python / **cpython** Public

Code Issues 5k+ Pull requests 2.1k Actions Projects Security and q

Commit b4ec174

3 people authored on Aug 19, 2025 · ✕ 65 / 71 · Verified

```
[3.11] gh-130577: tarfile now validates archives to ensure member offsets are
non-negative (GH-137027) (#137172)

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>
```

3.11 (#98846, #137172) · v3.11.15 v3.11.14

1 parent 3511c2e commit b4ec174

3 files changed +162

Top

Filter files…

- Lib
  - tarfile.py
  - test
    - test_tarfile.py
- Misc/NEWS.d/next/Library
  - 2025-07-23-00-35-29.gh-issue-130577.c7EITy.rst

Search within code

Lib/tarfile.py

```
@@ -1614,6 +1614,9 @@ def _block(self, count):
```

```
1614  1614          """Round up a byte count by BLOCKSIZE and return it,
1615  1615             e.g. _block(834) => 1024.
1616  1616          """
      1617  +       # Only non-negative offsets are allowed
      1618  +       if count < 0:
      1619  +           raise InvalidHeaderError("invalid offset")
1617  1620          blocks, remainder = divmod(count, BLOCKSIZE)
1618  1621          if remainder:
1619  1622              blocks += 1
```

Lib/test/test_tarfile.py

```
      @@ -50,6 +50,7 @@ def sha256sum(data):
50    50      xzname = os.path.join(TEMPDIR, "testtar.tar.xz")
51    51      tmpname = os.path.join(TEMPDIR, "tmp.tar")
52    52      dotlessname = os.path.join(TEMPDIR, "testtar")
      53  +   SPACE = b" "
53    54
54    55      sha256_regtype = (
55    56          "e09e4bc8b3c9d9177e77256353b36c159f5f040531bbd4b024a8f9b9196c71ce"
      @@ -4386,6 +4387,161 @@ def extractall(self, ar):
4386  4387          ar.extractall(self.testdir, filter='fully_trusted')
4387  4388
4388  4389
      4390  + class OffsetValidationTests(unittest.TestCase):
      4391  +     tarname = tmpname
      4392  +     invalid_posix_header = (
      4393  +         # name: 100 bytes
      4394  +         tarfile.NUL * tarfile.LENGTH_NAME
      4395  +         # mode, space, null terminator: 8 bytes
      4396  +         + b"000755" + SPACE + tarfile.NUL
      4397  +         # uid, space, null terminator: 8 bytes
      4398  +         + b"000001" + SPACE + tarfile.NUL
      4399  +         # gid, space, null terminator: 8 bytes
      4400  +         + b"000001" + SPACE + tarfile.NUL
      4401  +         # size, space: 12 bytes
      4402  +         + b"\xff" * 11 + SPACE
      4403  +         # mtime, space: 12 bytes
      4404  +         + tarfile.NUL * 11 + SPACE
```

```
4405 +         # chksum: 8 bytes
4406 +         + b"0011407" + tarfile.NUL
4407 +         # type: 1 byte
4408 +         + tarfile.REGTYPE
4409 +         # linkname: 100 bytes
4410 +         + tarfile.NUL * tarfile.LENGTH_LINK
4411 +         # magic: 6 bytes, version: 2 bytes
4412 +         + tarfile.POSIX_MAGIC
4413 +         # uname: 32 bytes
4414 +         + tarfile.NUL * 32
4415 +         # gname: 32 bytes
4416 +         + tarfile.NUL * 32
4417 +         # devmajor, space, null terminator: 8 bytes
4418 +         + tarfile.NUL * 6 + SPACE + tarfile.NUL
4419 +         # devminor, space, null terminator: 8 bytes
4420 +         + tarfile.NUL * 6 + SPACE + tarfile.NUL
4421 +         # prefix: 155 bytes
4422 +         + tarfile.NUL * tarfile.LENGTH_PREFIX
4423 +         # padding: 12 bytes
4424 +         + tarfile.NUL * 12
4425 +     )
4426 +     invalid_gnu_header = (
4427 +         # name: 100 bytes
4428 +         tarfile.NUL * tarfile.LENGTH_NAME
4429 +         # mode, null terminator: 8 bytes
4430 +         + b"0000755" + tarfile.NUL
4431 +         # uid, null terminator: 8 bytes
4432 +         + b"0000001" + tarfile.NUL
4433 +         # gid, space, null terminator: 8 bytes
4434 +         + b"0000001" + tarfile.NUL
4435 +         # size, space: 12 bytes
4436 +         + b"\xff" * 11 + SPACE
4437 +         # mtime, space: 12 bytes
4438 +         + tarfile.NUL * 11 + SPACE
4439 +         # chksum: 8 bytes
4440 +         + b"0011327" + tarfile.NUL
4441 +         # type: 1 byte
4442 +         + tarfile.REGTYPE
4443 +         # linkname: 100 bytes
4444 +         + tarfile.NUL * tarfile.LENGTH_LINK
```

```
4445 +         # magic: 8 bytes
4446 +         + tarfile.GNU_MAGIC
4447 +         # uname: 32 bytes
4448 +         + tarfile.NUL * 32
4449 +         # gname: 32 bytes
