

SICTF

Write-Up

Linux-Long Challenge :

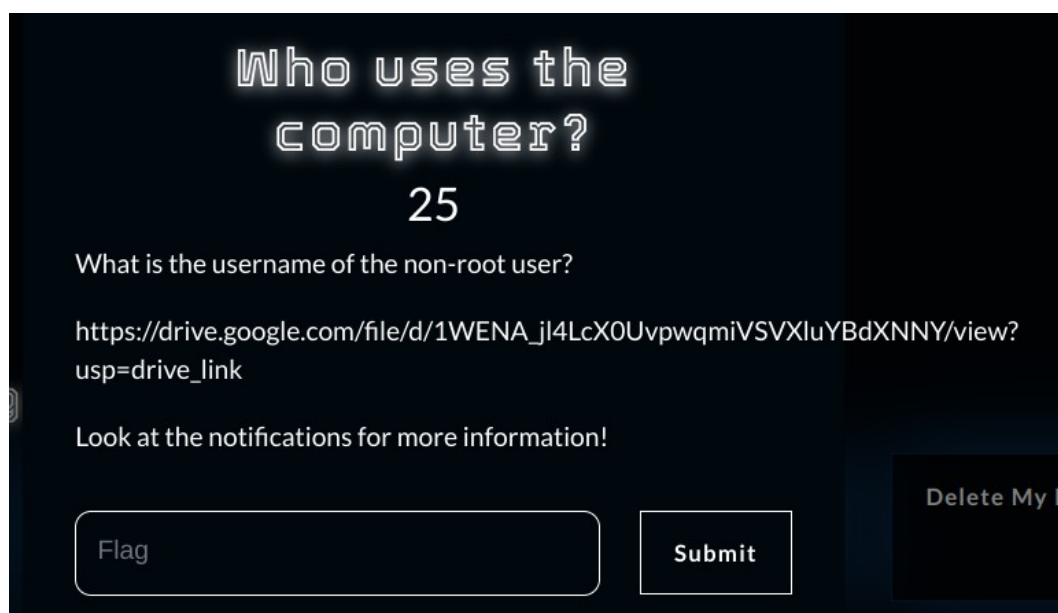
We were provided with a linux .img file. Lets access this file so that we can go through its contents and solve the challenges. A basic google search gave me information about the file. So, I first made a dircetory, mounted the file and then used chroot command to gain access to the .img file.

```
(base) pakambo@gabimaru:~$ mkdir ~/ctf
(base) pakambo@gabimaru:~$ sudo mount -o loop /home/pakambo/Downloads/
Futureman_ctf.img ~/ctf
[sudo] password for pakambo:
(base) pakambo@gabimaru:~$ sudo chroot ~/ctf
chroot: failed to run command '/bin/bash': No such file or directory
```

As we can see in the above image, we were given an error message saying there is no `/bin/bash` directory inside the chroot. This might be either because of missing of some necessary libraries such as `/lib` and `/lib64` or there is no bash in `/bin`. So i tried running the same command but specified the path to `/bin/sh`. And it worked! Now we have access to the given img file.

```
(base) pakambo@gabimaru:~$ sudo chroot ~/ctf /bin/sh
/ #
```

Problem 1: Who uses the computer?



Who uses the computer?

25

What is the username of the non-root user?

https://drive.google.com/file/d/1WENA_jl4LcX0UvpwqmiVSVXluYBdXNNY/view?usp=drive_link

Look at the notifications for more information!

