

python / cpython Public

<> Code Issues 5k+ Pull requests 2.2k Actions Projects Security and q

# Commit 05ed7ce



sethmlarson and illia-v authored 3 days ago · ✓ 94 / 97 · Verified



gh-146211: Reject CR/LF in HTTP tunnel request headers (#146212)

Co-authored-by: Illia Volochii <illia.volochii@gmail.com>

main (#146212)

1 parent 266247c commit 05ed7ce

3 files changed +57 -1 lines changed

↑ 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

3 files changed +57 -1 lines changed

Search within code ⚙️

Lib/http/client.py



```

... @@ -976,13 +976,22 @@ def _wrap_ipv6(self, ip):
976 976         return ip
977 977
978 978     def _tunnel(self):
979 +         if _contains_disallowed_url_pchar_re.search(self._tunnel_host):

```

```

980 +         raise ValueError('Tunnel host can\'t contain control characters %r'
981 +                             % (self._tunnel_host,))
979 982         connect = b"CONNECT %s:%d %s\r\n" % (
980 983             self._wrap_ipv6(self._tunnel_host.encode("idna")),
981 984             self._tunnel_port,
982 985             self._http_vsn_str.encode("ascii"))
983 986         headers = [connect]
984 987         for header, value in self._tunnel_headers.items():
985 -             headers.append(f"{header}: {value}\r\n".encode("latin-1"))
988 +             header_bytes = header.encode("latin-1")
989 +             value_bytes = value.encode("latin-1")
990 +             if not _is_legal_header_name(header_bytes):
991 +                 raise ValueError('Invalid header name %r' % (header_bytes,))
992 +             if _is_illegal_header_value(value_bytes):
993 +                 raise ValueError('Invalid header value %r' % (value_bytes,))
994 +             headers.append(b"%s: %s\r\n" % (header_bytes, value_bytes))
986 995         headers.append(b"\r\n")
987 996         # Making a single send() call instead of one per line encourages
988 997         # 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
415 + contain control characters'):
416 +                     conn._tunnel() # Called in .connect()
417 +
418 +     def test_headers_debuglevel(self):
419 +         body = (
420 +             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