Private File IDOR via raw/direct endpoints

GHSA : https://github.com/FlintSH/Flare/security/advisories/GHSA-gwqr-xf5c-5569

CVE : CVE-2026-30231

Summary

The raw and direct file routes only block unauthenticated users from accessing private files. Any authenticated, non-owner user who knows the file URL can retrieve the content, which is inconsistent with stricter checks used by other endpoints

ezgif-80eb9d3b19d68298.gif

Evidence (Code References)

- raw route predicate: raw/route.ts:L92-L97
- direct route predicate: direct/route.ts:L35-L40
- download route correct check: download/route.ts:L36-L45
- thumbnail route correct check: thumbnail/route.ts:L32-L45

Impact

- Confidential private files can be accessed by any authenticated user if the URL is known.
- Violates access control consistency across file endpoints.

Expected vs Actual

- Expected: Private files should be accessible only to the owner or admin.
- Actual: Private files are accessible to any authenticated user on raw/direct routes.

Minimal Logic Reproduction (non? network)

This demonstrates the misclassification using the same predicates.

```
type Session = { user?: { id: string; role?: 'ADMIN' | 'USER' } } | null

function rawIsPrivate(visibility: 'PUBLIC' | 'PRIVATE', session: Session) {
  return visibility === 'PRIVATE' && !session?.user
}

function directIsPrivate(visibility: 'PUBLIC' | 'PRIVATE', session: Session) {
  return visibility === 'PRIVATE' && !session?.user
}

// Setup: private file owned by A; authenticated B session
const fileVisibility = 'PRIVATE' as const
const sessionB: Session = { user: { id: 'user-B', role: 'USER' } }

// Current behavior in raw/direct:
rawIsPrivate(fileVisibility, sessionB)     // false → grants access
directIsPrivate(fileVisibility, sessionB)  // false → grants access

// Expected: deny unless owner or admin
function expectedIsPrivate(
  visibility: 'PUBLIC' | 'PRIVATE',
  session: Session,
  ownerId: string
) {
  const isOwner = session?.user?.id === ownerId
  const isAdmin = session?.user?.role === 'ADMIN'
  return visibility === 'PRIVATE' && !isOwner && !isAdmin
}
```

Affected Components

- raw file endpoint: raw/route.ts
- direct file endpoint: direct/route.ts

Remediation

Align raw/direct with download/thumbnail:

```
const isOwner = session?.user?.id === file.userId
const isAdmin = session?.user?.role === 'ADMIN'
const isPrivate = file.visibility === 'PRIVATE' && !isOwner && !isAdmin

if (isPrivate) {
  return new Response(null, { status: 404 }) // or 403
}
```

Consider centralizing access checks in a shared utility to ensure consistency across endpoints.

Verification Checklist

- Private file as Owner A:
  - Unauthenticated user → denied (404/401)
  - Authenticated non-owner User B → denied
  - Admin → allowed
  - Owner → allowed
- Behavior matches download/thumbnail routes.
- Automated tests added for owner/admin/non-owner/unauthenticated across raw/direct.

Revision #3
Created 2026-03-02 14:22:11 UTC by Aryma
Updated 2026-03-06 01:22:36 UTC by Aryma