WWBN / **AVideo** Public

Code Issues 13 Pull requests Actions Projects Wiki Security a

Commit ee56151

Daniel Neto committed last week

```
fix: Add request validation to prevent untrusted access in category and plugin scripts

GHSA-ffw8-fwxp-h64w
```

master

1 parent f9492f5 commit ee56151

15 files changed +15 -3 lines changed

Top

Filter files…

objects

- categoryAddNew.json.php
- categoryDelete.json.php
- configurationClearCache.json.php
- configurationGenerateSiteMap.json.php
- notifySubscribers.json.php
- pluginAddDataObject.json.php
- pluginRunUpdateScript.json.php
- userDelete.json.php
- userGroupsAddNew.json.php
- userGroupsDelete.json.php
- videoDelete.json.php
- videoRefresh.json.php
- videoRotate.json.php
- videoStatus.json.php

videoSwap.json.php

15 files changed +15 -3 lines changed

Search within code

objects/categoryAddNew.json.php

```
@@ -19,6 +19,7 @@
19 19      $obj->msg = __("Permission denied");
20 20      die(json_encode($obj));
21 21  }
   22 + forbidIfIsUntrustedRequest('categoryAddNew');
22 23
23 24  $objCat = new Category(intval(@$_POST['id']));
24 25  $objCat->setName($_POST['name']);
```

objects/categoryDelete.json.php

```
@@ -10,6 +10,7 @@
10 10  if (!Category::canCreateCategory()) {
11 11      die('{"error":"' . __("Permission denied") . '"}');
12 12  }
   13 + forbidIfIsUntrustedRequest('categoryDelete');
13 14  require_once 'category.php';
14 15  $obj = new Category($_POST['id']);
15 16
```

objects/configurationClearCache.json.php

```
@@ -11,6 +11,7 @@
11 11  $obj->clearCache = false;
12 12  $obj->deleteALLCache = false;
13 13  $obj->deleteAllSessionCache = false;
   14 + forbidIfIsUntrustedRequest('configurationClearCache');
14 15  $_SESSION['user']['sessionCache']['getAllCategoriesClearCache'] = 1;
15 16
16 17  if (!Permissions::canClearCache() || !empty($_REQUEST['sessionOnly'])) {
```

objects/configurationGenerateSiteMap.json.php

```
@@ -16,6 +16,7 @@
16 16      $obj->msg = __("Permission denied");
17 17      die(json_encode($obj));
18 18  }
   19 + forbidIfIsUntrustedRequest('configurationGenerateSiteMap');
19 20  $sitemap = siteMap();
20 21
21 22  if (empty($sitemap)) {
```

objects/notifySubscribers.json.php

```
@@ -10,6 +10,7 @@
10 10  if (!User::canUpload()) {
11 11      forbiddenPage('You can not notify');
12 12  }
   13 + forbidIfIsUntrustedRequest('notifySubscribers');
13 14  $user_id = User::getId();
14 15  // if admin bring all subscribers
15 16  if (User::isAdmin()) {
```

objects/pluginAddDataObject.json.php

```
@@ -9,6 +9,7 @@
9  9  if (!User::isAdmin()) {
10 10      die('{"error":"'.__("Permission denied").'"}');
11 11  }
   12 + forbidIfIsUntrustedRequest('pluginAddDataObject');
12 13  if (empty($_POST['id'])) {
13 14      die('{"error":"'.__("ID can't be blank").'"}');
14 15  }
```

objects/pluginRunUpdateScript.json.php

```
@@ -9,6 +9,7 @@
9  9  if (!User::isAdmin()) {
10 10      forbiddenPage('Permission denied');
```

```
11  11      }
    12  +   forbidIfIsUntrustedRequest('pluginRunUpdateScript');
12  13      if (empty($_POST['name'])) {
13  14          forbiddenPage('Name can\'t be blank');
14  15      }
```

objects/userDelete.json.php

```
        @@ -9,5 +9,6 @@
9   9       if (!User::isAdmin() || empty($_POST['id'])) {
10  10          die('{"error":"'.__("Permission denied").'"}');
11  11      }
    12  +   forbidIfIsUntrustedRequest('userDelete');
12  13      $item = new UserGroups($_POST['id']);
13  14      echo '{"status":"'.$item->delete().'"}';
```

objects/userGroupsAddNew.json.php

```
        @@ -8,6 +8,7 @@
8   8       if (!Permissions::canAdminUserGroups()) {
9   9           die('{"error":"' . __("Permission denied") . '"}');
10  10      }
    11  +   forbidIfIsUntrustedRequest('userGroupsAddNew');
11  12
12  13      require_once 'userGroups.php';
13  14      $obj = new UserGroups(@$_POST['id']);
```

objects/userGroupsDelete.json.php

```
        @@ -8,6 +8,7 @@
8   8       if (!User::isAdmin() || empty($_POST['id'])) {
9   9           die('{"error":"'.__("Permission denied").'"}');
10  10      }
    11  +   forbidIfIsUntrustedRequest('userGroupsDelete');
11  12      require_once 'userGroups.php';
12  13      $obj = new UserGroups($_POST['id']);
13  14      echo '{"status":"'.$obj->delete().'"}';
```

objects/videoDelete.json.php

```
@@ -8,6 +8,7 @@
8  8   if (empty($_POST['id'])) {
9  9       forbiddenPage("Id is empty");
10 10  }
   11 + forbidIfIsUntrustedRequest('videoDelete');
11 12  require_once 'video.php';
12 13  if (!is_array($_POST['id'])) {
13 14      $_POST['id'] = [$_POST['id']];
```

objects/videoRefresh.json.php

```
@@ -11,7 +11,7 @@
11 11  if (!Permissions::canModerateVideos() || empty($_POST['id'])) {
12 12      die('{"error":"'.__("Permission denied").'"}');
13 13  }
14    -
   14 + forbidIfIsUntrustedRequest('videoRefresh');
15 15  require_once 'video.php';
16 16  $obj = new Video("", "", $_POST['id']);
17 17  if (empty($obj)) {
```

objects/videoRotate.json.php

```
@@ -9,7 +9,7 @@
9  9   if (!User::canUpload() || empty($_POST['id'])) {
10 10      die('{"error":"'.__("Permission denied").'"}');
11 11  }
12    -
   12 + forbidIfIsUntrustedRequest('videoRotate');
13 13  $type = !empty($_POST['type']) ? $_POST['type'] : "";
14 14
15 15  require_once 'video.php';
```

objects/videoStatus.json.php

```
@@ -9,6 +9,7 @@
9  9   if (!User::canUpload() || empty($_POST['id'])) {
10 10      die('{"error":"' . __("Permission denied") . '"}');
11 11  }
```

```
      12  + forbidIfIsUntrustedRequest('videoStatus');
12    13    if (!is_array($_POST['id'])) {
13    14        $_POST['id'] = [$_POST['id']];
14    15    }
```

objects/videoSwap.json.php

```
            @@ -18,7 +18,7 @@
18    18        $obj->msg = __("Permission denied");
19    19        die(json_encode($obj));
20    20    }
21        -
      21  + forbidIfIsUntrustedRequest('videoSwap');
22    22    if (empty($_POST['videos_id_1']) || empty($_POST['videos_id_2'])) {
23    23        $obj->msg = __("Mou MUST select 2 videos to swap");
24    24        die(json_encode($obj));
```

Comments 0

Please sign in to comment.