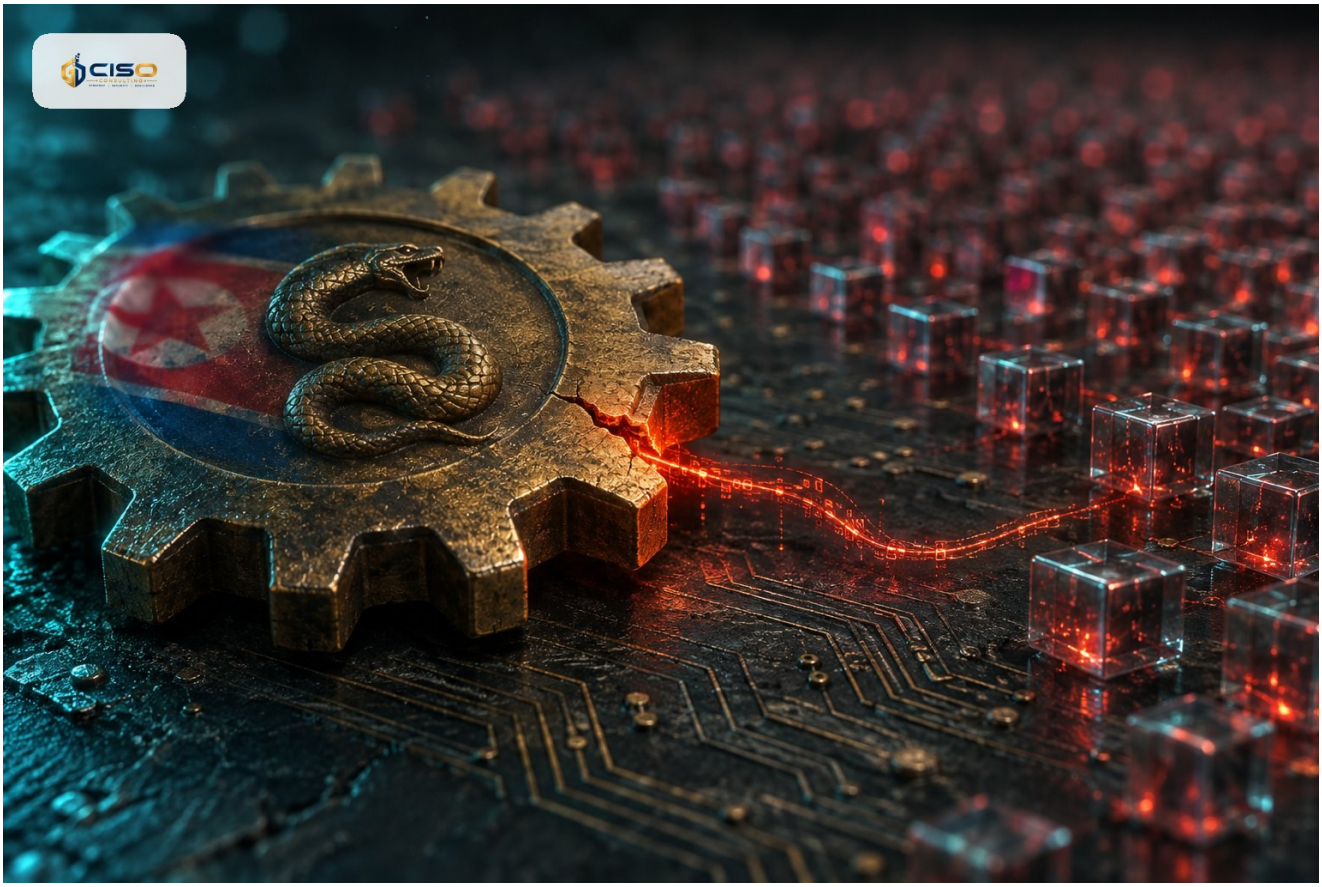
PolinRider campaign plants Python loaders in over 7,600 GitHub repositories



A North Korea-linked campaign called PolinRider is reported to have poisoned over 7,600 GitHub repositories with a Python loader, possibly since February 2025, according to SafeDep.

- The PolinRider campaign is linked to North Korea and has affected more than 7,600 GitHub repositories.
- The malicious Python loader is placed in a small utils package or in conftest.py.
- The loader runs when Python imports the file.
- SafeDep's report suggests the activity may date back to February 2025.

What was reported

The Hacker News, citing a SafeDep report published this week, describes a campaign named **PolinRider**, attributed to North Korea-linked actors. They are reported to have planted a Python loader in GitHub repositories, with more than 7,600 repositories affected.

SafeDep says the loader hides in a small utils package or in conftest.py. When Python imports the file, the loader runs. The source excerpt is truncated, so details of what the loader fetches or delivers are not available here.

There is also evidence that the activity may have begun as early as February 2025.

Why it matters

The technique relies on code that developers and build pipelines import routinely. A helper module or a test configuration file rarely draws close review, so a malicious change there can execute inside developer machines or CI systems without an obvious trigger.

Institutions that depend on open-source Python code, or that clone and test third-party repositories, should treat this as a supply chain exposure worth checking, without assuming they are affected.

Suggested approach

Review what your teams pull from public repositories, look for unexpected changes in utility and test-configuration files, and consult SafeDep's full report for indicators once you have access to it.

What to do now

1. Inventory which public GitHub repositories your developers and CI pipelines clone or depend on.
2. Search internal and cloned Python code for unexpected content in utils packages and conftest.py files.
3. Run untrusted repositories and their tests only in isolated environments without access to production credentials.
4. Obtain SafeDep's full report and check its indicators against your repositories and endpoints.
5. Brief development teams on reviewing third-party code before importing or testing it.