CVE-2021-29921 – python stdlib “ipaddress” – Improper Input Validation of octal literals in python 3.8.0 thru v3.10 results in indeterminate SSRF & RFI vulnerabilities. – “ipaddress leading zeros in IPv4 address”

by Sick Codes – April 30, 2021 - Updated on October 4, 2021 in Security 8

python stdlib "ipaddress" CVE-2021-29921

Title

python stdlib “ipaddress” – Improper Input Validation of octal literals in python 3.8.0 thru v3.10 results in indeterminate SSRF & RFI vulnerabilities. – “ipaddress leading zeros in IPv4 address”

CVE ID

CVE-2021-29921

CVSS Score

9.8

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Internal ID

SICK-2021-014

Vendor

python

Product

ipaddress stdlib

Product Versions

3.8.0 thru v3.10

Vulnerability Details

Improper input validation of octal strings in Python 3.8.0 thru v3.10 stdlib ipaddress allows unauthenticated remote attackers to perform indeterminate SSRF, RFI, and LFI attacks on many programs that rely on Python stdlib ipaddress. IP address octects are left stripped instead of evaluated as valid IP addresses. For example, an attacker submitting an IP address to a web application that relies on stdlib ipaddress, could cause SSRF via inputting octal input data; An attacker can submit exploitable IP addresses if the octet is 3 digits, with the minimum exploitable octect being 08 (Denial of Service) and the maximum exploitable octet is 099. For example, an attacker can submit 010.8.8.8, which is 8.8.8.8, yet Python ipaddress builtin will evaluate this as 10.8.8.8.

Vendor Response

Currently unpatched - due to be addressed in next release.

Proof of Concept

Vulnerability added in python3.8

https://github.com/python/cpython/pull/12577

Documentated to be vulnerable in the changelog:

https://github.com/python/cpython/blob/63298930fb531ba2bb4f23bc3b915dbf1e17e9e1/Misc/NEWS.d/3.8.0a4.rst

Stop rejecting IPv4 octets for being ambiguously octal. Leading zeros are ignored, and no longer are assumed to specify octal octets. Octets are always decimal numbers. Octets must still be no more than three digits, including leading zeroes.

```
#!/usr/bin/env python
# Authors:      sickcodes, Victor Viale
# License:      GPLv3+
# Reference:    https://docs.python.org/3.10/library/ipaddress.html#ipaddress.IPv4Address

# Leading zeroes are tolerated only for values less than 8 (as there is no ambiguity between the decimal and octal interpretations of such strings).

import subprocess
import ipaddress

SUSPECT = '010.8.8.8'

print(ipaddress.ip_network(SUSPECT, strict=True))

BAD_IP = ipaddress.ip_address(SUSPECT)

print('http://'+str(BAD_IP))

print(str(subprocess.check_output("ping -W3 -v -c1 "+str(SUSPECT), shell=True, universal_newlines=True).strip()))

print(str(subprocess.check_output("ping -W3 -v -c1 "+str(BAD_IP), shell=True, universal_newlines=True).strip()))
```

Disclosure Timeline

- **2019-03-20** - Issue created in https://bugs.python.org/issue36384
- **2021-03-29** - Researchers discover vulnerability
- **2021-03-29** - Vendor notified
- **2021-03-29** - CVE requested
- **2021-04-30** - CVE Assigned CVE-2021-29921 https://bugs.python.org/issue36384#msg392423
- **2021-04-30** - CVE published

Links

https://github.com/python/cpython

https://github.com/sickcodes/security/blob/master/advisories/SICK-2021-014.md https://sick.codes/sick-2021-014

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

https://bugs.python.org/issue36384

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

https://github.com/python/cpython/pull/12577

https://github.com/python/cpython/pull/25099

https://github.com/python/cpython/blob/63298930fb531ba2bb4f23bc3b915dbf1e17e9e1/Misc/NEWS.d/3.8.0a4.rst

Researchers

Joel Croteau: https://github.com/TV4Fun

Victor Viale: https://github.com/koroeskohr || https://twitter.com/koroeskohr

Sick Codes: https://github.com/sickcodes || https://twitter.com/sickcodes

Kelly Kaoudis: https://github.com/kaoudis || https://twitter.com/kaoudis

John Jackson https://www.twitter.com/johnjhacking

Nick Sahler: https://github.com/nicksahler || https://twitter.com/tensor_bodega

Christian Heimes: https://github.com/tiran

Victor Stinner: https://github.com/vstinner

CVE Links

https://sick.codes/sick-2021-014

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29921

https://nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-29921

Comments 8

Pingback: Vulnerabilidad en Python (Python) - CVE-2021-29921 - Información y Soluciones

ralf 1 year ago

Hi there !
I may be at sea, but I wanted to know ; what's the problem with this "vulnerability" ? What does it allow to do, knowing that 'ipaddress' just strips leading zeros in IP's bytes while the 'ping' command seems to handle the leading zero in "010" as the octal value 8 ? I can't understand why it is considered as a vulnerability…

Reply

азизбек 2 years ago

я хакер

Reply

JRG 2 years ago

Is anyone else having deja vu, or is it just me?

Reply

Anonymous 2 years ago

Dates of assignment and disclosure seem to be mixed up in timeline.

Reply

admin 2 years ago

Thanks fixed!

Reply

Jeff Silverman 2 years ago

I'm still a little confused by the timeline. It currently reads:
2019-03-20 - Issue created in https://bugs.python.org/issue36384
2021-03-29 - Researchers discover vulnerability
2021-03-29 - Vendor notified
2021-03-29 - CVE requested
2021-04-30 - CVE Assigned CVE-2021-29921 https://bugs.python.org/issue36384#msg392423

I think it should read:
2019-03-20 - Researchers discover vulnerability
2021-03-20 - Issue created in https://bugs.python.org/issue36384
2021-03-20 - Vendor notified
2021-03-29 - CVE requested
2021-03-30 - Fix merged in to main code branch
2021-04-30 - CVE Assigned CVE-2021-29921 https://bugs.python.org/issue36384#msg392423

I would argue that if a vendor has a bug report on a publicly available bug tracking system, then the vendor has been notified. In fact, looking at bug 392423, I see that Joel Croteau started working on the problem March 20th. I see that Nick Coghlan merged the fix into python 3.8.
The reason why I am being pedantic about this is because I want an organization that is open and honest to be lauded when they respond to an obscure problem quickly.

Reply

admin 2 years ago

Hey Jeff,

Thanks for reaching out, we didn’t originally know that Joel has created an issue 2 years prior, on 2019-03-20.

```
2021-03-20 – Issue created in https://bugs.python.org/issue36384
```

It’s confusing as it was exactly 2 years between initial issue and fix.

What ended up happening was Joel created the issue in March 2019, then a half fix was applied here Aug 30, 2019,
https://github.com/python/cpython/blob/63298930fb531ba2bb4f23bc3b915dbf1e17e9e1/Misc/NEWS.d/3.8.0a4.rst

Stop rejecting IPv4 octets for being ambiguously octal. Leading zeros are ignored, and no longer are assumed to specify octal octets. Octets are always decimal numbers. Octets must still be no more than three digits, including leading zeroes.

However it was still vulnerable as indicated by the changelog.

Then there was this message: https://bugs.python.org/issue36384#msg389826

Which refers directly to our other CVE-2021-28918: https://github.com/sickcodes/security/blob/master/advisories/SICK-2021-011.md

Which goes into detail here: https://sick.codes/universal-netmask-npm-package-used-by-270000-projects-vulnerable-to-octal-input-data-server-side-request-forgery-remote-file-inclusion-local-file-inclusion-and-more-cve-2021-28918/

We submitted all of them on pretty much the same day, and then a week or so ago the CVE was assigned.

I’d say it took a while, 2 years to fix, which is where the “discovered vulnerability” part comes into play.

vstinner was right though, within 1 day of publishing this (as soon as we saw he had posted the CVE publicly) there was a fix available.

Let me know if I’m missing something, but yeah there’s a 2 year gap between issue opening and the actual fix, jumpstarted by CVE-2021-28918 and then this received CVE-2021-29921.

We submitted both xD

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name

Email

Website

Privacy - Terms
I am human

POST COMMENT

@sickcodes

@sickcodes

@sickcodes

Discord Server

sickcodes.slack.com

t.me/sickcodeschat

./contact_form