

New issue

[Jump to bottom](#)

ReDoS vulnerability in svnurl.py #287

🟢 **Open** SCH227 opened this issue on Sep 21 · 30 comments

SCH227 commented on Sep 21

Good night!

I found that [this regex](#) is vulnerable to Regular Expression Denial of Service.

PoC:

```
>>> from py._path.svnurl import InfoSvnCommand
>>> payl = " 2256      hpk      165 Nov 24 17:55 __init__.py" + " " * 5000
>>> InfoSvnCommand(payl)
```

Attack vector:

An user accessing a (possibly remote) subversion repository that provides malicious "info" data.
Or an attacker injecting 'svn ls http://...' output (less realistic).

Fix:

Use a pattern with non-overlapping groups. I can help in finding a better regex and testing if needed.



2



16

The-Compiler commented on Sep 22

Member

Related: [#256](#)

bluetech commented on Oct 16

Member

I doubt there's anyone using this code, so I don't think it would warrant any sort of security notification that users should bother with, but if you prepare a PR for this I'll merge & release it.



skialpine commented on Oct 18

[GHSA-w596-4wvx-j9j6](#)
<https://nvd.nist.gov/vuln/detail/CVE-2022-42969>

jenstroeger commented on Oct 19

As @skialpine mentions, the advisory now triggers [pip-audit](#) and thus can fail CI runs (like [this one](#)).

Is there a fix on the horizon?

RonnyPfannschm... commented on Oct 19

Member

The issue is not considered critical, im not aware of anyone working towards a fix

The-Compiler commented on Oct 19 • edited ▾

Member

Well, congratulations to whoever it is that decided that the right path of action here is getting a CVE for...

- Something that seems very questionable to be titled a "vulnerability" in the first place (if I can control an SVN repo/server, I might as well just [Slowloris](#) the initial info request, I suppose... though admittedly I didn't try that.)
- For some ~18 year old code...
- ...which is only there for historical reasons and [discouraged to use](#)
- ...which, to the best of our knowledge, is [not used anywhere in the wild](#) outside of some [old PyPy development scripts](#) nobody probably uses anymore (and certainly not against random SVN servers). Note the search results seem to be copies of those PyPy scripts, as far as I can tell.
- ...which, given the above, nobody is terribly interested in maintaining
- ...yet you ended up generating nothing but noise for [hundreds of thousands of pytest users](#), which [for historical reasons](#) depends on pylib (since it came from the PyPy project, like pylib does). Yes, not everyone of the ~~half~~ quarter a million projects there will monitor CVEs against dependencies, but at the same time, lots of people that do (in companies and such) are probably not in that list on GitHub.

Given that it will still take some time for [pytest to get fully rid of its py internals](#), and people seem to like getting CVEs for this project (which just happens to be very popular via pytest, but pretty much unused outside of pytest), but without really making an attempt to understanding the context... can we please consider:

- Either vendoring the remaining code (`py.path` I assume) in pytest, and archiving pylib...
- ...or releasing a pylib 2.0.0 which simply drops all the code that isn't used in pytest?

Sorry if I sound frustrated. But requesting a CVE without understanding the context of the project/issue you're reporting, and then generating false reports for hundreds of thousands of pytest users is doing a major disservice to both pytest/pylib maintainers and all those users.



The-Compiler mentioned this issue on Oct 19

CVE-2022-42969: security issue in library py used by this project [pytest-dev/pytest#10392](#)

🔒 Closed

RonnyPfannschm... commented on Oct 19

Member

I'd go as far as to label this cve report a supply chain attack

Pytest should keep dropping pylib as it does

The cve should be added as common false positive



The-Compiler commented on Oct 19

Member

I'd go as far as to label this cve report a supply chain attack

I would not, but I'd certainly add it to my "examples of behavior causing open source maintainer burnout" list, given that we're now getting the [first pytest issue](#) about this, and it almost certainly won't be the last one.

The-Compiler mentioned this issue on Oct 19

[GHSA-w596-4wvx-j9j6] ReDoS in py library [github/advisory-database#761](#)

🔗 Merged

bluetech commented on Oct 19

Member

I should have known better that this was going to happen when I replied as I did...

If someone can get the notification retracted that'd be best.

But I guess since they've already put out a security notification to the entire world and their grandmother through GitHub, the path of least resistance would be to issue a release to fix this "security issue". I'll try to do it today.



RonnyPfannschm... commented on Oct 19

Member

@bluetech @The-Compiler i wouldnt mind to cut a release that drops the svn wc stuff altogether

  **The-Compiler** mentioned this issue on Oct 19

Plan for dropping/deprecating submodules of `py` and releasing v2.0 #288

 Open

The-Compiler commented on Oct 19

Member

@RonnyPfannschmidt agreed, and perhaps a bit more even - I opened #288 with some overview on that.

The-Compiler commented on Oct 19

Member

FWIW, I've also proposed adding a note to the GitHub advisory ([github/advisory-database#761](#)) and [tweeted a PSA](#).



  **shirayu** mentioned this issue on Oct 19

CVE-2022-42969 shirayu/whispering#40

 Open

The-Compiler commented on Oct 19

Member

GitHub have [now reacted](#) and amended [the advisory](#):

The particular codepath in question is the regular expression at `py._path.svnurl.InfoSvnCommand.lspattern` and is only relevant when dealing with subversion (svn) projects. Notably the codepath is not used in the popular pytest project.

and apparently also added that information for Dependabot alerts:

I've also added the codepath to the advisory so that [dependabot can more intelligently target alerts](#).

woodruffw commented on Oct 19

`pip-audit` maintainer here. I have no say in it, but it's a shame (in my personal opinion) that this kind of low-quality finding was assigned a CVE ID without any significant cross-checking.

I've filed a CVE rejection request with MITRE, since they're the CVE CNA in this case. If they successfully reject it, getting it removed from GHSA should also be easy.



woodruffw commented on Oct 19

Actually, looks like I'll be able to propose a GHSA withdrawal even if the CVE itself hasn't been retracted. I'll open a PR for that in a moment.



woodruffw mentioned this issue on Oct 19

GHSA-w596-4wvx-j9j6: mark as withdrawn [github/advisory-database#762](#)

Closed

woodruffw commented on Oct 19

I've filed [github/advisory-database#762](#) to mark the GHSA report as withdrawn.

This was referenced on Oct 19

Skip vulnerability reports that are marked as "withdrawn" [pypa/pip-audit#385](#)

Closed

Contested advisory: GHSA-w596-4wvx-j9j6 / CVE-2022-42969 [pypa/advisory-database#104](#)

Open

jenstroeger commented on Oct 20 • edited ▼

@The-Compiler agreeable and funny [#287 \(comment\)](#) 😄

@woodruffw thanks for following up on the advisory! That means that `pip-audit` won't pick up that CVE any longer once [pypa/pip-audit#385](#) has merged, correct?

🔗  RonnyPfannschmidt mentioned this issue on Oct 20

RFE: Replace use of `pkg_resources` with `importlib.metadata` [#282](#)

🔒 Closed

woodruffw commented on Oct 20

@woodruffw thanks for following up on the advisory! That means that `pip-audit` won't pick up that CVE any longer once [pypa/pip-audit#385](#) has merged, correct?

Once that's merged *and* the OSV entry is marked as withdrawn, yeah.

👍 3

Denelvo commented on Oct 21

@The-Compiler @bluetechnology @woodruffw & other maintainers: don't sweat it! Even though 18 years ago, someone wrote a regex, not thinking of 2022-levels of security paranoia, your efforts are still very much appreciated and your contributions are incredibly productive.

The CVE did trigger some alarms in builds of valuable systems used by financial institutions. But that's good. It doesn't necessarily mean that `pytest` is suddenly unusable. It just forces downstream developers in highly secured places to stop and think. Alerts can be ignored, assessed, categorized, accepted,...

Please look at CVEs as mostly just a catalyst. Having the CVE removed is a harsh reaction, but the right one if we're absolutely sure that that particular regex is not exploitable. But try to think of it from the point of view of a security officer, with full paranoia goggles on (and no intimate knowledge of the module). The regex is present in the code. It could be executed. A hacker could penetrate, set up a SVN repo, craft some commands, take advantage of other existing vulnerabilities on the system,... You have to think worst case and then some.

In the end, I think a correctly worded CVE with a low severity score which clearly includes all the IFs, would have been the best option here.

Thanks!

The-Compiler commented on Oct 21

Member

Taking the freedom to quote what I wrote over at [github/advisory-database#762 \(comment\)](#):

but to my eye the advisory is in fact talking about a real redos vulnerability. I do understand the annoyance with what you call CVE spam, but if the advisory is valid then we want to include it in our data set.

In theory, when viewed in a vacuum: indeed.

However, even a [very broad search](#) on GitHub for the affected code yields 92 results. I've looked through them all, and from what I can tell, all the matches fall into one of those categories:

- Copies of `py` using it (in `svnur1()` , which the search captures as well)
- Copies of `py` (where the `py` library originates from historically), in it's [development tools directory](#). However, PyPy has [moved away from SVN](#) in 2010, so they're probably just bitrotting. In any case, they use the repository the current file is in, or are hardcoded to run on a specific old PyPy team server ([codespeak.net](#)). If someone has control over that, they might as well just change those scripts. Even if there are certain scripts which actually run on arbitrary SVN servers, the context they are run in is unlikely to make a DoS a real problem. But given that they won't actually work anymore for their intended original purpose, the chance of anyone running them is pretty much zero.

On the other side of the coin, you have [at least half a million projects](#) depending on `pylib` via `pytest`, which got noise in their inbox.

I believe the main point of a CVE and security advisories is to make people aware of problems which have a real (even if small) chance of affecting them. An advisory which just adds noise to what I believe will be 100.0% of the receivers is just going to hurt the whole system and community.

In this particular case, and when viewed in context, the sheer amount of noise the advisory generates vs. its real usage is so high that the word "spam" is unfortunately very fitting. I can't help but think that "oh, I get a shiny CVE in a popular project" was the only motivation behind it.

woodruffw commented on Oct 21

Just to make sure there's no confusion: I'm not a maintainer of this project or of `pytest` , so my opinions are not those of the maintainers. I'm the maintainer of a separate dependency scanning project, so I have an interest in dependency feeds having a high signal-to-noise ratio. After this comment, I'll step out of this thread now (since I'm not a maintainer and I only stepped in to coordinate on the `pip-audit` side).

With that being said: I disagree that it's good that a CVE was filed for this behavior. The fact that one was filed *and* published seemingly without any review represents a series of communication and authority breakdowns; the fact that the project's own maintainers have done more investigation into exploitability potential than the original reporter seems to have is an indicator of this.

As for why that matters: not everybody is a bank, with roles dedicated to reviewing a constant deluge of security reports. In the context of more limited resources, managing security fatigue is **far** more important than reporting weakness classes like ReDoS, which don't manifest as exploitable vulnerabilities in the overwhelming majority of cases. When users get tired of useless reports, they disable their security tools entirely.



SCH227 commented on Oct 24

Author

First of all, I did fill for a CVE, but I didn't publish it. Someone else did it. I find illogical it was you, but the GH advisory says "Credits - @The-Compiler".

The other thing I can think of, is that GH made it automatically because this is described in a public issue. But I find it strange, because I already reported other security issues to MITRE and they never got published before I informed they were made public

37 hidden items

[Load more...](#)

 KSchopmeyer pushed a commit to pywbem/pywbemtools that referenced this issue 23 days ago


Fix issues from Nov 2022 security issue changes ...

