Issue36384

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/80565

classification

Title:	[security] CVE-2021-29921: ipaddress Should not reject IPv4 addresses with leading zeroes as ambiguously octal		
Type:	security	**Stage:**	resolved
Components:	Library (Lib)	**Versions:**	Python 3.10, Python 3.9, Python 3.8

process

Status:	closed	**Resolution:**	fixed
Dependencies:		**Superseder:**	
Assigned To:		**Nosy List:**	Joel Croteau, Julian, achraf-mer, christian.heimes, docs@python, eric.smith, gc2, lukasz.langa, mgorny, miss-islington, ncoghlan, ned.deily, pablogsal, pmoody, python-dev, serhiy.storchaka, steve.dower, vstinner
Priority:	deferred blocker	**Keywords:**	3.8regression, 3.9regression, patch

Created on **2019-03-20 23:40** by **Joel Croteau**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 12577	merged	python-dev, 2019-03-27 02:05	
PR 25099	merged	christian.heimes, 2021-03-30 14:51	
PR 25815	merged	miss-islington, 2021-05-02 12:00	
PR 27801	merged	python-dev, 2021-08-17 20:13	
PR 27824	merged	lukasz.langa, 2021-08-19 08:39	
PR 27825	merged	lukasz.langa, 2021-08-19 08:41	
PR 27826	merged	miss-islington, 2021-08-19 08:55	
PR 27827	merged	miss-islington, 2021-08-19 08:56	

msg338514 - **(view)**

I understand to a certain extent the logic in not allowing IPv4 octets that might ambiguously be octal, but in practice, it just seems like i

msg338694 - **(view)**

I agree that this is not a useful check.

msg339204 - **(view)**

New changeset e653d4d8e820a7a004ad399530af0135b45db27a by Nick Coghlan (Joel Croteau) in branch 'master':
bpo-36384: Remove check for leading zeroes in IPv4 addresses (GH-12577)
https://github.com/python/cpython/commit/e653d4d8e820a7a004ad399530af0135b45db27a

msg339205 - **(view)**

I've merged the change for Python 3.8 (thanks Joel!).

I'm not sure whether to classify it as an enhancement or as an interoperability bug fix, though, so I've put the status to pending and added

msg339206 - **(view)**

ipaddress is behaving as documented:

"The following constitutes a valid IPv4 address:

A string in decimal-dot notation, consisting of four decimal integers in the inclusive range 0–255, separated by dots (e.g. 192.168.0.1). Eac

https://docs.python.org/3/library/ipaddress.html

I can sort of understand imposing that restriction in a Python 2 world where leading zeros implied octal and Python 3 outright rejects such f

But, in any case, the documentation for 3.8 and/or 3.7 needs to be updated.

msg339210 - **(view)**

See also the article "Ping and FTP Resolve IP Address with Leading Zero as Octal" (https://web.archive.org/web/20061206211851/http://support.

This is still true in Windows 10.

So it is safer to reject IPv4 addresses with leading zeros that can be ambiguously interpreted. Otherwise this can create a security hole.

msg339211 - **(view)**

I think it should be 3.8 only, and the docs should be updated. Apologies for not catching that earlier: I searched via Google, which was a mi

msg339572 - **(view)**

The recommended handling in the article that Serhiy mentions is to strip the leading zeroes, which the ipaddress module will still do - it's

So that means the one part we missed is the docs update (together with a versionchanged note in the module docs themselves)

msg389826 - **(view)**

Serhiy was right, this is a security issue.

The patch should not have landed in 3.8. At a bare minimum the patch should have been postponed until documentation was updated. Since 3.8 th

I proposed to not only revert the change, but also tighten the check for leading zeros so it behaves like glibc's inet_pton(). It refuses any

```
>>> socket.inet_pton(socket.AF_INET, "01.1.1.1")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: illegal IP address string passed to inet_pton
>>> socket.inet_pton(socket.AF_INET, "1.1.1.01")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: illegal IP address string passed to inet_pton
```

msg389847 - **(view)**

> The patch should not have landed in 3.8. At a bare minimum the patch should have been postponed until documentation was updated. Since 3.8

Link: https://sick.codes/sick-2021-011

msg390046 - **(view)**

Deferred the blocker to the next regular release due to lack of activity in time for the current expedited releases.

msg390128 - **(view)**

(Copied from my comment on the PR, following the one where I said this was ready to go.)

Withdrawing the readiness - @ambv and I would prefer to see this behind a flag (probably "strict" parsing), on by default for 3.10, and maybe

The main reasoning being that this isn't our vulnerability, but an inconsistency with other vulnerable libraries. The current fix is the best

msg390331 - **(view)**

> Withdrawing the readiness - @ambv and I would prefer to see this behind a flag (probably "strict" parsing), on by default for 3.10, and may

Last time we added a new parameter in a stable branch, it didn't go well:
https://bugs.python.org/issue42967#msg387638

msg390340 - **(view)**

The important quote from the linked issue seems to be:

> Our new separator= parameter does not allow one to achieve the previous behavior if mixing and matching & And ; was intended to be allowed,

So arguably, we added _the wrong_ parameter in that case, because it only allowed choosing between behaviours not including the "bad" behavic

In this case, having it off by default goes further to prevent breakage, and I wouldn't be opposed to a process level opt-in (e.g. a module-l

msg390353 - **(view)**

> In this case, having it off by default goes further to prevent breakage

PyYAML was unsafe by default: it allowed to execute arbitary Python code by default. It took years to change the default to "safe". I don't t

I prefer to break code and force people to manually get back the old behavior. It's better to make 90% safe by default but make 10% of people

It's uncommon to pass IPv4 addresses with leading zeros.

If you want to tolerate leading zeros, you don't have to modify the ipaddress for that, you can pre-process your inputs: it works on any Pyth

```
>>> def reformat_ip(address): return '.'.join(part.lstrip('0') if part != '0' else part for part in address.split('.'))
...
>>> reformat_ip('0127.0.0.1')
'127.0.0.1'
```

Or with an explicit loop for readability:

```
def reformat_ip(address):
    parts = []
    for part in address.split('.'):
        if part != "0":
            part = part.lstrip('0')
        parts.append(part)
    return '.'.join(parts)
```

msg390361 - **(view)**

> I don't think that adding a parameter for opt-in for security is a good approach.

I meant to have it set by default on 3.10, when we do not have to worry
about breaking users.

If it takes years for users to get to 3.10, we should reevaluate our
release cycle, not whether we aggressively break maintenance releases.

msg392423 - **(view)**

The CVE-2021-29921 was assigned to this vulnerability.

msg392572 - **(view)**

> If it takes years for users to get to 3.10, we should reevaluate our
> release cycle, not whether we aggressively break maintenance releases.

I don't really understand how that would help. The problem is that users have major inertia for switching to newer Python versions. A part

I can't imagine deliberately leaving 3.8 and 3.9 vulnerable when 3.10 isn't going to reach final release in the next half year. Gentoo stabl

I dare say this has less breakage potential than the &/; change. It should be fixed on all affected versions. If you don't do that, distrib

msg392684 - **(view)**

Due to the relative obscurity of the bug and potential disruption of the fix, I decided not to include it in 3.8.

However, Michał's argument about 3.10 not being released for another five months is resonating with me and so we will be backporting the chan

In any case, the issue will be solved in Python 3.10.0 Beta 1 and Python 3.9.5. Having the fixed behavior "in 3.9.5 and newer" makes for easy

msg392692 - **(view)**

New changeset 60ce8f0be6354ad565393ab449d8de5d713f35bc by Christian Heimes in branch 'master':
bpo-36384: Leading zeros in IPv4 addresses are no longer tolerated (GH-25099)
https://github.com/python/cpython/commit/60ce8f0be6354ad565393ab449d8de5d713f35bc

msg392696 - **(view)**

New changeset 5374fbc31446364bf5f12e5ab88c5493c35eaf04 by Miss Islington (bot) in branch '3.9':
bpo-36384: Leading zeros in IPv4 addresses are no longer tolerated (GH-25099) (GH-25815)
https://github.com/python/cpython/commit/5374fbc31446364bf5f12e5ab88c5493c35eaf04

msg392697 - **(view)**

Łukasz, thanks for pushing the PR over the finish line!

msg394162 - **(view)**

Is there anything more to be done for this issue or can it be closed?

msg394316 - **(view)**

I'm closing this, if someone thinks something is missing, please, reopen

msg394322 - **(view)**

I created https://python-security.readthedocs.io/vuln/ipaddress-ipv4-leading-zeros.html to track this vulnerability.

