New issue

Server-Side Request Forgery Vulnerability in directus-api-extended #36

Open

BruceJqs opened 2 weeks ago Owner

Server-Side Request Forgery Vulnerability in directus-mcp

1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: Apr 18, 2026

2) Reporter Contact

- Reporter name: `BruceJin`
- Reporter email: `brucejin@zju.edu.cn`
- Permission to share contact with vendor: `Yes`

3) Vendor / Product Identification

- Vendor: pixelsock
- Product: directus-mcp
- Repository: https://github.com/pixelsock/directus-mcp
- Affected component(s):
- `index.ts`

4) Vulnerability Type

- CWE: CWE-918 (Server-Side Request Forgery)
- Short title: SSRF in MCP `uploadFile` URL handling

5) Affected Versions

- Confirmed affected: 1.0.0, commit `77758625355d105364eeaeac9afec2f743fe369b`
- Suspected affected range: revisions containing the same request-to-sink flows listed below
- Fixed version: Not available at time of report

6) Vulnerability Description

A server-side request forgery (SSRF) vulnerability (CWE-918) has been identified in directus-api-extended (directus-mcp) version 1.0.0, specifically within the uploadFile MCP tool. The tool accepts a user-supplied fileUrl argument and passes it directly to axios.get without URL allowlisting, private-address blocking, or redirect validation. An attacker with network access to the MCP interface can cause the server to make arbitrary outbound HTTP requests to loopback, internal, or attacker-controlled destinations, potentially leading to information disclosure or further exploitation. No fixed version is available at the time of reporting.

7) Technical Root Cause

1. `js/request-forgery-from-request`
   - Source: `index.ts:377` (`uploadFile` tool)
   - Source argument: `index.ts:390` (`fileUrl`)
   - Tool argument extraction: `index.ts:802`
   - Sink: `index.ts:812`
   - Sink code: `const fileResponse = await axios.get(fileUrl, { responseType: 'arraybuffer' });`
2. Related untrusted target selection
   - Source argument: `index.ts:382` (`url`)
   - Per-request API URL selection: `index.ts:522`
   - Authentication request sink: `index.ts:73`
   - Sink code: `const response = await axios.post(`${url}/auth/login`, { email, password });`

8) Attack Prerequisites

- Attacker can invoke the MCP `uploadFile` tool.
- The MCP server has network egress to attacker-chosen, loopback, or internal destinations.

- The deployment does not otherwise block private, loopback, link-local, metadata, or internal host requests at the network layer.
- For later Directus upload completion, valid Directus credentials or token may be required, but the SSRF request to `fileUrl` occurs before the upload request.

9) Proof of Concept / Reproduction Guidance

This proof of concept provides a concise, CVE-style reproduction example for the reported issue.

1. Reproduction request

```
{"jsonrpc":"2.0","id":1,"method":"tools/call","params":{"name":"uploadFile","arguments"
```

2. Validation

- Start a controlled HTTP listener, for example on `127.0.0.1:18080`, serving a small file named `probe.txt`.
- Start the affected MCP server and invoke the `uploadFile` tool through `mcp-inspector` with `fileUrl` pointing to the controlled listener.
- Confirm that the controlled listener receives a request for `/probe.txt` from the MCP server process.
- The tool may later fail when it attempts to upload to a dummy Directus URL; the SSRF evidence is the server-side fetch of `fileUrl` before that later upload step.
- The reproduction has been manually confirmed with `mcp-inspector`.

10) Security Impact

- Confidentiality: High (the MCP server can be induced to request internal or loopback resources and may return or process fetched content depending on tool behavior and downstream errors).
- Integrity: Low (the demonstrated `fileUrl` path performs a GET request; state-changing impact depends on reachable internal endpoints and HTTP semantics).
- Availability: Low (an attacker may cause outbound requests to slow or unavailable services, depending on timeout behavior and deployment limits).
- Scope: Changed.

11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:L`
- Suggested base score: 7.9 (High)
- Adjust `AV` to `N` if the affected MCP tool is exposed through a remotely reachable MCP bridge or service.

12) Workarounds / Mitigations

- Do not expose the MCP server to untrusted clients until a fix is available.
- Restrict access to `uploadFile` and other URL-taking tools to trusted users only.
- Disable `fileUrl` uploads and require caller-provided `fileData` until outbound URL validation is implemented.
- Block loopback, link-local, RFC1918, metadata-service, and other internal destinations at the network layer.
- Add egress allowlists for expected Directus and file-hosting domains.

13) Recommended Fix

- Validate `fileUrl` before making any outbound request.
- Allow only expected schemes, hosts, and ports for file downloads.
- Resolve DNS and reject loopback, link-local, RFC1918, multicast, and metadata-service addresses both before and after redirects.
- Disable redirects or revalidate every redirect target.
- Remove or tightly restrict per-request `url` overrides for Directus API calls.
- Add regression tests proving that `fileUrl` and `url` cannot target `127.0.0.1`, `localhost`, private IP ranges, link-local addresses, or cloud metadata endpoints.
- Publish a maintainer security advisory once a patch is released.

14) References

- Repository: https://github.com/pixelsock/directus-mcp
- Reviewed source file: `index.ts`
- CWE-918: https://cwe.mitre.org/data/definitions/918.html

15) Credits

- Discoverer: `BruceJin`
- Discovery method: Static analysis (CodeQL), repository source-code audit, and manual reproduction with `mcp-inspector`

16) Additional Notes for Form Mapping

- Audit verdict: Manually reproduced: attacker-controlled MCP `fileUrl` reaches an outbound HTTP request sink.
- Dynamic exploit replay status: completed with `uploadFile` and a controlled `fileUrl`; a server-side request from the MCP process to the controlled listener was observed.
- Maintainer should validate release mapping before coordinated disclosure.

BruceJqs 2 weeks ago Owner Author •••

Tools

List Tools

Clear

login	Login to Directus and get an access token
getActivity	Get activity logs from Directus
getFields	Get fields for a collection
getRelations	Get relations for a collection
getFiles	Get files from Directus
uploadFile	Upload a file to Directus
getUsers	Get users from Directus
getCurrentUser	Get the current user info

uploadFile

Upload a file to Directus

url
http://127.0.0.1:59999

token
aaa

fileUrl
http://127.0.0.1:18080/probe.txt

fileData
Base64 encoded file data (either fileUrl or fileData must be provided)

fileName
probe.txt

mimeType
text/plain

storage
Storage location (optional)

title
File title (optional)

Run Tool

Tool Result: Success

```
"Error: connect ECONNREFUSED 127.0.0.1:59999"
```

```
❯ python3 -m http.server 18080
Serving HTTP on :: port 18080 (http://[::]:18080/) ...
::ffff:127.0.0.1 - - [18/Apr/2026 13:48:04] "GET /probe.txt HTTP/1.1" 200 -
```

By exploiting the vulnerability, a server-side request has been successfully forged.

BruceJqs mentioned this 2 weeks ago

Server-Side Request Forgery Vulnerability in directus-mcp pixelsock/directus-mcp#13

Copilot mentioned this 2 weeks ago

fix: patch SSRF vulnerability and upgrade vulnerable dependencies pixelsock/directus-mcp#14

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants