

python / cpython Public

<> Code Issues 5k+ Pull requests 2.1k Actions Projects Security and q

# Commit b1cf901



3 people authored 2 weeks ago · ✖ 89 / 93 · Partially verified

[3.14] [gh-146211](#): Reject CR/LF in HTTP tunnel request headers ([GH-146212](#)) ([#148342](#))

[gh-146211](#): Reject CR/LF in HTTP tunnel request headers ([GH-146212](#)) (cherry picked from commit [05ed7ce](#))

Co-authored-by: Seth Larson <seth@python.org>  
Co-authored-by: Illia Volochii <illia.volochii@gmail.com>

🔑 3.14 (#148342) · 📦 v3.14.5rc1

1 parent [c235654](#) commit b1cf901 📄

**3 files changed** +57 -1 🟢🟢🟢🟢

📄 ↑ Top ⚙️

🔍 Filter files... ☰

- ✓ 📁 Lib
  - ✓ 📁 http
    - 📄 client.py
  - ✓ 📁 test
    - 📄 test\_httplib.py
- ✓ 📁 Misc/NEWS.d/next/Security
  - 📄 2026-03-20-09-29-42.gh-issue-146211.PQVbs7.rst

📄 🔍 Search within code ⚙️

```

Lib/http/client.py
@@ -972,13 +972,22 @@ def _wrap_ipv6(self, ip):

```

```

972 972         return ip
973 973
974 974     def _tunnel(self):
975 +         if _contains_disallowed_url_pchar_re.search(self._tunnel_host):
976 +             raise ValueError('Tunnel host can\'t contain control characters %r'
977 +                               % (self._tunnel_host,))
978 978         connect = b"CONNECT %s:%d %s\r\n" % (
979 979             self._wrap_ipv6(self._tunnel_host.encode("idna")),
980 980             self._tunnel_port,
981 981             self._http_vsn_str.encode("ascii"))
982 982         headers = [connect]
983 983         for header, value in self._tunnel_headers.items():
984 -             headers.append(f"{header}: {value}\r\n".encode("latin-1"))
985 +             header_bytes = header.encode("latin-1")
986 +             value_bytes = value.encode("latin-1")
987 +             if not _is_legal_header_name(header_bytes):
988 +                 raise ValueError('Invalid header name %r' % (header_bytes,))
989 +             if _is_illegal_header_value(value_bytes):
990 +                 raise ValueError('Invalid header value %r' % (value_bytes,))
991 +             headers.append(b"%s: %s\r\n" % (header_bytes, value_bytes))
992 992         headers.append(b"\r\n")
993 993         # Making a single send() call instead of one per line encourages
994 994         # the host OS to use a more optimal packet size instead of

```



Lib/test/test\_httplib.py



@@ -369,6 +369,51 @@ def test\_invalid\_headers(self):

```

369 369         with self.assertRaisesRegex(ValueError, 'Invalid header'):
370 370             conn.putheader(name, value)
371 371
372 +     def test_invalid_tunnel_headers(self):
373 +         cases = (
374 +             ('Invalid\r\nName', 'ValidValue'),
375 +             ('Invalid\rName', 'ValidValue'),
376 +             ('Invalid\nName', 'ValidValue'),
377 +             ('\r\nInvalidName', 'ValidValue'),
378 +             ('\rInvalidName', 'ValidValue'),
379 +             ('\nInvalidName', 'ValidValue'),
380 +             (' InvalidName', 'ValidValue'),
381 +             ('\tInvalidName', 'ValidValue'),

```

```
382 +         ('Invalid:Name', 'ValidValue'),
383 +         (':InvalidName', 'ValidValue'),
384 +         ('ValidName', 'Invalid\r\nValue'),
385 +         ('ValidName', 'Invalid\rValue'),
386 +         ('ValidName', 'Invalid\nValue'),
387 +         ('ValidName', 'InvalidValue\r\n'),
388 +         ('ValidName', 'InvalidValue\r'),
389 +         ('ValidName', 'InvalidValue\n'),
390 +     )
391 +     for name, value in cases:
392 +         with self.subTest((name, value)):
393 +             conn = client.HTTPConnection('example.com')
394 +             conn.set_tunnel('tunnel', headers={
395 +                 name: value
396 +             })
397 +             conn.sock = FakeSocket('')
398 +             with self.assertRaisesRegex(ValueError, 'Invalid header'):
399 +                 conn._tunnel() # Called in .connect()
400 +
401 +     def test_invalid_tunnel_host(self):
402 +         cases = (
403 +             'invalid\r.host',
404 +             '\ninvalid.host',
405 +             'invalid.host\r\n',
406 +             'invalid.host\x00',
407 +             'invalid host',
408 +         )
409 +         for tunnel_host in cases:
410 +             with self.subTest(tunnel_host):
411 +                 conn = client.HTTPConnection('example.com')
412 +                 conn.set_tunnel(tunnel_host)
413 +                 conn.sock = FakeSocket('')
414 +                 with self.assertRaisesRegex(ValueError, 'Tunnel host can\'t
contain control characters'):
415 +                     conn._tunnel() # Called in .connect()
416 +
372 417     def test_headers_debuglevel(self):
373 418         body = (
374 419             b'HTTP/1.1 200 OK\r\n'
```



```
...6-03-20-09-29-42.gh-issue-146211.PQVbs7.rst
```

```
@@ -0,0 +1,2 @@  
1 + Reject CR/LF characters in tunnel request headers for the  
2 + HTTPConnection.set_tunnel() method.
```

**Comments** 0