4450 +         + tarfile.NUL * 32
4451 +         # devmajor, null terminator: 8 bytes
4452 +         + tarfile.NUL * 8
4453 +         # devminor, null terminator: 8 bytes
4454 +         + tarfile.NUL * 8
4455 +         # padding: 167 bytes
4456 +         + tarfile.NUL * 167
4457 +     )
4458 +     invalid_v7_header = (
4459 +         # name: 100 bytes
4460 +         tarfile.NUL * tarfile.LENGTH_NAME
4461 +         # mode, space, null terminator: 8 bytes
4462 +         + b"000755" + SPACE + tarfile.NUL
4463 +         # uid, space, null terminator: 8 bytes
4464 +         + b"000001" + SPACE + tarfile.NUL
4465 +         # gid, space, null terminator: 8 bytes
4466 +         + b"000001" + SPACE + tarfile.NUL
4467 +         # size, space: 12 bytes
4468 +         + b"\xff" * 11 + SPACE
4469 +         # mtime, space: 12 bytes
4470 +         + tarfile.NUL * 11 + SPACE
4471 +         # chksum: 8 bytes
4472 +         + b"0010070" + tarfile.NUL
4473 +         # type: 1 byte
4474 +         + tarfile.REGTYPE
4475 +         # linkname: 100 bytes
4476 +         + tarfile.NUL * tarfile.LENGTH_LINK
4477 +         # padding: 255 bytes
4478 +         + tarfile.NUL * 255
4479 +     )
4480 +     valid_gnu_header = tarfile.TarInfo("filename").tobuf(tarfile.GNU_FORMAT)
4481 +     data_block = b"\xff" * tarfile.BLOCKSIZE
4482 +
4483 +     def _write_buffer(self, buffer):
4484 +         with open(self.tarname, "wb") as f:
```

```
4485 +             f.write(buffer)
4486 +
4487 +     def _get_members(self, ignore_zeros=None):
4488 +         with open(self.tarname, "rb") as f:
4489 +             with tarfile.open(
4490 +                 mode="r", fileobj=f, ignore_zeros=ignore_zeros
4491 +             ) as tar:
4492 +                 return tar.getmembers()
4493 +
4494 +     def _assert_raises_read_error_exception(self):
4495 +         with self.assertRaisesRegex(
4496 +             tarfile.ReadError, "file could not be opened successfully"
4497 +         ):
4498 +             self._get_members()
4499 +
4500 +     def test_invalid_offset_header_validations(self):
4501 +         for tar_format, invalid_header in (
4502 +             ("posix", self.invalid_posix_header),
4503 +             ("gnu", self.invalid_gnu_header),
4504 +             ("v7", self.invalid_v7_header),
4505 +         ):
4506 +             with self.subTest(format=tar_format):
4507 +                 self._write_buffer(invalid_header)
4508 +                 self._assert_raises_read_error_exception()
4509 +
4510 +     def test_early_stop_at_invalid_offset_header(self):
4511 +         buffer = self.valid_gnu_header + self.invalid_gnu_header +
     self.valid_gnu_header
4512 +         self._write_buffer(buffer)
4513 +         members = self._get_members()
4514 +         self.assertEqual(len(members), 1)
4515 +         self.assertEqual(members[0].name, "filename")
4516 +         self.assertEqual(members[0].offset, 0)
4517 +
4518 +     def test_ignore_invalid_archive(self):
4519 +         # 3 invalid headers with their respective data
4520 +         buffer = (self.invalid_gnu_header + self.data_block) * 3
4521 +         self._write_buffer(buffer)
4522 +         members = self._get_members(ignore_zeros=True)
4523 +         self.assertEqual(len(members), 0)
```

```
4524 +
4525 +     def test_ignore_invalid_offset_headers(self):
4526 +         for first_block, second_block, expected_offset in (
4527 +             (
4528 +                 (self.valid_gnu_header),
4529 +                 (self.invalid_gnu_header + self.data_block),
4530 +                 0,
4531 +             ),
4532 +             (
4533 +                 (self.invalid_gnu_header + self.data_block),
4534 +                 (self.valid_gnu_header),
4535 +                 1024,
4536 +             ),
4537 +         ):
4538 +             self._write_buffer(first_block + second_block)
4539 +             members = self._get_members(ignore_zeros=True)
4540 +             self.assertEqual(len(members), 1)
4541 +             self.assertEqual(members[0].name, "filename")
4542 +             self.assertEqual(members[0].offset, expected_offset)
4543 +
4544 +
4389 4545     def setUpModule():
4390 4546         os_helper.unlink(TEMPDIR)
4391 4547         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