Key commitment policy bypass via shared key cache in AWS Encryption SDK for Python

Moderate **antonf-amzn** published **GHSA-v638-38fc-rhfv** 1 hour ago

Package

aws-encryption-sdk (pip)

Affected versions	Patched versions
< 2.5.1	4.0.5
< 3.3.0	3.3.1
< 4.0.4	4.0.5

Description

Summary

AWS Encryption SDK (ESDK) for Python is a client-side encryption library. An issue exists where, under certain circumstances, a specific cryptographic algorithm downgrade in the caching layer might allow an authenticated local threat actor to bypass key commitment policy enforcement via a shared key cache, resulting in ciphertext that can be decrypted to multiple different plaintexts.

Impact

This issue requires all of the following conditions to be true: (1) Two ESDK for Python clients with different commitment policies share a single CachingCryptoMaterialsManager instance within the same process. (2) The client with the weaker commitment policy encrypts first, warming the cache. (3) Both clients use matching encryption contexts. (4) Both clients use the pre-configured default algorithm suite.

These conditions may occur during a migration from ESDK for Python v1 to newer versions, as v1 did not support key commitment.

When the weaker-policy client encrypts first, the cache stores encryption materials that do not enforce key commitment. Subsequent callers — including those configured to require key commitment — are served these cached materials instead of generating new ones that satisfy their policy. This results in encryption without key commitment, meaning the same ciphertext can be validly decrypted to different plaintexts under different keys (the "Invisible Salamanders" issue; see GHSA-wqgp-vphw-hphf). A threat actor who controls ciphertext can cause a recipient to decrypt a message different from what the sender encrypted, breaking message integrity.

Impacted versions

- From 2.0 to 2.5.1
- From 3.0 to 3.3.0
- From 4.0 to 4.0.4

Patches

This issue has been addressed in ESDK for Python versions 3.3.1 and 4.0.5. We recommend upgrading to the latest version and ensuring any forked or derivative code is patched to incorporate the new fixes.

Workarounds

If a customer requires operating multiple instances of the Python ESDK each with differently configured key commitment policies, they must not share a key cache.

References
If you have any questions or comments about this advisory, we ask that you contact AWS Security via our vulnerability reporting page or directly via email to aws-security@amazon.com. Please do not create a public GitHub issue.

Acknowledgement

We would like to thank 1seal.org for collaborating on this issue through the coordinated vulnerability disclosure process.

Severity

Moderate 5.7 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Local

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

Learn more about base metrics

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

CVE ID

CVE-2026-6550

Weaknesses

No CWEs