SCENARIO

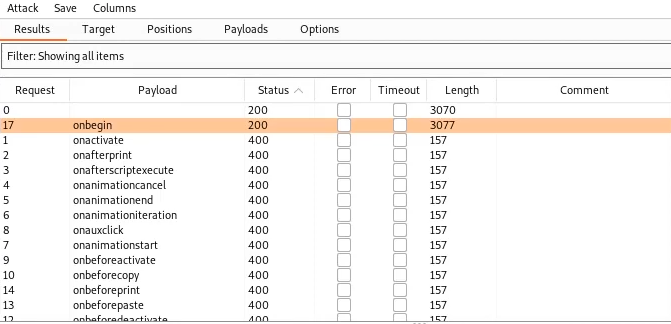
The application contains a reflected cross site scripting vulnerability in the search box functionality but it uses a web application firewall to protect against common XSS vectors by blocking all the HTML tags but misses some svg tags and events. We will try to trigger an alert message by injecting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

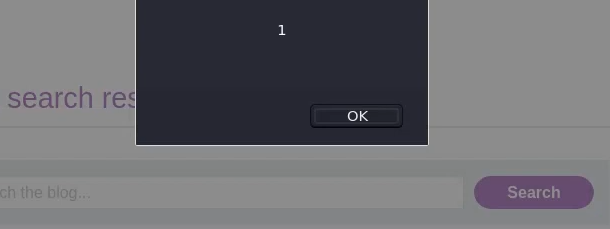
**PROCEDURE**

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 as the possible HTML tags enclosed within <§§> from the cheat sheet of XSS.
3. Then mount a brute force attack after appending Payload 1 in the search query.
4. Then we found that only onbegin tag returned the 200 HTML response and all other returned 400.
5. 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.
6. Now, according to the tags and events we found we will create a 2nd Payload and then by typing it into the search bar we will trigger the alert.

**PAYLOAD**

1. <svg><animatetransform%20§§=1>
2. %22%3E%3Csvg%3E%3Canimatetransform%20onbegin=alert(1)%3E

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Comprehensive Whitelisting:** Implement a strict whitelist of accepted characters or strings for the search functionality. By denying everything except a strict set of accepted values, you significantly reduce the risk of XSS. This list should not only cover regular HTML tags but also SVG and its associated events.
2. **Encode Data:** Always HTML-encode every piece of data that is reflected back to the web page. This means that characters like <, >, and & are turned into their HTML-encoded equivalents such as &lt;, &gt;, and &amp;. By doing this, even if an attacker can inject a payload, it will not be interpreted by the browser as code.
3. **Content Security Policy:** Implement a strict Content Security Policy (CSP) that disallows inline scripts. A proper CSP can prevent most XSS attacks by prohibiting the execution of inline scripts and restricting script sources.