» Issue36260



This issue tracker **has been migrated to GitHub**, and is currently **read-only**.
For more information, **see the GitHub FAQs in the Python's Developer Guide.**



This issue has been migrated to GitHub:
https://github.com/python/cpython/issues/80441

classification

Title: [security] CVE-2019-9674: Zip Bomb vulnerability	
Type: security	**Stage**: resolved
Components: Library (Lib)	**Versions**: Python 3.8, Python 3.7, Python 3.6, Python 3.4, Python 3.5

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To:	**Nosy List**: 18z, Victor Kung, christian.heimes, jaraco, krnick, serhiy.storchaka, vstinner, xtreak
Priority: normal	**Keywords**: patch

Created on **2019-03-11 07:16** by **krnick**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 13378	merged	krnick, 2019-05-17 07:50	
PR 15976	merged	miss-islington, 2019-09-11 15:04	

Messages (19)

msg337650 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-03-11 07:16**

Dear Python Community,

We've found a vulnerability in cpython Lib and already received a cve number (CVE-2019-9674)
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9674

We also have a patch for this vulnerability, please tell us what to do next.
Since we don't want to uncover the vulnerability before it get fixed.

JUN-WEI SONG

msg337651 - (view) **Author: KunYu Chen (18z) *** **Date: 2019-03-11 07:19**

Dear community,

I am one of the discoverer of this vulnerability, please tell us what to do next :D

Kunyu Chen

msg337652 - (view) **Author: Karthikeyan Singaravelan (xtreak) *** **Date: 2019-03-11 07:22**

You can find the process to report security vulnerabilities at **https://www.python.org/news/security/** . Please email the details to **security@python.org** and who will analyze the report before public disclosure.

msg337835 - (view) **Author: KunYu Chen (18z) *** **Date: 2019-03-13 05:39**

Thank you Karthikeyan Singaravelan.
We're working on it :D

Kunyu Chen

msg339061 - (view) **Author: Christian Heimes (christian.heimes) *** **Date: 2019-03-28 16:54**

Issue **~~#36462~~** contains more information. The reporter claims that the zipfile module is inherent insecure because it does not provide any heuristics to make zipbomb attacks harder.

I'm -1 to implement such a heuristic. The zipfile module is a low level module and should not limit extraction by defaykt. Instead we should improve documentation and maybe implement some method that simplifies detection of zipbomb attacks. I'm thinking about a method that returns total count of files, total compressed size and total uncompressed size.

msg339062 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2019-03-28 17:05**

All these are simple one-liners:

```
    len(zf.infolist())
    sum(zi.compress_size for zi in zf.infolist())
    sum(zi.file_size for zi in zf.infolist())
```

msg339083 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-03-28 23:46**

Thank you python community, these two issues are indeed the same problem.

I also think that it is good to make a related document to reduce such problems.

msg339084 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-03-28 23:46**

Thank you python community, these two issues are indeed the same problem.

I also think that it is good to make a related document to reduce such problems.

msg339087 - (view) **Author: KunYu Chen (18z) *** **Date: 2019-03-29 00:20**

Thank you for the responses.

I agree with Christian Heimes.

It's indeed better to improve the documentation rather than directly implement the heuristic.

msg339316 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-04-02 06:14**

Hello Python community,

With Christian Heimes' suggestion, we manipulate appropriate warning to inform users that they may encounter zip bomb issues when using the zipfile module.

The warning we would like to add in the zipfile documentation is shown below :

https://github.com/python/cpython/blob/3.7/Doc/library/zipfile.rst

```
   .. warning::

    Never extract files from untrusted sources without prior
    inspection. It is possible that the file may contain zip bomb
    issues such as 42.zip. The zip bomb will usually be a small file
    before decompression, but once it is decompressed, it will
    exhaust system resources.
```

You can protect your system by limiting system resources, limiting compression ratio (zip bombs are usually quite high), and checking for nested zip files.

We are also pleasure to provide a patch to enhance the zipfile module to provide basic information.

In zipfile.py

https://github.com/python/cpython/blob/master/Lib/zipfile.py

Inside the ZipFile class :

```
def filecount(self):
    """Return total count of files in the archive."""
```

```
    return len(self.filelist)

def total_compressed_size(self):
    """Return total compressed size in the archive."""
    return sum([data.compress_size for data in self.filelist])

def total_uncompressed_size(self):
    """Return total uncompressed size in the archive."""
    return sum([data.file_size for data in self.filelist])
```

msg339329 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2019-04-02 11:09**

I am against such trivial methods in ZipFile. Its interface is already complicate. The advantage of Python is that you do not need tons of methods for every possible query -- you can just combine few Python features into a one-line expression.

As for the documentation change, it could be useful to add more general note about possible pitfalls. What happen when interrupt extracting or adding to the archive, what happen when extract into existing tree or overwrite an existing file, what happen when the file system does not support some file names, what happen when extract to case-insensitive file system, what happen when extract encrypted file with wrong password, etc. We do not have to tell the user what he should not do, just to warn about the possible consequences.

msg339406 - (view) **Author: Victor Kung (Victor Kung)** **Date: 2019-04-03 17:52**

Hello Python community,

I’m curious why the patch or pitfall prevention in ZipFile
 are not suggested. I have no idea if everyone read documentation in detail. It seems straightforward to add the methods in ZipFile with well documented rather than just warn in documentation. Any comment would be appreciated.

Victor Kung

msg339408 - (view) **Author: Christian Heimes (christian.heimes) *** **Date: 2019-04-03 18:09**

The suggested approach is merely a heuristic that reduces the impact of a zipbomb. An attacker can circumvent the heuristic. In best case scenario, the approach just increases the cost factor for a successful DoS. For example an attacker may have to upload 10 larger zip files instead of one smaller zip file to fill up the disk space of a server.

