SCENARIO

The application contains a reflected cross site scripting vulnerability in the search query tracking functionality because angle brackets and double quotes are HTML encoded and single quotes are escaped. We will try to trigger an alert message by injecting a payload into the application.

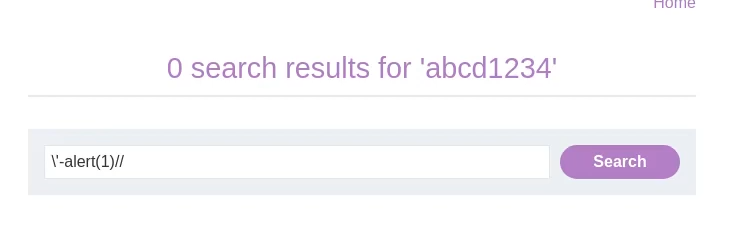
**PROCEDURE**

1. Go to the vulnerable application and search for anything.
2. Inspect the source of the HTML page and we can see that in our provided string the single quotes are back slashed and angle brackets are encoded.
3. At the end, we will create a payload to inject into the search to trigger an alert.

**PAYLOAD**

'\';alert(“HACKED!”)//

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Avoid Direct User Input:** Never use direct user input within scripts without properly validating and encoding it first. Untrusted data should not be used as part of a script until it's been sanitized.
2. **Content Security Policy (CSP):** Implement a strict CSP that disallows inline scripts. A well-configured CSP can prevent many injection attacks, including XSS.
3. **Use JavaScript Frameworks Safely:** If using frameworks like AngularJS, React, or Vue.js, always adhere to the framework's best practices. These modern frameworks have built-in mechanisms to prevent many types of XSS attacks when used correctly.
4. **Use Escape Functions:** Use escape functions suitable for the context in which the data is being inserted into the page. For example, when inserting data into a JavaScript block, use JavaScript escaping functions.