CVE-2026-36340

Remote Code Execution (RCE) Vulnerability in Krayin CRM v2.1.5

Remote Code Execution (RCE) Vulnerability in Krayin CRM v2.1.5

CVE: CVE-2026-36340
Severity: Critical
Affected Product: Krayin CRM v2.1.5
Vulnerability Type: Remote Code Execution (RCE)
Authentication Required: Yes

Summary

A critical Remote Code Execution (RCE) vulnerability exists in **Krayin CRM v2.1.5**.

The vulnerability allows an authenticated user to upload arbitrary PHP files through the email composition feature. Uploaded attachments are stored inside a publicly accessible directory without proper validation or execution restrictions.

As a result, an attacker can upload a malicious PHP payload and execute it remotely by accessing the uploaded file URL. Successful exploitation may allow full compromise of the server.

Vulnerability Details

The vulnerable functionality exists in the **Email** → **Compose** feature.

Affected route:

```
POST /admin/mail/create
```

When attaching files to an email, the backend does not properly perform the following checks:

- File extension validation
- MIME type verification
- Restriction against executable file types
- Sanitization or blocking of PHP code
- Storage outside a publicly accessible directory

Because of this, the application accepts `.php` files and stores them directly inside the following publicly accessible path:

```
/public/storage/emails/<mail_id>/<filename>.php
```

Since this directory is served by the web server, the uploaded PHP file can be executed simply by visiting its URL.

Proof of Concept

> This proof of concept should only be used in an authorized testing environment.

1. Log in to Krayin CRM.
2. Go to **Email** → **Compose**.
3. Upload a `.php` file as an attachment.
4. Submit the email form.

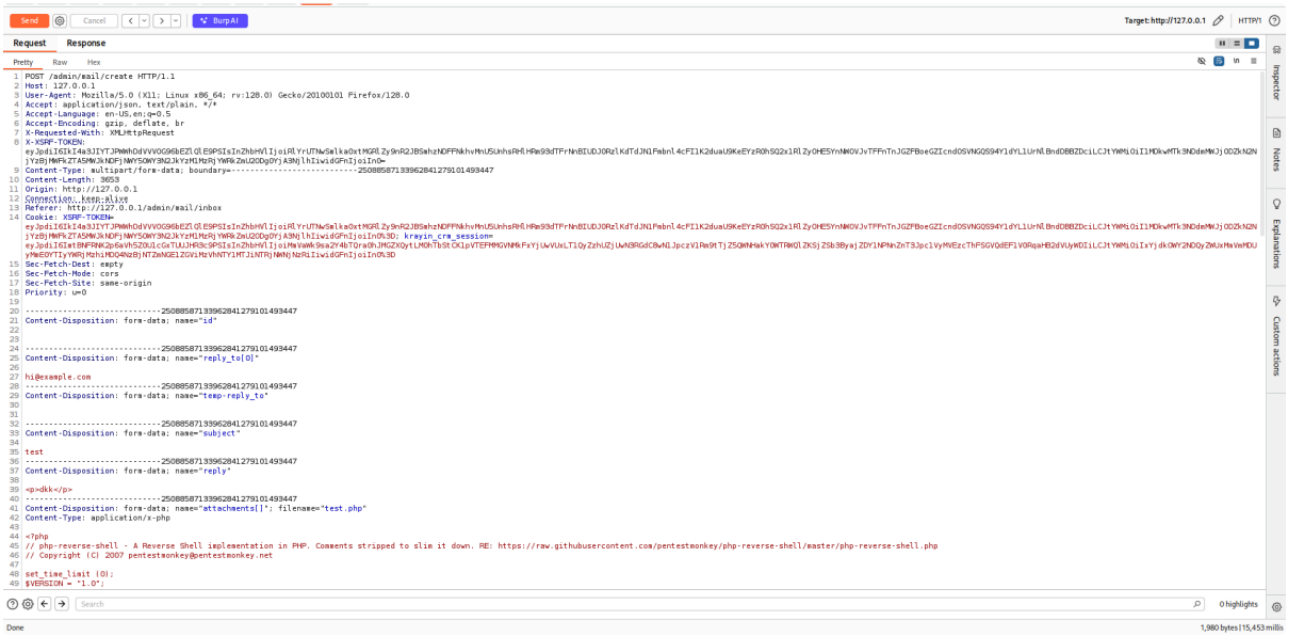
5. The uploaded file is stored under the following path:

```
/storage/emails/<id>/test.php
```

6. Accessing the uploaded file URL executes the uploaded payload.

Request

The following request shows the email attachment upload process.



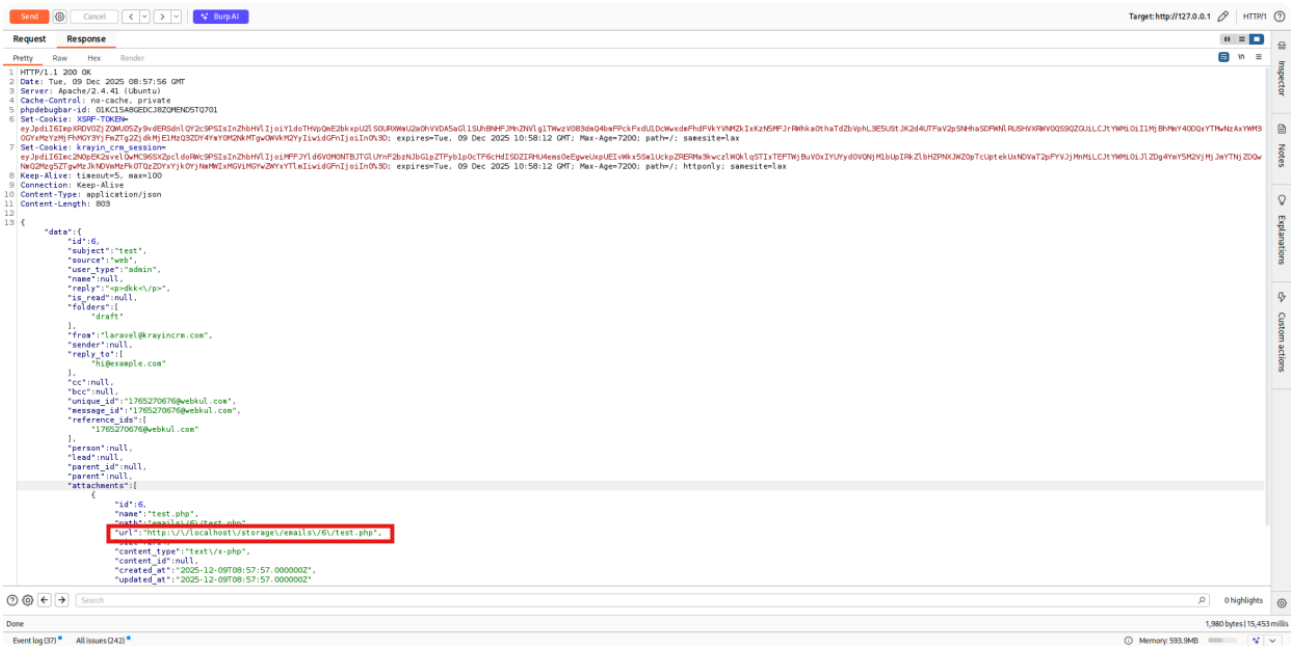
Example affected endpoint:

```
POST /admin/mail/create
```

The request includes the uploaded PHP attachment.

Response

The server stores the uploaded file and returns a publicly accessible file path.



