Path traversal via double-slash URI prefix in TemplateLookup

Moderate **zzzeek** published **GHSA-v92g-xgxw-vvmm** last week

Package

Mako (pip)

Affected versions

<= 1.3.10

Patched versions

1.3.11

Description

Summary

`TemplateLookup.get_template()` is vulnerable to path traversal when a URI starts with `//` (e.g., `//../../../secret.txt`). The root cause is an inconsistency between two slash-stripping implementations:

- `Template.__init__` strips **one** leading `/` using `if`/slice
- `TemplateLookup.get_template()` strips **all** leading `/` using `re.sub(r"^\/+", "")`

When a URI like `//../../../../etc/passwd` is passed:

1. `get_template()` strips all `/` → `../../../../etc/passwd` → file found via `posixpath.join(dir_, u)`
2. `Template.__init__` strips one `/` → `/../../../../etc/passwd` → `normpath` → `/etc/passwd`
3. `/etc/passwd`.startswith(`..`) → `False` → **check bypassed**

Impact

Arbitrary file read: any file readable by the process can be returned as rendered template content when an application passes untrusted input directly to `TemplateLookup.get_template()`.

Note: this is exploitable at the library API level. HTTP-based exploitation is mitigated by Python's `BaseHTTPRequestHandler` which normalizes double-slash prefixes since CPython gh-87389. Applications using other HTTP servers that do not normalize paths may be affected.

Fix

Changed `Template.__init__` to use `lstrip("/")` instead of stripping only a single leading slash, so both code paths handle leading slashes consistently.

Severity

Moderate

CVE ID

CVE-2026-41205

Weaknesses

▶ CWE-22

Credits

 **0xHunSec** Reporter