SCENARIO

The application is vulnerable to web cache poisoning due to discrepancies in how the cache and the back-end application handle ambiguous requests. We will try to poison the internal cache so that the home page executes alert(document.cookie) in the victim's browser.

**PROCEDURE**

1. Open the web application and send the request for **homepage** to BurpSuite’s Repeater.
2. Observe that we can use cache buster if we append **?cb=121** into the URL we can query the server.
3. Notice that we can add a second Host header with an arbitrary value, this appears to be ignored when validating and routing your request. Crucially, notice that the arbitrary value of your second Host header is reflected in an absolute URL.
4. Go to the exploit server and create a file at Payload 1 containing the Payload 2 in the body. Store the exploit.
5. Back in Burp Repeater, remove the cache buster and try to mimic the request by replaying the request in browser.
6. 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**

1. /resources/js/tracking.js
2. alert(document.cookie)

**REMEDIATION**