Python 3.8 is left unchanged (accept leading zeros). Python 3.7 and older are not affected.

msg394327 - **(view)**

The timeline there is wrong. This issue's creation time isn't the disclosure time, it's when the bug was introduced. The disclosure was on 30

msg394379 - **(view)**

George-Cristian Bîrzan: "The timeline there is wrong."

Fixed: https://python-security.readthedocs.io/vuln/ipaddress-ipv4-leading-zeros.html#timeline

The strange part is "2019-03-20 (-741 days): Python issue bpo-36384 reported by Joel Croteau".

The problem is that this issue was "reused" for two different things: the initial change and the vulnerability.

Maybe I can removed the reference to the bpo to remove it from the timeline (and put it in links).

msg394380 - **(view)**

I think the only thing I'd improve would be to mention that this issue is the one that introduced the bug, otherwise it looks a bit weird.

msg394390 - **(view)**

> I think the only thing I'd improve would be to mention that this issue is the one that introduced the bug, otherwise it looks a bit weird.

Ok, done: https://python-security.readthedocs.io/vuln/ipaddress-ipv4-leading-zeros.html#timeline

msg399784 - **(view)**

Can we backport the security fix from this issue https://bugs.python.org/issue36384#msg392684 to version 3.8
The comment explicitly says that it was decided to not include in 3.8, I am not sure this is best, since it prevents using 3.8 because of thi

msg399798 - **(view)**

> it prevents using 3.8 because of this open vulnerability

What do you mean by this?

Our understanding is that this is a low-severity CVE because in order for this to be a vulnerability, you'd have to have both:

1. user access to IP address input; and
2. control over two addresses sharing numerical representation with leading zeroes: the first resolving when leading zeroes are treated as oc

Access to both then allows you at best to circumvent IP address-based access control or denial of service. However, access to just 1. allows

Hence low-severity.

> it does not seem to be a breaking change

It is a bona fide breaking change. Any IP address configuration saved in files or databases which might have used leading zeroes would be rej

If you still feel this ought to be fixed in 3.8, please elaborate.

msg399801 - **(view)**

>> it prevents using 3.8 because of this open vulnerability

>What do you mean by this?

>Our understanding is that this is a low-severity CVE because in order for this to be a vulnerability, you'd have to have both:

>1. user access to IP address input; and
>2. control over two addresses sharing numerical representation with leading zeroes: the first resolving when leading zeroes are treated as c

>Access to both then allows you at best to circumvent IP address-based access control or denial of service. However, access to just 1. allows

>Hence low-severity.

Even though I agree with you assessment on the root cause of the issue itself, it is listed as critical in https://nvd.nist.gov/vuln/detail/C

>> it does not seem to be a breaking change

>It is a bona fide breaking change. Any IP address configuration saved in files or databases which might have used leading zeroes would be re

>If you still feel this ought to be fixed in 3.8, please elaborate.

