New issue

[Bug]: .ovpack import may allow unsafe archive member paths (ZIP Slip / path traversal) #342

Closed

Labels bug

13ernkastel opened last month Contributor

Bug Description

Importing a crafted .ovpack archive can process unsafe ZIP member names (for example ../, absolute paths, or unexpected root prefixes) before strict validation. This may allow path traversal-style behavior during import if malicious archive entries are accepted.

Steps to Reproduce

Steps to Reproduce
Initialize an OpenViking client.

Create a malicious .ovpack/ZIP with entries such as:

pkg/../../escape.txt

/abs/path/file.txt

C:\evil.txt

other/file.txt (different root than expected archive root)

Call import_ovpack("evil.ovpack", "viking://resources/...").

Observe that import can proceed into unsafe/invalid path handling flow (or fail late/inconsistently, depending on archive content and environment).

Expected Behavior

Expected Behavior
import_ovpack should reject unsafe archive entries immediately and consistently before any write/mkdir operation:

reject absolute paths

reject drive-prefixed paths

reject any .. traversal segment

reject entries outside the expected archive root folder

Actual Behavior

Actual Behavior
Unsafe member paths are not uniformly blocked at the earliest validation point, creating a potential security gap and inconsistent behavior for malicious archives.

Minimal Reproducible Example

```
import io
import zipfile
import asyncio
from openviking.storage.local_fs import import_ovpack

class DummyFS:
    async def stat(self, parent, ctx=None): return None
    async def mkdir(self, uri, exist_ok=False, ctx=None): return None
    async def ls(self, uri, ctx=None): raise FileNotFoundError(uri)
    async def write_file_bytes(self, uri, data, ctx=None): return None

def build_zip(path, entries):
    buf = io.BytesIO()
    with zipfile.ZipFile(buf, "w") as zf:
        for name, content in entries.items():
            zf.writestr(name, content)
    with open(path, "wb") as f:
        f.write(buf.getvalue())

async def main():
    build_zip("evil.ovpack", {
        "pkg/_._meta.json": '{"uri":"viking://resources/pkg"}',
        "pkg/../../escape.txt": "pwned"
    })
    await import_ovpack(DummyFS(), "evil.ovpack", "viking://resources", ctx=None, vectorize=F

asyncio.run(main())
```

Error Logs

```
Error Logs
Possible observed errors (environment-dependent):

ValueError related to unsafe path handling (if validation exists)

runtime/import errors unrelated to bug reproduction in constrained environments (e.g. missing
```

OpenViking Version

OpenViking CLI v0.2.0

Python Version

3.11

Operating System

Linux

Model Backend

OpenAI

Additional Context

This is a security-sensitive input validation issue. .ovpack is a ZIP container, so member path validation should be strict and centralized to prevent ZIP Slip/path traversal patterns across all supported path formats (Unix and Windows style).

👍 1

13ernkastel added **bug** last month

github-project-automation added this to OpenViking project last month

github-project-automation moved this to Backlog in OpenViking project last month

qin-ctx last month Collaborator

Thanks for the detailed report and the clear reproduction script!

You are right — `import_ovpack` currently lacks validation on ZIP member paths before any `mkdir` / `write_file_bytes` operations. The attack vectors you identified (path traversal via `../`, absolute paths, drive prefixes, root mismatch) are all valid concerns.

A fix should add a centralized validation step inside the import loop that **rejects** (not sanitizes) unsafe entries before any I/O occurs:

- Reject `.\.` in any path segment
- Reject absolute paths (`/` -prefixed)
- Reject Windows drive prefixes (`C:` etc.) and backslashes
- Reject entries whose root directory does not match the expected archive root

Would you be interested in submitting a PR for this? We would be happy to review it. The relevant code is in `openviking/storage/local_fs.py`, and the existing tests live in `tests/client/test_import_export.py` — adding targeted test cases for each unsafe pattern would be great.

13ernkastel last month Contributor Author •••

Fixed and committed

`46b3e76`

13ernkastel closed this as completed last month

github-project-automation moved this from Backlog to Done in OpenViking project last month

LEEKIYOON-SEC mentioned this 28 days ago

[Argus] CVE-2026-28518: OpenViking .ovpack 가져오기 ZIP 경로 탐색 취약점 LEEKIYOON-SEC/Argus-AI-Threat-Intelligence#74

Metadata

Assignees

No one assigned

Labels

bug

Type

No type

Projects

OpenViking project

Status Done

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants