WWBN / AVideo Public

Code Issues 13 Pull requests Actions Projects Wiki Security a

Commit 184f36b

Daniel Neto committed last week · 11 / 11

```
fix: Add request validation to prevent untrusted access in comment deletion

GHSA-8qm8-g55h-xmqr
```

master

1 parent 8e2a2f5 commit 184f36b

5 files changed +108 -2 lines changed

Top

Filter files…

- CreatePlugin/templates
  - add.json.php
  - delete.json.php
- objects
  - commentDelete.json.php
  - functionsSecurity.php
  - include_config.php

5 files changed +108 -2 lines changed

Search within code

CreatePlugin/templates/add.json.php

```
@@ -8,12 +8,14 @@
8  8   $obj->msg = "";
9  9
10 10  $plugin = AVideoPlugin::loadPluginIfEnabled('{pluginName}');
11    -
   11  +
```

```
12  12    if(!User::isAdmin()){
13  13        $obj->msg = "You cant do this";
14  14        die(json_encode($obj));
15  15    }
16  16
    17  + forbidIfIsUntrustedRequest('{pluginName}::{classname}::add');
    18  +
17  19    $o = new {classname}(@$_POST['id']);
18  20    {columnsAdd}
19  21
```

CreatePlugin/templates/delete.json.php

```
    @@ -13,8 +13,10 @@
13  13        die(json_encode($obj));
14  14    }
15  15
    16  + forbidIfIsUntrustedRequest('{pluginName}::{classname}::delete');
    17  +
16  18    $id = intval($_POST['id']);
17  19    $row = new {classname}($id);
18  20    $obj->error = !$row->delete();
19  21    die(json_encode($obj));
20      - ?>
        ⊖
    22  + ?>
```

objects/commentDelete.json.php

```
    @@ -5,6 +5,7 @@
5   5         require_once '../videos/configuration.php';
6   6     }
7   7     require_once $global['systemRootPath'] . 'objects/comment.php';
    8   + require_once $global['systemRootPath'] . 'objects/functions.php';
8   9
9   10    $obj = new stdClass();
10  11    $obj->error = true;
    @@ -21,6 +22,8 @@
21  22        die(_json_encode($obj));
22  23    }
```

```
23  24
    25  + forbidIfIsUntrustedRequest('commentDelete');
    26  +
24  27      $objC = new Comment("", 0, $obj->id);
25  28      $obj->videos_id = $objC->getVideos_id();
26  29      $obj->status = $objC->delete();
```

objects/functionsSecurity.php

```
    @@ -786,3 +786,89 @@ function enforceRateLimit(string $operation = '', int
    $maxAttempts = 20, int $ti
786 786         ObjectYPT::setCacheGlobal($key, $attempts + 1);
787 787     }
788 788
    789 + /**
    790 +  * Automatic CSRF guard invoked once by include_config.php for every POST
    791 +  * to a *.json.php endpoint.
    792 +  *
    793 +  * Bypass options (pick the one that fits your use-case):
    794 +  *
    795 +  *  1. $global['bypassSameDomainCheck'] = 1  — existing flag; disables all
    796 +  *     same-domain checks globally (encoder-to-encoder calls, etc.).
    797 +  *     Must be set BEFORE require configuration.php.
    798 +  *
    799 +  *  2. $global['skipAutoCSRFCheck'] = true  — disables only this auto-guard
    800 +  *     for the current request.  Must be set BEFORE require configuration.php.
    801 +  *
    802 +  *  3. $global['csrfBypassFiles'][] = 'myfile.json.php'  — persistent per-file
    803 +  *     opt-out; add it in videos/configuration.php (or a plugin config file).
    804 +  *
    805 +  * @param string $baseName  basename of the currently executing script
    806 +  */
    807 + function autoCSRFGuard($baseName)
    808 + {
    809 +     global $global;
    810 +
    811 +     // Respect existing full bypass (encoder callbacks, CLI, etc.)
    812 +     if (!empty($global['bypassSameDomainCheck']) || isCommandLineInterface()) {
    813 +         return;
    814 +     }
```

```
815 +
816 +     // Per-request opt-out — must be set before configuration.php loads
817 +     if (!empty($global['skipAutoCSRFCheck'])) {
818 +         return;
819 +     }
820 +
821 +     // Built-in bypass list.
822 +     // Groups:
823 +     //   auth/signup  — accept calls from mobile apps & external clients
824 +     //   public reads — use POST params for filtering, but mutate nothing
825 +     //   public write — like/view/subscribe actions open to all users
826 +     //   encoder      — authenticated via video-hash token, not session
827 +     static $builtinBypass = [
828 +         // Auth & account management
829 +         'login.json.php',
830 +         'userCreate.json.php',
831 +         'userRecoverPassSave.json.php',
832 +         // Public write actions
833 +         'sendEmail.json.php',
834 +         'subscribe.json.php',
835 +         'subscribeNotify.json.php',
836 +         'like.json.php',
837 +         'videoAddViewCount.json.php',
838 +         // Read-only endpoints that accept POST params
839 +         'categories.json.php',
840 +         'comments.json.php',
841 +         'users.json.php',
842 +         'videos.json.php',
843 +         'videosAndroid.json.php',
844 +         'plugins.json.php',
845 +         'playlistsPublic.json.php',
846 +         'playlistsVideos.json.php',
847 +         'playlistsFromUserVideos.json.php',
848 +         'mention.json.php',
849 +         'notifications.json.php',
850 +         'listFiles.json.php',
851 +         // Encoder upload callbacks (auth via video-hash, not session)
852 +         'aVideoEncoder.json.php',
853 +         'aVideoEncoderLog.json.php',
854 +         'aVideoEncoderNotifyIsDone.json.php',
```

```
+         'aVideoEncoderReceiveImage.json.php',
+     ];
+
+     if (in_array($baseName, $builtinBypass, true)) {
+         return;
+     }
+
+     // Allow operators to extend the bypass list in videos/configuration.php:
+     //   $global['csrfBypassFiles'] = ['myWebhook.json.php'];
+     if (
+         !empty($global['csrfBypassFiles']) &&
+         is_array($global['csrfBypassFiles']) &&
+         in_array($baseName, $global['csrfBypassFiles'], true)
+     ) {
+         return;
+     }
+
+     forbidIfIsUntrustedRequest("autoCSRF::{$baseName}");
+ }
+
```

objects/include_config.php

```
@@ -317,4 +317,17 @@ function includeConfigLog($line, $desc = '')
317 317      $metaDescription = " {$_GET['showOnly']}";
318 318  }
319 319
320 + // ── Auto CSRF guard ──────────────────────────────────────────────────
321 + // Blocks cross-origin POST to every *.json.php endpoint unless the file is
322 + // whitelisted.  See objects/functionsSecurity.php → autoCSRFGuard() for the
323 + // full bypass documentation.
324 + if (
325 +     isset($_SERVER['REQUEST_METHOD']) &&
326 +     $_SERVER['REQUEST_METHOD'] === 'POST' &&
327 +     substr($baseName, -9) === '.json.php'
328 + ) {
329 +     autoCSRFGuard($baseName);
330 + }
331 + //
    ─────────────────────────────────────────────────────────────────────
```

```
     332 +
320  333     includeConfigLog(__LINE__);
```

Comments 0

Please sign in to comment.