» Issue42103



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/86269

classification

Title:	[security] DoS (MemError via CPU and RAM exhaustion) when processing malformed Apple Property List files in binary format		
Type:	resource usage	**Stage**:	resolved
Components:	Library (Lib)	**Versions**:	Python 3.10, Python 3.9, Python 3.8, Python 3.7, Python 3.6

process

Status:	closed	**Resolution**:	fixed
Dependencies:		**Superseder**:	
Assigned To:		**Nosy List**:	lukasz.langa, miss-islington, ned.deily, pablogsal, ronaldoussoren, serhiy.storchaka, wessen
Priority:	release blocker	**Keywords**:	patch, security_issue

Created on **2020-10-21 00:25** by **wessen**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 22882	merged	serhiy.storchaka, 2020-10-22 09:27	
PR 23115	merged	miss-islington, 2020-11-02 21:02	
PR 23116	merged	serhiy.storchaka, 2020-11-02 21:17	
PR 23117	merged	serhiy.storchaka, 2020-11-02 21:26	
PR 23118	merged	serhiy.storchaka, 2020-11-02 21:40	

Messages (12)

msg379175 - (view) **Author: Robert Wessen (wessen)** **Date: 2020-10-21 00:25**

```
In versions of Python from 3.4-3.10, the Python core plistlib library supports
Apple's binary plist format. When given malformed input, the implementation can be
forced to create an argument to struct.unpack() which consumes all available CPU
and memory until a MemError is thrown as it builds the 'format' argument to
unpack().
```

This can be seen with the following malformed example binary plist input:

```
$ xxd binary_plist_dos.bplist
00000000: 6270 6c69 7374 3030 d101 0255 614c 6973  bplist00...UaLis
00000010: 74a5 0304 0506 0000 00df 4251 4351 44a3  t.........BQCQD.
00000020: 0809 0a10 0110 0210 0308 0b11 1719 1b1d  ................
00000030: 0000 0101 0000 0000 0000 000b 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0029            ...........)
```

The error is reached in the following lines of plistlib.py: (**https://github.com/python/cpython/blob/e9959c71185d0850c84e3aba0301fbc238f194a9/Lib/plistlib.py#L485**)

```
    def _read_ints(self, n, size):
        data = self._fp.read(size * n)
        if size in _BINARY_FORMAT:
            return struct.unpack('>' + _BINARY_FORMAT[size] * n, data)
```

When the malicious example above is opened by plistlib, it results in 'n' being controlled by the input and it can be forced to be very large. Plistlib attempts to build a string which is as long as 'n', consuming excessive resources.

Apple's built in utilities for handling plist files detects this same file as malformed and will not process it.

msg379238 - (view) **Author: Ronald Oussoren (ronaldoussoren) *** **Date: 2020-10-21 19:44**

Thanks for the report. I can reproduce the issue.

msg379243 - (view) **Author: Ronald Oussoren (ronaldoussoren) *** **Date: 2020-10-21 20:12**

One Apple implementation of binary plist parsing is here: **https://opensource.apple.com/source/CF/CF-550/CFBinaryPList.c**.

That seems to work from a buffer (or mmap) of the entire file, making consistency checks somewhat easier, and I don't think they have a hardcoded limit on the number of items in the plist (but: it's getting late and might have missed that).

An easy fix for this issue is to limit the amount of values read by _read_ints, but that immediately begs the question what that limit should be. It is clear that 1B items in the array is too much (in this case 1B object refs for dictionary keys).

I don't know what a sane limit would be. The largest plist file on my system is 10MB. Limiting *n* to 20M would be easily enough to parse that file, and doesn't consume too much memory (about 160MB for the read buffer at most).

Another thing the current code doesn't do is check that the "ref" argument to _read_object is in range. That's less problematic because the code will raise an exception regardless (IndexErrox, instead of the nicer InvalidFileException).

msg379255 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2020-10-21 21:36**

There are two issues here.

The simple one is building a large format string for struct.unpack(). It has simple solution: use f'>{n}{_BINARY_FORMAT[size]}'.

The hard issue is that read(n) allocates n bytes in memory even if there are not so many bytes in the file. It affects not only plistlib and should be fixed in the file implementation itself. There is an open issue for this.

msg379283 - (view) **Author: Ronald Oussoren (ronaldoussoren) *** **Date: 2020-10-22 09:40**

Serhiy, thanks. Just the change in the format string would fix this particular example.

I see you're working on a PR with better validation. The current state of the draft looks good to me, but I haven't checked yet if there are other potential problems that can be added to the list of invalid binary plists.

msg379285 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2020-10-22 09:47**

~~PR 22882~~ fixes problem in _read_ints(), adds validation for string size, and adds many tests for mailformed binary Plists.

