Contextual Code Execution in nuitka/nuitka

1

Valid Reported on Jun 4th 2022

Description

The `main()` function uses the `eval()` function which can lead to contextual code execution, allowing an attacker to gain access to a system and execute commands with the privileges of the running program by setting `NUITKA_PYTHONPATH`, `NUITKA_NAMESPACES` or `NUITKA_PTH_IMPORTED` to a malicious payload string. This can lead to backdoors, reverse shells or reading/writing to privileged files.
One example of a similar vulnerability is *CVE-2022-0845* in the popular pytorch-lightning repository. [See References]

Proof of Concept

Set malicious payload

```
$ export NUITKA_PYTHONPATH='os.system("touch rickroll")'
```

Run `nuitka/__main__.py`
Code gets executed!

```
$ ls rickroll
rickroll
```

Impact

This vulnerability is capable of executing code on the target system in the context of the user running the program. This can allow an attacker to gain access to systems, read/write malicious files, etc

Remediation

A safe for patching the said vulnerability while preserving it's functionality would be to

A safe for patching the said vulnerability while preserving it's functionality would be to manually parse the environment variable and iterating over it to resolve the value of `sys.path`

Occurrences

__main__.py L108

```
setPreloadedPackagePaths(eval(os.environ["NUITKA_NAMESPACES"]))
```

__main__.py L45

```
sys.path = eval(os.environ["NUITKA_PYTHONPATH"])
```

__main__.py L117

```
setPthImportedPackages(eval(os.environ["NUITKA_PTH_IMPORTED"]))
```

References

- Code Injection in GitHub repository pytorchlightning/pytorch-lightning prior to 1.6.0.

CVE
CVE-2022-2054
(Published)

Vulnerability Type
CWE-77: Command Injection

Severity
High (8.4)

Registry
Pypi

Affected Version
0.9

Visibility
Public

Status
Fixed

Found by

whokilleddb
@whokilleddb
master

This report was seen 714 times.

We are processing your report and will contact the **nuitka** team within 24 hours. 6 months ago

whokilleddb modified the report 6 months ago

We have contacted a member of the **nuitka** team and are waiting to hear back 6 months ago

We have sent a follow up to the **nuitka** team. We will try again in 7 days. 6 months ago

A **nuitka/nuitka** maintainer 6 months ago — Maintainer

I acknowledge the issue. I do not have an immediate fix. I guess using e.g. JSON dumps would be safe?

whokilleddb 6 months ago — Researcher

Well, yes, that could be safe. Also, what type of data do these fields contain? If it's booleans or numbers(integers/floats) you can use `ast.literal_eval` . If you can give me a general idea about the nature and type of data, I can try to come up with a fix. Thank You 😃

A **nuitka/nuitka** maintainer 6 months ago — Maintainer

I have made a change for factory https://nuitka.net/doc/factory.html where I replaced all usages in Nuitka that are expecting to work on literals with `ast.literal_eval` which is a shame I didn't

know exists. It will be in the 0.9 release, I do not plan any hotfixes for 0.8 right now being on holiday, i.e. less infrastructure.

Not sure what actions to take, or if I should make this a thing in the changelog, or how to credit you in the commit or changelog. I welcome suggestions. Trying to be a good upstream here.

A **nuitka/nuitka** maintainer 6 months ago Maintainer

I just saw there is a fix bounty, in my mind pointing out that function should mean you get that.

whokilleddb 6 months ago Researcher

I'm just happy to help secure a repository I have been so intrigued by in the past. As for any credits, I believe @admin can assign this bug a `CVE` and that will be really helpful for users and organizations using the repository alike.

Thank you for maintaining and developing such an amazing project 😀

Jamie Slome 6 months ago Admin

@whokilleddb - if you submit a fix using the controls on this report page, the maintainer will be able to credit you with finding and fixing the issue.

@maintainer - if you think this is a legitimate vulnerability, I encourage you to resolve this as valid using the `Resolve` button below 👇 👇 👇

Once you have a commit SHA for the fix, you can confirm the fix against the report as well. This will make the report public and publish a CVE for the report.

A **nuitka/nuitka** maintainer validated this vulnerability 5 months ago

whokilleddb has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

A **nuitka/nuitka** maintainer marked this as fixed in **0.9** with commit **096477** 5 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

__main__.py#L117 has been validated ✔

__main__.py#L45 has been validated ✔

__main__.py#L108 has been validated ✔

Sign in to join this conversation