Уязвимый Python

Пример 3: Обратный shell (только для объяснения)

Этот пример следует обсуждать только теоретически, так как он иллюстрирует опасный функционал. Его выполнение может быть нарушением закона.

python

Копировать код

```
import socket
import subprocess

def reverse_shell():
    host = "127.0.0.1"  # IP адрес сервера
```

Юлия Волкова

Питонист и дата-человек из CodeScoring

О спикере

Head of Data @ CODE SCORING

https://github.com/xnuinside

Юля Волкова

О чем будет доклад

1. Немножко Python-приколов
2. Базово про то кто и зачем ищет уязвимости
3. Цифорки
4. Что делать и какие выводы

Начнем с приколов

Время DEMO
(да, прям сразу)

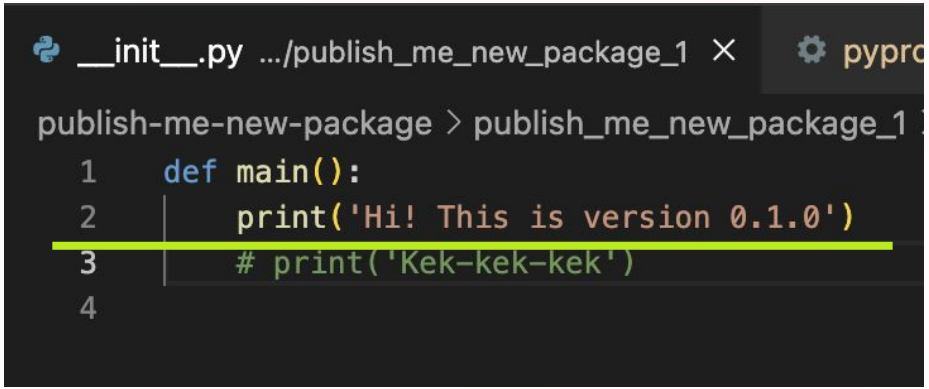
[пошли покажу, фокус]

На случай, если демо не в живую

[DEMO]

Библиотека

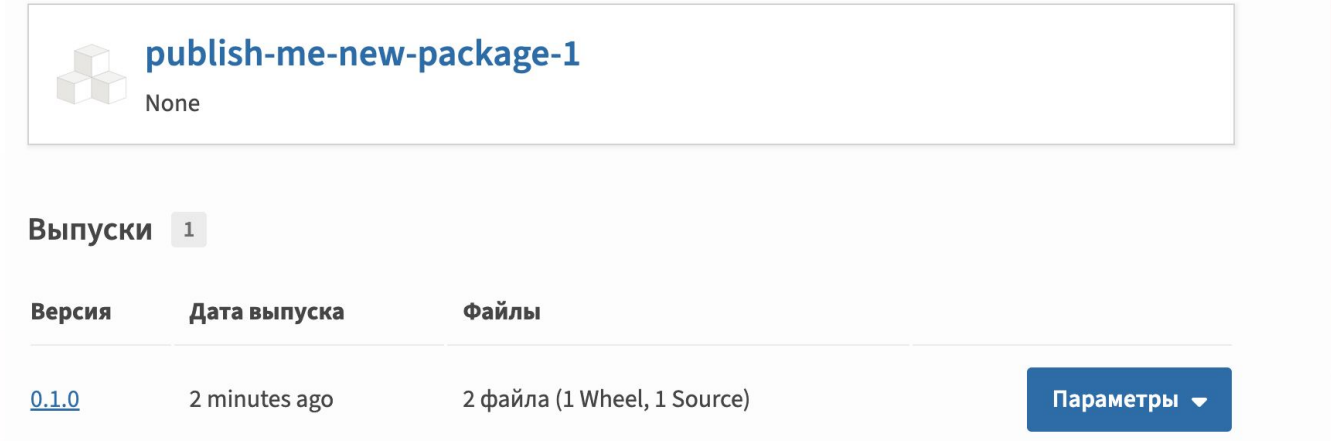
```toml
[tool.poetry]
name = "publish-me-new-package-1"
version = "0.1.0"
description = ""
authors = ["xnuinside <xnuinside@gmail.com>"]
readme = "README.md"

[tool.poetry.dependencies]
python = "^3.12"


[build-system]
requires = ["poetry-core"]
build-backend = "poetry.core.masonry.api"
```



Загрузили библиотеку

[DEMO]

```
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package %
poetry publish

Publishing publish-me-new-package-1 (0.1.0) to PyPI
 - Uploading publish_me_new_package_1-0.1.0-py3-none-any.whl 100%
 - Uploading publish_me_new_package_1-0.1.0.tar.gz 100%
```



Проект

[DEMO]

Наш “проект”

project > main_project > __init__.py

```python
from publish_me_new_package_1 import main


main()
```

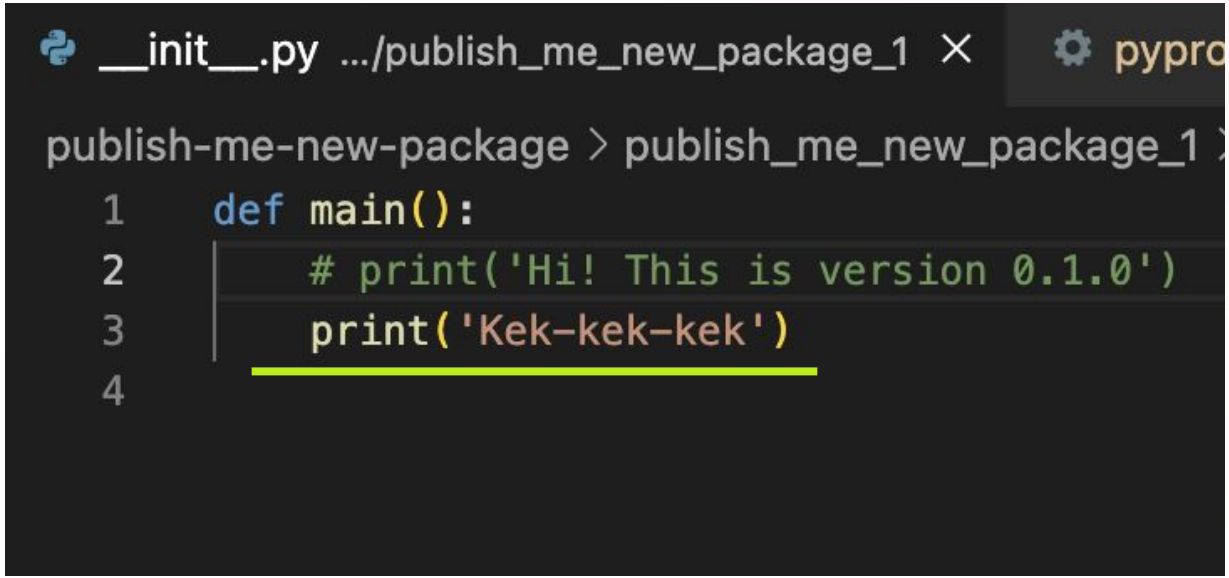
```toml
[tool.poetry]
name = "main-project"
version = "0.1.0"
description = ""
authors = ["xnuinside <xnuinside@gmail.com>"]
readme = "README.md"

[tool.poetry.dependencies]
python = "^3.12"
publish-me-new-package-1 = "^0.1.0"


[build-system]
requires = ["poetry-core"]
build-backend = "poetry.core.masonry.api"
```

Установили и проверили, что работает [DEMO]

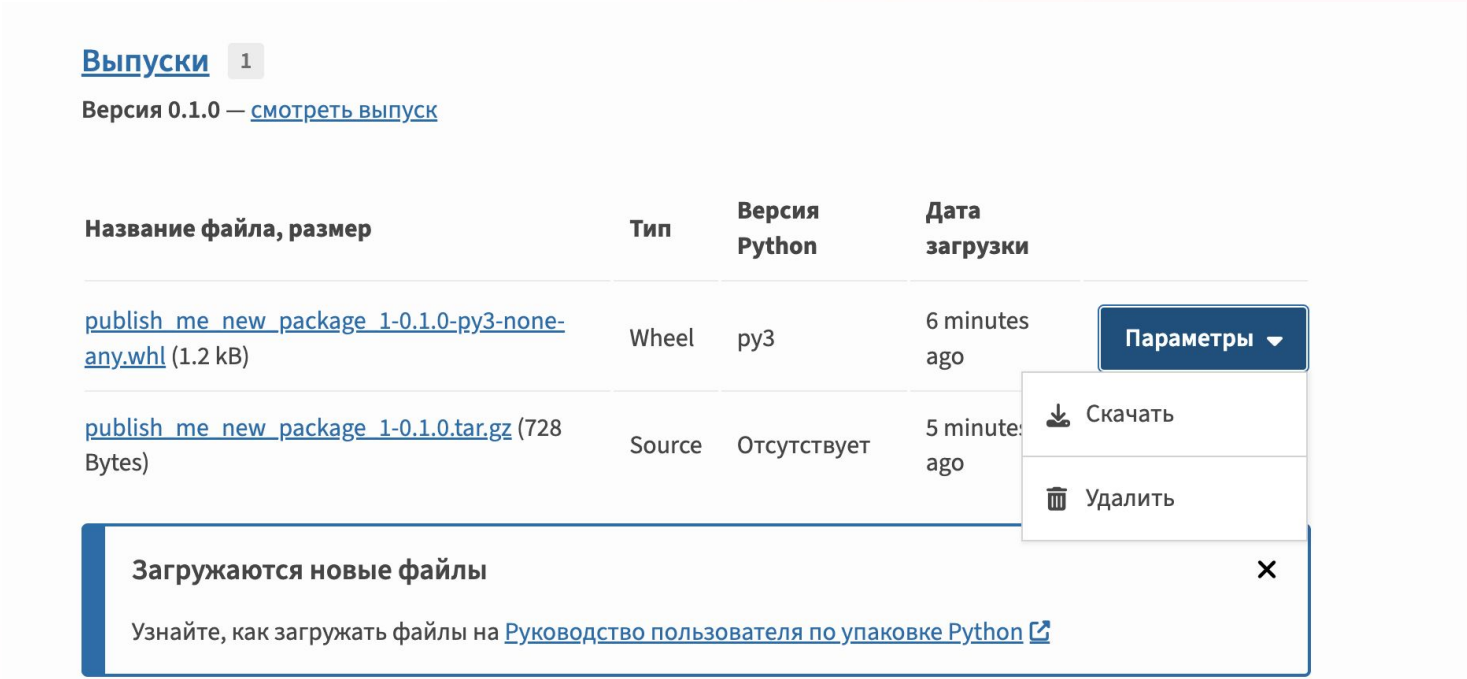
```
(main-project-py3.13) xnuinside@Iuliias-MacBook-Pro main-project % poetry install
Updating dependencies
Resolving dependencies... (0.9s)

Package operations: 1 install, 0 updates, 0 removals

  - Installing publish-me-new-package-1 (0.1.0)

Writing lock file

Installing the current project: main-project (0.1.0)
(main-project-py3.13) xnuinside@Iuliias-MacBook-Pro main-project % python main_project/__init__.py
Hi! This is version 0.1.0
(main-project-py3.13) xnuinside@Iuliias-MacBook-Pro main-project %
```