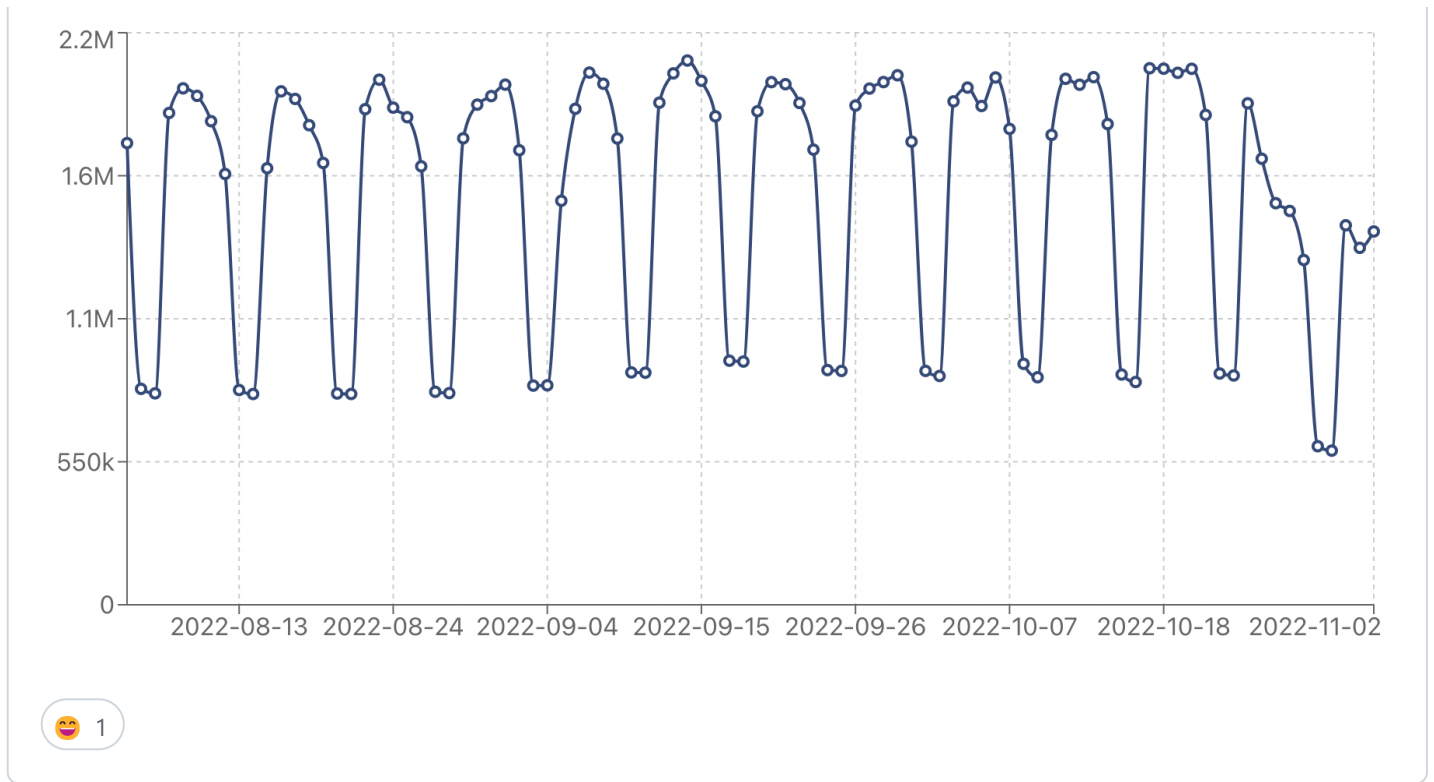
 de3b490

hugovk commented 23 days ago

Member

- As a result of this issue, today pytest 7.2.0 was released, which [vendors the parts of py](#) it needs. With that being slowly adopted, I'm assuming the number of projects depending on `py` will be much lower soon.

 Midweek [py PyPI numbers](#) on are down from ~2M/day to ~1.4M/day:



🔗 **KSchopmeyer** mentioned this issue 22 days ago

Update the requirements files when correct versions of these packages released

pywbem/pywbemtools#1228

👍 Closed

🔗 **KSchopmeyer** pushed a commit to pywbem/pywbemtools that referenced this issue 22 days ago

Fix issues from Nov 2022 security issue changes ...

✖ 268eb43

🔗 **KSchopmeyer** pushed a commit to pywbem/pywbemtools that referenced this issue 22 days ago

Fix issues from Nov 2022 security issue changes ...

✖ 83570e8

🔗 **KSchopmeyer** pushed a commit to pywbem/pywbemtools that referenced this issue 22 days ago

Security issues from Nov 2022 security issue changes ...

✖ ec6dab9

🔗 **KSchopmeyer** pushed a commit to pywbem/pywbemtools that referenced this issue 22 days ago

Security issues from Nov 2022 security issue changes ...

✖ 3aba88f





 hajapy mentioned this issue 21 days ago

[Housekeeping] Remove unused retry dependency in flytekit flyteorg/flyte#3052

 Open

 2 tasks



 jiwaszki mentioned this issue 20 days ago

[PyOV] Enable Python 3.10 on Azure CI openvinotoolkit/openvino#12578

 Merged



 amitgalitz mentioned this issue 18 days ago

CVE-2022-42969 (High) detected in py-1.11.0-py2.py3-none-any.whl opensearch-project/anomaly-detection#694

 Closed



 sp-luciano-chinke mentioned this issue 18 days ago

Add support to pytest>=7.2.0 pytest-dev/pytest-selenium#305

 Open



andy-maier added a commit to pywbem/pywbemtools that referenced this issue 18 days ago



Fix issues from Nov 2022 security issue changes ([#1229](#)) ...

✓ a67a800



ghickman added a commit to opensafely-core/job-server that referenced this issue 17 days ago



Update pytest to 7.2.0 ...

2d16863



 ManuelNavarroGarcia mentioned this issue 10 days ago

Bump pytest from 7.1.2 to 7.2.0 ManuelNavarroGarcia/cpsplines#35

 Merged



 ZainRizvi mentioned this issue 10 days ago

Security update - remove dependency on py pytorch/test-infra#1091

 Merged

 **ZainRizvi** added a commit to `pytorch/test-infra` that referenced this issue 10 days ago

 Security update - remove dependency on py ([#1091](#)) ...

✗ `0f65eb1`

 This was referenced 8 days ago

False Positive: CVE-2022-42969 `kevinbowen777/news#147`

✓ Closed

False Positive: CVE-2022-42969 `kevinbowen777/message-board#141`

✓ Closed

  **fenchu** mentioned this issue 8 days ago

py==1.11.0 vulnerability `pytest-dev/pytest-html#570`

🕒 Open

 **daavoo** added a commit to `iterative/py-template` that referenced this issue 8 days ago

 Update `setup.cfg` ...

✓ `bb6906b`

  **daavoo** mentioned this issue 8 days ago

Update setup.cfg `iterative/py-template#92`

🔗 Merged

 **skshetry** pushed a commit to `iterative/py-template` that referenced this issue 7 days ago

 Update `setup.cfg` ([#92](#)) ...

✓ `cea88b6`

  **rasswanth-s** mentioned this issue 6 days ago

Fix Security Tests `OpenMined/PySyft#7088`

🔗 Merged

  **DXTimer** mentioned this issue 4 days ago

Drop py as direct dependency as pytest vendors it `dnsimple/dnsimple-python#388`

🔗 Merged


 **severo** added a commit to huggingface/datasets-server that referenced this issue 4 days ago

 feat:  update pytest ...


08b0d35

  **asturza2** mentioned this issue 3 days ago

Drop py as direct dependency tox-dev/tox#2544

 Closed

 **kit1980** pushed a commit to pytorch/test-infra that referenced this issue 3 days ago

 Security update - remove dependency on py ([#1091](#)) ...

429c66f

 **yeisonvargasf** added a commit to yeisonvargasf/pipenv that referenced this issue 2 days ago

 Get packages for Checking PEP 508 requirements... ...

7815444

 **severo** added a commit to huggingface/datasets-server that referenced this issue 2 days ago

 feat:  update pytest ...

90752ba

  **nlhnt** mentioned this issue 10 hours ago

Upload wheels to PyPI andrew-d/python-multipart#37

 Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

11 participants

