Microsoft Agent Framework PyPI package compromised to steal credentials



Attackers compromised Microsoft's official Agent Framework Python package on PyPI and published two malicious versions designed to steal cloud credentials and other sensitive data from developers' systems.

- Microsoft's official Agent Framework package on PyPI was compromised.
- Two malicious versions were published.
- The malicious code is designed to steal cloud credentials and other sensitive information.
- The targets are developers' systems.

What happened

According to BleepingComputer, attackers compromised Microsoft's official Agent Framework Python package on PyPI. They published two malicious versions of the package, built to harvest cloud credentials and other sensitive information from the systems of developers who installed them.

Why it matters

The package is distributed under Microsoft's name through an official channel, so teams may install or update it without extra scrutiny. This is a software supply chain risk: a trusted dependency becomes the delivery route. Stolen cloud credentials can in turn give an attacker access to cloud environments well beyond the developer's own machine.

The source excerpt available to us does not specify the affected version numbers, the exact date range, or the full technical behaviour of the malware. Readers should consult the original report and any Microsoft or PyPI notice for those details before acting on specific version checks.

What to do next

Institutions that use Python for AI or agent development should establish whether the package is present in any project, build pipeline or developer workstation, and whether either malicious version was installed. If so, treat the affected environment as potentially exposed and rotate the cloud credentials reachable from it.

What to do now

1. Search repositories, build pipelines and developer machines for the Microsoft Agent Framework package and identify installed versions.
2. Check the original report and vendor or PyPI notices to confirm which versions are malicious.
3. If a malicious version was installed, isolate the system and rotate all cloud credentials, tokens and keys accessible from it.
4. Review cloud audit logs for unusual activity linked to the exposed credentials.
5. Pin dependency versions and route package installs through a vetted internal mirror or approval process.