python / **cpython** Public

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

Commit 57e88c1

StanFromIreland and **vstinner** authored 3 weeks ago · 69 / 72 · Verified

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

Reject control characters in `http.cookies.Morsel.update()` and
`http.cookies.BaseCookie.js_output`.

Co-authored-by: Victor Stinner <vstinner@python.org>
Co-authored-by: Victor Stinner <victor.stinner@gmail.com>
```

main (#145600) · v3.15.0a8

1 parent 77632f0 commit 57e88c1

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):
```

```
337  337                key = key.lower()
338  338                if key not in self._reserved:
339  339                    raise CookieError("Invalid attribute %r" % (key,))
     340  +             if _has_control_character(key, val):
     341  +                 raise CookieError("Control characters are not allowed in "
     342  +                                   f"cookies {key!r} {val!r}")
340  343                data[key] = val
341  344            dict.update(self, data)
342  345
     346  +     def __ior__(self, values):
     347  +         self.update(values)
     348  +         return self
     349  +
343  350        def isReservedKey(self, K):
344  351            return K.lower() in self._reserved
345  352
          @@ -365,9 +372,15 @@ def __getstate__(self):
365  372            }
366  373
367  374        def __setstate__(self, state):
368       -         self._key = state['key']
369       -         self._value = state['value']
370       -         self._coded_value = state['coded_value']
     375  +         key = state['key']
     376  +         value = state['value']
     377  +         coded_value = state['coded_value']
     378  +         if _has_control_character(key, value, coded_value):
     379  +             raise CookieError("Control characters are not allowed in cookies "
     380  +                               f"{key!r} {value!r} {coded_value!r}")
     381  +         self._key = key
     382  +         self._value = value
     383  +         self._coded_value = coded_value
371  384
372  385        def output(self, attrs=None, header="Set-Cookie:"):
373  386            return "%s %s" % (header, self.OutputString(attrs))
          @@ -379,13 +392,16 @@ def __repr__(self):
379  392
380  393        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

```
                @@ -604,6 +604,14 @@ def test_control_characters(self):
604             604               with self.assertRaises(cookies.CookieError):
605             605                   morsel["path"] = c0
606             606
                607 +             # .__setstate__()
                608 +             with self.assertRaises(cookies.CookieError):
                609 +                 morsel.__setstate__({'key': c0, 'value': 'val', 'coded_value':
                    'coded'})
                610 +             with self.assertRaises(cookies.CookieError):
                611 +                 morsel.__setstate__({'key': 'key', 'value': c0, 'coded_value':
                    'coded'})
                612 +             with self.assertRaises(cookies.CookieError):
                613 +                 morsel.__setstate__({'key': 'key', 'value': 'val',
                    'coded_value': c0})
                614 +
607             615               # .setdefault()
608             616               with self.assertRaises(cookies.CookieError):
609             617                   morsel.setdefault("path", c0)
                @@ -618,6 +626,18 @@ def test_control_characters(self):
618             626               with self.assertRaises(cookies.CookieError):
619             627                   morsel.set("path", "val", c0)
620             628
                629 +             # .update()
                630 +             with self.assertRaises(cookies.CookieError):
```

```
631 +                 morsel.update({"path": c0})
632 +             with self.assertRaises(cookies.CookieError):
633 +                 morsel.update({c0: "val"})
634 +
635 +             # .__ior__()
636 +             with self.assertRaises(cookies.CookieError):
637 +                 morsel |= {"path": c0}
638 +             with self.assertRaises(cookies.CookieError):
639 +                 morsel |= {c0: "val"}
640 +
621 641         def test_control_characters_output(self):
622 642             # Tests that even if the internals of Morsel are modified
623 643             # that a call to .output() has control character safeguards.
    @@ -638,6 +658,24 @@ def test_control_characters_output(self):
638 658                 with self.assertRaises(cookies.CookieError):
639 659                     cookie.output()
640 660
661 +         # Tests that .js_output() also has control character safeguards.
662 +         for c0 in support.control_characters_c0():
663 +             morsel = cookies.Morsel()
664 +             morsel.set("key", "value", "coded-value")
665 +             morsel._key = c0  # Override private variable.
666 +             cookie = cookies.SimpleCookie()
667 +             cookie["cookie"] = morsel
668 +             with self.assertRaises(cookies.CookieError):
669 +                 cookie.js_output()
670 +
671 +             morsel = cookies.Morsel()
672 +             morsel.set("key", "value", "coded-value")
673 +             morsel._coded_value = c0  # Override private variable.
674 +             cookie = cookies.SimpleCookie()
675 +             cookie["cookie"] = morsel
676 +             with self.assertRaises(cookies.CookieError):
677 +                 cookie.js_output()
678 +
641 679
642 680     def load_tests(loader, tests, pattern):
643 681         tests.addTest(doctest.DocTestSuite(cookies))
```

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

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

Comments 0