The correct approach is to always verify all data from untrusted sources. It's the 101 of application security.

msg339587 - (view) **Author: Victor Kung (Victor Kung)** **Date: 2019-04-08 02:32**

I see.

@Christian Heimes Thank you for the response.

msg341256 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-05-02 08:22**

Thank you very much for your reply.

Based on discussions above, consensuses are improving the zipfile documentation.

And we (JUN-WEI SONG & KunYu Chen) would like to work on this.

With opinions of Serhiy Storchaka, Christian Heimes and the ideas we have, possible pitfalls are listed below.

1. From file itself:

   Decompression may fail due to an incorrect password, an incorrect CRC checksum, an incorrect PKZIP format, an unsupported compression method, or an unsupported decryption.

2. File system:

   Each file system has different limitations such as allowable characters in directory entries, the max length of file name, the max length of path name, the max size of single file, the max number of files, the max number of files in a single directory, etc. Decompression will fail as long as these limitations are exceeded.

3. Operating system:

   The lack of memory or disk space would lead to decompression failed (see also Zip Bomb).

4. Interrupt:

   Users should be careful in interrupting the process of decompression, such as control-C or killing the process during decompression, which may result in incomplete decompression of the archive.

5. Different default behaviors:

   Users should figure out different default extraction behaviors, such as when extracting into the existing tree, it will overwriting an existing file without asking, or when in a case-insensitive file system, it keeps only one file when extracting an archive which contains many files that have the same name but different case.

Please let us know if anything’s missing.

msg342693 - (view) **Author: JUN-WEI SONG (krnick) *** **Date: 2019-05-17 07:59**

Dear friends,

We moved a little bit forward to improve the writing. :)

msg351921 - (view) **Author: Jason R. Coombs (jaraco) *** **Date: 2019-09-11 15:04**

New changeset **3ba51d587f6897a45301ce9126300c14fcd4eba2** by Jason R. Coombs (JunWei Song) in branch 'master':
~~**bpo-36260**~~: Add pitfalls to zipfile module documentation (**#13378**)
https://github.com/python/cpython/commit/3ba51d587f6897a45301ce9126300c14fcd4eba2

msg351964 - (view) **Author: Jason R. Coombs (jaraco) *** **Date: 2019-09-11 16:03**

New changeset **c5a672315dffbc95acc1ca28584ec84ddb56626f** by Jason R. Coombs (Miss Islington (bot)) in branch '3.8':
~~**bpo-36260**~~: Add pitfalls to zipfile module documentation (~~**GH-13378**~~) (~~**GH-15976**~~)
https://github.com/python/cpython/commit/c5a672315dffbc95acc1ca28584ec84ddb56626f

msg361673 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-02-10 07:59**

I marked ~~**bpo-39341**~~ as a duplicate of this issue: "[security] zipfile: ZIP Bomb vulnerability, don't check announced uncompressed size".

History

Date	User	Action	Args
2022-04-11 14:59:12	admin	set	github: 80441
2020-02-10 07:59:41	vstinner	set	messages: + **msg361673**
2020-02-10 07:59:22	vstinner	link	**issue39341 superseder**
2019-09-11 16:03:21	jaraco	set	nosy: + **jaraco** messages: + **msg351964**
2019-09-11 15:04:49	jaraco	set	status: open -> closed nosy: - **jaraco** resolution: remind -> fixed stage: patch review -> resolved
2019-09-11 15:04:37	miss-islington	set	pull_requests: + **pull_request15609**
2019-09-11 15:04:14	jaraco	set	nosy: + **jaraco** messages: + **msg351921**
2019-05-17 09:00:53	vstinner	set	title: Zip Bomb vulnerability -> [security] CVE-2019-9674: Zip Bomb vulnerability
2019-05-17 07:59:20	krnick	set	messages: + **msg342693**
2019-05-17 07:50:04	krnick	set	keywords: + **patch** stage: resolved -> patch review pull_requests: + **pull_request13288**
2019-05-02 13:06:15	vstinner	set	title: Cpython/Lib vulnerability found and request a patch submission -> Zip Bomb vulnerability

2019-05-02 08:22:05	krnick	set	messages: + **msg341256**
2019-04-08 02:32:47	Victor Kung	set	messages: + **msg339587**
2019-04-03 18:09:44	christian.heimes	set	messages: + **msg339408**
2019-04-03 17:52:22	Victor Kung	set	nosy: + **Victor Kung** messages: + **msg339406**
2019-04-02 11:09:02	serhiy.storchaka	set	messages: + **msg339329**
2019-04-02 06:14:37	krnick	set	status: closed -> open resolution: remind messages: + **msg339316**
2019-03-29 00:20:57	18z	set	messages: + **msg339087**
2019-03-28 23:46:36	krnick	set	status: closed messages: + **msg339084** stage: resolved
2019-03-28 23:46:13	krnick	set	status: open -> (no value) messages: + **msg339083**
2019-03-28 17:05:32	serhiy.storchaka	set	nosy: + **serhiy.storchaka** messages: + **msg339062**
2019-03-28 16:54:47	christian.heimes	set	nosy: + **christian.heimes** messages: + **msg339061**
2019-03-28 16:43:08	brett.cannon	link	**issue36462 superseder**
2019-03-13 05:39:08	18z	set	messages: + **msg337835**
2019-03-11 09:17:13	vstinner	set	nosy: + **vstinner**
2019-03-11 07:22:30	xtreak	set	nosy: + **xtreak** messages: + **msg337652**
2019-03-11 07:19:41	18z	set	nosy: + **18z** messages: + **msg337651**
2019-03-11 07:16:58	krnick	create	