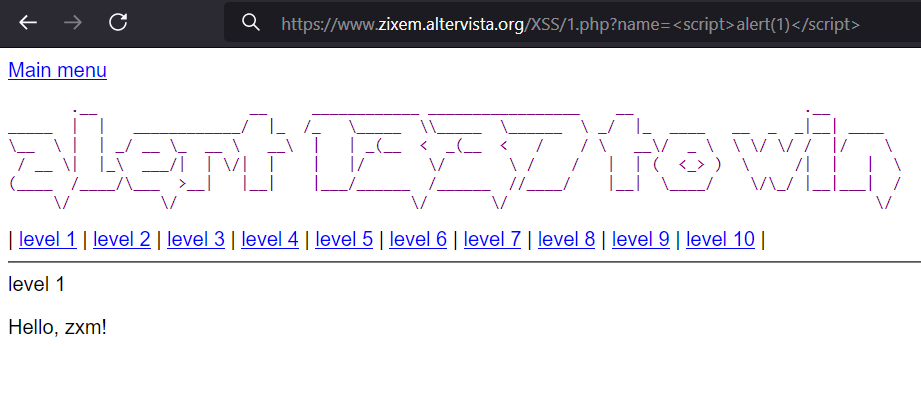
Target Link: [**https://www.zixem.altervista.org/XSS/1.php?name=zxm**](https://www.zixem.altervista.org/XSS/1.php?name=zxm)

In the target application perform a cross-site scripting attack that calls the alert function.

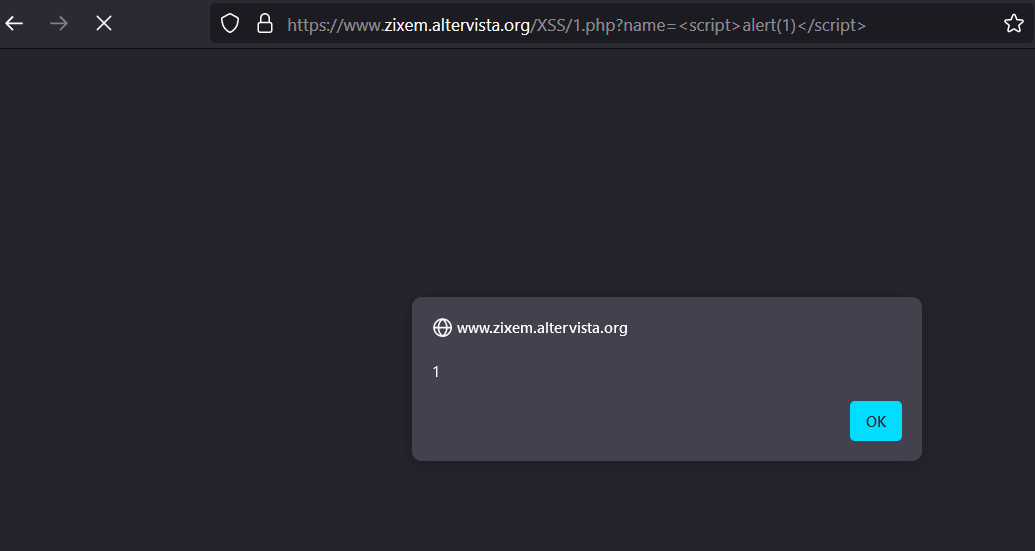
****

1. Copy and paste the script into the **name** query parameter: **<script>alert(1)</script>**

**Command:** [**https://www.zixem.altervista.org/XSS/1.php?name=%3Cscript%3Ealert(1)%3C/script%3E**](https://www.zixem.altervista.org/XSS/1.php?name=%3Cscript%3Ealert(1)%3C/script%3E)

****

1. Press Enter.



An alert/dialog box will appear showing reflected XSS exploited successfully.

We can prevent XSS by:

1. Strictly filtering user input upon arrival.
2. Encoding user data before outputting it in HTTP responses.
3. Using appropriate response headers like Content-Type and X-Content-Type-Options.
4. Implementing Content Security Policy (CSP) as a final defense.