IMHO I still think this should be solved in 3.8, otherwise there is really no other alternative but to upgrade to python 3.9 which is a hassl

msg399803 - **(view)**

I was unaware of the "CRITICAL" base score assigned by NIST to this. Alright, let's port this back then. There are a few things the PR will n

msg399804 - **(view)**

"CRITICAL" is a ridiculous high assessment for this bug. Somebody ticked all the scary boxes in the CVSS form like "total loss of control".

msg399805 - **(view)**

The CVE was rated https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1, which is equiv

I would rate the issue https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N&version=3.1, maybe A:L

msg399809 - **(view)**

New changeset 03dd89d62413c4a92831ed1b36e2ae8983bcb2d4 by achraf-mer in branch '3.8':
[3.8] bpo-36384: Leading zeros in IPv4 addresses are no longer tolerated (GH-25099) (GH-27801)
https://github.com/python/cpython/commit/03dd89d62413c4a92831ed1b36e2ae8983bcb2d4

msg399893 - **(view)**

New changeset 6ebfe8da6331bfcf54057f6e22a6f353a5621d35 by Łukasz Langa in branch '3.8':
[3.8] bpo-36384: [doc] Correct typos in CVE-2021-29921 fix description (GH-27825)
https://github.com/python/cpython/commit/6ebfe8da6331bfcf54057f6e22a6f353a5621d35

msg399895 - **(view)**

New changeset 0fd66e46b2f472d0d206a185dc8892f4f0347cb6 by Łukasz Langa in branch 'main':
bpo-36384: [doc] Mention CVE-2021-29921 fix in 3.8.12 (GH-27824)
https://github.com/python/cpython/commit/0fd66e46b2f472d0d206a185dc8892f4f0347cb6

msg399898 - **(view)**

New changeset 1204dfc89cb3ed5e21dce32aed0339b7569fe1f9 by Miss Islington (bot) in branch '3.10':
bpo-36384: [doc] Mention CVE-2021-29921 fix in 3.8.12 (GH-27824)
https://github.com/python/cpython/commit/1204dfc89cb3ed5e21dce32aed0339b7569fe1f9

msg399899 - **(view)**

New changeset 98820250a3c9c131d3c2d57c4fc5260aebd8aa1d by Miss Islington (bot) in branch '3.9':
bpo-36384: [doc] Mention CVE-2021-29921 fix in 3.8.12 (GH-27824) (GH-27827)
https://github.com/python/cpython/commit/98820250a3c9c131d3c2d57c4fc5260aebd8aa1d

History

Date	User	Action	Args
2022-04-11 14:59:12	admin	set	github: 80565
2021-08-19 09:20:50	lukasz.langa	set	messages: + msg399899
2021-08-19 09:16:18	miss-islington	set	messages: + msg399898
2021-08-19 08:56:02	miss-islington	set	pull_requests: + pull_request26292
2021-08-19 08:55:58	miss-islington	set	pull_requests: + pull_request26291
2021-08-19 08:55:53	lukasz.langa	set	messages: + msg399895
2021-08-19 08:50:47	lukasz.langa	set	messages: + msg399893
2021-08-19 08:41:05	lukasz.langa	set	pull_requests: + pull_request26290
2021-08-19 08:39:57	lukasz.langa	set	pull_requests: + pull_request26289
2021-08-19 08:15:17	lukasz.langa	set	versions: + Python 3.9, Python 3.10
2021-08-17 23:46:43	lukasz.langa	set	messages: + msg399809
2021-08-17 23:13:39	christian.heimes	set	messages: + msg399805
2021-08-17 23:08:43	christian.heimes	set	messages: + msg399804
2021-08-17 22:59:24	lukasz.langa	set	messages: + msg399803
2021-08-17 22:51:52	achraf-mer	set	messages: + msg399801
2021-08-17 22:22:15	lukasz.langa	set	messages: + msg399798
2021-08-17 20:13:46	python-dev	set	nosy: + python-dev pull_requests: + pull_request26269

