SCENARIO

The application is vulnerable to web cache poisoning because cookie isn’t included in the cache key. We will try to poison the cache with a response that executes alert(1) in the visitor's browser.

**PROCEDURE**

1. Open the web application and in the BurpSuite’s Proxy tab send the request to BurpSuite’s Repeater and study the GET request to the home URL.
2. We observe that the value from the **fehost** cookie is reflected inside a double-quoted JavaScript object in the response, try sending requests to the endpoint by using some cache busters and we see that the value is reflected in the source code of the website.
3. Replace the value of the **fehost** parameter in the cookie with the payload to trigger an alert.
4. Send the malicious request and keep replaying the request until we see our exploit server URL being reflected in the response and **X-Cache: hit** in the headers.

**PAYLOAD**

"-alert(1)-"qwertyuiop

**REMEDIATION**

1. **Include Cookies in Cache Key:** When caching, ensure that the cookies (especially those used in the application logic) are part of the cache key. This means each unique cookie will have its own unique cache entry.
2. **Sanitize Input:** Always sanitize user input, including values from cookies. Never trust data that comes from the client side, even if it's stored in cookies.
3. **Content Security Policy (CSP):** Implement a strict CSP. This will mitigate the risk of malicious payloads being executed, even if they manage to be injected.
4. **Avoid Direct Reflection:** If certain cookie values are meant to be reflected in the page's HTML or JavaScript, make sure to escape them properly to prevent any executable code from running.
5. **Vary HTTP Responses:** Include the "Vary: Cookie" HTTP response header to indicate that different cookie values can produce different outputs. This ensures that the caching mechanism knows that responses vary based on cookies.
6. **Limit the Cache:** Do not cache responses that contain sensitive or user-specific data, especially if they are derived from cookies.