Flag Submit Delete My Entry

In linux, the `/etc/passwd` file store user account information. So let's use the `cat` (concatenate) command to view its contents. The final command is `cat /etc/passwd`

```
/ # cat /etc/passwd
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21:ftp:/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
cyrus:x:85:12:cy:/usr/cyrus:/sbin/nologin
vpopmail:x:89:89:Vpopmail:/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmisp:x:209:209:smmisp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
chrony:x:100:101:chrony:/var/log/chrony:/sbin/nologin
emmettbrown:x:1000:1000:Doctor Emmett Lathrop Brown:/home/emmettbrown:/bin/ash
/ # id
uid=0(root) gid=0(root) groups=0(root)
```

We will get the output as shown in the above figure. The contents are in this format `username: passwd : user id : group id : GECOS : home directory : command/shell`. Generally, root users have their UID as “0” and non root users will have the UID equal to 1000 or greater. In the obtained output, we can see the user “emmettbrown” with UID ‘1000’.

Flag : SICTF{emmettbrown}

Problem 2: Password check

Challenge

11 Solves

✕

Password Check

25

When was the last password changed for the non-root user?

Flag

Submit

Since the information related to the passwords of the users is stored in `/etc/shadow` file, I used `cat` command to view its contents but couldn't find anything useful. Then tried using `grep` command on logs to find information related the passwords.

```
grep -R -i passwd /var/log/*
```

```
/ # grep -R -i passwd /var/log/*  
/var/log/messages:Sep  9 20:38:58 futureman auth.info passwd: password for  
emmettbrown changed by root
```

Since it was mentioned that time is supposed to be in epoch format, I converted the time in GMT format to the epoch format and submitted the flag

Flag: SICTF{1694291938}

Problem 3: Browsing the Interwebs

The image shows a challenge interface with a dark blue background. At the top, the title 'Browsing the Interwebs' is written in a large, glowing, monospace-style font. Below the title is the number '25'. Underneath that is the question 'When did the user install a browser?'. At the bottom, there are two buttons: a rounded rectangle labeled 'Flag' and a square labeled 'Submit'.

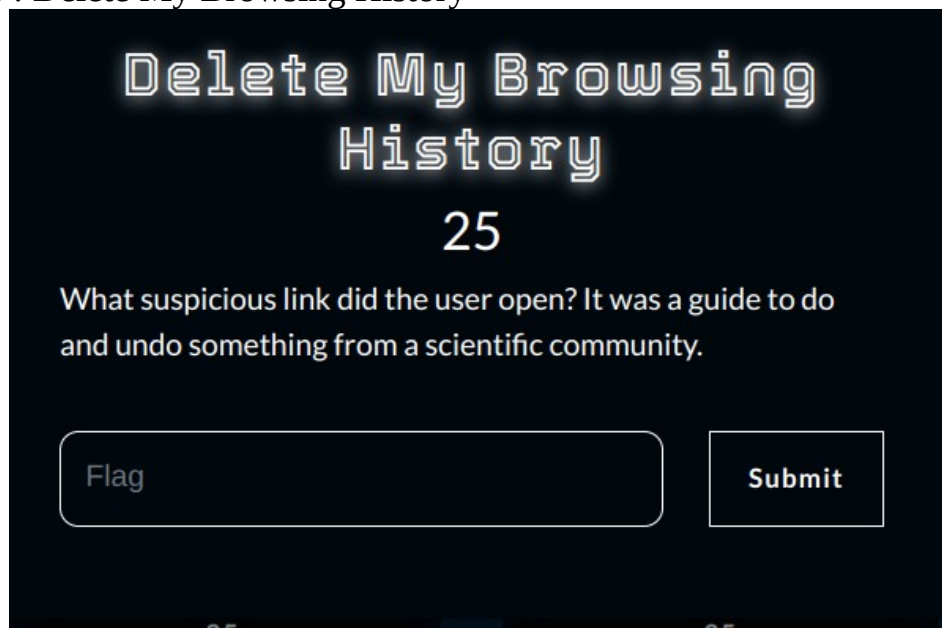
I checked the bash history to find information about the installation with the command `history`, but didn't find anything useful there. After a brief search in Google, I found out that the information about installations is stored in logs. So I opened `/var/log` in the terminal and used the command `cat <file name> | grep browser` initially, where I replaced the `<file name>` with all the files in the folder (there are only 6 files). I didn't get anything useful, so this time I used the command `cat <file name> | grep firefox` and repeated the same process, where I finally got the time stamp when I ran the command with file name as 'messages'.

```
/ # cd /var/log
/var/log # cat messages | grep browser
/var/log # cat messages | grep firefox
Sep  9 21:19:51 futureman authpriv.info : emmettbrown ran command apk add firefox
as root from /home/emmettbrown
/var/log #
```

Now, it's time to convert the obtained timestamp from GMT format to EPOCH time format and submit the flag.

Flag: SICTF{1694294391}

Problem 4 : Delete My Browsing History



In my first try, I went to the `/home/emmettbrown/.mozilla/firefox/slaowkrn.default-release` location and used `cat` command on all `.txt`, `.ini` and `.json` files and used `strings` command on `.sqlite` files and went through all the outputs hoping to find something related to the challenge, but couldn't find anything. This is a very tedious process. Since it was mentioned that the url is a guide to something, i searched for files with the word 'guide' in their storage. Finally i found two urls along with their timestamp related to the word guide when i ran this command `cat AlternateServices.txt | grep guide`. Submitted these urls in the specified flag format, but they turned out to be the wrong answers. After a proper google search, found that the history related to the Firefox browser is stored in the file `places.sqlite`.

>Used the `cat` command (`cat places.sqlite`) on the file, but the given output is very large. so used `strings` command (`strings places.sqlite`) on the file to only extraxt human readable information. scrolled through the out put , but couldn't find anything because its too messy.

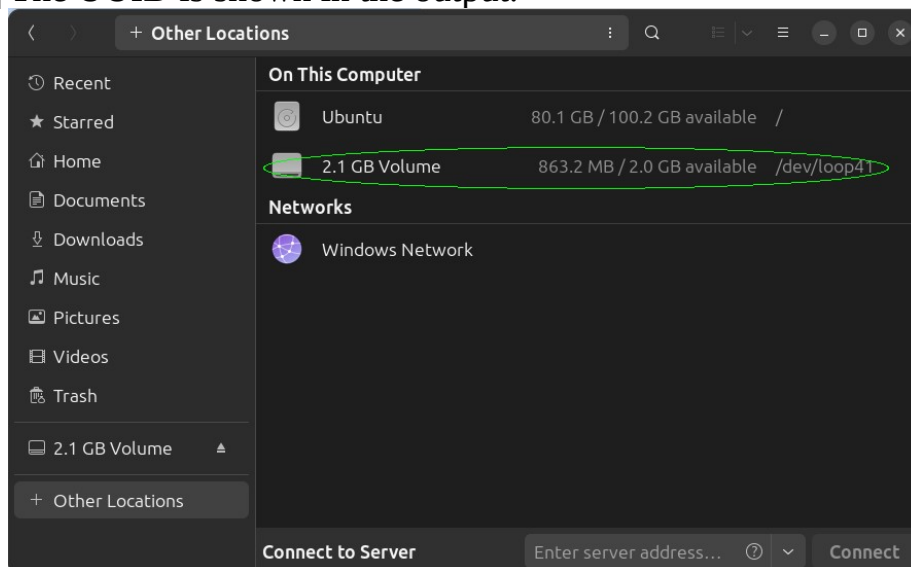
> Searched for how to read `.sqlite` files in google and found out about `sqlitebrowser`. **I moved the file to my laptop's home folder as the application cant be installed via the .img file's terminal.** installed it via terminal with `sudo apt install sqlitebrowser` and opened it by running the command `sqlitebrowser` in terminal. The pplication opened. Then i opened the `places.sqlite` file in this application and went through the table named `mozh_places`. found the suspicious site with the prefix `http`, converted the given timestamp into epoch time in seconds and submitted the flag only for it to be the wrong flag. since ts the wrong flag, i tried submitting the flag with url and timestamp (in seconds) of the same site but with `https` prefix and it worked!!

