python / **cpython** Public

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

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit `73f03e4`

3 people authored on Sep 13, 2025 · ✕ 37 / 43 · Verified

```
[3.9] gh-130577: tarfile now validates archives to ensure member offsets are non-
negative (GH-137027) (GH-137645)

gh-130577: tarfile now validates archives to ensure member offsets are non-negative (GH-
137027)

(cherry picked from commit 7040aa5)

Co-authored-by: Alexander Urieles <aeurielesn@users.noreply.github.com>
Co-authored-by: Gregory P. Smith <greg@krypto.org>
```

· v3.9.25 ••• 3.9

1 parent 06fc882 commit 73f03e4

3 files changed +162 -0 lines changed

Top

Filter files…

- Lib
  - tarfile.py
  - test
    - test_tarfile.py
- Misc/NEWS.d/next/Library
  - 2025-07-23-00-35-29.gh-issue-130577.c7ElTy.rst

3 files changed +162 -0 lines changed

Search within code

Lib/tarfile.py

```
@@ -1602,6 +1602,9 @@ def _block(self, count):
1602 1602          """Round up a byte count by BLOCKSIZE and return it,
1603 1603             e.g. _block(834) => 1024.
1604 1604          """
     1605 +        # Only non-negative offsets are allowed
     1606 +        if count < 0:
     1607 +            raise InvalidHeaderError("invalid offset")
1605 1608          blocks, remainder = divmod(count, BLOCKSIZE)
1606 1609          if remainder:
1607 1610              blocks += 1
```

