GOVERNMENT OF THE REPUBLIC OF VANUATU

PRIME MINISTER'S OFFICE

CERTVU
DEPARTMENT OF COMMUNICATIONS & DIGITAL TRANSFORMATION

PM B 9108 Port Vila, Vanuatu

Tel: (678) 33380

GOUVERNEMENT DE LA REPUBLIQUE DU VANUATU

BUREAU DU PREMIER MINISTRE

CERTVU

DEPARTMENT DE COMMUNICATION ET DE TRANSFORMATION NUMERIQUE

SPP 9108 Port Vila, Vanuatu
Tel: (678) 33380

3 September 2026

Advisory 238: Starlette "BadHost" Host-Header Authentication Bypass (CVE-2026-48710).

Release Date: 2nd September 2026
Impact: MEDIUM
TLP: CLEAR

The Department of Communications and Digital Transformation (DCDT) through CERT Vanuatu (CERTVU), provides the following advisory.

This alert is relevant to Organizations, System/Network administrators, and DevOps/software engineering teams that build or operate Python web services on Starlette or on frameworks built on top of it, including FastAPI. This alert is intended to be understood by technical users and systems administrators.

What is it?

CVE-2026-48710 ("BadHost") is a Host-header authentication-bypass vulnerability in Starlette, the ASGI framework underpinning FastAPI and used directly by vLLM, LiteLLM (CVE-2026-59822/Advisory 237), Hugging Face text-generation-inference, and MCP servers. Starlette builds request.url by concatenating the raw, unvalidated Host header, so an attacker can inject characters such as "/", "?", or "#" into that header to make request.url.path — the value most security middleware inspects — disagree with the path the server actually routed. In a published proof of concept, a request to a protected path such as /admin is correctly blocked (403) with a clean Host header, but the identical request is served (200) once a single extra character is appended to the Host header, because the routing layer and the middleware's view of the URL no longer match.

CERTVU notes the practical impact can be materially worse downstream: because this flaw lets an attacker defeat path-based access controls, it can expose administrative, metrics, or shutdown

endpoints, and - on services that expose tool-execution or code-evaluation functionality behind such a check, as some AI-serving and MCP-based applications do - a successful bypass could be chained into a more serious compromise than the base score alone suggests. CERTVU found no confirmed reports of this CVE being exploited in the wild.

What are the systems affected?

The following version(s) are affected:

- Starlette 0.8.3 through 1.0.0 (inclusive) – (Affected)

Starlette 1.0.1 and later – (Not affected, patched)

Starlette is a dependency pulled into an application at build time rather than a standalone service that is centrally patched, so any framework or application that pins an affected Starlette version - directly, or indirectly through FastAPI or an AI-serving/MCP framework that depends on it - remains vulnerable until that dependency is explicitly upgraded and redeployed..

What does this mean?

Typical attack flow:

1. **Send a request with a malformed Host header**
    - An attacker sends an HTTP request to a Starlette-based application, setting the Host header to a value containing an extra character such as “/”, “?”, or “#” (for example, Host: foo?) rather than a normal hostname.
2. **Exploit the path mismatch this creates**
    - Starlette reconstructs request.url by concatenating the unvalidated Host header into it, shifting the parsed path/query/fragment boundaries so that request.url.path no longer matches the path the ASGI server actually routed to. Security middleware that checks request.url.path — rather than the raw ASGI scope path — sees a sanitized, allow-listed-looking path and lets the request through to the real, protected endpoint.

Attack vectors:

- A network-based, unauthenticated request against any internet- or network-reachable Starlette or FastAPI application whose security middleware relies on request.url.path for path-based access control.
- No user interaction, no privileges, and no special access conditions are required (CVSS AV:N/AC:L/PR:N/UI:N) — a single crafted HTTP request with a modified Host header is sufficient to trigger the bypass.

Successful exploitation may allow attackers to:

- Bypass path-based authentication and access-control middleware to reach administrative, metrics, debug, or shutdown endpoints that were intended to be blocked.
- On applications that expose tool-execution, code-evaluation, or other high-privilege functionality behind a path-based check — including some AI-serving and Model Context

Protocol (MCP) applications built on Starlette - potentially chain the bypass into a more serious compromise of the underlying service.

Mitigation process

CERTVU recommends the following:

1. Apply the vendor patch without delay

- Upgrade Starlette to version 1.0.1 or later, where this vulnerability is fixed (GitHub Security Advisory GHSA-86qp-5c8j-p5mr); the fixed version validates the Host header against RFC 9112 §3.2 grammar before using it to construct request.url, and falls back to the ASGI scope's own server value for malformed input. Where Starlette is pulled in indirectly (e.g. via FastAPI, vLLM, LiteLLM, text-generation-inference, or an MCP server framework), upgrade that framework to a release that itself depends on a patched Starlette version, and redeploy — this fix will not apply itself.

2. Apply the vendor's interim workaround if immediate upgrade isn't possible
3. Audit the estate for affected applications
4. Harden security middleware in the meantime
5. Review access logs for suspicious Host headers

Report suspected compromise to CERTVU at incident@cert.gov.vu or on telephone (678) 33380.

Reference

1. https://www.cve.org/CVERecord?id=CVE-2026-48710
2. https://github.com/Kludex/starlette/security/advisories/GHSA-86qp-5c8j-p5mr