aio-libs / **aiohttp** Public

Code Issues 185 Pull requests 89 Discussions Actions Security an

Commit 9370b97

rodrigobnogueira and **rodrigo.nogueira** authored 3 weeks ago · 35 / 39 · Verified

```
[PR #12231/7043bc56 backport][3.13] Adjust header value character checks to RFC
9110 (#12235)

Co-authored-by: rodrigo.nogueira <rodrigo.nogueira@prf.gov.br>
```

master (#12235) · v3.13.5 v3.13.4

1 parent cbb774f commit 9370b97

3 files changed +20 -1 lines changed

Top

Filter files...

- CHANGES
  - 12231.bugfix.rst
- aiohttp
  - http_parser.py
- tests
  - test_http_parser.py

3 files changed +20 -1 lines changed

Search within code

CHANGES/12231.bugfix.rst

```
@@ -0,0 +1,2 @@
+ Adjusted pure-Python request header value validation to align with RFC 9110
  control-character handling, while preserving lax response parser behavior, and
  added regression tests for Host/header control-character cases.
+ -- by :user:`rodrigobnogueira`.
```

aiohttp/http_parser.py

```
@@ -81,6 +81,10 @@
#     token = 1*tchar
_TCHAR_SPECIALS: Final[str] = re.escape("!#$%&'*+-.^_`|~")
TOKENRE: Final[Pattern[str]] = re.compile(f"[0-9A-Za-z{_TCHAR_SPECIALS}]+")
+# https://www.rfc-editor.org/rfc/rfc9110#section-5.5-5
+_FIELD_VALUE_FORBIDDEN_CTL_RE: Final[Pattern[str]] = re.compile(
+    r"[\x00-\x08\x0a-\x1f\x7f]"
+)
VERSRE: Final[Pattern[str]] = re.compile(r"HTTP/(\d)\.(\d)", re.ASCII)
DIGITS: Final[Pattern[str]] = re.compile(r"\d+", re.ASCII)
HEXDIGITS: Final[Pattern[bytes]] = re.compile(rb"[0-9a-fA-F]+")
@@ -208,7 +212,10 @@ def parse_headers(
                value = bvalue.decode("utf-8", "surrogateescape")

                # https://www.rfc-editor.org/rfc/rfc9110.html#section-5.5-5
-               if "\n" in value or "\r" in value or "\x00" in value:
+               if self._lax:
+                   if "\n" in value or "\r" in value or "\x00" in value:
+                       raise InvalidHeader(bvalue)
+               elif _FIELD_VALUE_FORBIDDEN_CTL_RE.search(value):
                    raise InvalidHeader(bvalue)

                headers.add(name, value)
```

tests/test_http_parser.py

```
@@ -221,6 +221,9 @@ def test_bad_header_name(parser: Any,
rfc9110_5_6_2_token_delim: str) -> None:
            "Foo : bar",  # https://www.rfc-editor.org/rfc/rfc9112.html#section-
5.1-2
            "Foo\t: bar",
            "\xffoo: bar",
+           "Foo: abc\x01def",  # CTL bytes forbidden per RFC 9110 §5.5
+           "Foo: abc\x7fdef",  # DEL is also a CTL byte
+           "Foo: abc\x1fdef",
        ),
```

```
225  228      )
226  229      def test_bad_headers(parser: Any, hdr: str) -> None:
          @@ -229,6 +232,13 @@ def test_bad_headers(parser: Any, hdr: str) -> None:
229  232          parser.feed_data(text)
230  233
231  234
     235  + def test_ctl_host_header_bad_characters(parser: HttpRequestParser) -> None:
     236  +     """CTL byte in Host header must be rejected."""
     237  +     text = b"GET /test HTTP/1.1\r\nHost: trusted.example\x01@bad.test\r\n\r\n"
     238  +     with pytest.raises(http_exceptions.BadHttpMessage):
     239  +         parser.feed_data(text)
     240  +
     241  +
232  242  def test_unpaired_surrogate_in_header_py(loop: Any, protocol: Any) -> None:
233  243      parser = HttpRequestParserPy(
234  244          protocol,
```

Comments 0

Please sign in to comment.