New issue

Stored XSS via Paragraph Content in MaxKB #28

Open

AnalogyC0de opened last month Owner

Stored XSS via Paragraph Content in MaxKB

Identification

- **Project:** MaxKB
- **Repository:** https://github.com/1Panel-dev/MaxKB
- **Affected Version/Commit:** <= v2.6.1

CVE Description

A Stored Cross-Site Scripting (XSS) vulnerability exists in MaxKB. Authenticated users with dataset management permissions can create paragraphs containing arbitrary HTML/Markdown content. The application stores this content without sanitization and subsequently renders it in the UI using the `md-editor-v3` `MdPreview` component, which does not sanitize HTML by default. This allows attackers to inject malicious JavaScript that executes when other users, including administrators, view the paragraph or execution details.

Affected Component

- **File(s):** `apps/knowledge/views/paragraph.py`, `apps/knowledge/serializers/paragraph.py`, `ui/src/views/paragraph/component/ParagraphCard.vue`
- **Function / Method:** `post()`, `get_paragraph_problem_model()`, `MdPreview` rendering
- **Entry Point:** `content` parameter in POST `/api/workspace/{workspace_id}/knowledge/{knowledge_id}/document/{document_id}/paragraph`

Reproduction Summary

1. An authenticated attacker with dataset management permissions sends a POST request to create a paragraph, including a malicious payload in the `content` field (e.g., `<img src=x onerror=alert('XSS')>`).
2. The backend stores the unsanitized `content` payload directly in the database.
3. A victim views the paragraph execution details in the UI.
4. The `ParagraphCard` component renders the unsanitized HTML via `MdPreview`, causing the browser to execute the injected JavaScript.

Technical Details

Entry Point Code (`apps/knowledge/views/paragraph.py:72-75`):

```
def post(self, request: Request, workspace_id: str, knowledge_id: str, document_id: str
    return result.success(ParagraphSerializers.Create(
        data={'workspace_id': workspace_id, 'knowledge_id': knowledge_id, 'document_id': docu
    ).save(request.data))
```

Missing Sanitization in Serializer (`apps/knowledge/serializers/paragraph.py:323-329`):

```
paragraph = Paragraph(
    id=uuid.uuid7(),
    document_id=document_id,
    content=instance.get("content"),  # ← User input, NO sanitization
    knowledge_id=knowledge_id,
    title=instance.get("title") if 'title' in instance else ''
)
```

Sink / Vulnerable Component (`ui/src/views/paragraph/component/ParagraphCard.vue:136-143`):

```
<MdPreview
  ref="editorRef"
  editorId="preview-only"
  :modelValue="data.content"
  class="maxkb-md"
  style="background: none"
  @clickPreview="handleClickCard(data)"
/>
```

Proof of Concept - Malicious Request:

```
POST /api/workspace/{workspace_id}/knowledge/{knowledge_id}/document/{document_id}/para
Host: target.com
Authorization: Bearer <attacker_token>
```

```
Content-Type: application/json

{
  "content": "<img src=x onerror=alert(document.cookie)>",
  "title": "XSS Payload"
}
```

Validation Notes

```
Verification Notes:
- `sanitize-html` package is installed in `package.json` but **NOT used** in the codeba
- `MdPreview` component does NOT pass any sanitize option to `md-editor-v3`
```

shaohuzhang1 3 weeks ago

The XSS rendered by MdPreview has been fixed in version 2.5.0 1Panel-dev/MaxKB#4578
Did you reproduce it on version 2.6.1

AnalogyC0de 3 weeks ago Owner Author

I started by conducting vulnerability mining on previous version 2.2.1 of the code, when writing the vulnerability report, though I did code review of the latest version, but I have missed this fix. I have checked the XSS is fixed in current version, sorry for the false positive.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants