SCENARIO

The application contains a reflected cross site scripting vulnerability in the search query tracking functionality because it encodes angle brackets before processing. We will try to trigger an alert message by injecting a payload into the application.

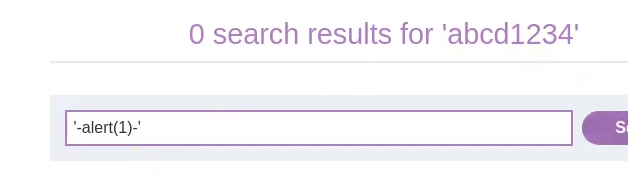
**PROCEDURE**

1. Go to the vulnerable application and search for anything.
2. Inspect the source of the HTML page and we can see that our provided string was provided into a javascript string.
3. At the end, we will create a payload to inject into the search.

**PAYLOAD**

';-alert("HACKED!");’

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Input Validation:** Implement strict validation for all user-supplied data to ensure it adheres to the expected format. Reject any data that doesn't match the specified criteria.
2. **Encode Data:** Ensure all user data is properly escaped before it is reflected back on the page. This means encoding special characters like <, >, &, and ".
3. **Use Safe JavaScript Methods:** When working with JavaScript, make sure to use methods that are immune to injections. Instead of using risky methods like eval() or document.write(), utilize safer alternatives like textContent or setAttribute.
4. **Avoid Inline JavaScript:** Using inline JavaScript can expose the application to various attacks, including DOM-based XSS. Externalize scripts when possible.