SCENARIO

The application contains a reflected cross site scripting vulnerability in the search box functionality but it uses a web application firewall to protect against common XSS vectors by whitelisting some HTML tags with all events and href attributes blocked. We will try to trigger an alert message by injecting a payload into the search box.

**PROCEDURE**

1. Go to the vulnerable application and try searching for anything.
2. Get the request in BurpSuite’s Intruder and mount a brute force attack by setting the payloads from the cheat sheet of XSS.
3. We got only <svg>, <a>, <text> and <animate> tag with the HTTP response of 200.
4. We will now create a payload to inject in the search box by using the appropriate tags which will force the application to show a button which shows an alert when clicked;.

**PAYLOAD**

https://YOUR-LAB-ID.web-security-academy.net/?search=<svg><a><animate attributeName=href values=javascript:alert(1) /><text x=20 y=20>Click me</text></a>

**REMEDIATION**

1. **Strict Whitelisting:** Update the web application firewall (WAF) to use a strict whitelisting approach. Instead of only blocking certain tags and attributes, define a strict list of allowed tags, attributes, and their values. Everything else should be rejected or sanitized.
2. **Escape Output:** Regardless of any whitelisting, ensure that all user-generated content displayed in the application is escaped. Convert special characters (like <, >, and &) to their HTML encoded equivalents (like &lt;, &gt;, and &amp;).
3. **Length Restrictions:** Implement a length restriction on input fields where appropriate. This won't prevent XSS outright, but it can make exploitation more challenging for an attacker.