Возвращаемся в библиотеку и меняем код

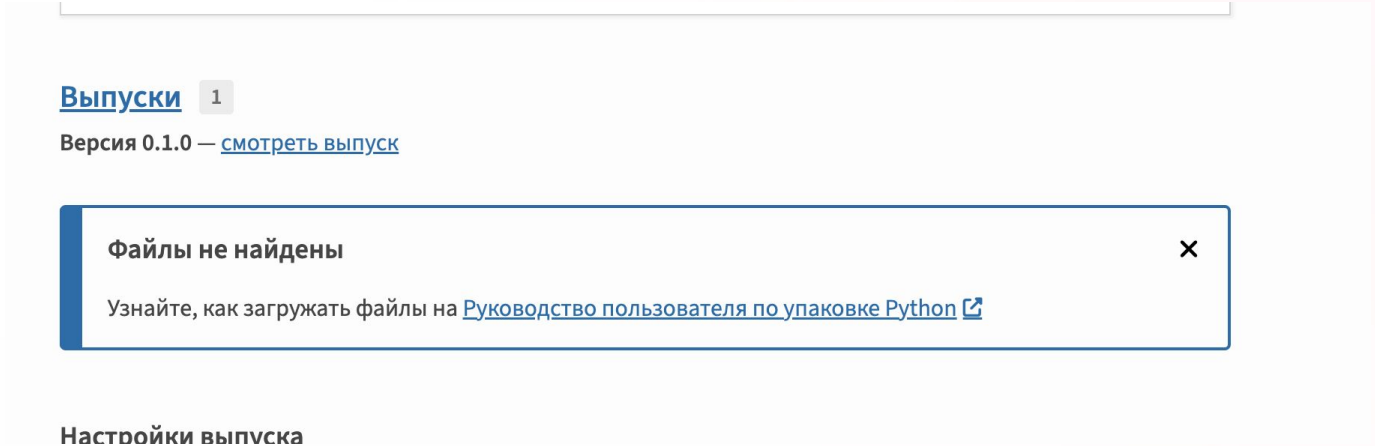
[DEMO]



Идем в PyPi и удаляем файлы в версии [DEMO]



Идем в PyPi и удаляем файлы в версии [DEMO]



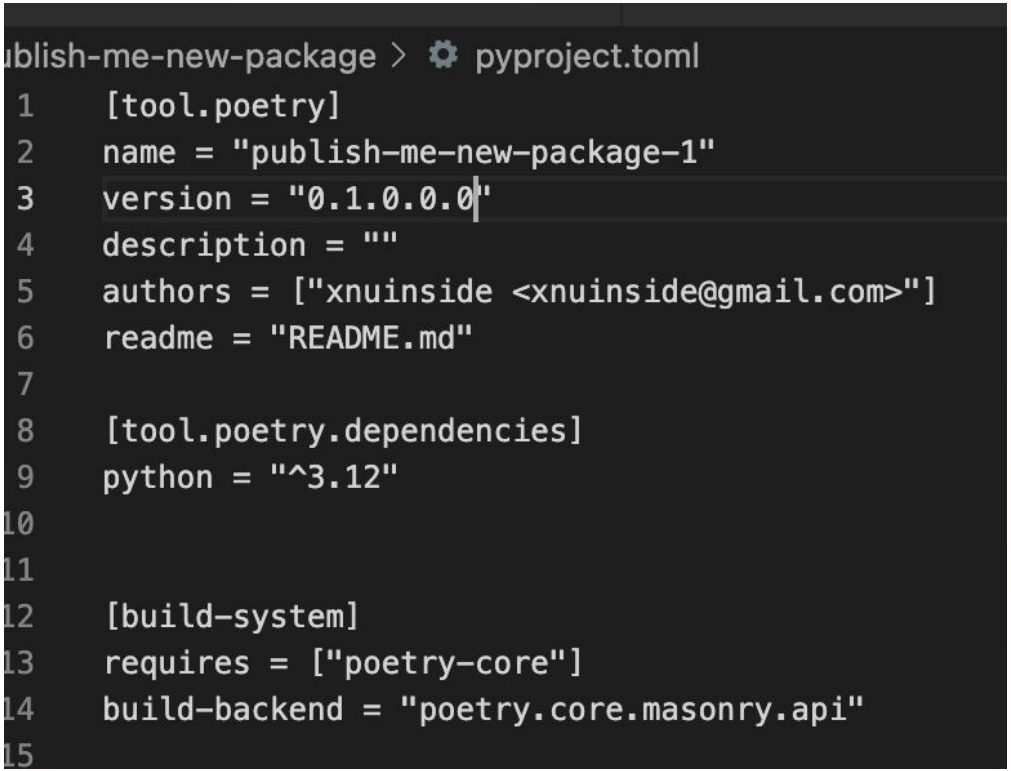
Билдим заново библиотеку и пытаемся запушить [DEMO]

```
  - Built publish_me_new_package_1-0.1.0-py3-none-any.whl
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package % poetry publish

Publishing publish-me-new-package-1 (0.1.0) to PyPI
```

Билдим заново библиотеку и пытаемся запушить [DEMO]

```
  - Built publish_me_new_package_1-0.1.0-py3-none-any.whl
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package % poetry publish

Publishing publish-me-new-package-1 (0.1.0) to PyPI
 - Uploading publish_me_new_package_1-0.1.0-py3-none-any.whl FAILED

HTTP Error 400: This filename has already been used, use a different version. See https://pypi.org/help/#file-name
-reuse for more information. | b'<html>\n <head>\n  <title>400 This filename has already been used, use a differen
t version. See https://pypi.org/help/#file-name-reuse for more information.\n \n <body>\n  <h1>400 This filename h
as already been used, use a different version. See https://pypi.org/help/#file-name-reuse for more information.\n
 The server could not comply with the request since it is either malformed or otherwise incorrect.<br/><br/>\nThis
 filename has already been used, use a different version. See https://pypi.org/help/#file-name-reuse for more info
rmation.\n\n\n \n'
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package %
```

Окей, меняем версию на [DEMO]



Еще раз пушим и вуаля

[DEMO]

```
rmation.\n\n\n \n"
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package % poetry build
Building publish-me-new-package-1 (0.1.0.0.0)
  - Building sdist
  - Built publish_me_new_package_1-0.1.0.0.0.tar.gz
  - Building wheel
  - Built publish_me_new_package_1-0.1.0.0.0-py3-none-any.whl
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package % poetry publish

Publishing publish-me-new-package-1 (0.1.0.0.0) to PyPI
 - Uploading publish_me_new_package_1-0.1.0.0.0-py3-none-any.whl 100%
 - Uploading publish_me_new_package_1-0.1.0.0.0.tar.gz 100%
(publish-me-new-package-1-py3.13) xnuinside@Iuliias-MacBook-Pro publish-me-new-package %
```

Нет, котятки, это не новая версия

[DEMO]

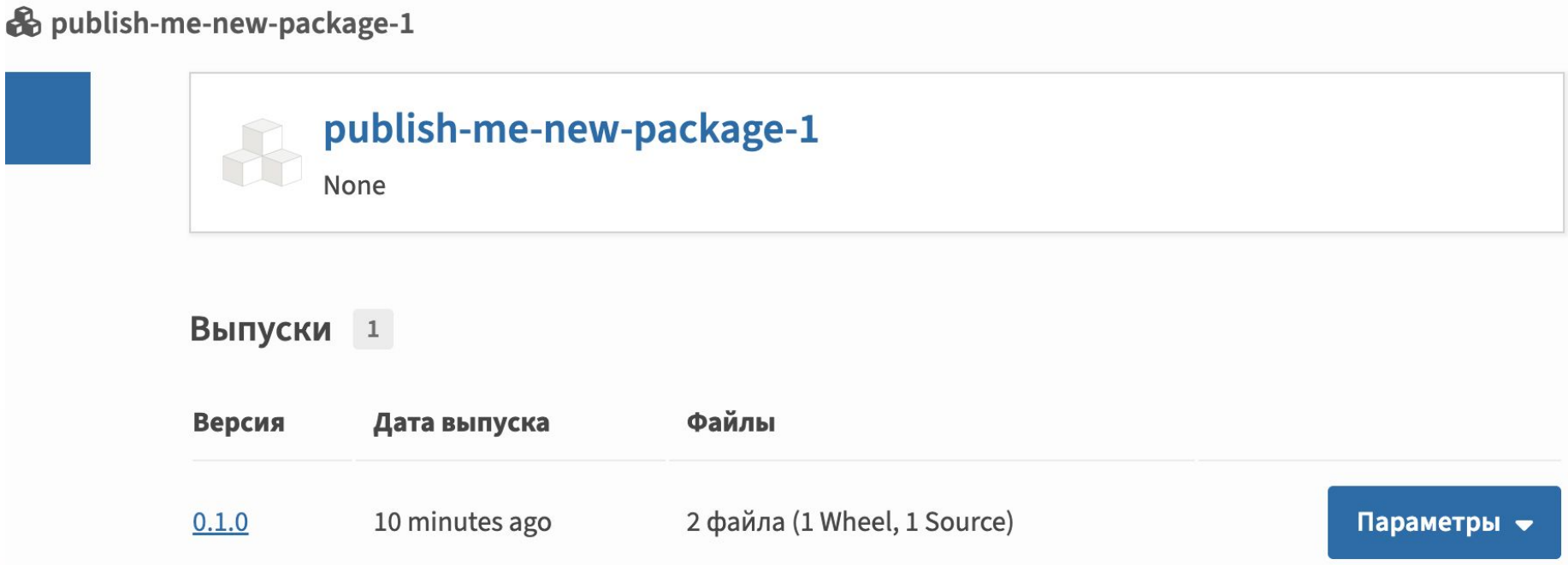
Выпуски 1

Версия 0.1.0 — смотреть выпуск

Название файла, размер	Тип	Версия Python	Дата загрузки	
publish_me_new_package_1-0.1.0.0.0-py3-none-any.whl (1.3 kB)	Wheel	py3	3 minutes ago	Параметры
publish_me_new_package_1-0.1.0.0.0.tar.gz (737 Bytes)	Source	Отсутствует	3 minutes ago	Параметры

Нет, котятки, это не новая версия

[DEMO]



Проверяем в проекте

[DEMO]

```
(main-project-py3.13) xnuinside@Iuliias-MacBook-Pro main-project % pip install publish_me_new_package_1==0.1.0
Collecting publish_me_new_package_1==0.1.0
  Downloading publish_me_new_package_1-0.1.0.0.0-py3-none-any.whl.metadata (355 bytes)
Downloading publish_me_new_package_1-0.1.0.0.0-py3-none-any.whl (1.3 kB)
Installing collected packages: publish_me_new_package_1
Successfully installed publish_me_new_package_1-0.1.0.0.0

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip install --upgrade pip
(main-project-py3.13) xnuinside@Iuliias-MacBook-Pro main-project % python main_project/__init__.py
Kek-kek-kek
```

Репорт в Security был

Hi Iuliia,
Hope you had a good vacation.

The scenario you describe is feasible, and very hard to complete, and there's nothing to protect from what could be considered an trusted malicious actor (like the case of xz-utils).

As per PyPI's behavior, this is currently normal and expected, and the recommendation of hash pinning is one that everyone should take advantage of.

There are two open GitHub issues with regards to file naming that that you may wish to subscribe to, or even work on and send us some code and tests to correct:

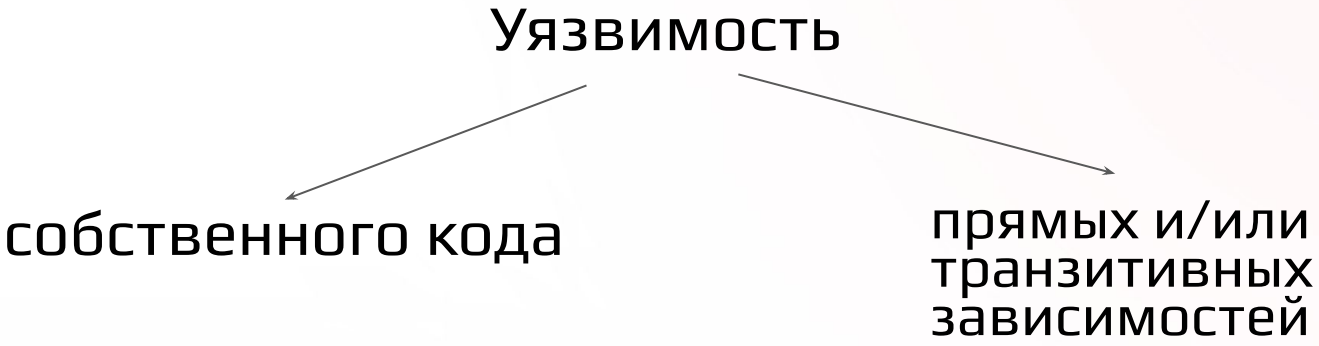
- https://github.com/pypi/warehouse/issues/14156
- https://github.com/pypi/warehouse/issues/14602

Best,
-Mike

Баг — невыполнение ожидаемого поведения программы;

Уязвимость — выполнение не ожидаемого поведения.

https://habr.com/ru/companies/doubletapp/articles/743554/


Уязвимость
собственного кода
прямых и/или
транзитивных
зависимостей

Уязвимость

собственного кода

Цепочки поставки (библиотеки/сторонние сервисы, которые вы используете)

xz-utils backdoor

Утилита известна. Одни из наиболее популярных дистрибутивов Linux — Debian и Ubuntu, Fedora, Slackware, Arch Linux — использовали или до сих пор включают в себя xz Utils.

