

f / prompts.chat Public

&lt;&gt; Code Issues 10 Pull requests 35 Actions Projects Models S

# fix(security): Security Report Wiro `inputImageUrl` Blind SSRF #1102

Merged f merged 1 commit into `f:main` from `mdisec:fix-ssrf-wiro` 3 weeks ago

Conversation 1 Commits 1 Checks 8 Files changed 1



mdisec commented 3 weeks ago • edited by coderabbitai (bot)

Contributor

## Security Report — Wiro `inputImageUrl` Blind SSRF

**Date:** 2026-03-25**Severity:** Critical**Status:** Fixed**Affected path:** `POST /api/media-generate` → `src/lib/plugins/media-generators/wiro.ts`

### Summary

The Wiro plugin performed `fetch(request.inputImageUrl)` server-side with a user-controlled URL, then uploaded the fetched bytes to Wiro as multipart `inputImage`. This is a **blind SSRF** — the response body is not returned to the attacker, but the server-side fetch still enables internal network probing, timing attacks, and out-of-band exfiltration via Wiro.

**Reachability:** The stock frontend does not wire up `inputImageUrl`, but any authenticated user can exploit it via direct `POST /api/media-generate` with crafted JSON.

### Vulnerable Code

```
src/lib/plugins/media-generators/wiro.ts :
```

```

if (request.imageUrl) {
  const imageResponse = await fetch(request.imageUrl); // SSRF sink
  if (imageResponse.ok) {
    const imageBlob = await imageResponse.blob();
    formData.append("inputImage", imageBlob, "input.jpg");
  }
}

```



## Fix: Use Wiro's Native `inputImageUrl` Parameter

The server never needed to fetch the image. Per [Wiro Model Parameters docs](#), any `fileinput` parameter supports a `{id}Url` suffix to pass a URL string instead of a file blob:

**Tip:** For `fileinput` and `multifileinput` parameters, use the `{id}Url` suffix to send URLs (e.g., `inputImageUrl`).

By passing the URL directly to Wiro via `formData.append("inputImageUrl", ...)`, the fetch responsibility shifts to Wiro's infrastructure, completely eliminating the server-side SSRF.

### Why this is the ideal fix

1. **Zero server-side fetch** — no outbound request to user-controlled URLs
2. **No SSRF hardening needed** — no private IP blocking, DNS rebinding defenses, or redirect handling required
3. **Feature preserved** — reference-image generation continues to work
4. **Simpler code** — one `formData.append` replaces fetch + blob conversion

### Why other approaches are inferior

Approach	Problem
Remove <code>inputImageUrl</code> entirely	Kills a useful feature the API supports natively
Add <code>isPrivateUrl()</code> guard	Bypassable via DNS rebinding, IPv4-mapped IPv6! Since its only used for webhook for the admin notification, I dont wanna waste my time to fix this. It's a lot of work to do

## Summary by CodeRabbit

- Refactor

- Streamlined image processing in Wiro media generator by optimizing how image URLs are handled, reducing server-side operations for improved efficiency.



[fix\(security\): wiro ssrf](#)

✓ [66de14d](#)

**coderabbitai** (bot) commented [3 weeks ago](#) • edited ▾

- ▶ Walkthrough
- ▶ Pre-merge checks | 3
- ▶ Finishing Touches

Thanks for using [CodeRabbit!](#) It's free for OSS, and your support helps us grow. If you like it, consider giving us a shout-out.

- ▶ Share

Comment [@coderabbitai help](#) to get the list of available commands and usage tips.



**coderabbitai** (bot) reviewed [3 weeks ago](#)

[View reviewed changes](#)



**coderabbitai** (bot) left a comment

- ▶ Nitpick comments (1)
  - ▶ Prompt for all review comments with AI agents
- 
- ▶ Review info



**f** merged commit **289c52c** into [f:main](#) [3 weeks ago](#)

9 checks passed

[View details](#)

Sign up for free

to join this conversation on GitHub. Already have an account? [Sign in to](#)

[comment](#)

### Reviewers



coderabbitai[bot]



### Assignees

No one assigned

### Labels

None yet

### Projects

None yet

### Milestone

No milestone

### Development

Successfully merging this pull request may close these issues.

None yet

### 3 participants

