

python / cpython Public

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

Commit 5af6ce3



3 people authored 8 hours ago · ✓ 42 / 45 · Verified

[3.13] [gh-146121](#): Clarify security model of pkgutil.getdata; revert checks ([GH-148197](#)) (#148205)

[gh-146121](#): Clarify security model of pkgutil.getdata; revert checks ([GH-148197](#))

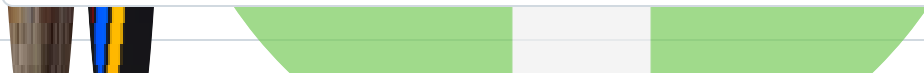
This reverts commit [bcdf231](#), and clarifies get_data's security model.

(cherry picked from commit [cf59bf7](#))

Co-authored-by: Petr Viktorin <encukou@gmail.com>
Co-authored-by: Stan Ulbrych <stan@python.org>

3.13 (AcreationOS-Linux/python#2, #148205)

1 parent [7fc8507](#) commit 5af6ce3



1 file changed +26 -2 lines changed

↑ Top ⚙️

Filter files...

Doc/library

pkgutil.rst

1 file changed +26 -2 lines changed

Search within code ⚙️

Doc/library/pkgutil.rst



```

@@ -191,24 +191,48 @@ support.
191 191      :meth:`get_data` <importlib.abc.ResourceLoader.get_data>` API. The
192 192      *package* argument should be the name of a package, in standard module
      format
193 193      (foo.bar). The *resource* argument should be in the form of a relative
194 - filename, using .. as the path separator. The parent directory name

```

195 - ``..`` is not allowed, and nor is a rooted name (starting with a ``/``).

194 + filename, using ``/`` as the path separator.

196 195

197 196 The function returns a binary string that is the contents of the specified
198 197 resource.

199 198

199 + This function uses the `:term:`loader`` method

200 + `:func:`~importlib.abc.FileLoader.get_data``

201 + to support modules installed in the filesystem, but also in zip files,

202 + databases, or elsewhere.

203 +

200 204 For packages located in the filesystem, which have already been imported,
201 205 this is the rough equivalent of::

202 206

203 207 `d = os.path.dirname(sys.modules[package].__file__)`

204 208 `data = open(os.path.join(d, resource), 'rb').read()`

205 209

210 + Like the `:func:`open`` function, `:func:`~get_data`` can follow parent

211 + directories (```../```) and absolute paths (starting with ```/``` or ```C:/```,
212 + for example).

213 + It can open compilation/installation artifacts like ```.py``` and ```.pyc```

214 + files or files with `:func:`reserved filenames <os.path.isreserved>``.

215 + To be compatible with non-filesystem loaders, avoid using these features.

216 +

217 + .. `warning::`

218 +

219 + This function is intended for trusted input.

220 + It does not verify that `*resource*` "belongs" to `*package*`.

221 +

222 + If you use a user-provided `*resource*` path, consider verifying it.

223 + For example, require an alphanumeric filename with a known extension, or

224 + install and check a list of known resources.

225 +

206 226 If the package cannot be located or loaded, or it uses a `:term:`loader``

207 227 which does not support `:meth:`get_data``

`<importlib.abc.ResourceLoader.get_data>``,

208 228 then ```None``` is returned. In particular, the `:term:`loader`` for

209 229 `:term:`namespace packages <namespace package>`` does not support

210 230 `:meth:`get_data <importlib.abc.ResourceLoader.get_data>``.

211 231

```
232 + .. seealso::
233 +
234 +     The :mod:`importlib.resources` module provides structured access to
235 +     module resources.
212 236
213 237 .. function:: resolve_name(name)
214 238
.....
↓
```

Comments 0