Unauthenticated Access to Role-Restricted documents via neutralized .htaccess

High **Fasse** published **GHSA-7fh7-8xqm-3g88** last week

Package

admidio/admidio:latest (Docker image)

Affected versions

>= 5.0.0

Patched versions

5.0.8

Description

Reported by: Juan Felipe Oz @JF0x0r

> LinkedIn

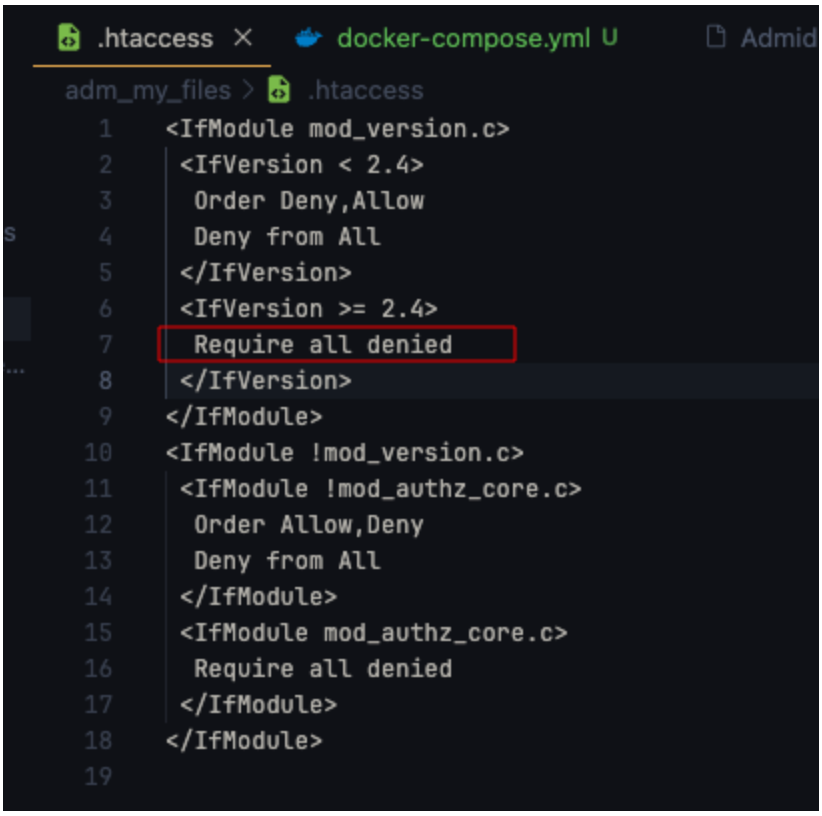
Summary

Admidio relies on `adm_my_files/.htaccess` to deny direct HTTP access to uploaded documents. The Docker image ships with `AllowOverride None` in the Apache configuration, which causes Apache to silently ignore all `.htaccess` files. As a result, any file uploaded to the documents module regardless of the *role-based* permissions configured in the UI, is directly accessible over HTTP without authentication by anyone who knows the file path. The file path is disclosed in the upload response JSON.

Root Cause

File 1: Intended protection (ignored):
`adm_my_files/.htaccess`

```
Require all denied
```

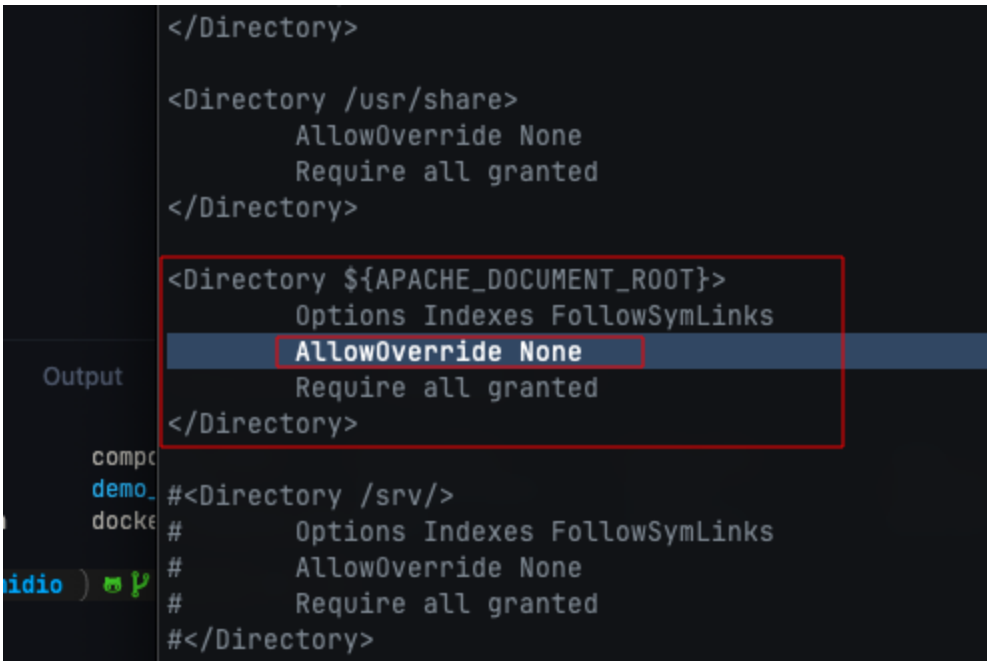


File 2: Apache config that neutralizes it:

- Command in order to search in Docker container: `docker exec admidio-sec-app cat /etc/apache2/apache2.conf`

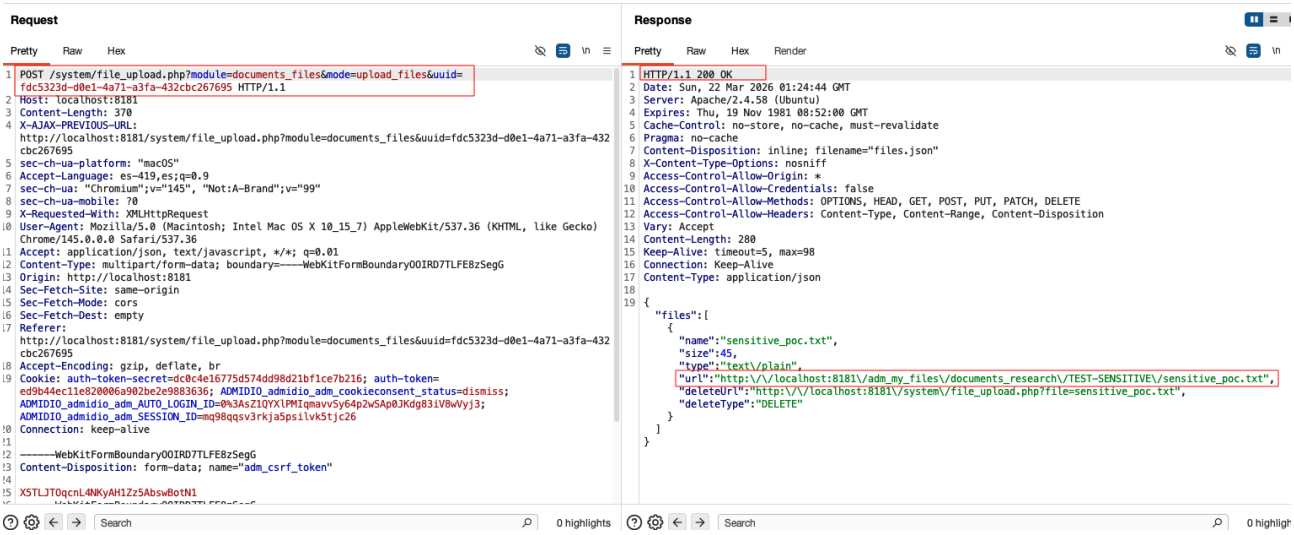
`/etc/apache2/apache2.conf` (Docker image)

```
<Directory ${APACHE_DOCUMENT_ROOT}>
    AllowOverride None
</Directory>
```



`AllowOverride None` instructs Apache to skip `.htaccess` processing entirely, the deny rule never executes. The upload directory is inside the web root at `/opt/app-root/src/adm_my_files/` and returns **HTTP 200** for direct requests.

File 3: Upload response leaks the direct URL: `system/file_upload.php`, upload response JSON:

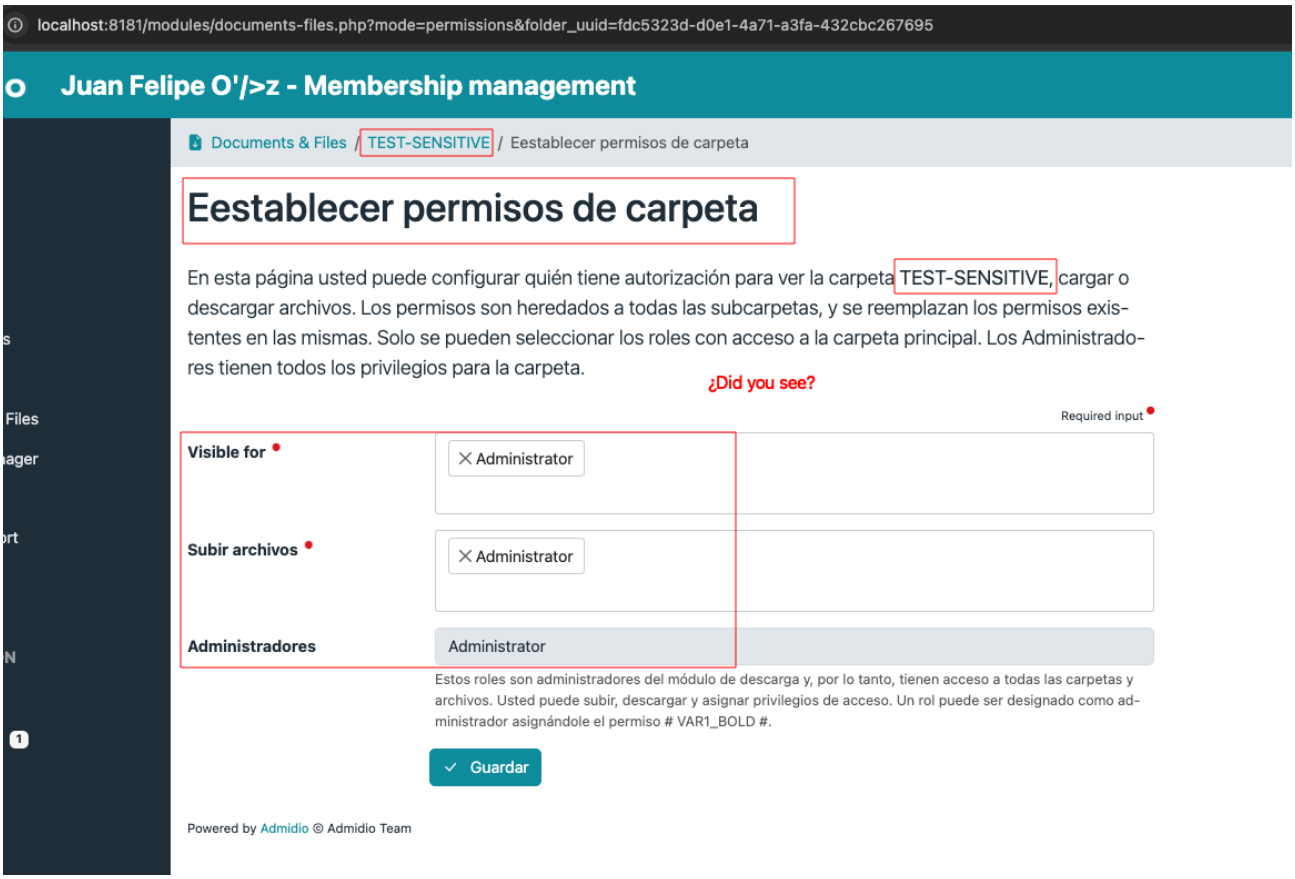



```
{
  "files": [{
    "name": "sensitive_poc.txt",
    "url": "http://TARGET/adm_my_files/documents_research/TEST-SENSITIVE/sensitive_poc.t
  }]
}
```

Verified PoC

Step 1: Admin creates a restricted folder (visible only to Administrator role):

