SYNACKTIV
DIGITAL SECURITY

Insecure password reset in Sulu < 1.6.35, 2.0.10 & 2.1.1

Security advisory

2020-08-26

Julien Legras

Vulnerabilities description

1. Presentation of *Sulu*

"Sulu is a content management platform based on Symfony made for businesses. It's a flexible CMS to create and manage enterprise multi-sites and a reliable development environment for high-performance apps. With powerful features for developers and a simple UI for editors it's the ideal engine for state-of-the-art business websites and web-based software."[1]

2. The issues

Synacktiv discovered multiple issues in the password reset feature of *Sulu*:

- cleartext password reset token storage that can be used with a SQL injection to reset an admin account
- admin email address disclosure in password reset feature
- user enumeration in password reset feature

3. Affected versions

The versions before 1.6.35, 2.0.10 & 2.1.1 are affected.

4. Timeline

Date	Action
2020-06-22	Vulnerabilities identified.
2020-06-22	Advisory writing.
2020-06-23	First contact with developers.
2020-06-24	Advisory sent to security@sulu.io
2020-07-29	Release of the fix https://sulu.io/blog/sulu-release-1-6-35-2-0-10-2-1-1
2020-08-26	Public release of this advisory.

1 https://sulu.io/

Technical description and proof-of-concept

1. User enumeration and mail disclosure

Description

When users try to reset their password, the message will change if the user exists or not. If the user exists, the server responds with the user email address:

```
POST /admin/security/reset/email HTTP/1.1
Host: 172.17.0.4:8000
Content-Length: 16
Origin: http://172.17.0.4:8000
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.162 Safari/537.36
Content-Type: application/json
Accept: */*
Referer: http://172.17.0.4:8000/admin/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{"user":"admin"}


HTTP/1.1 200 OK
Host: 172.17.0.4:8000
Content-Type: application/json

{"email":"admin@localhost.local"}
```

Otherwise, an error is returned:

```
HTTP/1.1 400 Bad Request
Host: 172.17.0.4:8000
Content-Type: application/json

{"code":0,"message":"Entity with the type \u0022Sulu\\Bundle\\SecurityBundle\\Entity\\User\
u0022 and the id \u0022nope\u0022 not found."}
```

Impact

By comparing the server responses, attackers can create a list of valid usernames and email addresses. This information can then be reused to perform a login bruteforce for instance. With the email addresses, it is also possible to look into public leaks to find a valid password.

2. Reset token not hashed

Description

When a user asks for a password reset, a token is generated, stored in the database and sent in an email to the user. This token is not hashed, the user is directly retrieved using the token (see *sulu/src/Sulu/Bundle/SecurityBundle/Entity/UserRepository.php*):

```
public function findUserByToken($token)
    {
        $qb = $this->createQueryBuilder('user')
            ->where('user.passwordResetToken=:token');

        $query = $qb->getQuery();
        $query->setParameter('token', $token);

        return $query->getSingleResult();
    }
```

Impact

If the application is impacted by a SQL injection, the attackers could ask for a password reset and extract the reset token by exploiting the injection. Reset token should be considered as critical as passwords, which are correctly hashed.