

python / cpython Public

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

# Commit cbba611



Pinky-Chaudhary and sethmlarson authored 3 weeks ago · ✖ 65 / 79 · Partially verified

[3.12] gh-143930: Reject leading dashes in webbrowser URLs (GH-146360)  
(cherry picked from commit [82a24a4](#))

Co-authored-by: Seth Michael Larson <seth@python.org>

3.12 (#146360)

1 parent [57cc1bd](#) commit cbba611

3 files changed +18 -0 lines changed

↑ Top ⚙

Filter files...

- Lib
  - test
    - test\_webbrowser.py
    - webbrowser.py
- Misc/NEWS.d/next/Security
  - 2026-01-16-12-04-49.gh-issue-143930.zYC5x3.rst

3 files changed +18 -0 lines changed

Search within code ⚙

Lib/test/test\_webbrowser.py

```

... @@ -59,6 +59,11 @@ def test_open(self):
59 59         options=[],
60 60         arguments=[URL])
61 61
62 +     def test_reject_dash_prefixes(self):
63 +         browser = self.browser_class(name=CMD_NAME)

```

```

64 +         with self.assertRaises(ValueError):
65 +             browser.open(f"--key=val {URL}")
66 +
62 67
63 68     class BackgroundBrowserCommandTest(CommandTestMixin, unittest.TestCase):
64 69

```

Lib/webbrowser.py

```

@@ -158,6 +158,12 @@ def open_new(self, url):
158 158     def open_new_tab(self, url):
159 159         return self.open(url, 2)
160 160
161 +     @staticmethod
162 +     def _check_url(url):
163 +         """Ensures that the URL is safe to pass to subprocesses as a
164 +         parameter"""
164 +         if url and url.lstrip().startswith("-"):
165 +             raise ValueError(f"Invalid URL: {url}")
166 +
161 167
162 168     class GenericBrowser(BaseBrowser):
163 169         """Class for all browsers started with a command
@@ -175,6 +181,7 @@ def __init__(self, name):
175 181
176 182     def open(self, url, new=0, autoraise=True):
177 183         sys.audit("webbrowser.open", url)
184 +         self._check_url(url)
178 185         cmdline = [self.name] + [arg.replace("%s", url)
179 186                     for arg in self.args]
180 187         try:
@@ -195,6 +202,7 @@ def open(self, url, new=0, autoraise=True):
195 202         cmdline = [self.name] + [arg.replace("%s", url)
196 203                     for arg in self.args]
197 204         sys.audit("webbrowser.open", url)
205 +         self._check_url(url)
198 206         try:
199 207             if sys.platform[:3] == 'win':
200 208                 p = subprocess.Popen(cmdline)

```

```

↑... @@ -260,6 +268,7 @@ def _invoke(self, args, remote, autoraise, url=None):
260 268
261 269     def open(self, url, new=0, autoraise=True):
262 270         sys.audit("webbrowser.open", url)
271 + self._check_url(url)
263 272     if new == 0:
264 273         action = self.remote_action
265 274     elif new == 1:
↓...
↑... @@ -350,6 +359,7 @@ class Konqueror(BaseBrowser):
350 359
351 360     def open(self, url, new=0, autoraise=True):
352 361         sys.audit("webbrowser.open", url)
362 + self._check_url(url)
353 363         # XXX Currently I know no way to prevent KFM from opening a new win.
354 364         if new == 2:
355 365             action = "newTab"
↓...
↑... @@ -554,6 +564,7 @@ def register_standard_browsers():
554 564     class WindowsDefault(BaseBrowser):
555 565         def open(self, url, new=0, autoraise=True):
556 566             sys.audit("webbrowser.open", url)
567 + self._check_url(url)
557 568         try:
558 569             os.startfile(url)
559 570         except OSError:
↓...
↑... @@ -638,6 +649,7 @@ def _name(self, val):
638 649
639 650     def open(self, url, new=0, autoraise=True):
640 651         sys.audit("webbrowser.open", url)
652 + self._check_url(url)
641 653         if self.name == 'default':
642 654             script = 'open location "%s"' % url.replace('"', '%22') # opens
        in default browser
643 655         else:
↓...

```

...6-01-16-12-04-49.gh-issue-143930.zYC5x3.rst



... @@ -0,0 +1 @@

1 + Reject leading dashes in URLs passed to `:func:`webbrowser.open``

**Comments** 0