

PrefectHQ / fastmcp Public[Code](#) [Issues](#) 211 [Pull requests](#) 12 [Discussions](#) [Actions](#) [Projects](#)

# fix: URL-encode path params to prevent SSRF/path traversal (GHSA-vv7q-7jx5-f767) #3507

**Merged** [jlowin](#) merged 6 commits into [main](#) from [security/openapi-path-traversal-s...](#) on Mar 15

[Conversation](#) 9 [Commits](#) 6 [Checks](#) 7 [Files changed](#) 3



[jlowin](#) commented on [Mar 14](#)

Member

`RequestDirector._build_url()` substituted path parameter values directly into URL templates without URL-encoding. Since `urllib.parse.urljoin()` interprets `../` sequences, an attacker who controls a path parameter value (e.g. `../../../../admin/delete-all?`) can traverse out of the intended API prefix and reach arbitrary backend endpoints — inheriting any auth credentials the provider is configured with.

The fix applies `urllib.parse.quote(str(param_value), safe='')` to every path parameter value before substitution. Using `safe=""` ensures all special characters — including `/`, `.`, `?`, and `#` — are percent-encoded, so `urljoin` treats the substituted value as a single opaque path segment rather than interpreting path navigation sequences.

```
# Before (vulnerable):
url_path = url_path.replace(placeholder, str(param_value))

# After (safe):
safe_value = quote(str(param_value), safe='')
url_path = url_path.replace(placeholder, safe_value)
```



Query and header parameters are not affected — they are passed to `httpx.Request` via `params=` and `headers=`, which handle encoding internally.

Ref: [GHSA-vv7q-7jx5-f767](#)



[fix: URL-encode path params in OpenAPI provider to prevent SSRF/path ...](#)



[3583c26](#)

 **jlowin** added the **bug** label on Mar 14

 **marvin-context-protocol** (Bot) added **high-priority** **http** labels on Mar 14

 **chatgpt-codex-connector** (Bot) reviewed on Mar 14

[View reviewed changes](#)

 **chatgpt-codex-connector** (Bot) left a comment

### Codex Review

Here are some automated review suggestions for this pull request.

**Reviewed commit:** `3583c26b55`

▶  [About Codex in GitHub](#)

src/fastmcp/utilities/openapi/director.py **Outdated**

211	212		placeholder = f"{{{param_name}}}"
212	213		if placeholder in url_path:
213	-		url_path = url_path.replace(placeholder, str(param_value))
	214	+	safe_value = quote(str(param_value), safe="")

 **chatgpt-codex-connector** (Bot) on Mar 14

#### **P1** Escape dot-segment path params before URL join

`quote(str(param_value), safe="")` does not escape `.` characters, so attacker input like `".."` is still substituted as a real dot segment. When `urljoin()` combines the URL, it normalizes `/../` and can climb out of the intended route prefix (for example, `/api/v1/users/{id}/profile` becomes `/api/v1/profile`), so path traversal remains possible for this input class.

Useful? React with  / .

 **jlowin** added 2 commits [last month](#)

[Exempt too-long from core-category requirement in triage](#) ✓ [6b829a7](#)

[fix: also encode dots in path params to prevent bare .. traversal](#) ✗ [8b18db9](#)

**chatgpt-codex-connector** Bot reviewed [on Mar 14](#)

[View reviewed changes](#)

**chatgpt-codex-connector** Bot left a comment

### Codex Review

Here are some automated review suggestions for this pull request.

**Reviewed commit:** [8b18db9fff](#)

▶ [About Codex in GitHub](#)

▼ [tests/utilities/openapi/test\\_director.py](#) Outdated

```
526 +         path_route, {"id": "../admin"}, "https://api.example.com"
527 +     )
528 +     url = str(request.url)
529 +     assert "%2Fadmin" in url or "%2fadmin" in url
```

**chatgpt-codex-connector** Bot [on Mar 14](#)

#### P1 Update dot-dot test to match encoded dot segments

After `_build_url` started encoding `.` as `%2E`, this assertion no longer matches the produced URL (`%2E%2E%2Fadmin`), so `uv run pytest tests/utilities/openapi/test_director.py` now fails in `test_dot_dot_slash_is_encoded`. This leaves the new security regression suite red and blocks the required test workflow for the commit.

Useful? React with / .

**marvin-context-protocol** Bot commented [on Mar 14](#) • edited ▼

Contributor

## Test Failure Analysis

Edited to reflect the latest CI run ([#23096236334](#)).

**Summary:** A single test in `tests/cli/test_run.py` is timing out on **Windows / Python 3.10** only. All other matrix jobs (Ubuntu Python 3.10, Ubuntu Python 3.13, lowest-direct deps, integration) are green.

**Root Cause:** `TestMCPConfig::test_run_mcp_config` spawns a subprocess via `StdioMCPServer(command="python", args=[script_path])`, then connects to it using FastMCP's stdio transport to call `list_tools()`. The test is hitting the 5-second unit-test timeout while blocked in `asyncio.selector_events` waiting on the subprocess's stdout. This failure is **unrelated to the PR's changes** (URL-encoding path params in `director.py`) — nothing in the diff touches subprocess or stdio transport handling.

This looks like a pre-existing Windows subprocess/stdio flakiness. The Python subprocess appears to connect (the debug log shows `Stdio transport connected`) but never sends back a `list_tools` response within the timeout window.

**Suggested Solution:** This is not caused by this PR. Options:

- Re-run the failed job — if it's a transient flake, it should pass.
- If it's consistently failing on Windows for this test, the test may need a longer timeout or a `@pytest.mark.skip` on Windows (`sys.platform == "win32"`), but that's a separate issue from this PR.

► Detailed Analysis

► Related Files

  [fix: only encode .. \(not all dots\) to preserve valid dotted values](#) ✖ [20d0a9e](#)

 **chatgpt-codex-connector** Bot reviewed [on Mar 14](#)

[View reviewed changes](#)

 **chatgpt-codex-connector** Bot left a comment

### Codex Review

Here are some automated review suggestions for this pull request.

**Reviewed commit:** `20d0a9e31f`

►  About Codex in GitHub

```

  ✓ src/fastmcp/utilities/openapi/director.py Outdated
  211 |   212 |   placeholder = f"{{{param_name}}}"

```

212	213		if placeholder in url_path:
213		-	url_path = url_path.replace(placeholder, str(param_value))
	214	+	safe_value = quote(str(param_value), safe="").replace("..",

**chatgpt-codex-connector** Bot on Mar 14**P1 Encode single-dot path segments before urljoin**

`quote(str(param_value), safe="")` still leaves `.` unescaped, and this patch only rewrites `..`, so a path param of `"."` is substituted literally. In `urljoin`, a `"/./"` segment is normalized away (e.g. `/api/v1/users/{id}/profile` becomes `/api/v1/users/profile`), which means caller-controlled params can still rewrite the target route and bypass path-segment boundaries the fix is meant to enforce.

Useful? React with 👍 / 👎.



fix: encode all dots in path params to prevent single-dot normalization

✓ 66693f0

**chatgpt-codex-connector** Bot reviewed on Mar 15

View reviewed changes

**chatgpt-codex-connector** Bot left a comment**Codex Review**

Here are some automated review suggestions for this pull request.

**Reviewed commit:** 66693f068e▶ [About Codex in GitHub](#)

▼
tests/utilities/openapi/test\_director.py
Outdated

585	+	"https://api.example.com",
586	+	)
587	+	url = str(request.url)
588	+	assert "/admin" not in unquote(unquote(url)).split("/api/v1/users/")

**chatgpt-codex-connector** Bot on Mar 15**P2 Assert traversal check on decoded path, not URL prefix**

The `test_double_encoded_traversal` assertion currently inspects `split("/api/v1/users/")` `[0]`, which is only the part *before* that marker (typically just the scheme/host). That means a decoded URL like `/api/v1/users/../../../../admin/profile` would still satisfy this check, so the test does not actually validate the traversal condition it is meant to cover. Checking the decoded path portion after `/api/v1/users/` (or the full decoded path) would make this regression test effective.

Useful? React with 👍 / 👎.

[fix: check decoded path stays within prefix in double-encoding test](#) ✓ [4b4f119](#)

**jlowin** merged commit `40bdfb6` into `main` on Mar 15 View details  
7 checks passed

**jlowin** deleted the `security/openapi-path-traversal-ssrf` branch [last month](#)

**jlowin** added the `security` label on Mar 15

**jlowin** added a commit that referenced this pull request [3 weeks ago](#)

[Publish docs for v3.2.0 \(#3713\)](#) ✓ [3a9717e](#)

Sign up for free
 to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

**Reviewers**

**chatgpt-codex-connector[bot]**

**Assignees**

No one assigned

**Labels**

bug high-priority http security

**Projects**

None yet

### Milestone

No milestone

---

### Development

Successfully merging this pull request may close these issues.

None yet

---

### 1 participant