Problem 5: Say My Name



Did a google search to find info about UUIDs. Found out about the blkid command. The `blkid` command when run, lists the names, UUID, type, size and other information of the partitions of the linux. I ran `blkid` command in host system's terminal, but only got information about the partitions in the system.

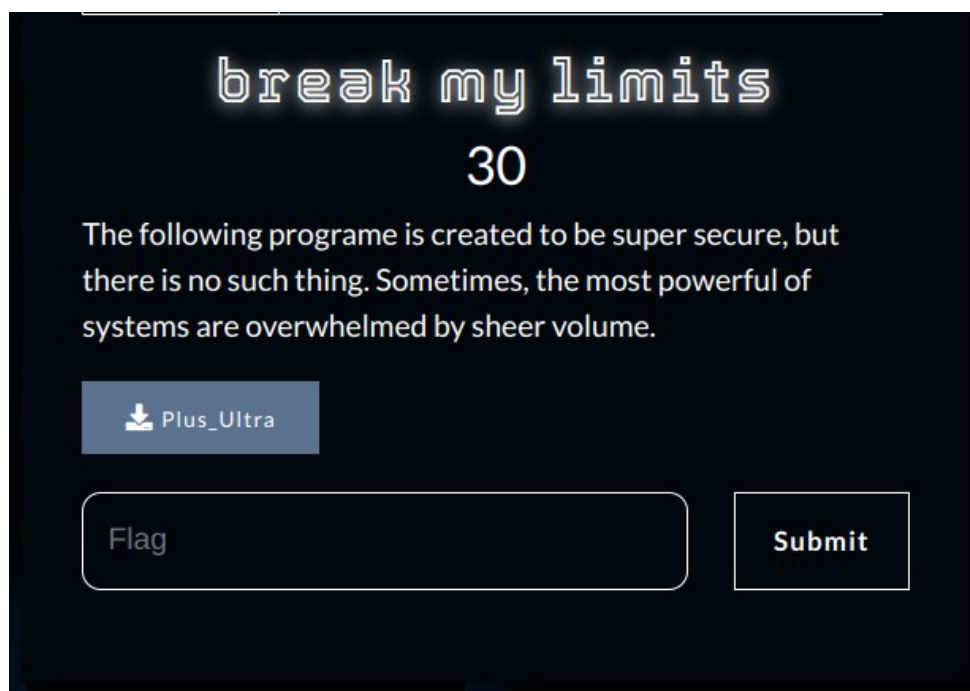
In linux(ubuntu), the system displays the name of the block device in its files. Found out the name of the block device with .img file is `/dev/loop41`. After finding the name, i ran the following command specifying the name of the device, `sudo blkid /dev/loop41` The UUID is shown in the output.



```
(base) pakambo@gabimaru:~$ sudo blkid /dev/loop41
[sudo] password for pakambo:
/dev/loop41: UUID="0d67f47e-a234-41b4-918a-f7e1e66d20db" BLOCK_SIZE="4096" TYPE="ext4"
```

Flag : SICTF{0d67f47e-a234-41b4-918a-f7e1e66d20db}

ELF :



Downloaded the given file. In search of the flag, I first viewed the contents of the Plus_Ultra file using cat command (`cat Plus_Ultra`). The out put was a mix of symbols, text and numbers and it was too messy to read. So used strings command (`strings Plus_Ultra`) to go through the printable characters of the file. Since the output is too large to go through, i used the grep command with some key words like super, secure, overwhelmed etc but wasn't able to find anything useful. Then i used the grep command with the keyword SICTF as that is a part of flag format, lucky for me, i got the flag.

