SCENARIO

The application implements a Content Security Policy and contains a reflected cross site scripting vulnerability in the search box functionality. We will try to bypass the CSP and exploit the vulnerability to make the application trigger an alert.

**PROCEDURE**

1. Go the vulnerable web applications inject the first payload in the search box.
2. Observe the request in BurpSuite’s Interceptor to find out that we control the CSP’s token/directive value and can change it accordingly.
3. Based on the observations we will create our second payload to inject into the URL which will force the application to generate and alert.
4. The injection uses the script-src-elem directive in CSP. This directive allows you to target just script elements. Using this directive, you can overwrite existing script-src rules enabling you to inject unsafe-inline, which allows you to use inline scripts.

**PAYLOAD**

1. <img src=1 onerror=alert(1)>
2. https://YOUR-LAB-ID.web-security-academy.net/?search=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&token=;script-src-elem%20%27unsafe-inline%27

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Implement Strict Content Security Policy (CSP):** Ensure that your CSP doesn't have overly permissive policies such as 'unsafe-inline'. Instead, use nonces or hashes to allow specific inline scripts. Always make sure to review and test your CSP implementation to ensure it's effective.
2. **Avoid Allowing User Input to Influence CSP Directives:** Never trust user input when it comes to security-sensitive operations. By ensuring that user input can't influence your CSP directives, you prevent attackers from weakening or bypassing your security measures.
3. **Encode Data on Output:** Always encode user-generated data before displaying it back to the user to prevent potential malicious scripts from executing.
4. **Sanitize and Validate User Input:** Before processing any user-generated data, make sure to validate and sanitize it. Make use of libraries and frameworks that offer these functionalities out of the box.
5. **Avoid Inline Scripts:** Whenever possible, avoid using inline scripts in your web applications. Instead, use external scripts with proper nonces or hashes as required by your CSP.