

aio-libs / aiohttp Public

<> Code Issues 184 Pull requests 90 Discussions Actions Security an

Commit e00ca3c



patchback[bot] and rodrigonogueira authored 3 weeks ago · ✓ 35 / 39 · Verified



[PR #12240/345d2537 backport][3.13] Reject duplicate singleton headers in C extension parser (#12241)

****This is a backport of PR #12240 as merged into master (345d253).****

Co-authored-by: Rodrigo Nogueira <rodrigo.b.nogueira@gmail.com>

🔗 master (#12241) · 📦 v3.13.5 v3.13.4

1 parent 9370b97 commit e00ca3c 📄

📁 3 files changed +68 -0 lines changed

↑ Top ⚙️

🔍 Filter files...

- ✓ 📁 CHANGES
 - 📄 12240.bugfix.rst
- ✓ 📁 aiohttp
 - 📄 _http_parser.pyx
- ✓ 📁 tests
 - 📄 test_http_parser.py

📁 3 files changed +68 -0 lines changed

🔍 Search within code ⚙️

✓ CHANGES/12240.bugfix.rst

<> 📄 ...

```

... @@ -0,0 +1,5 @@
1 + Rejected duplicate singleton headers (`Host`, `Content-Type`,
2 + `Content-Length`, etc.) in the C extension HTTP parser to match
3 + the pure Python parser behavior, preventing potential host-based

```

```

4 + access control bypasses via parser differentials
5 + -- by :user:`rodrigobnogueira`.

```

▼ aiohttp/_http_parser.pyx

```

@@ -71,6 +71,20 @@ cdef object StreamReader = _StreamReader
71 71 cdef object DeflateBuffer = _DeflateBuffer
72 72 cdef bytes EMPTY_BYTES = b""
73 73
74 + # https://www.rfc-editor.org/rfc/rfc9110.html#section-5.5-6
75 + cdef tuple SINGLETON_HEADERS = (
76 +     hdrs.CONTENT_LENGTH,
77 +     hdrs.CONTENT_LOCATION,
78 +     hdrs.CONTENT_RANGE,
79 +     hdrs.CONTENT_TYPE,
80 +     hdrs.ETAG,
81 +     hdrs.HOST,
82 +     hdrs.MAX_FORWARDS,
83 +     hdrs.SERVER,
84 +     hdrs.TRANSFER_ENCODING,
85 +     hdrs.USER_AGENT,
86 + )
87 +
74 88 cdef inline object extend(object buf, const char* at, size_t length):
75 89     cdef Py_ssize_t s
76 90     cdef char* ptr
@@ -430,6 +444,14 @@ cdef class HttpParser:
430 444         raw_headers = tuple(self._raw_headers)
431 445         headers = CIMultiDictProxy(CIMultiDict(self._headers))
432 446
447 +         # https://www.rfc-editor.org/rfc/rfc9110.html#name-collected-abnf
448 +         bad_hdr = next(
449 +             (h for h in SINGLETON_HEADERS if len(headers.getall(h, ())) > 1),
450 +             None,
451 +         )
452 +         if bad_hdr is not None:
453 +             raise BadHttpMessage(f"Duplicate '{bad_hdr}' header found.")
454 +
433 455         if self._cparser.type == cparser.HTTP_REQUEST:
434 456             h_upgrade = headers.get("upgrade", "")

```

```
435 457         allowed = upgrade and h_upg.isascii() and h_upg.lower() in
ALLOWED_UPGRADES
```



tests/test_http_parser.py



```
@@ -265,6 +265,47 @@ def test_content_length_transfer_encoding(parser: Any)
-> None:
```

```
265 265         parser.feed_data(text)
```

```
266 266
```

```
267 267
```

```
268 + @pytest.mark.parametrize(
269 +     "hdr",
270 +     (
271 +         "Content-Length",
272 +         "Content-Location",
273 +         "Content-Range",
274 +         "Content-Type",
275 +         "ETag",
276 +         "Host",
277 +         "Max-Forwards",
278 +         "Server",
279 +         "Transfer-Encoding",
280 +         "User-Agent",
281 +     ),
282 + )
283 + def test_duplicate_singleton_header_rejected(
284 +     parser: HttpRequestParser, hdr: str
285 + ) -> None:
286 +     val1, val2 = ("1", "2") if hdr == "Content-Length" else ("value1",
287 +         "value2")
288 +     text = (
289 +         f"GET /test HTTP/1.1\r\n"
290 +         f"Host: example.com\r\n"
291 +         f"{hdr}: {val1}\r\n"
292 +         f"{hdr}: {val2}\r\n"
293 +         f"\r\n"
294 +     ).encode()
295 +     with pytest.raises(http_exceptions.BadHttpMessage, match="Duplicate"):
296 +         parser.feed_data(text)
```

```
297 +
298 + def test_duplicate_host_header_rejected(parser: HttpRequestParser) -> None:
299 +     text = (
300 +         b"GET /admin HTTP/1.1\r\n"
301 +         b"Host: admin.example\r\n"
302 +         b"Host: public.example\r\n"
303 +         b"\r\n"
304 +     )
305 +     with pytest.raises(http_exceptions.BadHttpMessage,
306 +                         match="Duplicate.*Host"):
307 +         parser.feed_data(text)
308 +
268 309 def test_bad_chunked(parser: HttpRequestParser) -> None:
269 310     """Test that invalid chunked encoding doesn't allow content-length to be
270 311     used."""
270 311     text = (
```



Comments 0



Please [sign in](#) to comment.