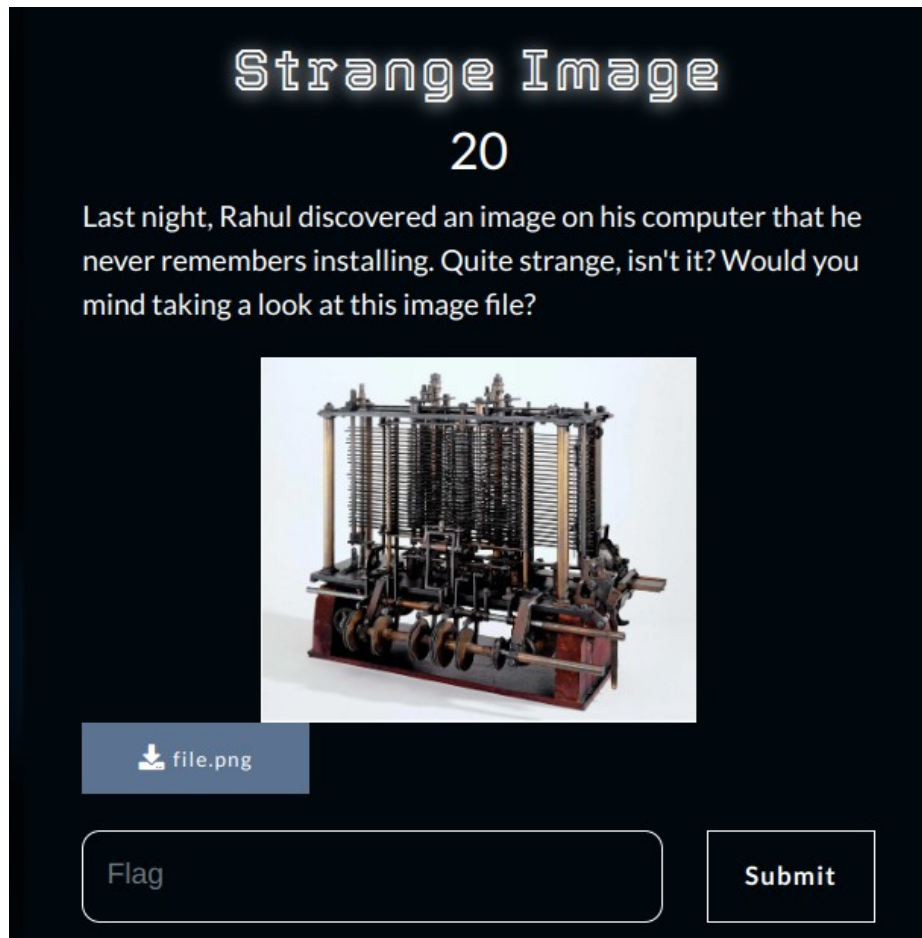
```
(base) pakambo@gabimaru:~/Downloads$ strings Plus_Ultra | grep SICTF
SICTF{br0k3n_h34r75_4nd_m3nd3d_5y573m5}
```

Alternatively, we can also go through the output that we got from running the command `strings Plus_Ultra`, but it is not easy to go through the whole output.

Flag : SICTF{br0k3n_h34r75_4nd_m3nd3d_5y573m5}

Steganography :

Problem 1 : Strange Image



I downloaded the file and used the command `cat file.png`. As usual, the output has many symbols, alphabets and numbers. So ran the command `strings file.png` and got some printable characters as output. Since the output is too large, used `grep` with keywords such as SICTF, flag etc. When i ran the command `strings file.png | grep flag`, i got output as shown in the below image

```
(base) pakambo@gabimaru:~/Downloads$ strings file.png | grep flag
this may be flag: 89488595684968954984oV0}
```

I put this string as a whole in the `cycberchef` website and performed all the operations, but failed in obtaining the flag. After a while, realized that the output for some formats (like decimal, hex etc) are given in a string with space for every 2 digits. so i removed the symbols and alphabets from the input string and added space after every pair of digits. After performing every operation, i finally got the flag (Y0U_D1D_1T) when performing the "from decimal" operation in the site

Flag : SICTF{Y0U_D1D_1T}

Problem 2: Boom Boom


Boom Boom


20

After the historic Trinity Test, one name became more famous than any living being ever - that of Sir J. Robert Oppenheimer, the brilliant physicist behind the Manhattan Project. His legacy lives on, and one of his most iconic moments was captured in an image.

