## **TDL-001 – HTTP Request Smuggling – {Medium} {Open}**

|  |  |
| --- | --- |
| **Vulnerable URLs** | N/A |
| **Vulnerable Parameter** | N/A |
| **Payload** | N/A |
| **Vulnerability Class** | A05:2021 – Security Misconfiguration |
| **CVSS Score 3.1** | **Medium - 5.3**  **CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N** |
| **CWE-ID** | **CWE-444** |
| **Vulnerability Explanation:** | HTTP Request Smuggling exploits inconsistencies in parsing HTTP requests between different servers (e.g., proxy and backend), allowing attackers to smuggle a malicious request within another. |
| **Vulnerability Impact:** | Can lead to bypassing security controls, cache poisoning, request spoofing, or even XSS. Attackers can hijack user sessions or manipulate requests sent to the backend. |
| **Remediation** | * Disable conflicting HTTP headers like Content-Length and Transfer-Encoding together. Patch and configure web servers correctly, and apply input validation and normalization. |
| **Reference** | https://portswigger.net/web-security/request-smuggling |

**Steps to Reproduce & Proof of Concept:**

1. Navigate to the target URL and capture the HTTP request using Burp Suite or a proxy tool.
2. Observe that the application supports both HTTP/1.1 and HTTP/2 protocols.
3. Downgrade the connection from HTTP/2 to HTTP/1.1 to perform the smuggling test.