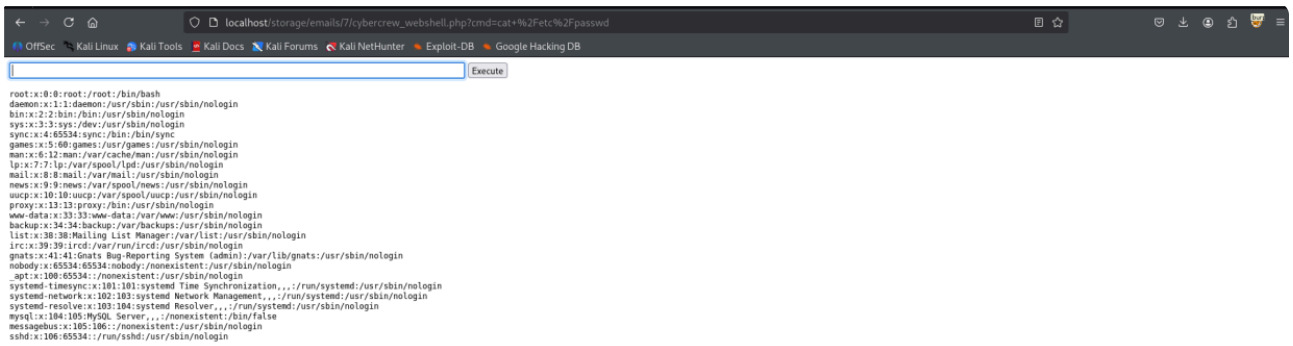
Example uploaded file path:

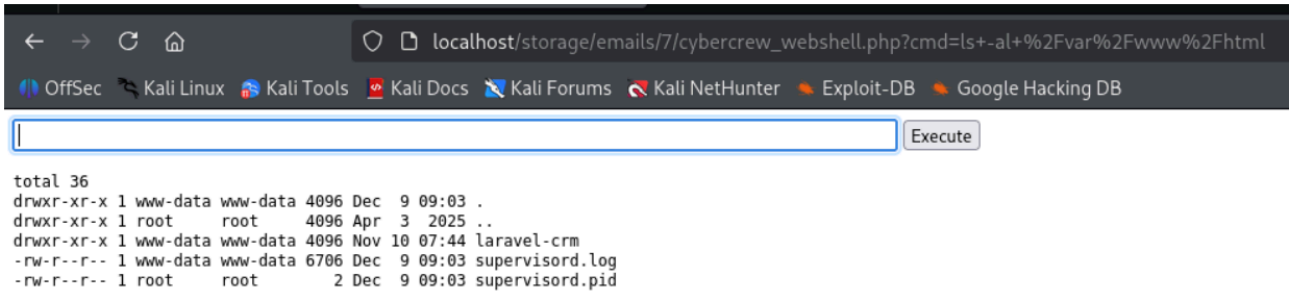
```
/storage/emails/<id>/test.php
```

Remote Code Execution

After the PHP file is uploaded, it can be accessed directly from the browser.

This results in remote execution of the uploaded PHP payload.





Video PoC

[Video PoC]

https://cyber.spool.co.jp/wp-content/uploads/2026/04/RCE-krayin.mp4

Impact

Successful exploitation may allow an attacker to:

- Execute arbitrary OS commands
- Upload and run web shells
- Access, modify, or delete server files
- Pivot into internal network resources
- Steal database contents
- Compromise customer data
- Fully take over the CRM server

Recommendations

To mitigate this vulnerability, the following measures are recommended:

1. Restrict allowed file extensions.
2. Validate MIME types on the server side.
3. Store uploaded attachments outside publicly accessible directories.
4. Block execution of uploaded files through web server rules.
5. Add strict validation in backend controllers.
6. Reject executable file types such as `.php`, `.phtml`, `.phar`, and similar extensions.
7. Rename uploaded files using safe generated filenames.
8. Apply least-privilege permissions to uploaded files and directories.
9. Monitor upload directories for suspicious files.
10. Review all existing uploaded attachments for executable files.

References

Releases

No releases published

Packages

No packages published

Contributors 1

Ishan-Nim D4rkElves