

python / cpython Public

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

# Commit 62ceb39



3 people authored 3 weeks ago · 66 / 70 · Verified

[3.14] [gh-145599](#), CVE 2026-3644: Reject control characters in `http.cookies.Morsel.update()` ([GH-145600](#)) ([#146023](#))

[gh-145599](#), CVE 2026-3644: Reject control characters in ``http.cookies.Morsel.update()`` ([GH-145600](#))

Reject control characters in ``http.cookies.Morsel.update()`` and ``http.cookies.BaseCookie.js_output``.  
(cherry picked from commit [57e88c1](#))

Co-authored-by: Stan Ulbrych <89152624+StanFromIreland@users.noreply.github.com>  
Co-authored-by: Victor Stinner <vstinner@python.org>  
Co-authored-by: Victor Stinner <victor.stinner@gmail.com>

3.14 (#146023) · v3.14.4

1 parent [8bc3aa9](#) commit 62ceb39

3 files changed +62 -4 lines changed

↑ Top ⚙️

Filter files...

- Lib
  - http
    - cookies.py
  - test
    - test\_http\_cookies.py
- Misc/NEWS.d/next/Security
  - 2026-03-06-17-03-38.gh-issue-145599.kchwZV.rst

3 files changed +62 -4 lines changed

Search within code

```

Lib/http/cookies.py
@@ -337,9 +337,16 @@ def update(self, values):
    key = key.lower()
    if key not in self._reserved:
        raise CookieError("Invalid attribute %r" % (key,))
+       if _has_control_character(key, val):
+           raise CookieError("Control characters are not allowed in "
+                               f"cookies {key!r} {val!r}")
    data[key] = val
    dict.update(self, data)

+   def __ior__(self, values):
+       self.update(values)
+       return self

    def isReservedKey(self, K):
        return K.lower() in self._reserved

@@ -365,9 +372,15 @@ def __getstate__(self):
    }
    def __setstate__(self, state):
-       self._key = state['key']
-       self._value = state['value']
-       self._coded_value = state['coded_value']
+       key = state['key']
+       value = state['value']
+       coded_value = state['coded_value']
+       if _has_control_character(key, value, coded_value):
+           raise CookieError("Control characters are not allowed in cookies "
+                               f"{key!r} {value!r} {coded_value!r}")
+       self._key = key
+       self._value = value
+       self._coded_value = coded_value

    def output(self, attrs=None, header="Set-Cookie:"):
        return "%s %s" % (header, self.OutputString(attrs))

@@ -379,13 +392,16 @@ def __repr__(self):
    def js_output(self, attrs=None):

```

```

381 394         # Print javascript
395 +         output_string = self.OutputString(attrs)
396 +         if _has_control_character(output_string):
397 +             raise CookieError("Control characters are not allowed in cookies")
382 398         return ""
383 399         <script type="text/javascript">
384 400         <!-- begin hiding
385 401         document.cookie = \"%s\";
386 402         // end hiding -->
387 403         </script>
388 -         """ % (self.OutputString(attrs).replace("'", r'\'))
404 +         """ % (output_string.replace("'", r'\'))
389 405
390 406     def OutputString(self, attrs=None):
391 407         # Build up our result

```



Lib/test/test\_http\_cookies.py



```

@@ -581,6 +581,14 @@ def test_control_characters(self):
581 581         with self.assertRaises(cookies.CookieError):
582 582             morsel["path"] = c0
583 583
584 +         # .__setstate__()
585 +         with self.assertRaises(cookies.CookieError):
586 +             morsel.__setstate__({'key': c0, 'value': 'val', 'coded_value':
'coded'})
587 +         with self.assertRaises(cookies.CookieError):
588 +             morsel.__setstate__({'key': 'key', 'value': c0, 'coded_value':
'coded'})
589 +         with self.assertRaises(cookies.CookieError):
590 +             morsel.__setstate__({'key': 'key', 'value': 'val',
'coded_value': c0})
591 +
584 592         # .setdefault()
585 593         with self.assertRaises(cookies.CookieError):
586 594             morsel.setdefault("path", c0)
@@ -595,6 +603,18 @@ def test_control_characters(self):
595 603         with self.assertRaises(cookies.CookieError):
596 604             morsel.set("path", "val", c0)
597 605

```

```

606 +         # .update()
607 +         with self.assertRaises(cookies.CookieError):
608 +             morsel.update({"path": c0})
609 +         with self.assertRaises(cookies.CookieError):
610 +             morsel.update({c0: "val"})
611 +
612 +         # .__ior__()
613 +         with self.assertRaises(cookies.CookieError):
614 +             morsel |= {"path": c0}
615 +         with self.assertRaises(cookies.CookieError):
616 +             morsel |= {c0: "val"}
617 +
598 618         def test_control_characters_output(self):
599 619             # Tests that even if the internals of Morsel are modified
600 620             # that a call to .output() has control character safeguards.
615 635             with self.assertRaises(cookies.CookieError):
616 636                 cookie.output()
617 637
638 +         # Tests that .js_output() also has control character safeguards.
639 +         for c0 in support.control_characters_c0():
640 +             morsel = cookies.Morsel()
641 +             morsel.set("key", "value", "coded-value")
642 +             morsel._key = c0 # Override private variable.
643 +             cookie = cookies.SimpleCookie()
644 +             cookie["cookie"] = morsel
645 +             with self.assertRaises(cookies.CookieError):
646 +                 cookie.js_output()
647 +
648 +             morsel = cookies.Morsel()
649 +             morsel.set("key", "value", "coded-value")
650 +             morsel._coded_value = c0 # Override private variable.
651 +             cookie = cookies.SimpleCookie()
652 +             cookie["cookie"] = morsel
653 +             with self.assertRaises(cookies.CookieError):
654 +                 cookie.js_output()
655 +
618 656
619 657         def load_tests(loader, tests, pattern):
620 658             tests.addTest(doctest.DocTestSuite(cookies))

```



...6-03-06-17-03-38.gh-issue-145599.kchwZV.rst



```
... @@ -0,0 +1,4 @@
1 + Reject control characters in :class:`http.cookies.Morsel`
2 + :meth:`~http.cookies.Morsel.update` and
3 + :meth:`~http.cookies.BaseCookie.js_output`.
4 + This addresses :cve:`2026-3644`.
```

**Comments** 0