Argument Injection can lead to Local Code Execution

Moderate **neersighted** published **GHSA-9xgj-fcgf-x6mw** on Aug 31

Package

poetry (pip)

Affected versions	Patched versions
< 1.1.9	1.1.9

Description

Observation

When handling dependencies that come from a Git repository instead of a registry, Poetry uses various commands, such as `git clone`. These commands are being constructed using user input (e.g. the repository URL). When building the commands, Poetry correctly avoids Command Injection vulnerabilities by passing an array of arguments instead of a command string. However, there is the possibility that a user input starts with a dash (`-`) and is therefore treated as an optional argument instead of a positional one. This can lead to Code Execution because some of the commands have options that can be leveraged to run arbitrary executables.

To clone a repository, Poetry builds a git clone command, but fails to validate or sanitize the repository location properly:

`poetry/core/vcs/git.py`:

```
  def clone(self, repository: str, dest: Path) -> str:
      return self.run("clone", "--recurse-submodules", repository, str(dest))
```

Since this value comes from the `pyproject.toml` file, it can contain any character, including a leading dash.

Impact

This vulnerability can lead to Arbitrary Code Execution, which would lead to the takeover of the system. If a developer is exploited, the attacker could steal credentials or persist their access. If the exploit happens on a server, the attackers could use their access to attack other internal systems.
Since this vulnerability requires a fair amount of user interaction, it is not as dangerous as a remotely exploitable one. However, it still puts developers at risk when dealing with untrusted files in a way they think is safe, because the exploit still works when the victim tries to make sure nothing can happen, e.g. by vetting any Git or Poetry config files that might be present in the directory.
This kind of attack vector has been used in the past to target security researchers by sending them projects to collaborate on, so we believe that there is a non-negligible risk.

Patches

1.1.8 || 1.2.0b1

Remediation

Upgrade to version 1.1.9 || 1.2.0b1

References

Fix PR

For more information

If you have any questions or comments about this advisory, email us at security@python-poetry.org

Severity

Moderate

CVE ID

CVE-2022-36069

Weaknesses

CWE-88

Credits


paul-gerste-sonarsource