fix: Inject application name through XSS #4919

Merged **shaohuzhang1** merged 1 commit into `v2` from `pr@v2@fix_application_xss`

3 weeks ago

shaohuzhang1 commented 3 weeks ago — Contributor

fix: Inject application name through XSS

`fix: Inject application name through XSS` ✓ `7aa8e65`

f2c-ci-robot bot commented 3 weeks ago

Adding the "do-not-merge/release-note-label-needed" label because no release-note block was detected, please follow our release note process to remove it.

▶ Details

f2c-ci-robot bot added the do-not-merge/release-note-label-needed label 3 weeks ago

f2c-ci-robot bot commented 3 weeks ago

[APPROVALNOTIFIER] This PR is **NOT APPROVED**

This pull-request has been approved by:

The full list of commands accepted by this bot can be found here.

▼ Details

Needs approval from an approver in each of these files:

- **OWNERS**

Approvers can indicate their approval by writing `/approve` in a comment
Approvers can cancel approval by writing `/approve cancel` in a comment

shaohuzhang1 commented 3 weeks ago

View reviewed changes

apps/common/middleware/chat_headers_middleware.py

```
     29 +                application_name = escape(application_access_token.get(
29   30                  if white_active:
30   31                      # 添加自定义的响应头
31   32                      response[
```

shaohuzhang1 3 weeks ago Contributor Author

The provided Python code looks generally correct, but there are a few suggested improvements:

1. **Escape Functionality**: It's good that you're using `escape` to handle any HTML content in the headers. This helps prevent XSS (Cross-Site Scripting) attacks.

2. **Check for Non-existence**: You've added an optional fallback with an empty string (`''`) before applying `escape`. While this prevents errors, it might be better to explicitly check for non-existent keys to avoid raising exceptions. You can use `get()` with a default value instead of checking individually after getting each key.

3. **Consistency and Style**: Ensure consistent formatting throughout the code. PEP 8 guidelines recommend spacing around operators and making sure lines don't exceed 79 characters.

Here's an updated version of the middleware with these suggestions:

```python
from common.cache_data.application_access_token_cache import get_applica a
from maxkb.const import CONFIG
import html

class ChatHeadersMiddleware(MiddlewareMixin):

    def process_response(self, request, response):
        application_access_token = get_application_access_token()

        if application_access_token is not None:
            white_active = application_access_token.get('white_active', False)
            white_list = application_access_token.get('white_list', [])
```

```
            # Use get() with default values to safely access keys
            application_icon = escape(application_access_token.get('applicatio
            application_name = escape(application_access_token.get('applicatio

            if white_active:
                # Add custom response header
                response['X-CustomHeader'] = f"White Active: {application_icon
```

By adding checks for key existence using `get()` and handling them gracefully, you enhance the robustness and maintainability of your code.

shaohuzhang1 merged commit **`026a2d6`** into `v2` 3 weeks ago

View details

3 checks passed

shaohuzhang1 deleted the `pr@v2@fix_application_xss` branch 3 weeks ago

This was referenced 3 weeks ago

Stored XSS in Maxkb via Middleware HTML Injection AnalogyC0de/public_exp#23

Open

Stored XSS in MaxKB AnalogyC0de/public_exp#24

Open

Reviewers

No reviews

Assignees

No one assigned

Labels

do-not-merge/release-note-label-needed

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

1 participant