python / **cpython** Public

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

Commit cdae923

3 people authored on Jul 28, 2025 · 103 / 106 · Verified

```
[3.13] gh-130577: tarfile now validates archives to ensure member offsets are
non-negative (GH-137027) (#137170)

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>
```

3.13 (AcreetionOS-Linux/python#2, #137170) · v3.13.13 ••• v3.13.6

1 parent 3f57d9b commit cdae923

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

```
@@ -1636,6 +1636,9 @@ def _block(self, count):
1636  1636          """Round up a byte count by BLOCKSIZE and return it,
1637  1637             e.g. _block(834) => 1024.
```

```
1638  1638            """
      1639  +         # Only non-negative offsets are allowed
      1640  +         if count < 0:
      1641  +             raise InvalidHeaderError("invalid offset")
1639  1642            blocks, remainder = divmod(count, BLOCKSIZE)
1640  1643            if remainder:
1641  1644                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"
      @@ -4578,6 +4579,161 @@ def extractall(self, ar):
4578  4579            ar.extractall(self.testdir, filter='fully_trusted')
4579  4580
4580  4581
      4582  + class OffsetValidationTests(unittest.TestCase):
      4583  +     tarname = tmpname
      4584  +     invalid_posix_header = (
      4585  +         # name: 100 bytes
      4586  +         tarfile.NUL * tarfile.LENGTH_NAME
      4587  +         # mode, space, null terminator: 8 bytes
      4588  +         + b"000755" + SPACE + tarfile.NUL
      4589  +         # uid, space, null terminator: 8 bytes
      4590  +         + b"000001" + SPACE + tarfile.NUL
      4591  +         # gid, space, null terminator: 8 bytes
      4592  +         + b"000001" + SPACE + tarfile.NUL
      4593  +         # size, space: 12 bytes
      4594  +         + b"\xff" * 11 + SPACE
      4595  +         # mtime, space: 12 bytes
      4596  +         + tarfile.NUL * 11 + SPACE
      4597  +         # chksum: 8 bytes
      4598  +         + b"0011407" + tarfile.NUL
```

```
4599 +         # type: 1 byte
4600 +         + tarfile.REGTYPE
4601 +         # linkname: 100 bytes
4602 +         + tarfile.NUL * tarfile.LENGTH_LINK
4603 +         # magic: 6 bytes, version: 2 bytes
4604 +         + tarfile.POSIX_MAGIC
4605 +         # uname: 32 bytes
4606 +         + tarfile.NUL * 32
4607 +         # gname: 32 bytes
4608 +         + tarfile.NUL * 32
4609 +         # devmajor, space, null terminator: 8 bytes
4610 +         + tarfile.NUL * 6 + SPACE + tarfile.NUL
4611 +         # devminor, space, null terminator: 8 bytes
4612 +         + tarfile.NUL * 6 + SPACE + tarfile.NUL
4613 +         # prefix: 155 bytes
4614 +         + tarfile.NUL * tarfile.LENGTH_PREFIX
4615 +         # padding: 12 bytes
4616 +         + tarfile.NUL * 12
4617 +     )
4618 +     invalid_gnu_header = (
4619 +         # name: 100 bytes
4620 +         tarfile.NUL * tarfile.LENGTH_NAME
4621 +         # mode, null terminator: 8 bytes
4622 +         + b"0000755" + tarfile.NUL
4623 +         # uid, null terminator: 8 bytes
4624 +         + b"0000001" + tarfile.NUL
4625 +         # gid, space, null terminator: 8 bytes
4626 +         + b"0000001" + tarfile.NUL
4627 +         # size, space: 12 bytes
4628 +         + b"\xff" * 11 + SPACE
4629 +         # mtime, space: 12 bytes
4630 +         + tarfile.NUL * 11 + SPACE
4631 +         # chksum: 8 bytes
4632 +         + b"0011327" + tarfile.NUL
4633 +         # type: 1 byte
4634 +         + tarfile.REGTYPE
4635 +         # linkname: 100 bytes
4636 +         + tarfile.NUL * tarfile.LENGTH_LINK
4637 +         # magic: 8 bytes
4638 +         + tarfile.GNU_MAGIC
```

```
4639 +         # uname: 32 bytes
4640 +         + tarfile.NUL * 32
4641 +         # gname: 32 bytes
4642 +         + tarfile.NUL * 32
4643 +         # devmajor, null terminator: 8 bytes
4644 +         + tarfile.NUL * 8
4645 +         # devminor, null terminator: 8 bytes
4646 +         + tarfile.NUL * 8
4647 +         # padding: 167 bytes
4648 +         + tarfile.NUL * 167
4649 +     )
4650 +     invalid_v7_header = (
4651 +         # name: 100 bytes
4652 +         tarfile.NUL * tarfile.LENGTH_NAME
4653 +         # mode, space, null terminator: 8 bytes
4654 +         + b"000755" + SPACE + tarfile.NUL
4655 +         # uid, space, null terminator: 8 bytes
4656 +         + b"000001" + SPACE + tarfile.NUL
4657 +         # gid, space, null terminator: 8 bytes
4658 +         + b"000001" + SPACE + tarfile.NUL
4659 +         # size, space: 12 bytes
4660 +         + b"\xff" * 11 + SPACE
4661 +         # mtime, space: 12 bytes
4662 +         + tarfile.NUL * 11 + SPACE
4663 +         # chksum: 8 bytes
4664 +         + b"0010070" + tarfile.NUL
4665 +         # type: 1 byte
4666 +         + tarfile.REGTYPE
4667 +         # linkname: 100 bytes
4668 +         + tarfile.NUL * tarfile.LENGTH_LINK
4669 +         # padding: 255 bytes
4670 +         + tarfile.NUL * 255
4671 +     )
4672 +     valid_gnu_header = tarfile.TarInfo("filename").tobuf(tarfile.GNU_FORMAT)
4673 +     data_block = b"\xff" * tarfile.BLOCKSIZE
4674 +
4675 +     def _write_buffer(self, buffer):
4676 +         with open(self.tarname, "wb") as f:
4677 +             f.write(buffer)
4678 +
```

```
4679 +     def _get_members(self, ignore_zeros=None):
4680 +         with open(self.tarname, "rb") as f:
4681 +             with tarfile.open(
4682 +                 mode="r", fileobj=f, ignore_zeros=ignore_zeros
4683 +             ) as tar:
4684 +                 return tar.getmembers()
4685 +
4686 +     def _assert_raises_read_error_exception(self):
4687 +         with self.assertRaisesRegex(
4688 +             tarfile.ReadError, "file could not be opened successfully"
4689 +         ):
4690 +             self._get_members()
4691 +
4692 +     def test_invalid_offset_header_validations(self):
4693 +         for tar_format, invalid_header in (
4694 +             ("posix", self.invalid_posix_header),
4695 +             ("gnu", self.invalid_gnu_header),
4696 +             ("v7", self.invalid_v7_header),
4697 +         ):
4698 +             with self.subTest(format=tar_format):
4699 +                 self._write_buffer(invalid_header)
4700 +                 self._assert_raises_read_error_exception()
4701 +
4702 +     def test_early_stop_at_invalid_offset_header(self):
4703 +         buffer = self.valid_gnu_header + self.invalid_gnu_header +
     self.valid_gnu_header
4704 +         self._write_buffer(buffer)
4705 +         members = self._get_members()
4706 +         self.assertEqual(len(members), 1)
4707 +         self.assertEqual(members[0].name, "filename")
4708 +         self.assertEqual(members[0].offset, 0)
4709 +
4710 +     def test_ignore_invalid_archive(self):
4711 +         # 3 invalid headers with their respective data
4712 +         buffer = (self.invalid_gnu_header + self.data_block) * 3
4713 +         self._write_buffer(buffer)
4714 +         members = self._get_members(ignore_zeros=True)
4715 +         self.assertEqual(len(members), 0)
4716 +
4717 +     def test_ignore_invalid_offset_headers(self):
```

```
                4718 +         for first_block, second_block, expected_offset in (
                4719 +             (
                4720 +                 (self.valid_gnu_header),
                4721 +                 (self.invalid_gnu_header + self.data_block),
                4722 +                 0,
                4723 +             ),
                4724 +             (
                4725 +                 (self.invalid_gnu_header + self.data_block),
                4726 +                 (self.valid_gnu_header),
                4727 +                 1024,
                4728 +             ),
                4729 +         ):
                4730 +             self._write_buffer(first_block + second_block)
                4731 +             members = self._get_members(ignore_zeros=True)
                4732 +             self.assertEqual(len(members), 1)
                4733 +             self.assertEqual(members[0].name, "filename")
                4734 +             self.assertEqual(members[0].offset, expected_offset)
                4735 +
                4736 +
4581            4737     def setUpModule():
4582            4738         os_helper.unlink(TEMPDIR)
4583            4739         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