**SCENARIO**

This application involves a front-end and back-end server, and the front-end server doesn't support chunked encoding. The front-end server rejects requests that aren't using the GET or POST method. We will try to smuggle a request to the back-end server, so that the next request processed by the back-end server appears to use the method GPOST.

**PROCEDURE**

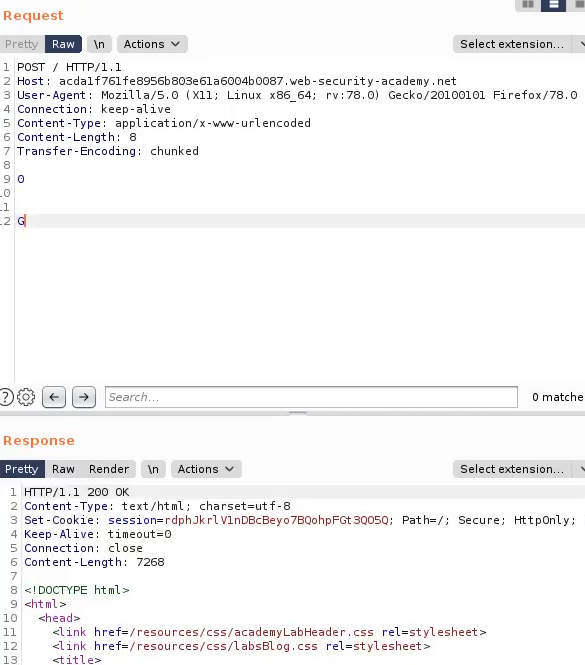
1. Open the web application and send the GET request for homepage to BurpSuite’s Repeater.
2. Now right click on the request and click **Change Request Method** to change the request to **POST** as we can not send body for **GET** requests.
3. Append the following Payload to the request and send it twice so that the first requests will end at 0 and the new request will start by G so it will become **GPOST.**

**PAYLOAD**

Transfer-Encoding: chunked

0

G

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Uniformly Handle HTTP Request Parsing:** Ensure both the front-end and back-end servers handle and parse HTTP requests in the same way. Differences in how servers interpret requests can lead to smuggling opportunities.
2. **Disallow Unsupported Encoding:** Implement server-side checks to disallow or strip unsupported or suspicious encoding techniques like "Transfer-Encoding: chunked" when they are not expected or supported.
3. **Employ Web Application Firewalls (WAFs):** Utilize WAFs to detect and block anomalous request patterns and encodings, adding an additional layer of security against smuggling attacks.
4. **Upgrade and Patch Regularly:** Ensure that both front-end and back-end servers are regularly updated and patched. Many modern servers have protections against HTTP Request Smuggling.
5. **Consistent Configuration:** Ensure that all servers in the application's infrastructure have consistent configurations, especially regarding how they handle HTTP request headers and encoding.