Broken Access Control on Asset Download

Moderate **enchant97** published **GHSA-p5w6-75f9-cc2p** 5 days ago

Package

note-mark

Affected versions

v0.19.1

Patched versions

v0.19.2

Description

Summary

A broken access control vulnerability allows unauthenticated users to retrieve note assets directly from the asset download endpoint when they know both the note UUID and asset UUID. This exposes the full contents of private note assets without authentication, even when the associated book is not public.

Details

The issue is caused by the asset download route being registered without authentication middleware.

Relevant route registration:

- `handlers/assets.go`, line 40

```
huma.Get(api, "/api/notes/{noteID}/assets/{assetID}", h.GetNoteAssetContentByID)
```

By contrast, other asset operations correctly apply authentication middleware. For example:

```
huma.Delete(api, "/api/notes/{noteID}/assets/{assetID}", h.DeleteNoteAsset,
    huma.WithMiddleware(h.authMiddleware.AuthRequiredMiddleware))
```

The backend service for asset retrieval also does not enforce ownership or visibility checks. According to the provided code references, the lookup only queries the asset table by asset ID and note ID:

```
SELECT * FROM note_assets WHERE id = ? AND note_id = ?
```

Because the retrieval path does not join against the related `notes` or `books` records, it does not verify:

- whether the requester owns the parent book
- whether the parent book is public or private
- whether the related note has been deleted

As a result, possession of a valid `noteID` and `assetID` is sufficient to retrieve the asset binary, regardless of whether the note belongs to a private book.

The exploitability is constrained by identifier knowledge. Both `noteID` and `assetID` are UUIDv4 values, so blind guessing is impractical. However, the endpoint remains vulnerable whenever those identifiers are disclosed through another channel, such as leaked links, browser history, proxy logs, shared URLs, or other application behaviors that expose internal asset references.

PoC

The issue can be reproduced by creating a private note with an attached asset, then requesting the asset download endpoint without authentication using the valid `noteID` and `assetID`. The server returns the asset content even though the associated note is private.

Impact

- **Type:** Broken access control / unauthenticated information disclosure
- **Who is impacted:** Any deployment exposing the affected asset download endpoint
- **Security impact:** Full binary contents of private note assets can be disclosed to unauthenticated users who know the required identifiers
- **Attack preconditions:** The attacker must know both the target `noteID` and `assetID`; no authentication is required
- **Attack complexity:** High, because successful exploitation depends on prior disclosure of both UUIDs rather than feasible online guessing

Severity

Moderate 5.9 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	High

Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	None
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N

CVE ID

CVE-2026-40265

Weaknesses

▶ CWE-862

Credits

QiaoNPC	Reporter
Across-Verticals-Malaysia	Reporter
enchant97	Remediation developer