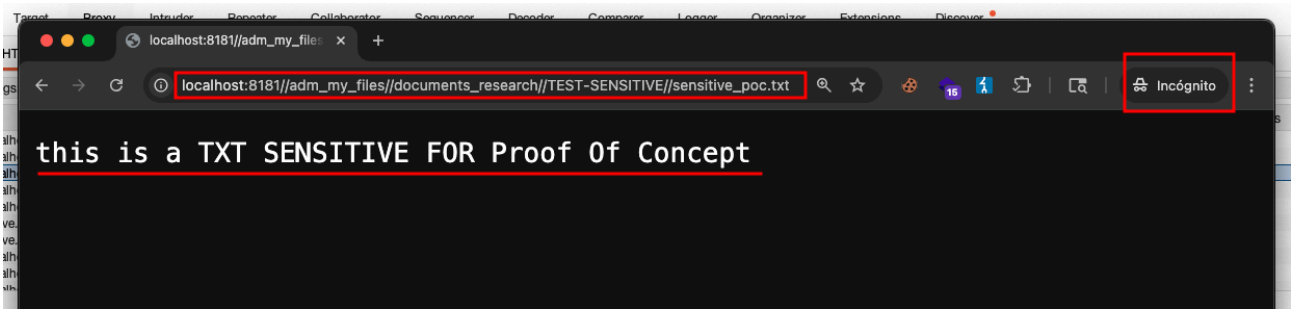
> `modules/documents-files.php` → permissions set to role `Administrator` only.



Step 2: Admin uploads a file to the restricted folder.

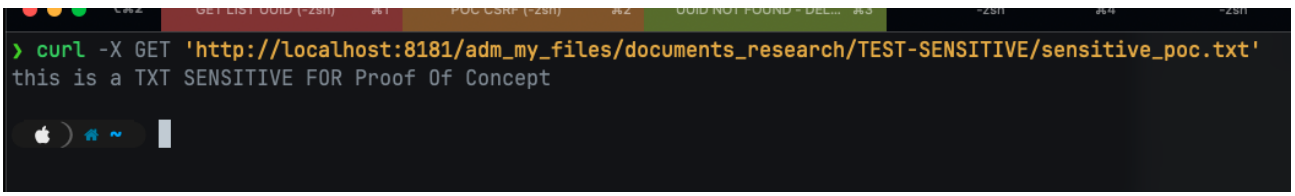
> Upload response returns:

```
http://TARGET/adm_my_files/documents_research/TEST-SENSITIVE/sensitive_poc.txt
```



Step 3: Unauthenticated request retrieves the file:

```
curl -X GET 'http://TARGET/adm_my_files/documents_research/TEST-SENSITIV
# Response: full file contents — no authentication required
```



Step 4: Confirm folder is role-restricted:

```
SELECT fil_name, fol_name, fol_public FROM adm_files JOIN adm_folders ON fil_fol_i o
ORDER BY fil_id DESC LIMIT 5; -- fol_public = 0, role restricted — yet file is pub
```

Impact

- Any document uploaded to **Admidio** including files restricted to specific roles is publicly accessible via direct HTTP request with no authentication required
- **Role-based** access control on the documents module is completely bypassed at the filesystem level
- Sensitive organizational documents (contracts, member data, financial records) are exposed to anyone who can guess or construct the file path
- The upload API response discloses the direct URL to the uploader, making path enumeration trivial

Recommended Fix

Option 1 (preferred): Enable AllowOverride in Apache config:

```
<Directory /opt/app-root/src/adm_my_files>
    AllowOverride All
</Directory>
```

Option 2: Move uploads outside the web root:
Store uploaded files in a directory outside `DOCUMENT_ROOT` and serve them exclusively through Admidio's download handler (`modules/documents-files.php?mode=download`), which enforces role checks before serving the file.

Option 3: Apache-level explicit deny (does not require .htaccess):

```
<Directory /opt/app-root/src/adm_my_files>
    Require all denied
</Directory>
```

> The most robust long-term fix is Option 2 — moving uploads outside the web root eliminates the dependency on Apache configuration correctness entirely.

Severity

High 7.5 / 10

CVSS v3 base metrics

Attack vector Network

Attack complexity	Low
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:L/PR:N/UI:N/S:U/C:H/I:N/A:N

CVE ID

CVE-2026-34381

Weaknesses

▶ CWE-284

Credits

JFOZ1010 Reporter