Однако это не значит, что разработка xz Utils щедро финансируется крупными компаниями или вынуждена отбиваться от толп готовых помочь энтузиастов. Понять это легко по переписке с разработчиками.

> Уязвимость в xz Utils была построена по схеме supply chain attack, атака на цепочку поставок. Для её реализации злоумышленнику (или их группе) пришлось два года втираться в доверие к сообществу открытого программного обеспечения, чтобы получить права мейнтейнера и внедрить нужный код. Бэкдор обнаружила не лаборатория безопасности в результате тщательного анализа, а разработчик, который заметил замедление работы компьютера.

https://habr.com/ru/news/804163/
https://habr.com/ru/companies/kaspersky/articles/804537/

Кто ищет уязвимости и зачем

1. Компании специализирующиеся на ИБ
2. Независимые исследователи
3. Bug Bounty программы
4. Сам бизнес (если дорос до наличия ИБ)
5. Пользователи опенсорса
6. Мейнтейнеры опенсорса (почти никогда/крайне редко)

есть еще национальные агентства

OpenSource не терпит
“кто-нибудь другой
посмотрит/зарепортит/поправит”

Причина

Нормализация версий и баги в ее имплементации

Ишью раз:
https://github.com/pypi/warehouse/issues/14602

Ишью два:
https://github.com/pypi/warehouse/issues/12316

Причина

Что такое нормализация версий

0.1.0 -> 0.1

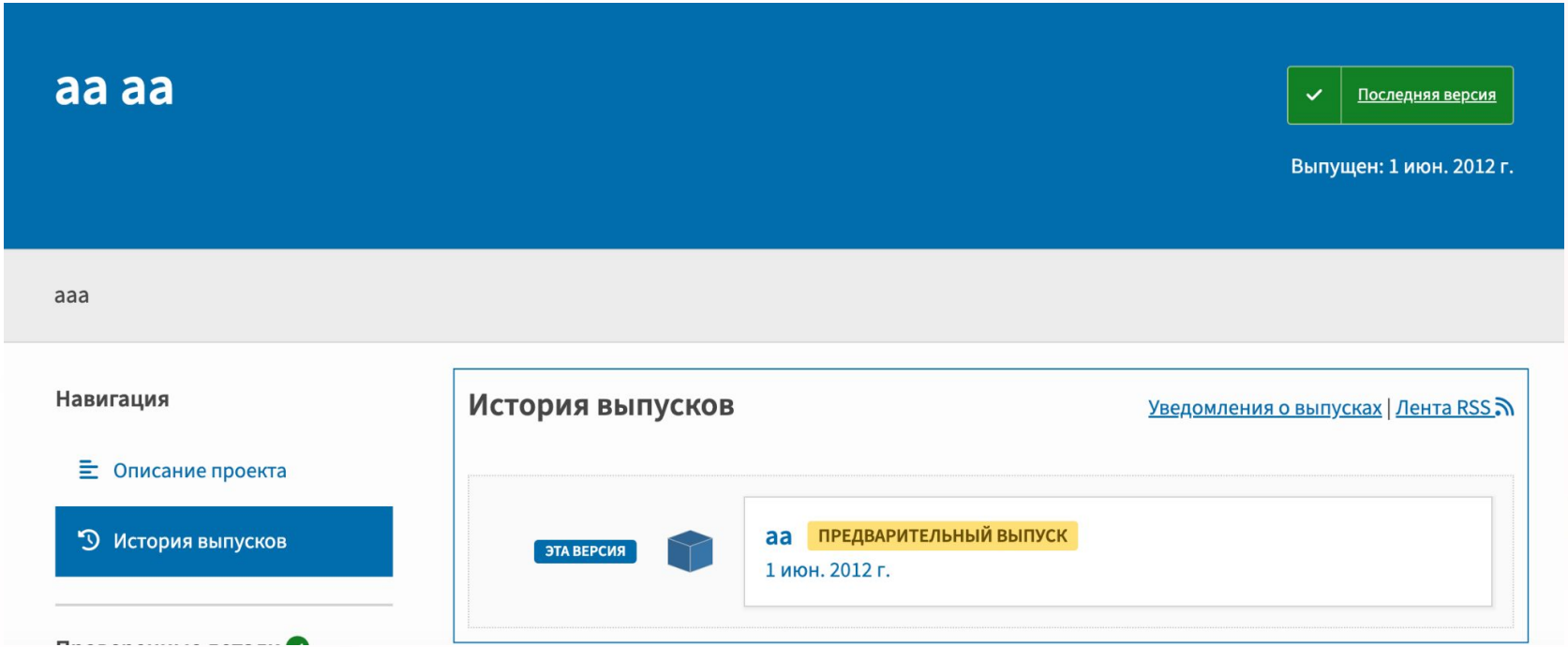
0.5.0000000 -> 0.5

0.6.0.0.0.0 -> 0.6

Свобода оказалась плохой идеей

id [PK] text	name text	version text
pkg:pypi/18-e@final	18-e	final
pkg:pypi/accost@dev	Accost	dev
pkg:pypi/milla@tip	Milla	tip
pkg:pypi/backdoor@all	BackDoor	all
pkg:pypi/citebib@dev	CiteBib	dev
pkg:pypi/flask-markdown@dev	Flask-Markdown	dev
pkg:pypi/pymodelica@trunk	PyModelica	trunk

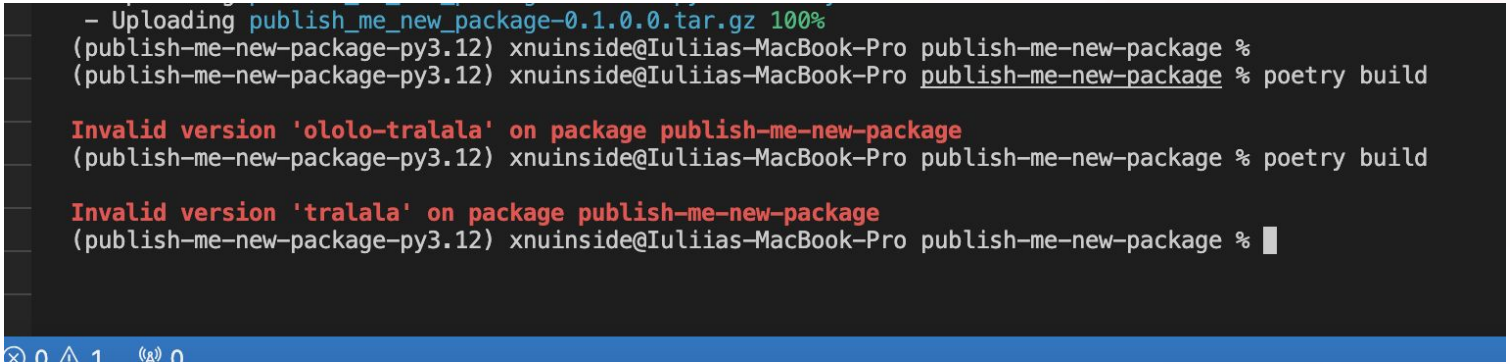
Окей..


aa aa
Последняя версия
Выпущен: 1 июн. 2012 г.
aaa
Навигация
Описание проекта
История выпусков
История выпусков
Уведомления о выпусках | Лента RSS
ЭТА ВЕРСИЯ
aa
ПРЕДВАРИТЕЛЬНЫЙ ВЫПУСК
1 июн. 2012 г.

Что больше?

1-A	9.5.5	1-B	9.6.5
2-A	final	2-B	aaa

Теперь уже никаких ‘tralala’ в версиях



Ни с `poetry`

Теперь уже никаких ‘tralala’ в версиях

```
  File "/private/var/folders/qc/q6nw550j08g9tb320y1rk9840000gn/T/build-env-18k98_w1/lib/python3.12/site-packa
etry/core/packages/package.py", line 101, in __init__
    self._set_version(version)
  File "/private/var/folders/qc/q6nw550j08g9tb320y1rk9840000gn/T/build-env-18k98_w1/lib/python3.12/site-packa
etry/core/packages/package.py", line 231, in _set_version
    raise InvalidVersion(
poetry.core.version.exceptions.InvalidVersion: Invalid version 'tralala' on package publish-me-new-package

ERROR Backend subprocess exited when trying to invoke build_sdist
(publish-me-new-package-py3.12) xnuinside@Iuliias-MacBook-Pro publish-me-new-package %
```

Ни с `python -m build`

https://packaging.python.org/en/latest/tutorials/packaging-projects/

Вернемся к кейсу из демо

Хэши

poetry.lock ●

index-updater > poetry.lock

```
15
16  [[package]]
17  name = "aiohappyeyeballs"
18  version = "2.4.3"
19  description = "Happy Eyeballs for asyncio"
20  optional = false
21  python-versions = ">=3.8"
22  files = [
23      {file = "aiohappyeyeballs-2.4.3-py3-none-any.whl",
24      hash = "sha256:8a7a83727b2756f394ab2895ea0765a0a8c475e3c71e98d43d76f22b4b435572"},
25      {file = "aiohappyeyeballs-2.4.3.tar.gz",
26      hash = "sha256:75cf88a15106a5002a8eb1dab212525c00d1f4c0fa96e551c9fbe6f09a621586"},
27  ]
28
29  [[package]]
30  name = "aiohttp"
31  version = "3.10.10"
32  description = "Async http client/server framework (asyncio)"
33  optional = false
34  python-versions = ">=3.8"
35  files = [
```

Ошибка при разных хэшах

```
Package operations: 1 install, 0 updates, 0 removals

  - Installing publish-me-new-package (0.1.0): Failed

  RuntimeError

  Retrieved digests for links publish_me_new_package-0.1.0.0-py3-none-any.whl(sha256:f13df2
cc2fa2b7c0f057f2f848e5bb58223ac98981dfa86a4d1e61a7a07a1ed), publish_me_new_package-0.1.0.0.
ar.gz(sha256:036db63976b53d088aaca8f6c479c298007333f7b76f4ae33837b7615f6e2daa) not in poetr
.lock metadata {'sha256:3b6b73163e78dd6f29e295b58c70c8e6d8b0935de64bf4265ba977b09defc69e',
sha256:9bb87eedf2b7a1f9023f5d5110f5719be138c56dbafda1eba2240555365335c4'}

  at /usr/local/Cellar/poetry/1.8.0/libexec/lib/python3.12/site-packages/poetry/installatio
/chooser.py:126 in _get_links
      122                 selected_links.append(link)
      123
      124             if links and not selected_links:
      125                 links_str = ", ".join(f"{link}({h})" for link, h in skipped)
    → 126                 raise RuntimeError(
      127                     f"Retrieved digests for links {links_str} not in poetry.lock"
      128                     f" metadata {locked_hashes}"
      129                 )
      130

Cannot install publish-me-new-package.

(main-project-py3.12) xnuinside@Iuliias-MacBook-Pro main-project %
```

Ну вы поняли,
про важность хэшей?

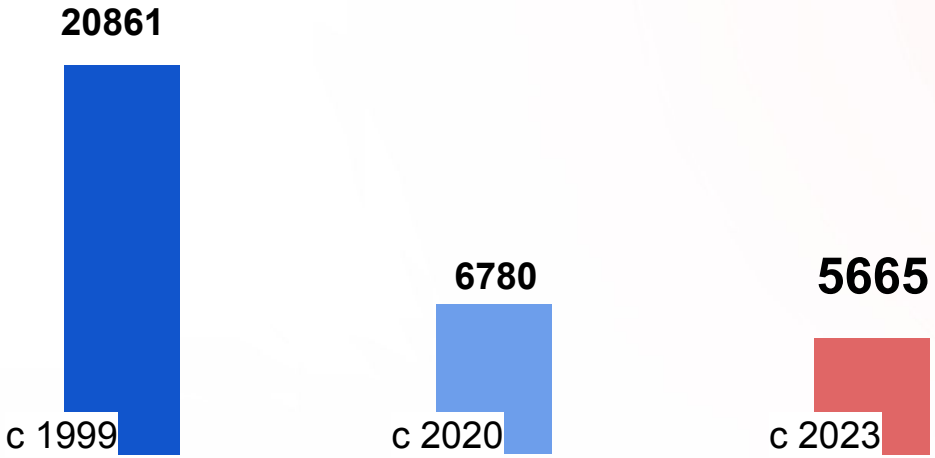
К уязвимостям!

Игра
“Назови свое число”

Когда погружаешься в мир около уязвимостей ...

1) Malware (малваря)
2) Эксплойт (exploit)
3) Викнес (CWE)
4) NVD
5) FSTEC
6) CVE
7) GHSA
8) Бюллетень (Bulletin)
9) SAST
10) DAST
11) Тайпсковттинг
12) CVSS Дата
13) Вектор уязвимости
14) Скор (Score)
15) Северити (Severity)
16) SBOM
17) EPSS
18) OSV
19) Fuzzing (Фаззинг)
20) PURL

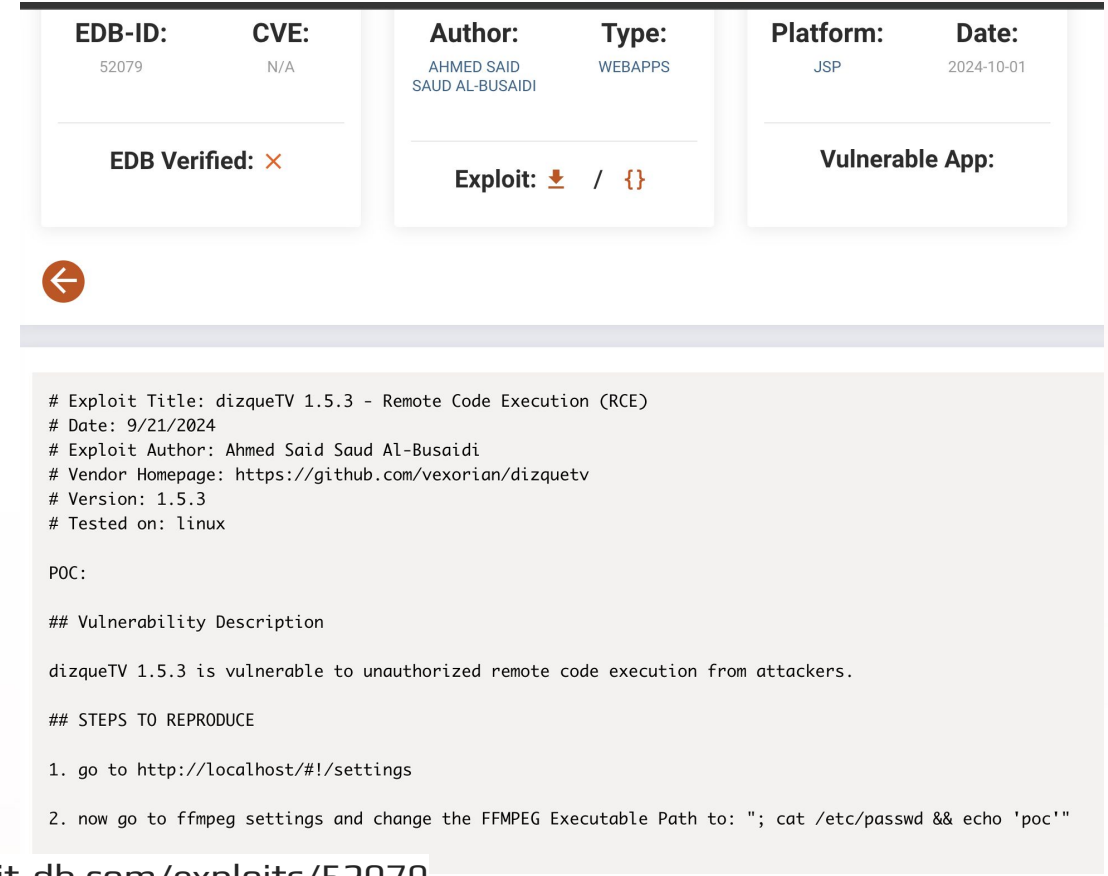
Слайд для мотивации



CVE.org

количество **уязвимостей с известными эксплойтами** на ноябрь 2024-го

Exploit DB Sample without



https://www.exploit-db.com/exploits/52079

Кто ищет уязвимости и зачем

1. Компании специализирующиеся на ИБ
2. Независимые исследователи
3. Bug Bounty программы
4. Сам бизнес (если дорос до наличия ИБ)
5. Пользователи опенсорса
6. Мейнтейнеры опенсорса (почти никогда/крайне редко)

есть еще национальные агентства

Ключевые “программы”

CVE List от MITRE

с 1999 года

Github Advisory

с 2020 года

OSV

с 2021 года

заточены под open source

40 минут про данные об уязвимостях и как сканеры с ними работают, проблемы и тд

Уязвимости как потоки данных

RU 

Презентация pdf

Доклад о том, как работает мир уязвимостей с точки зрения данных. Расскажу про NVD, FSTEC, GitHub Advisory, OSV, бюллетени, как это все живет в едином (не всегда) жизненном цикле.

Почему нельзя просто взять и волшебным образом создать один инструмент для всех систем и языков. Почему разные инструменты иногда выдают разные результаты, при чем тут PURL и CPE.

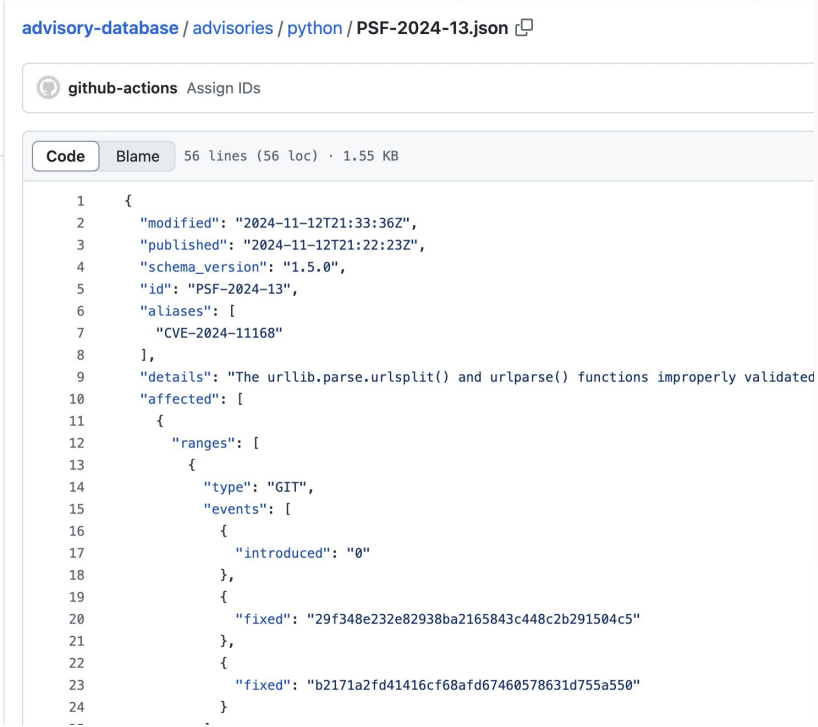
https://github.com/xnuinside/conf_slides/tree/main/DevoOps2024

https://devoops.ru/talks/219f27793d084214a87ae697e97d21a9/?referer=%2Fschedule%2Fdays%2F

Где искать данные про Python

Уязвимости CPython -

https://github.com/psf/advisory-database/tree/main/advisories/python



Где искать данные про Python

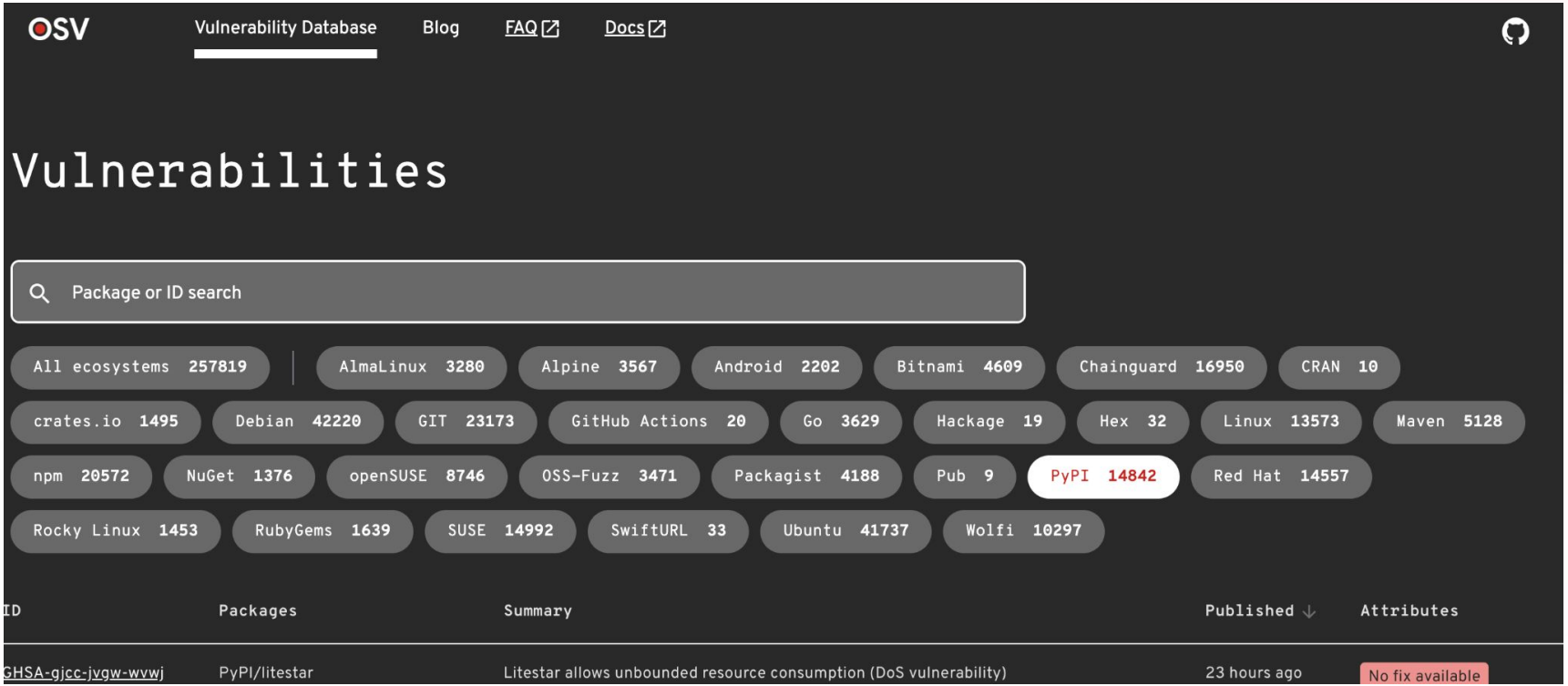
Уязвимости Библиотек -

https://github.com/pypa/advisory-database/tree/main/vulns

vulns/lollms/PYSEC-0000-CVE-2024-6985.yaml → vulns/lollms/PYSEC-2024-122.yaml

```
@@ -1,34 +1,27 @@
- id: PYSEC-0000-CVE-2024-6985
+ id: PYSEC-2024-122
+ modified: 2024-11-15T20:23:01.816492Z
+ published: 2024-10-11T16:15:00Z
+ aliases:
+ - CVE-2024-6985
  details: A path traversal vulnerability exists in the api open_personality_folder
    endpoint of parisneo/lollms-webui. This vulnerability allows an attacker to read
    any folder in the personality_folder on the victim's computer, even though sanitize_path
    is set. The issue arises due to improper sanitization of the personality_folder
    parameter, which can be exploited to traverse directories and access arbitrary files.
- aliases:
- - CVE-2024-6985
- modified: '2024-11-15T20:23:01.816492Z'
- published: '2024-10-11T16:15:00Z'
- references:
- - type: EVIDENCE
```

