In chaos engineering, "pod HTTP modify body" refers to deliberately altering the content of HTTP request or response bodies within a Kubernetes pod (container). This manipulation is performed to simulate scenarios where the data being transmitted over HTTP is modified unexpectedly.

There are several reasons why chaos engineers might want to modify HTTP request or response bodies:

1. \*\*Data corruption\*\*: Simulating situations where data is corrupted during transmission, such as through network errors or malicious interference.

2. \*\*Content manipulation\*\*: Testing how the system handles unexpected changes to the content of HTTP requests or responses, which may occur due to bugs or security vulnerabilities.

3. \*\*Injection attacks\*\*: Mimicking attacks such as SQL injection or Cross-Site Scripting (XSS) by injecting malicious content into HTTP requests or responses.

By modifying HTTP bodies, engineers can assess how the system reacts to such changes and whether it can maintain functionality and security. They may evaluate whether the system's input validation, data integrity checks, and error handling mechanisms are effective in detecting and mitigating unexpected modifications to HTTP data.

The purpose of conducting pod HTTP modify body experiments in chaos engineering is to identify vulnerabilities in the system's handling of HTTP traffic and to validate the effectiveness of mitigation strategies. This allows engineers to strengthen the system's resilience to HTTP-related threats and ensure reliable operation in real-world scenarios.

The impact of Pod HTTP Modify Body chaos attack can be seen using: **curl <IP>** command with IP of service frontend.

**Introduction**

* It injects http modify body chaos on the service whose port is provided as TARGET\_SERVICE\_PORT by starting proxy server and then redirecting the traffic through the proxy server.
* Can be used to overwrite the http response body by providing the new body value as RESPONSE\_BODY.
* It can test the application's resilience to error or incorrect http response body.