In this challenge, your mission is to uncover a hidden flag within this iconic image that holds the key to unlocking the Oppenheimer Enigma.

Remember, sometimes, secrets hide in plain sight.



 Oppenheimer...

Flag

Submit

Searched internet for image steganography tools and tried them on this image. Found out about the steghide tool and used it on this file. Ran the command `steghide extract -sf Oppenheimer.jpg`. It asked for the passphrase, but since we don't have any, I just pressed enter and got the output.

```
(base) pakambo@gabimaru:~/Downloads$ steghide extract -sf Oppenheimer.jpg
Enter passphrase:
wrote extracted data to "flag".
```

Ran the command `cat flag` to print the contents of the document.

```
(base) pakambo@gabimaru:~/Downloads$ cat flag
U0lDVEZ7MV80TV84M0MwTTNFRDM0N0hfN0gzX0QzNTdyMFkzc18wRl9XMHIxRDV9
```

This string is in some unknown format which we don't know about at present. So I used the `cyberchef` website and ran the operations it provided as mentioned in previous problem's writeup. It turned out that the string is in `base64`. After converting it from base64, we will directly get

"SICTF{1_4M_83C0M3_D347H_7H3_D357r0Y3r_0F_W0r1D5}"

Flag : SICTF{1_4M_83C0M3_D347H_7H3_D357r0Y3r_0F_W0r1D5}

Misc :



In this challenge, I was provided with a .zip file and the title “ Does this ever end?”. I opened the file and extracted the .zip file, I got another .zip file. I repeated the process another time and still I would only get .zip file. By this time, based on the file nature and the question, I realized that it's a nested zip file. So I searched the internet for how to unzip a nested zip file and found this script in a website.

```
while [ "`find . -type f -name '*.zip' | wc -l`" -gt 0 ]; do find -type f -name "*.zip" -exec unzip -- '{} ' \; -exec rm -- '{} ' \;; done
```

-->It continuously searches for files with a .zip extension in the current directory and its subdirectories. (I created a new folder and placed the zip file in it so that the zip files in the directory won't be affected by this code).

-->When it finds a .zip file, it uses the unzip command to extract the contents of the .zip file. After successfully extracting the contents, it uses the rm command to delete the original .zip file in the directory.

-->It repeats this process until there are no more .zip files left in the directory structure.

At the end of the process, we will get a file with name 'flag.txt'. I ran cat flag.txt and obtained the flag.

```
Archive:  ./jplmdbogzqmfybbf.zip
extracting: 7qehc8ehw8.zip
Archive:  ./7qehc8ehw8.zip
inflating: flag.txt
(base) pakambo@gabimaru:~/Downloads/zip$ car flag.txt
Command 'car' not found, but can be installed with:
sudo apt install ucommon-utils
(base) pakambo@gabimaru:~/Downloads/zip$ cat flag.txt
SICTF{17_6035_0n_4nd_0n_4nd_0n} (base) pakambo@gabimaru:~/Downloads/zip$
```

Flag : SICTF{17_6035_0n_4nd_0n_4nd_0n}

Learnings :

- Learned that using google to know locations of certain information in the system saves us a lot of time compared to blindly exploring every file in a directory.
- Learned about strings,blkid, chroot commands.
- Got to know various encryption and decryption tools and also learned how to use a few of them.
- Gained knowledge about logs and locations of various types of information of a system.
- Learned about various formats (base64, binary, decimal,hex etc) which the system uses to store the data.
- Learned about UUIDs and partitions in linux.
- Learned about EPOCH time format