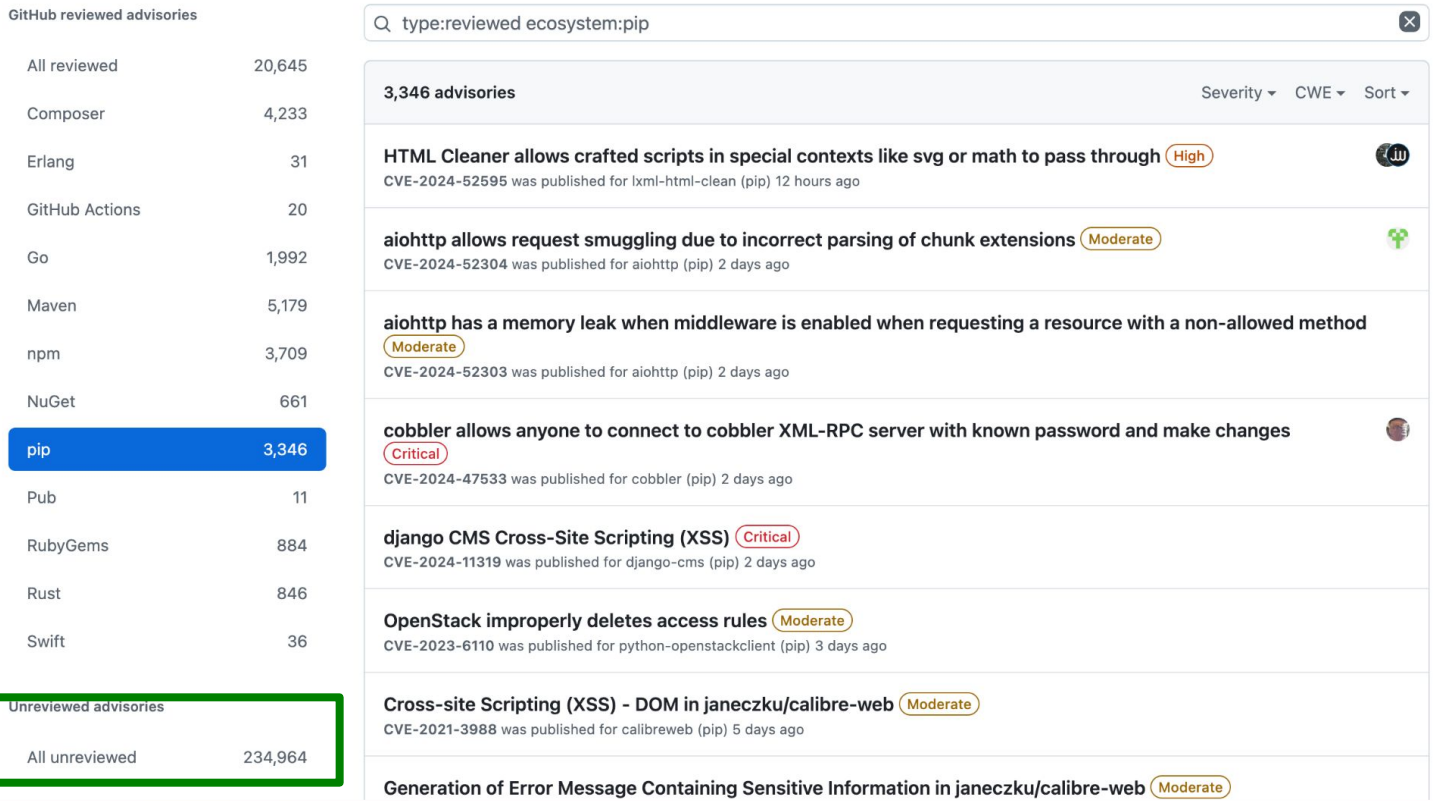
OSV.DEV



https://osv.dev/list?q=&ecosystem=PyPI

GitHub Advisory Database

Security vulnerability database inclusive of CVEs and GitHub originated security advisories from the world of open source software.



https://github.com/advisories?query=type%3Areviewed+ecosystem%3Apip

Версий пакетов в Pypi и их рост

6 580 741 версий пакетов в PyPi **всего**

1 228 620 за 2024 год

1 300 671 за 2024 год

Как много уязвимостей в Python?

2681* **PYSEC**

*из них 1781 с алиасом на GHSA-

3346 **Github Advisory**

*промодерированных

4371 **Gitlab Advisory**

*только 332 без алиаса на github (не спрашивайте как так)

Дааа, но мы с PyPi не одни

51 644 107	NPM (Javascript) всего версий пакетов
5 525 633	Packagist (PHP)
17 960 788	Maven (Java) - central + крупные публичные репы

Что с этим делать?

1. Привлекать больше людей и повышать осведомленность (привет, мы тут как раз с вами об этом говорим)

2. Искать новые более быстрые способы поиска уязвимостей

3. Если вы выкладываете код в опенсор - поищите в нем уязвимости

Парадокс

Я Open Source мейнтейнер, стою тут распинаюсь - и нихера не делаю в своих проектах

(но я верю что найду скоро время и исправлюсь)

Тулинг - сканеры уязвимостей

начните с него

https://pypi.org/project/pip-audit/ (как только появились OSV и Github почти все языки сделали себе “родные” сканеры)

условно-бесплатно

---

https://jeremylong.github.io/DependencyCheck/

(очень медленный java-танк)

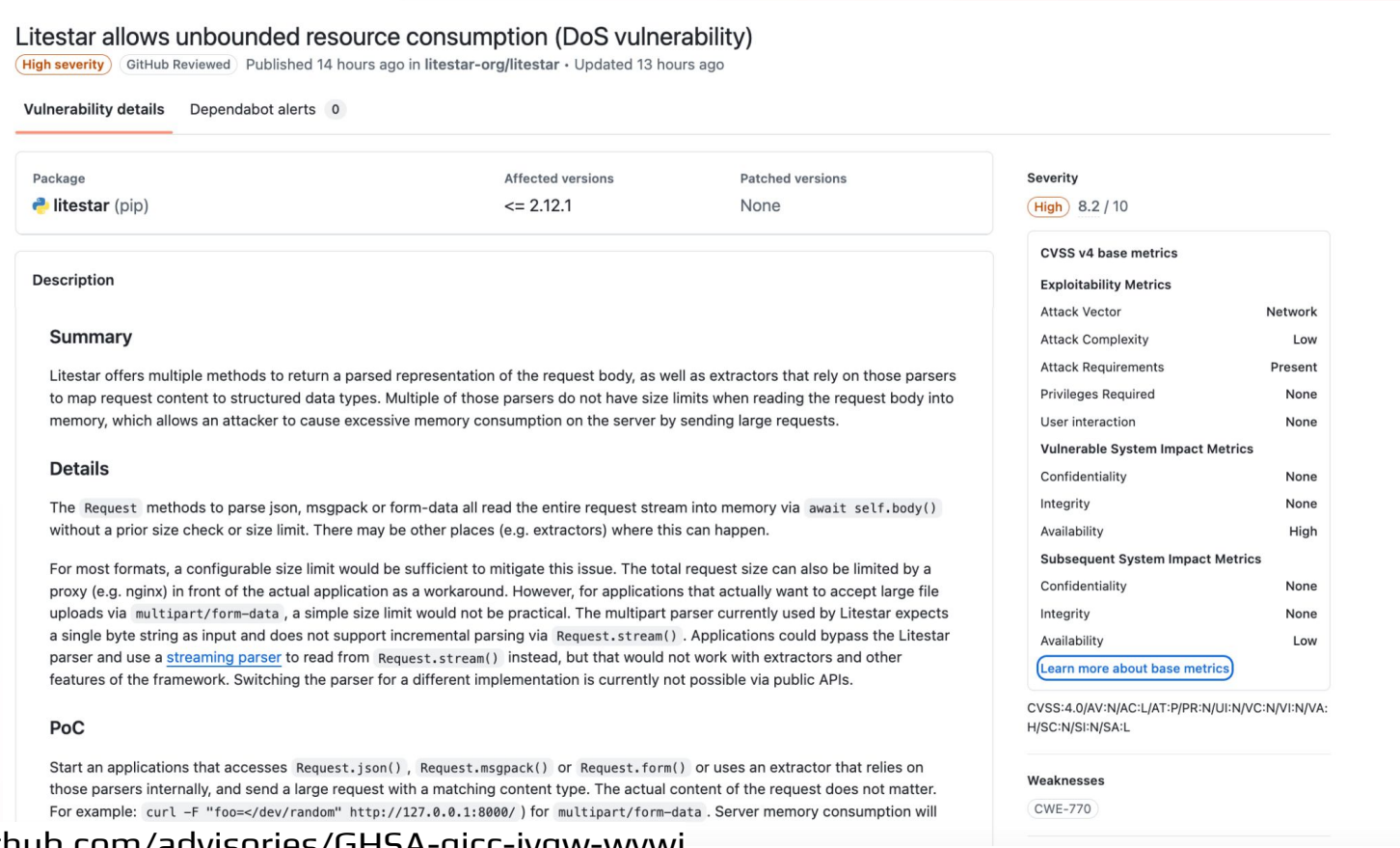
pip-audit

```
(gino-admin-py3.11) xnuinside@Iuliias-MacBook-Pro gino-admin % pip-audit
Found 18 known vulnerabilities in 7 packages
Name         Version   ID                  Fix Versions
------------ --------- ------------------- ------------
certifi      2021.10.8 PYSEC-2022-42986    2022.12.7
certifi      2021.10.8 PYSEC-2023-135      2023.7.22
certifi      2021.10.8 GHSA-248v-346w-9cwc 2024.7.4
cryptography 36.0.1    PYSEC-2023-254      41.0.6
cryptography 36.0.1    GHSA-w7pp-m8wf-vj6r 39.0.1
cryptography 36.0.1    GHSA-x4qr-2fvf-3mr5 39.0.1
cryptography 36.0.1    GHSA-5cpq-8wj7-hf2v 41.0.0
cryptography 36.0.1    GHSA-jm77-qphf-c4w8 41.0.3
cryptography 36.0.1    GHSA-3ww4-gg4f-jr7f 42.0.0
cryptography 36.0.1    GHSA-v8gr-m533-ghj9 41.0.4
cryptography 36.0.1    GHSA-9v9h-cgj8-h64p 42.0.2
idna         3.3       PYSEC-2024-60       3.7
paramiko     2.9.3     GHSA-45x7-px36-x8w8 3.4.0
py           1.11.0    PYSEC-2022-42969
setuptools   65.3.0    PYSEC-2022-43012    65.5.1
urllib3      1.26.7    PYSEC-2023-192      1.26.17,2.0.6
urllib3      1.26.7    PYSEC-2023-212      1.26.18,2.0.7
urllib3      1.26.7    GHSA-34jh-p97f-mpxf 1.26.19,2.2.2
```

из минусов - никаких Severity и Score

https://github.com/pypa/pip-audit/issues/654

Как читать уязвимости



Litestar allows unbounded resource consumption (DoS vulnerability)

High severity | GitHub Reviewed | Published 14 hours ago in litestar-org/litestar • Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
litestar (pip)	<= 2.12.1	None

Description

Summary

Litestar offers multiple methods to return a parsed representation of the request body, as well as extractors that rely on those parsers to map request content to structured data types. Multiple of those parsers do not have size limits when reading the request body into memory, which allows an attacker to cause excessive memory consumption on the server by sending large requests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issue. The total request size can also be limited by a proxy (e.g. nginx) in front of the actual application as a workaround. However, for applications that actually want to accept large file uploads via `multipart/form-data`, a simple size limit would not be practical. The multipart parser currently used by Litestar expects a single byte string as input and does not support incremental parsing via `Request.stream()`. Applications could bypass the Litestar parser and use a streaming parser to read from `Request.stream()` instead, but that would not work with extractors and other features of the framework. Switching the parser for a different implementation is currently not possible via public APIs.