Lib/test/test_tarfile.py

```
@@ -48,6 +48,7 @@ def sha256sum(data):
48 48  xzname = os.path.join(TEMPDIR, "testtar.tar.xz")
49 49  tmpname = os.path.join(TEMPDIR, "tmp.tar")
50 50  dotlessname = os.path.join(TEMPDIR, "testtar")
    51 + SPACE = b" "
51 52
52 53  sha256_regtype = (
53 54      "e09e4bc8b3c9d9177e77256353b36c159f5f040531bbd4b024a8f9b9196c71ce"
@@ -4234,6 +4235,161 @@ def valueerror_filter(tarinfo, path):
4234 4235              self.expect_exception(TypeError)  # errorlevel is not int
4235 4236
4236 4237
    4238 + class OffsetValidationTests(unittest.TestCase):
    4239 +     tarname = tmpname
    4240 +     invalid_posix_header = (
    4241 +         # name: 100 bytes
    4242 +         tarfile.NUL * tarfile.LENGTH_NAME
    4243 +         # mode, space, null terminator: 8 bytes
    4244 +         + b"000755" + SPACE + tarfile.NUL
    4245 +         # uid, space, null terminator: 8 bytes
    4246 +         + b"000001" + SPACE + tarfile.NUL
    4247 +         # gid, space, null terminator: 8 bytes
    4248 +         + b"000001" + SPACE + tarfile.NUL
    4249 +         # size, space: 12 bytes
```

```
4250 +          + b"\xff" * 11 + SPACE
4251 +          # mtime, space: 12 bytes
4252 +          + tarfile.NUL * 11 + SPACE
4253 +          # chksum: 8 bytes
4254 +          + b"0011407" + tarfile.NUL
4255 +          # type: 1 byte
4256 +          + tarfile.REGTYPE
4257 +          # linkname: 100 bytes
4258 +          + tarfile.NUL * tarfile.LENGTH_LINK
4259 +          # magic: 6 bytes, version: 2 bytes
4260 +          + tarfile.POSIX_MAGIC
4261 +          # uname: 32 bytes
4262 +          + tarfile.NUL * 32
4263 +          # gname: 32 bytes
4264 +          + tarfile.NUL * 32
4265 +          # devmajor, space, null terminator: 8 bytes
4266 +          + tarfile.NUL * 6 + SPACE + tarfile.NUL
4267 +          # devminor, space, null terminator: 8 bytes
4268 +          + tarfile.NUL * 6 + SPACE + tarfile.NUL
4269 +          # prefix: 155 bytes
4270 +          + tarfile.NUL * tarfile.LENGTH_PREFIX
4271 +          # padding: 12 bytes
4272 +          + tarfile.NUL * 12
4273 +      )
4274 +      invalid_gnu_header = (
4275 +          # name: 100 bytes
4276 +          tarfile.NUL * tarfile.LENGTH_NAME
4277 +          # mode, null terminator: 8 bytes
4278 +          + b"0000755" + tarfile.NUL
4279 +          # uid, null terminator: 8 bytes
4280 +          + b"0000001" + tarfile.NUL
4281 +          # gid, space, null terminator: 8 bytes
4282 +          + b"0000001" + tarfile.NUL
4283 +          # size, space: 12 bytes
4284 +          + b"\xff" * 11 + SPACE
4285 +          # mtime, space: 12 bytes
4286 +          + tarfile.NUL * 11 + SPACE
4287 +          # chksum: 8 bytes
4288 +          + b"0011327" + tarfile.NUL
4289 +          # type: 1 byte
```

```
4290 +         + tarfile.REGTYPE
4291 +         # linkname: 100 bytes
4292 +         + tarfile.NUL * tarfile.LENGTH_LINK
4293 +         # magic: 8 bytes
4294 +         + tarfile.GNU_MAGIC
4295 +         # uname: 32 bytes
4296 +         + tarfile.NUL * 32
4297 +         # gname: 32 bytes
4298 +         + tarfile.NUL * 32
4299 +         # devmajor, null terminator: 8 bytes
4300 +         + tarfile.NUL * 8
4301 +         # devminor, null terminator: 8 bytes
4302 +         + tarfile.NUL * 8
4303 +         # padding: 167 bytes
4304 +         + tarfile.NUL * 167
4305 +     )
4306 +     invalid_v7_header = (
4307 +         # name: 100 bytes
4308 +         tarfile.NUL * tarfile.LENGTH_NAME
4309 +         # mode, space, null terminator: 8 bytes
4310 +         + b"000755" + SPACE + tarfile.NUL
4311 +         # uid, space, null terminator: 8 bytes
4312 +         + b"000001" + SPACE + tarfile.NUL
4313 +         # gid, space, null terminator: 8 bytes
4314 +         + b"000001" + SPACE + tarfile.NUL
4315 +         # size, space: 12 bytes
4316 +         + b"\xff" * 11 + SPACE
4317 +         # mtime, space: 12 bytes
4318 +         + tarfile.NUL * 11 + SPACE
4319 +         # chksum: 8 bytes
4320 +         + b"0010070" + tarfile.NUL
4321 +         # type: 1 byte
4322 +         + tarfile.REGTYPE
4323 +         # linkname: 100 bytes
4324 +         + tarfile.NUL * tarfile.LENGTH_LINK
4325 +         # padding: 255 bytes
4326 +         + tarfile.NUL * 255
4327 +     )
4328 +     valid_gnu_header = tarfile.TarInfo("filename").tobuf(tarfile.GNU_FORMAT)
4329 +     data_block = b"\xff" * tarfile.BLOCKSIZE
```

```
4330 +
4331 +     def _write_buffer(self, buffer):
4332 +         with open(self.tarname, "wb") as f:
4333 +             f.write(buffer)
4334 +
4335 +     def _get_members(self, ignore_zeros=None):
4336 +         with open(self.tarname, "rb") as f:
4337 +             with tarfile.open(
4338 +                 mode="r", fileobj=f, ignore_zeros=ignore_zeros
4339 +             ) as tar:
4340 +                 return tar.getmembers()
4341 +
4342 +     def _assert_raises_read_error_exception(self):
4343 +         with self.assertRaisesRegex(
4344 +             tarfile.ReadError, "file could not be opened successfully"
4345 +         ):
4346 +             self._get_members()
4347 +
4348 +     def test_invalid_offset_header_validations(self):
4349 +         for tar_format, invalid_header in (
4350 +             ("posix", self.invalid_posix_header),
4351 +             ("gnu", self.invalid_gnu_header),
4352 +             ("v7", self.invalid_v7_header),
4353 +         ):
4354 +             with self.subTest(format=tar_format):
4355 +                 self._write_buffer(invalid_header)
4356 +                 self._assert_raises_read_error_exception()
4357 +
4358 +     def test_early_stop_at_invalid_offset_header(self):
4359 +         buffer = self.valid_gnu_header + self.invalid_gnu_header +
     self.valid_gnu_header
4360 +         self._write_buffer(buffer)
4361 +         members = self._get_members()
4362 +         self.assertEqual(len(members), 1)
4363 +         self.assertEqual(members[0].name, "filename")
4364 +         self.assertEqual(members[0].offset, 0)
4365 +
4366 +     def test_ignore_invalid_archive(self):
4367 +         # 3 invalid headers with their respective data
4368 +         buffer = (self.invalid_gnu_header + self.data_block) * 3
```

```
4369 +         self._write_buffer(buffer)
4370 +         members = self._get_members(ignore_zeros=True)
4371 +         self.assertEqual(len(members), 0)
4372 +
4373 +     def test_ignore_invalid_offset_headers(self):
4374 +         for first_block, second_block, expected_offset in (
4375 +             (
4376 +                 (self.valid_gnu_header),
4377 +                 (self.invalid_gnu_header + self.data_block),
4378 +                 0,
4379 +             ),
4380 +             (
4381 +                 (self.invalid_gnu_header + self.data_block),
4382 +                 (self.valid_gnu_header),
4383 +                 1024,
4384 +             ),
4385 +         ):
4386 +             self._write_buffer(first_block + second_block)
4387 +             members = self._get_members(ignore_zeros=True)
4388 +             self.assertEqual(len(members), 1)
4389 +             self.assertEqual(members[0].name, "filename")
4390 +             self.assertEqual(members[0].offset, expected_offset)
4391 +
4392 +
4237 4393   def setUpModule():
4238 4394       support.unlink(TEMPDIR)
4239 4395       os.makedirs(TEMPDIR)
```

...5-07-23-00-35-29.gh-issue-130577.c7EITy.rst

```
@@ -0,0 +1,3 @@
1 + :mod:`tarfile` now validates archives to ensure member offsets are
2 + non-negative.  (Contributed by Alexander Enrique Urieles Nieto in
3 + :gh:`130577`.)
```

Comments 0