**SCENARIO**

This application involves a front-end and back-end server, and the front-end server doesn't support chunked encoding. We will try to smuggle a request to the back-end server, so that a subsequent request for / (the web root) triggers a 404 Not Found response.

**PROCEDURE**

1. Open the web application and send the GET request for homepage to BurpSuite’s Repeater.
2. Now right click on the request and click **Change Request Method** to change the request to **POST** as we can not send body for **GET** requests.
3. Inject the payload in the Repeater tab and send the request twice, we see that the second response contains 404 not found.

**PAYLOAD**

POST / HTTP/1.1

Host: YOUR-LAB-ID.web-security-academy.net

Content-Type: application/x-www-form-urlencoded

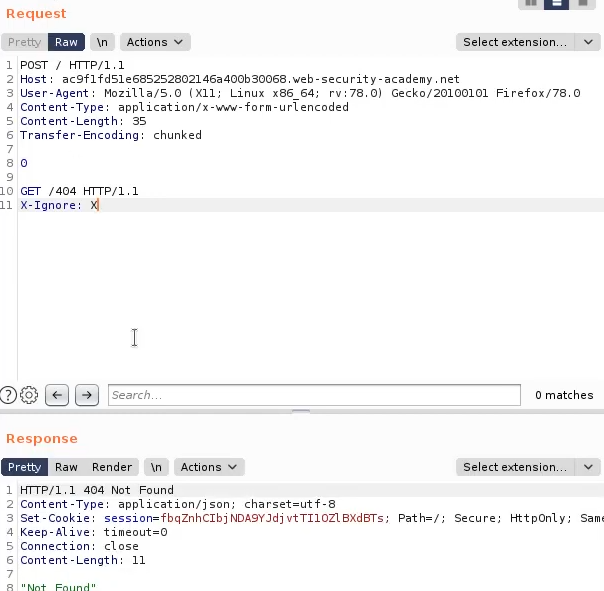
Content-Length: 35

Transfer-Encoding: chunked

0

GET /404 HTTP/1.1

X-Ignore: X

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Ensure Consistent Handling of Encodings:** Both the front-end and back-end servers should handle HTTP request encodings consistently. If the front-end server does not support chunked encoding, the back-end server should also be configured to reject chunked encoded requests.
2. **Use a Unified Server Architecture:** Consider using a unified server architecture where both the front-end and back-end functionalities are handled by a single server. This reduces discrepancies in request handling.
3. **Standardize Request Parsing:** Both the front-end and back-end servers should use the same libraries or methods for parsing HTTP requests. This ensures that the two servers interpret requests in the same way.
4. **Use Web Application Firewalls (WAFs):** WAFs can detect anomalies in the HTTP requests. Configuring a WAF to recognize and block potential smuggling attempts can help.
5. **Inspect for Hidden or Nested Requests:** Server-side logic should be implemented to detect and block requests that contain nested or hidden HTTP methods, like the injected "GET /404" in the payload.