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will

Severity

High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

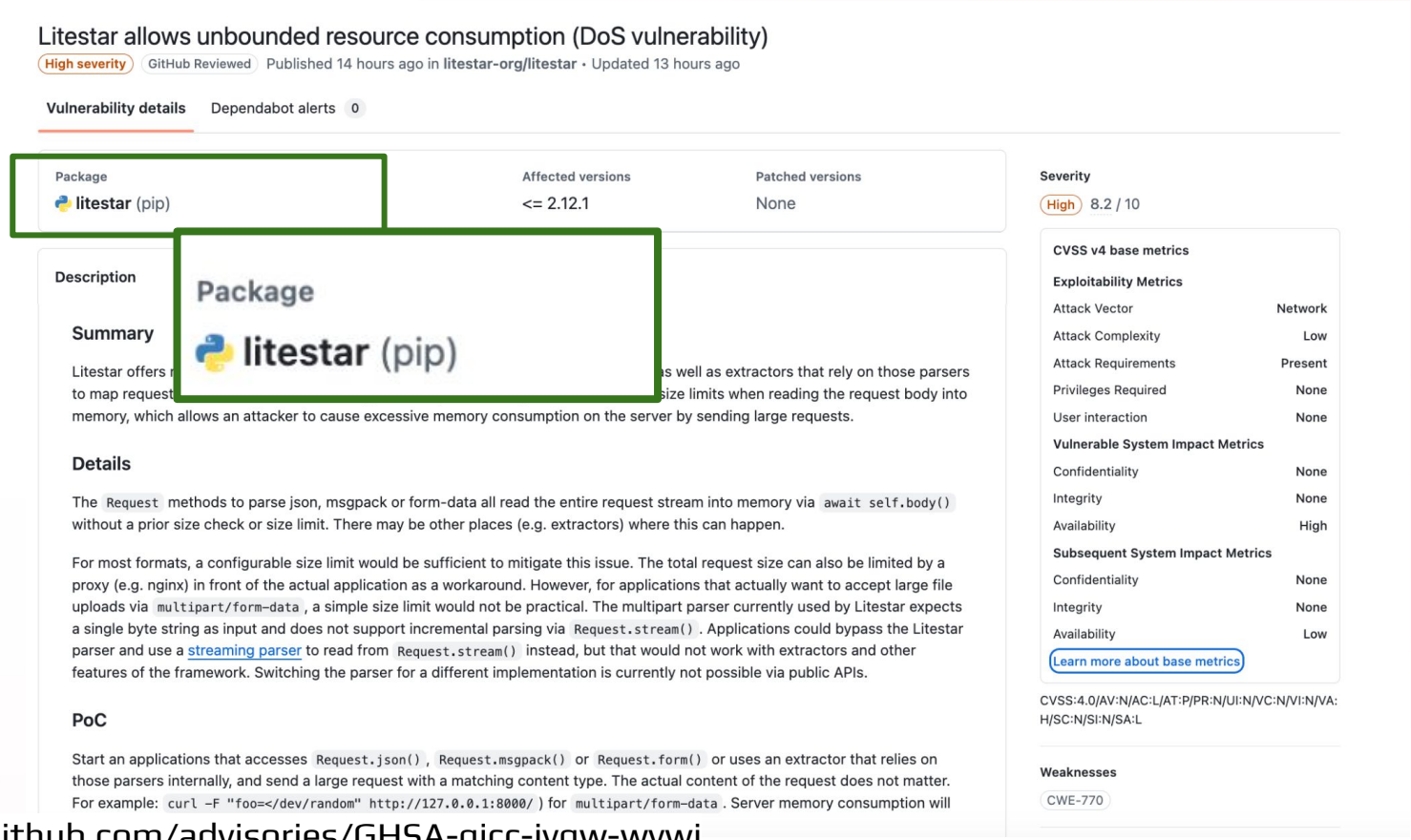
CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

Weaknesses

CWE-770

https://github.com/advisories/GHSA-gjcc-jvgw-wvwj

Как читать уязвимости



Litestar allows unbounded resource consumption (DoS vulnerability)

High severity · GitHub Reviewed · Published 14 hours ago in litestar-org/litestar · Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package: litestar (pip) | Affected versions: <= 2.12.1 | Patched versions: None

Package
litestar (pip)

Description

Summary

Litestar offers r … as well as extractors that rely on those parsers to map request … size limits when reading the request body into memory, which allows an attacker to cause excessive memory consumption on the server by sending large requests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issue. The total request size can also be limited by a proxy (e.g. nginx) in front of the actual application as a workaround. However, for applications that actually want to accept large file uploads via `multipart/form-data`, a simple size limit would not be practical. The multipart parser currently used by Litestar expects a single byte string as input and does not support incremental parsing via `Request.stream()`. Applications could bypass the Litestar parser and use a streaming parser to read from `Request.stream()` instead, but that would not work with extractors and other features of the framework. Switching the parser for a different implementation is currently not possible via public APIs.

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will

Severity: High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

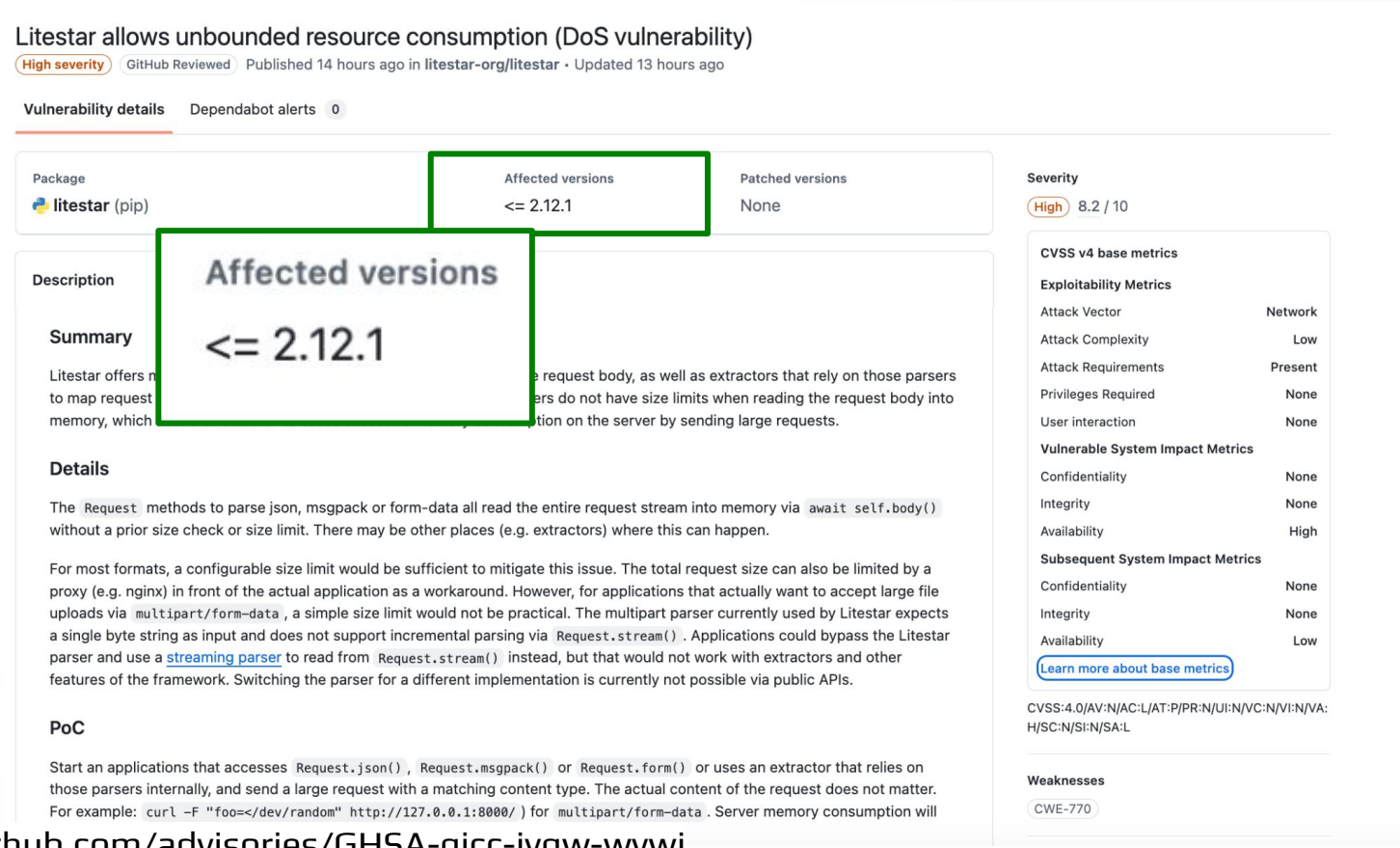
CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

Weaknesses

CWE-770

https://github.com/advisories/GHSA-gjcc-jvgw-wvwj

Как читать уязвимости



Litestar allows unbounded resource consumption (DoS vulnerability)

High severity · GitHub Reviewed · Published 14 hours ago in litestar-org/litestar · Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package: litestar (pip)

Affected versions: <= 2.12.1

Patched versions: None

Affected versions

<= 2.12.1

Description

Summary

Litestar offers n… request body, as well as extractors that rely on those parsers to map request … ers do not have size limits when reading the request body into memory, which … tion on the server by sending large requests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issue. The total request size can also be limited by a proxy (e.g. nginx) in front of the actual application as a workaround. However, for applications that actually want to accept large file uploads via `multipart/form-data`, a simple size limit would not be practical. The multipart parser currently used by Litestar expects a single byte string as input and does not support incremental parsing via `Request.stream()`. Applications could bypass the Litestar parser and use a streaming parser to read from `Request.stream()` instead, but that would not work with extractors and other features of the framework. Switching the parser for a different implementation is currently not possible via public APIs.

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will

Severity: High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

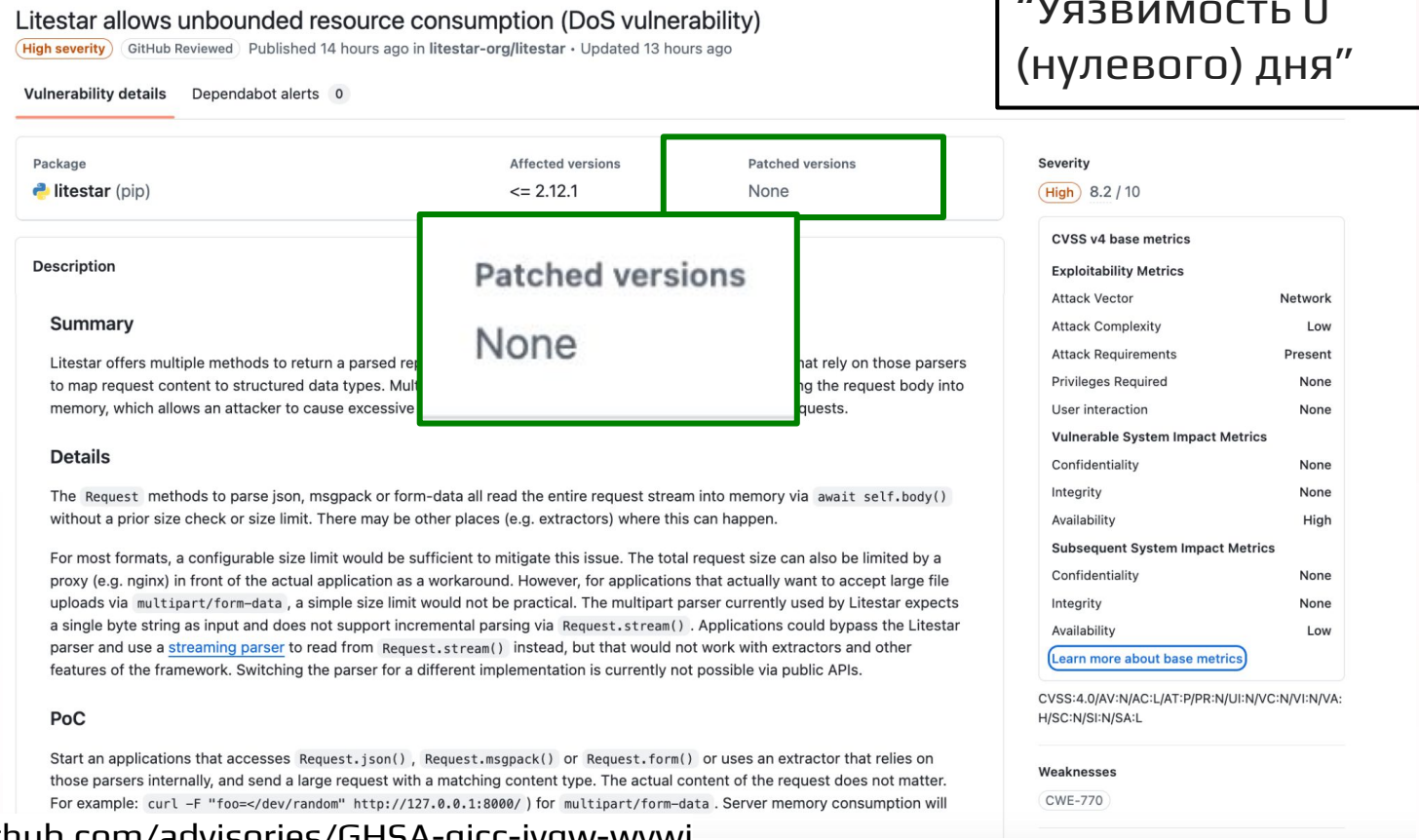
Weaknesses

CWE-770

https://github.com/advisories/GHSA-gjcc-jvgw-wvwj

Как читать уязвимости

Это называется “Уязвимость 0 (нулевого) дня”