There may be problems with recursive collections. I'll try to solve them too.

msg379286 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2020-10-22 09:59**

No, with recursive collections all is good.

Then I'll just add a NEWS entry and maybe few more tests.

msg380250 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2020-11-02 21:02**

New changeset **34637a0ce21e7261b952fbd9d006474cc29b681f** by Serhiy Storchaka in branch 'master':
~~bpo-42103~~: Improve validation of Plist files. (~~GH-22882~~)
https://github.com/python/cpython/commit/34637a0ce21e7261b952fbd9d006474cc29b681f

msg380252 - (view) **Author: miss-islington (miss-islington)** **Date: 2020-11-02 21:34**

New changeset **e277cb76989958fdbc092bf0b2cb55c43e86610a** by Miss Skeleton (bot) in branch '3.9':
~~bpo-42103~~: Improve validation of Plist files. (~~GH-22882~~)
https://github.com/python/cpython/commit/e277cb76989958fdbc092bf0b2cb55c43e86610a

msg380265 - (view) **Author: Serhiy Storchaka (serhiy.storchaka) *** **Date: 2020-11-03 07:32**

New changeset **547d2bcc55e348043b2f338027c1acd9549ada76** by Serhiy Storchaka in branch '3.8':
[3.8] ~~bpo-42103~~: Improve validation of Plist files. (~~GH-22882~~) (~~GH-23116~~)
https://github.com/python/cpython/commit/547d2bcc55e348043b2f338027c1acd9549ada76

msg380703 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2020-11-10 19:54**

New changeset **225e3659556616ad70186e7efc02baeebfeb5ec4** by Serhiy Storchaka in branch '3.7':
[3.7] ~~**bpo-42103**~~: Improve validation of Plist files. (~~**GH-22882**~~) (~~**#23117**~~)
https://github.com/python/cpython/commit/225e3659556616ad70186e7efc02baeebfeb5ec4

msg380704 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2020-11-10 19:57**

New changeset **a63234c49b2fbfb6f0aca32525e525ce3d43b2b4** by Serhiy Storchaka in branch '3.6':
[3.6] ~~**bpo-42103**~~: Improve validation of Plist files. (~~**GH-22882**~~) (~~**GH-23118**~~)
https://github.com/python/cpython/commit/a63234c49b2fbfb6f0aca32525e525ce3d43b2b4

History

Date	User	Action	Args
2022-04-11 14:59:37	admin	set	nosy: + **pablogsal** github: 86269
2020-11-10 20:17:26	serhiy.storchaka	set	status: open -> closed resolution: fixed stage: patch review -> resolved
2020-11-10 19:57:41	ned.deily	set	messages: + **msg380704**
2020-11-10 19:54:25	ned.deily	set	messages: + **msg380703**
2020-11-03 07:33:40	serhiy.storchaka	set	priority: normal -> release blocker nosy: + **lukasz.langa**
2020-11-03 07:32:31	serhiy.storchaka	set	messages: + **msg380265**
2020-11-02 21:40:36	serhiy.storchaka	set	pull_requests: + **pull_request22035**
2020-11-02 21:34:51	miss-islington	set	messages: + **msg380252**
2020-11-02 21:26:43	serhiy.storchaka	set	pull_requests: + **pull_request22034**
2020-11-02 21:17:38	serhiy.storchaka	set	pull_requests: + **pull_request22033**
2020-11-02 21:02:11	miss-islington	set	nosy: + **miss-islington** pull_requests: + **pull_request22032**
2020-11-02 21:02:11	serhiy.storchaka	set	messages: + **msg380250**
2020-10-22 09:59:32	serhiy.storchaka	set	messages: + **msg379286**
2020-10-22 09:47:54	serhiy.storchaka	set	messages: + **msg379285**
2020-10-22 09:40:45	ronaldoussoren	set	messages: + **msg379283**
2020-10-22 09:27:10	serhiy.storchaka	set	keywords: + **patch** stage: patch review pull_requests: + **pull_request21822**
2020-10-21 21:36:41	serhiy.storchaka	set	messages: + **msg379255**
2020-10-21 20:12:55	ronaldoussoren	set	messages: + **msg379243**

2020-10-21 19:44:21	ronaldoussoren	set	messages: + **msg379238**
2020-10-21 01:06:46	ned.deily	set	keywords: + **security_issue** nosy: + **ronaldoussoren**, **ned.deily**, **serhiy.storchaka** components: + Library (Lib), - Interpreter Core title: DoS (MemError via CPU and RAM exhaustion) when processing malformed Apple Property List files in binary format -> [security] DoS (MemError via CPU and RAM exhaustion) when processing malformed Apple Property List files in binary format
2020-10-21 00:25:11	wessen	create	