			versions: + Python 3.8, - Python 3.9, Python 3.10 nosy: + achraf-mer
2021-08-17 19:19:36	achraf-mer	set	messages: + msg399784
			components: - Documentation
2021-05-25 20:01:40	vstinner	set	messages: + msg394390
2021-05-25 17:10:43	gc2	set	messages: + msg394380
2021-05-25 16:44:50	vstinner	set	messages: + msg394379
2021-05-25 11:20:51	gc2	set	messages: + msg394327
2021-05-25 10:55:15	vstinner	set	messages: + msg394322
			status: open -> closed
2021-05-25 10:26:41	pablogsal	set	nosy: + pablogsal messages: + msg394316
			resolution: fixed stage: patch review -> resolved
2021-05-21 22:22:30	ned.deily	set	messages: + msg394162
2021-05-02 13:53:06	christian.heimes	set	messages: + msg392697
2021-05-02 13:49:09	lukasz.langa	set	messages: + msg392696
2021-05-02 12:00:52	lukasz.langa	set	messages: + msg392692
2021-05-02 12:00:50	miss-islington	set	nosy: + miss-islington pull_requests: + pull_request24501
2021-05-02 10:01:10	lukasz.langa	set	versions: - Python 3.8
2021-05-02 10:01:01	lukasz.langa	set	assignee: docs@python -> messages: + msg392684
2021-05-01 07:36:51	mgorny	set	nosy: + mgorny messages: + msg392572
2021-04-30 11:53:11	vstinner	set	messages: + msg392423 title: ipaddress Should not reject IPv4 addresses with leading zeroes as ambiguously octal -> [security] CVE-2021-29921: ipaddress Should not reject IPv4 addresses with leading zeroes as ambiguously octal
2021-04-11 23:01:06	Julian	set	nosy: + Julian
2021-04-06 19:03:09	steve.dower	set	messages: + msg390361
2021-04-06 17:19:01	vstinner	set	messages: + msg390353
2021-04-06 14:29:48	steve.dower	set	messages: + msg390340
2021-04-06 13:41:59	vstinner	set	messages: + msg390331
2021-04-03 16:41:45	steve.dower	set	nosy: + steve.dower messages: + msg390128
2021-04-02 09:35:18	lukasz.langa	set	messages: + msg390046
2021-04-02 09:34:39	lukasz.langa	set	priority: release blocker -> deferred blocker
2021-03-31 19:59:14	christian.heimes	set	priority: critical -> release blocker nosy: + lukasz.langa
2021-03-31 08:54:04	gc2	set	nosy: + gc2
2021-03-30 19:35:23	vstinner	set	nosy: + vstinner messages: + msg389847
2021-03-30 14:51:56	christian.heimes	set	keywords: + patch stage: needs patch -> patch review pull_requests: + pull_request23844
2021-03-30 14:23:11	christian.heimes	set	priority: normal -> critical type: behavior -> security components: + Library (Lib) versions: + Python 3.9, Python 3.10 keywords: + 3.8regression, 3.9regression, - 3.2regression nosy: + christian.heimes

			messages: + msg389826
2019-04-07 11:49:32	ncoghlan	set	nosy: + docs@python messages: + msg339572 assignee: docs@python components: + Documentation, - Library (Lib)
2019-03-30 16:16:17	eric.smith	set	messages: + msg339211
2019-03-30 16:15:02	serhiy.storchaka	set	nosy: + serhiy.storchaka messages: + msg339210
2019-03-30 15:44:32	ned.deily	set	status: pending -> open messages: + msg339206 keywords: + 3.2regression, - patch resolution: fixed -> (no value) stage: resolved -> needs patch
2019-03-30 15:12:39	xtreak	set	status: open -> pending
2019-03-30 15:12:26	xtreak	set	status: pending -> open nosy: + ned.deily
2019-03-30 14:57:36	ncoghlan	set	status: open -> pending resolution: fixed messages: + msg339205 stage: patch review -> resolved
2019-03-30 14:53:51	ncoghlan	set	messages: + msg339204
2019-03-27 06:32:51	serhiy.storchaka	set	nosy: + ncoghlan
2019-03-27 02:05:56	python-dev	set	keywords: + patch stage: patch review pull_requests: + pull_request12521
2019-03-23 19:35:05	eric.smith	set	nosy: + eric.smith messages: + msg338694
2019-03-21 08:16:51	SilentGhost	set	nosy: + pmoody versions: - Python 2.7, Python 3.5, Python 3.6, Python 3.7, Python 3.9
2019-03-20 23:40:09	Joel Croteau	create	

Supported by The Python Software Foundation,
Powered by Roundup
Copyright © 1990-2022, Python Software Foundation
Legal Statements