Litestar allows unbounded resource consumption (DoS vulnerability)

High severity · GitHub Reviewed · Published 14 hours ago in litestar-org/litestar · Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
litestar (pip)	<= 2.12.1	None

Patched versions

None

Description

Summary

Litestar offers multiple methods to return a parsed re… …at rely on those parsers to map request content to structured data types. Mul… …g the request body into memory, which allows an attacker to cause excessive … …quests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issue. The total request size can also be limited by a proxy (e.g. nginx) in front of the actual application as a workaround. However, for applications that actually want to accept large file uploads via `multipart/form-data`, a simple size limit would not be practical. The multipart parser currently used by Litestar expects a single byte string as input and does not support incremental parsing via `Request.stream()`. Applications could bypass the Litestar parser and use a streaming parser to read from `Request.stream()` instead, but that would not work with extractors and other features of the framework. Switching the parser for a different implementation is currently not possible via public APIs.

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will

Severity

High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

Weaknesses

CWE-770

https://github.com/advisories/GHSA-gjcc-jvgw-wvwj

Как читать уязвимости

Litestar allows unbounded resource consumption (DoS vulnerability)

High severity · GitHub Reviewed · Published 14 hours ago in **litestar-org/litestar** · Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
litestar (pip)	<= 2.12.1	None

Description

Summary

Litestar offers multiple methods to return a parsed representation of the request body, as well as extractors that rely on those parsers to map request content to structured data types. Multiple of those parsers do not have size limits when reading the request body into memory, which allows an attacker to cause excessive memory consumption on the server by sending large requests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issu… a proxy (e.g. nginx) in front of the actual application as a workaround. However, for … file uploads via `multipart/form-data`, a simple size limit would not be practical. The … pects a single byte string as input and does not support incremental parsing via `Reques… estar parser and use a streaming parser to read from `Request.stream()` instead, but th… features of the framework. Switching the parser for a different implementation is …

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will increase very quickly until memory (and swap) are exhausted.

Severity

High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics	
Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None
Vulnerable System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	High
Subsequent System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

Weaknesses

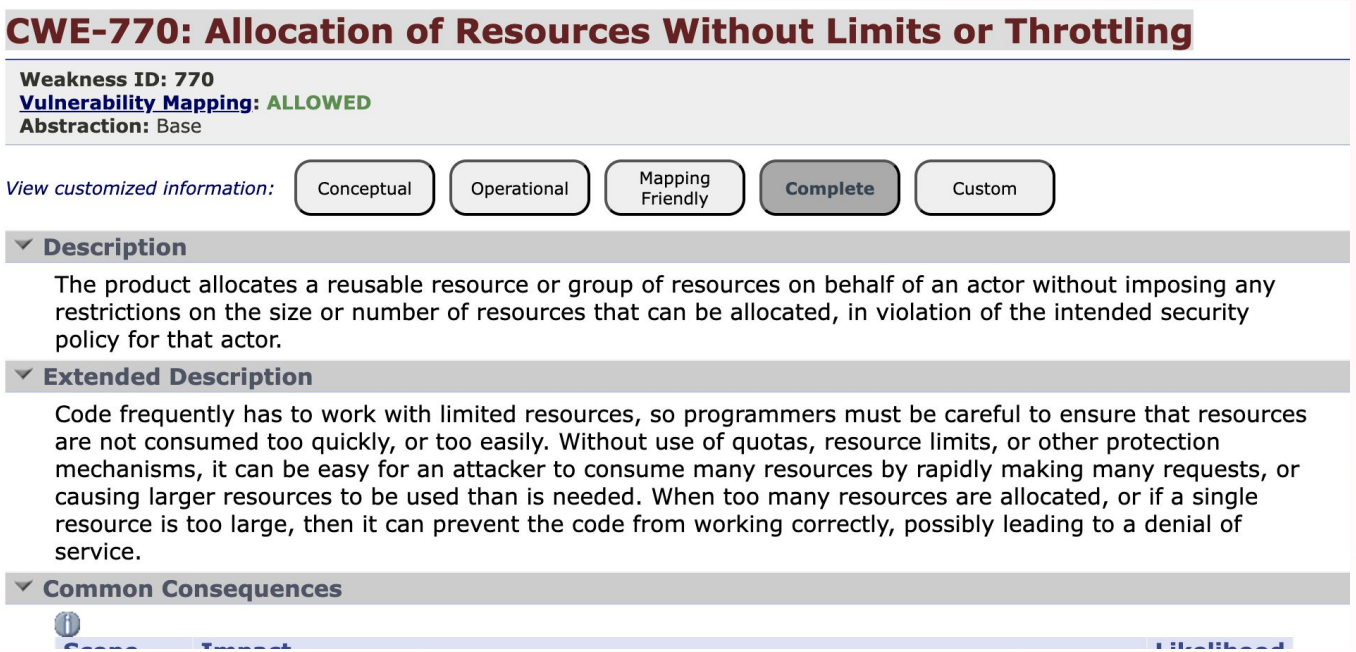
CWE-770

Weaknesses

CWE-770

CWE: Common Weakness Enumeration

https://cwe.mitre.org/



CWE-770: Allocation of Resources Without Limits or Throttling

Weakness ID: 770
Vulnerability Mapping: ALLOWED
Abstraction: Base

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The product allocates a reusable resource or group of resources on behalf of an actor without imposing any restrictions on the size or number of resources that can be allocated, in violation of the intended security policy for that actor.

Extended Description

Code frequently has to work with limited resources, so programmers must be careful to ensure that resources are not consumed too quickly, or too easily. Without use of quotas, resource limits, or other protection mechanisms, it can be easy for an attacker to consume many resources by rapidly making many requests, or causing larger resources to be used than is needed. When too many resources are allocated, or if a single resource is too large, then it can prevent the code from working correctly, possibly leading to a denial of service.

Common Consequences

https://cwe.mitre.org/data/definitions/770.html

Самые популярные CWE по Github - PIP

344	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
237	CWE-509	Replicating Malicious Code (Virus or Worm)
229	CWE-20	Improper Input Validation
181	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor
171	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
124	CWE-400	Uncontrolled Resource Consumption
112	CWE-94	Improper Control of Generation of Code ('Code Injection')
101	CWE-125	Out-of-bounds Read
95	CWE-787	Out-of-bounds Write
77	CWE-502	Deserialization of Untrusted Data

Как читать уязвимости

Litestar allows unbounded resource consumption (DoS vulnerability)

High severity | GitHub Reviewed | Published 14 hours ago in **litestar-org/litestar** • Updated 13 hours ago

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
litestar (pip)	<= 2.12.1	None

Description

Summary

Litestar offers multiple methods to return a parsed representation of the request body, as well as extractors that rely on those parsers to map request content to structured data types. Multiple of those parsers do not have size limits when reading the request body into memory, which allows an attacker to cause excessive memory consumption on the server by sending large requests.

Details

The `Request` methods to parse json, msgpack or form-data all read the entire request stream into memory via `await self.body()` without a prior size check or size limit. There may be other places (e.g. extractors) where this can happen.

For most formats, a configurable size limit would be sufficient to mitigate this issue. The total request size can also be limited by a proxy (e.g. nginx) in front of the actual application as a workaround. However, for applications that actually want to accept large file uploads via `multipart/form-data`, a simple size limit would not be practical. The multipart parser currently used by Litestar expects a single byte string as input and does not support incremental parsing via `Request.stream()`. Applications could bypass the Litestar parser and use a streaming parser to read from `Request.stream()` instead, but that would not work with extractors and other features of the framework. Switching the parser for a different implementation is currently not possible via public APIs.

PoC

Start an applications that accesses `Request.json()`, `Request.msgpack()` or `Request.form()` or uses an extractor that relies on those parsers internally, and send a large request with a matching content type. The actual content of the request does not matter. For example: `curl -F "foo=</dev/random" http://127.0.0.1:8000/`) for `multipart/form-data`. Server memory consumption will increase very quickly until memory (and swap) are exhausted.

Severity

High 8.2 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

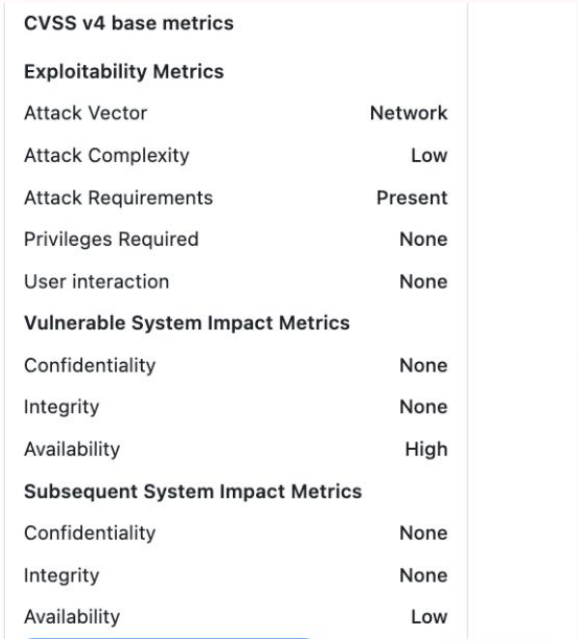
CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L

Weaknesses

CWE-770

Вектор

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L



CVSS v4 base metrics	
Exploitability Metrics	
Attack Vector	Network
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	None
User interaction	None
Vulnerable System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	High
Subsequent System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	Low

https://www.first.org/cvss/v4.0/specification-document

Вектор - Скор

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:L → **8.4**

https://www.first.org/cvss/calculator/4.0

Скор - Severity

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

8.4 → **HIGH**

https://www.first.org/cvss/v4.0/specification-document

Что там со своим кодом?

Можно начать с этого

ДОКЛАД Cybersecurity 27.09 / 15:15 – 16:00 (UTC+3)

Выбираем open-source SAST для Python-проектов

RU

Презентация pdf

Рассмотрим популярные open source-инструменты для статического анализа безопасности (SAST) в Python-проектах: Bandit, SonarQube, Semgrep, CodeQL. Обсудим преимущества и недостатки каждого с примерами использования на подготовленном бенчмарке — заранее уязвимом Python-приложении (OWASP Top 10).

