

python / cpython Public

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

Commit ad4d5ba



tomcruiseqi and sethmlarson authored 3 weeks ago · ✖ 55 / 66 · Partially verified

```
[3.10] gh-143930: Reject leading dashes in webbrowser URLs (GH-143931) (GH-146359)
(cherry picked from commit 82a24a4)
Co-authored-by: Seth Michael Larson <seth@python.org>
```

🔑 3.10 (#146359)

1 parent [ba6eba5](#) commit ad4d5ba



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
  ↑   @@ -57,6 +57,11 @@ def test_open(self):
57   57         options=[],
58   58         arguments=[URL])
59   59
60   + def test_reject_dash_prefixes(self):

```

```

61 +         browser = self.browser_class(name=CMD_NAME)
62 +         with self.assertRaises(ValueError):
63 +             browser.open(f"--key=val {URL}")
64 +
60 65
61 66     class BackgroundBrowserCommandTest(CommandTestMixin, unittest.TestCase):
62 67

```



Lib/webbrowser.py



```

↑... @@ -154,6 +154,12 @@ def open_new(self, url):
154 154     def open_new_tab(self, url):
155 155         return self.open(url, 2)
156 156
157 +     @staticmethod
158 +     def _check_url(url):
159 +         """Ensures that the URL is safe to pass to subprocesses as a
160 +         parameter"""
161 +         if url and url.lstrip().startswith("-"):
162 +             raise ValueError(f"Invalid URL: {url}")
163
157 163
158 164     class GenericBrowser(BaseBrowser):
159 165         """Class for all browsers started with a command
@@ -171,6 +177,7 @@ def __init__(self, name):
171 177
172 178     def open(self, url, new=0, autoraise=True):
173 179         sys.audit("webbrowser.open", url)
180 +         self._check_url(url)
174 181         cmdline = [self.name] + [arg.replace("%s", url)
175 182             for arg in self.args]
176 183         try:
@@ -191,6 +198,7 @@ def open(self, url, new=0, autoraise=True):
191 198         cmdline = [self.name] + [arg.replace("%s", url)
192 199             for arg in self.args]
193 200         sys.audit("webbrowser.open", url)
201 +         self._check_url(url)
194 202         try:
195 203             if sys.platform[:3] == 'win':
196 204                 p = subprocess.Popen(cmdline)

```

↓ ↑	@@ -256,6 +264,7 @@ def _invoke(self, args, remote, autoraise, url=None):
256	264
257	265
258	266
267	266
259	268
260	269
261	270
↓ ↑	@@ -357,6 +366,7 @@ class Konqueror(BaseBrowser):
357	366
358	367
359	368
369	368
360	370
361	371
362	372
↓ ↑	@@ -441,6 +451,7 @@ def _remote(self, action):
441	451
442	452
443	453
454	453
444	455
445	456
446	457
↓ ↑	@@ -603,6 +614,7 @@ def register_standard_browsers():
603	614
604	615
605	616
617	616
606	618
607	619
608	620
↓ ↑	@@ -633,6 +645,7 @@ def __init__(self, name):
633	645
634	646

```

635 647 sys.audit("webbrowser.open", url)
648 + self._check_url(url)
636 649 assert "" not in url
637 650 # hack for local urls
638 651 if not ':' in url:
@@ -670,6 +683,7 @@ def __init__(self, name):
670 683 self._name = name
671 684
672 685 def open(self, url, new=0, autoraise=True):
686 + self._check_url(url)
673 687 if self._name == 'default':
674 688     script = 'open location "%s"' % url.replace('"', '%22') # opens
        in default browser
675 689 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