SCENARIO

The application contains a stored cross site scripting vulnerability in the comment box functionality, website input field to be specific but inside a template string with angle brackets, single, and double quotes HTML encoded, and backticks escaped. We will try to trigger an alert message by injecting a payload into the application.

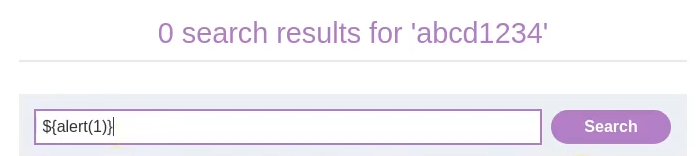
**PROCEDURE**

1. Go to the vulnerable application and comment anything in any blog with all fields filled.
2. As we know that the input is enclosed within a JavaScript template literal so we will create the payload accordingly.
3. It will force the application to trigger an alert whenever clicked on the commentor’s name.

**PAYLOAD**

${alert(1)}

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Strict Input Validation:** Always ensure user input adheres to expected patterns or values. Given that we're talking about a website input field, a regular expression should be in place to validate URLs.
2. **Safe Output Encoding:** While encoding is in place for certain characters, ensure that any content which can be controlled by a user and is displayed back on the site is correctly encoded to prevent any kind of script execution. This includes more comprehensive encoding to handle different contexts, like HTML content, attributes, and JavaScript contexts.
3. **Escape Template Literals:** If user input can be included in template literals, ensure characters significant in this context, such as ${, are escaped. This will prevent any JavaScript expressions from being evaluated.
4. **Adopt Non-Executable Contexts:** If user data is only intended to be displayed and never executed, try to place it in non-executable contexts. For instance, instead of inline JavaScript, use element properties to set values.