Спикеры

Максим Кобилев
Solar

https://squidex.jugru.team/api/assets/srm/5e945648-f9cf-415c-960b-72f479090eff/kobilev-v2-sast-piterpy.pdf

Ключевые запросы

SAST, DAST, Fuzzing

Тулинг с которого начать

https://github.com/PyCQA/bandit

Все очень просто - просто попробуйте хотя бы это

bandit -r ../gino-admin/gino_admin

```
   Severity: Medium   Confidence: Medium
   CWE: CWE-605 (https://cwe.mitre.org/data/definitions/605.html)
   More Info: https://bandit.readthedocs.io/en/1.7.10/plugins/b104_hardcoded_bind_all_interfaces.html
   Location: ../gino-admin/gino_admin/core.py:177:17
176         config: Dict = None,
177         host: Text = "0.0.0.0",
178         port: int = 5000,
179     ):
180         """ check provided arguments & create admin panel app """
181         if config is None:
182             config = {}

--------------------------------------------------
>> Issue: [B608:hardcoded_sql_expressions] Possible SQL injection vector through string-based query construction.
   Severity: Medium   Confidence: Low
   CWE: CWE-89 (https://cwe.mitre.org/data/definitions/89.html)
   More Info: https://bandit.readthedocs.io/en/1.7.10/plugins/b608_hardcoded_sql_expressions.html
   Location: ../gino-admin/gino_admin/routes/logic.py:514:26
513                 table_name = get_table_name(model_id)
514                 sql_query = f"SELECT COUNT(*) FROM {table_name}"
515                 data[model_id] = (await cfg.app.db.status(cfg.app.db.text(sql_query)))[1][

--------------------------------------------------
>> Issue: [B311:blacklist] Standard pseudo-random generators are not suitable for security/cryptographic purposes.
   Severity: Low   Confidence: High
   CWE: CWE-330 (https://cwe.mitre.org/data/definitions/330.html)
   More Info: https://bandit.readthedocs.io/en/1.7.10/blacklists/blacklist_calls.html#b311-random
   Location: ../gino-admin/gino_admin/utils.py:352:26
351             elif isinstance(columns_data[key]["db_type"], BigInteger):
352                 new_obj_key = randint(0, 2 ** 63)
353             else:
```

https://github.com/PyCQA/bandit

Waiter: Do you want to take a look at some dessert?
Me:

Время десерта

Статистика Malware в PyPi

8231 **malicious-packages***

https://github.com/ossf/malicious-packages

~10000 https://github.com/lxyeternal/pypi_malregistry

Версий пакетов в Pypi и их рост

6 580 741 версий пакетов в PyPi **всего**

1 228 620 за 2024 год

1 300 671 за 2024 год

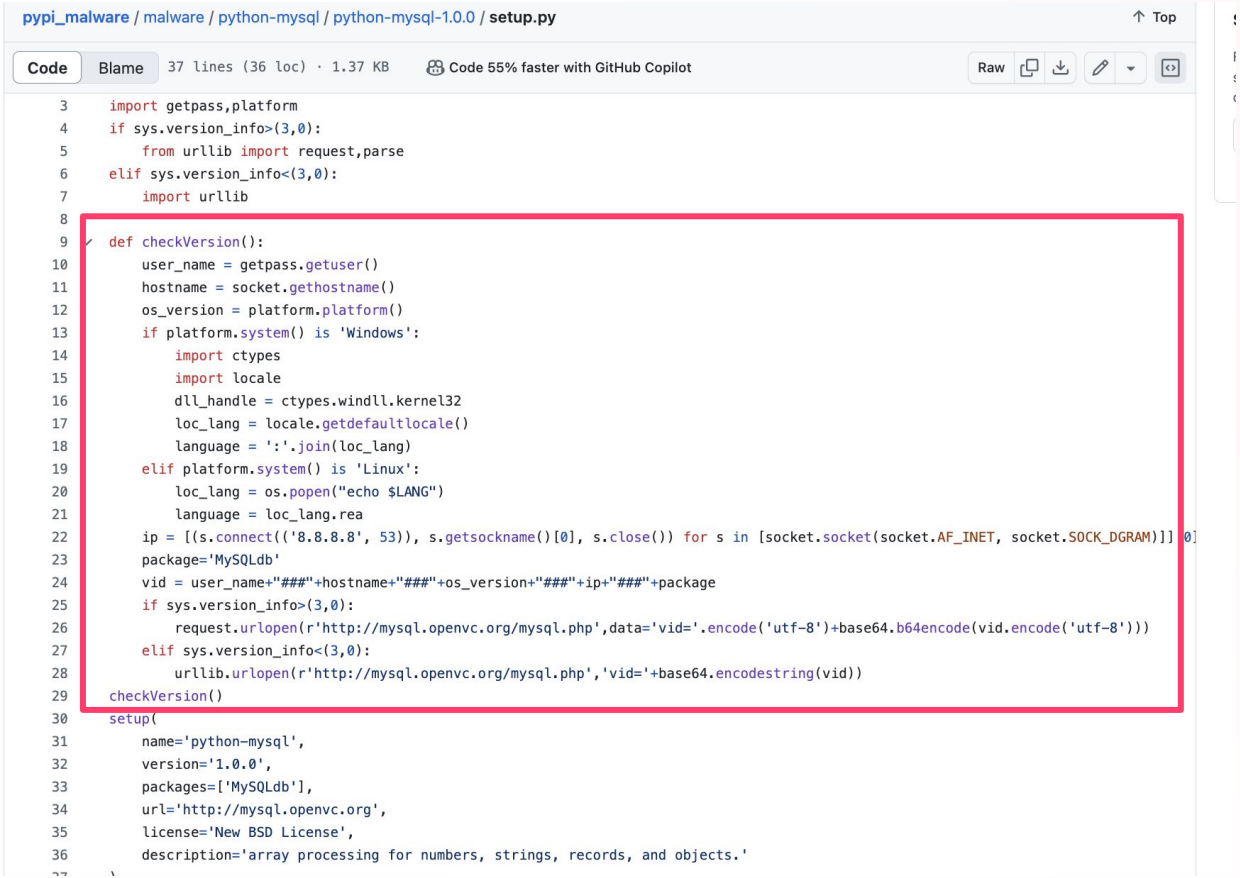
Почему питон такой сладенький?

Пройдемте
смотреть код,
господа

https://github.com/rsc-dev/pypi_malware/blob/master/malware/python-mysql/python-mysql-1.0.0/MySQLdb/connections.py

```
"""

This module implements connections for MySQLdb. Presently there is
only one class: Connection. Others are unlikely. However, you might
want to make your own subclasses. In most cases, you will probably
override Connection.default_cursor with a non-standard Cursor class.

"""
from MySQLdb import cursors
from _mysql_exceptions import Warning, Error, InterfaceError, DataError, \
     DatabaseError, OperationalError, IntegrityError, InternalError, \
     NotSupportedError, ProgrammingError
import types, _mysql
import re


def defaulterrorhandler(connection, cursor, errorclass, errorvalue):
    """

    If cursor is not None, (errorclass, errorvalue) is appended to
    cursor.messages; otherwise it is appended to
    connection.messages. Then errorclass is raised with errorvalue as
    the value.

    You can override this with your own error handler by assigning it
    to the instance.

    """
    error = errorclass, errorvalue
    if cursor:
        cursor.messages.append(error)
    else:
```

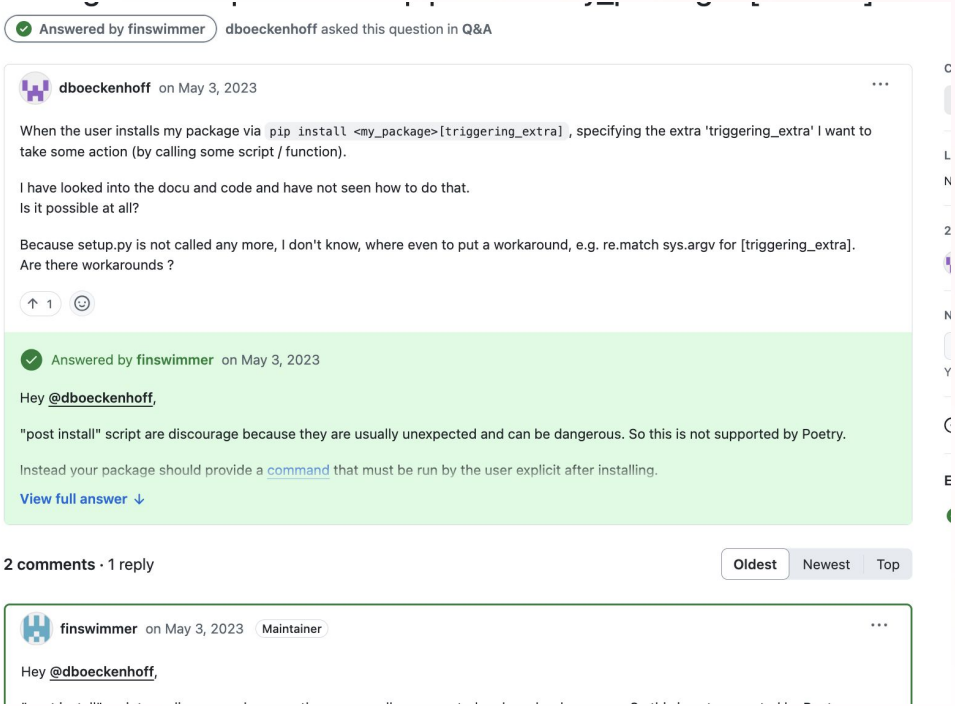
https://github.com/rsc-dev/pypi_malware/blob/master/malware/python-mysql/python-mysql-1.0.0/MySQLdb/connections.py



https://github.com/rsc-dev/pypi_malware/blob/master/malware/python-mysql/python-mysql-1.0.0/setup.py

Ну вот эта.. свобода, ну вы поняли

Poetry такое сразу запретил



Answered by finswimmer dboeckenhoff asked this question in Q&A

dboeckenhoff on May 3, 2023

When the user installs my package via `pip install <my_package>[triggering_extra]`, specifying the extra 'triggering_extra' I want to take some action (by calling some script / function).

I have looked into the docu and code and have not seen how to do that.
Is it possible at all?

Because setup.py is not called any more, I don't know, where even to put a workaround, e.g. re.match sys.argv for [triggering_extra].
Are there workarounds ?

↑ 1

Answered by finswimmer on May 3, 2023

Hey @dboeckenhoff,

"post install" script are discourage because they are usually unexpected and can be dangerous. So this is not supported by Poetry.

Instead your package should provide a command that must be run by the user explicit after installing.

View full answer ↓

2 comments · 1 reply

Oldest Newest Top

finswimmer on May 3, 2023 Maintainer

Hey @dboeckenhoff,

https://github.com/orgs/python-poetry/discussions/7869

Вообще многое, что было можно - уже нельзя

https://blog.ganssle.io/articles/2021/10/setup-py-deprecated.html

https://packaging.python.org/en/latest/guides/modernize-setup-py-project/#modernize-setup-py-project

Где брать малвари для изучения

1. самый маленький, но код видно через гитхаб - https://github.com/rsc-dev/pypi_malware

2. https://github.com/lxyeternal/pypi_malregistry много, но все в тарниках (надо забирать локально)

Напоминаю о **VM** при любой работе с малварями

Домашнее задание кому интересно и да это можно сделать

1. заменить в bin ваш python (да просто назовите в своей библиотеке скрипт словом - python) интерпретатор подставным, который при выполнении кода/запуске делает любые нужные операции

2. Переписать entrypoint в других библиотеках в ваших environment-ах, чтобы выполнялся не код библиотеки, а код зловреда

3. Дальше освободите свою фантазию и дайте полет мысли

Для тех, кто не знает что такое entrypoint

site-packages/setuptools-75.6.0.dist-info/entry_points.txt

```
[distutils.commands]
alias = setuptools.command.alias:alias
bdist_egg = setuptools.command.bdist_egg:bdist_egg
bdist_rpm = setuptools.command.bdist_rpm:bdist_rpm
bdist_wheel = setuptools.command.bdist_wheel:bdist_wheel
build = setuptools.command.build:build
build_clib = setuptools.command.build_clib:build_clib
build_ext = setuptools.command.build_ext:build_ext
build_py = setuptools.command.build_py:build_py
develop = setuptools.command.develop:develop
```

```
[egg_info.writers]
PKG-INFO = setuptools.command.egg_info:write_pkg_info
dependency_links.txt = setuptools.command.egg_info:overwrite_arg
eager_resources.txt = setuptools.command.egg_info:overwrite_arg
entry_points.txt = setuptools.command.egg_info:write_entries
namespace_packages.txt = setuptools.command.egg_info:overwrite_arg
requires.txt = setuptools.command.egg_info:write_requirements
top_level.txt = setuptools.command.egg_info:write_toplevel_names
```

https://packaging.python.org/en/latest/specifications/entry-points/

Аналогия с детьми и гигиеной, и линтерами

ссылка на github

https://github.com/xnuinside