

python / cpython Public

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

Commit ceac1ef



tomcruiseqi and sethmlarson authored 3 weeks ago · ✖ 61 / 74 · Partially verified

```
[3.11] gh-143930: Reject leading dashes in webbrowser URLs (GH-143931) (GH-146364)
(cherry picked from commit 82a24a4)
Co-authored-by: Seth Michael Larson <seth@python.org>
```

3.11 (#98846, #146364)

1 parent [86a67f8](#) commit [ceac1ef](#)



3 files changed +20 -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 +20 -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

```

@@ -155,6 +155,12 @@ def open_new(self, url):
155 155     def open_new_tab(self, url):
156 156         return self.open(url, 2)
157 157
158 +     @staticmethod
159 +     def _check_url(url):
160 +         """Ensures that the URL is safe to pass to subprocesses as a
161 +         parameter"""
162 +         if url and url.lstrip().startswith("-"):
163 +             raise ValueError(f"Invalid URL: {url}")
164
165     class GenericBrowser(BaseBrowser):
166         """Class for all browsers started with a command
@@ -172,6 +178,7 @@ def __init__(self, name):
172 178
173 179     def open(self, url, new=0, autoraise=True):
174 180         sys.audit("webbrowser.open", url)
181 +         self._check_url(url)
182         cmdline = [self.name] + [arg.replace("%s", url)
183                               for arg in self.args]
184         try:
@@ -192,6 +199,7 @@ def open(self, url, new=0, autoraise=True):
192 199         cmdline = [self.name] + [arg.replace("%s", url)
193                               for arg in self.args]
194 201         sys.audit("webbrowser.open", url)
202 +         self._check_url(url)
203         try:
204             if sys.platform[:3] == 'win':
205                 p = subprocess.Popen(cmdline)

```

↓ ↑	@@ -257,6 +265,7 @@ def _invoke(self, args, remote, autoraise, url=None):
257	265
258	266
259	267
268	267
260	269
261	270
262	271
↓ ↑	@@ -358,6 +367,7 @@ class Konqueror(BaseBrowser):
358	367
359	368
360	369
370	369
361	371
362	372
363	373
↓ ↑	@@ -442,6 +452,7 @@ def _remote(self, action):
442	452
443	453
444	454
455	454
445	456
446	457
447	458
↓ ↑	@@ -605,6 +616,7 @@ def register_standard_browsers():
605	616
606	617
607	618
619	618
608	620
609	621
610	622
↓ ↑	@@ -637,6 +649,7 @@ def __init__(self, name):
637	649
638	650

```

639 651 sys.audit("webbrowser.open", url)
652 + self._check_url(url)
640 653 assert "" not in url
641 654 # hack for local urls
642 655 if not ':' in url:
    ↓
    ↑ @@ -689,6 +702,7 @@ def _name(self, val):
689 702
690 703 def open(self, url, new=0, autoraise=True):
691 704 sys.audit("webbrowser.open", url)
705 + self._check_url(url)
692 706 if self.name == 'default':
693 707     script = 'open location "%s"' % url.replace('"', '%22') # opens
    in default browser
694 708 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