Writeup - Smog is shady (Forensics)

TL;DR

- Mount the `disk.raw` partition
- Recover the deleted files (`secret.vc` and `secret.key`) with *testdisk* or another tool
- Mount the `secret.vc` container with the `secret.key` and read the information inside
- Get the user's password hash in `/etc/shadow`
- Use the `wordlist.txt` file to perform a dictionary attack on the **yescrypt** hash
- Mount the hidden container with the password and get the flag

Statement

> We have conducted a raid at the home of an individual who seems to be planning a bad move. We were able to seize his computer, you will find a copy of the disk. We count on you to find information that would allow us to intervene in time. Maybe a date or a location? In short, hurry up.
> **P.S:** Our agents have been working on a dictionary that could help you if you encounter difficulties.

Files

- *disk.raw*
- *wordlist.txt*

Challenge analysis

In this challenge we are given 2 files. The first one is a copy of a disk that we will probably have to mount to see what it contains. The second file is a simple wordlist, so we can imagine that we will have to perform a bruteforce attack during the challenge. Let's begin !

Mount the disk

Before mounting the disk it's interesting to look at the characteristics of the image.

```
baddhack@shenron:~$ sudo file /media/sf_Tsurugi/disk.raw
/media/sf_Tsurugi/disk.raw: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "mkfs.fat",
sectors/cluster 8, Media descriptor 0xf8, sectors/track 63, heads 255, sectors 4194288 (volumes > 32 MB),
FAT (32 bit), sectors/FAT 4088, reserved 0x1, serial number 0xdd45fc38, label: "smog-deskto"
baddhack@shenron:~$ sudo fdisk -l /media/sf_Tsurugi/disk.raw
Disque /media/sf_Tsurugi/disk.raw : 2 GiB, 2147483648 octets, 4194304 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
```

```
Type d'étiquette de disque : dos
Identifiant de disque : 0x00000000
```

It seems that we have a single partition of **2GB** in **FAT32** format. We also note a label that corresponds to the name of the partition "smog-deskto".

> Yes, I forgot the "p" when I renamed the partition :')

Then we just have to mount the disk and see what's inside using the following commands :

```
baddhack@shenron:~$ sudo mount /media/sf_Tsurugi/disk.raw /media/sf_Tsurugi/volume
baddhack@shenron:~$ cd /media/sf_Tsurugi/volume/
baddhack@shenron:/media/sf_Tsurugi/volume$ ls
boot  cdrom  etc  home  lost+found  opt  srv  sys  tmp
```

It looks like a linux file system with a few less directories.

> As part of the challenge, I decided to create a reduced copy of the disk by removing some heavy directories. The idea of the challenge remains the same and my goal is not to make the players waste time with 20GB files.

Explore the disk

Now we have to browse the file system to find some clues about the individual's plans. Looking briefly at the directories, we notice quite quickly that the one we are interested in is `home`.

```
baddhack@shenron:/media/sf_Tsurugi/volume$ ls home/smog
Bureau     Doc_VeraCrypt.pdf  Modèles  Public         snap             Vidéos
Documents  Images             Musique  SecretMission  Téléchargements
```

"Smog" seems to be the nickname of the individual since it is the name of the user directory and we also saw it in the partiton label.

At this point two things looks suspicous :

- **Doc_Veracrypt.pdf**
  - It's a pdf copy of this site : https://arcanecode.com/2021/05/31/creating-and-using-hidden-containers-in-veracrypt/
- **SecretMission directory**
  - It contains a `confidential.txt` file. Which isn't very interesting after all 🙂

```
baddhack@shenron:/media/sf_Tsurugi/volume/home/smog/SecretMission$ cat confidential.txt
Pastebin : https://bit.ly/TBAi2j64GH
```

We could dig a little deeper into the other directories, but that would be a waste of time. One thing to do in this kind of situation is to look for files that have been deleted from the disk.

To do that, you cas use Autopsy, testdisk, PhotoRec, etc …
I choose to use testdisk.

So I just launched `testdisk` in the partition and followed the indications in the interactive menu. As you can see, the image is recognized as a **FAT32** partition.

```
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

Disk /dev/loop14 - 2147 MB / 2048 MiB - 4194304 sectors (RO)

     Partition                  Start        End    Size in sectors
>   P FAT32                        0    4194303    4194304 [smog-deskto]
```

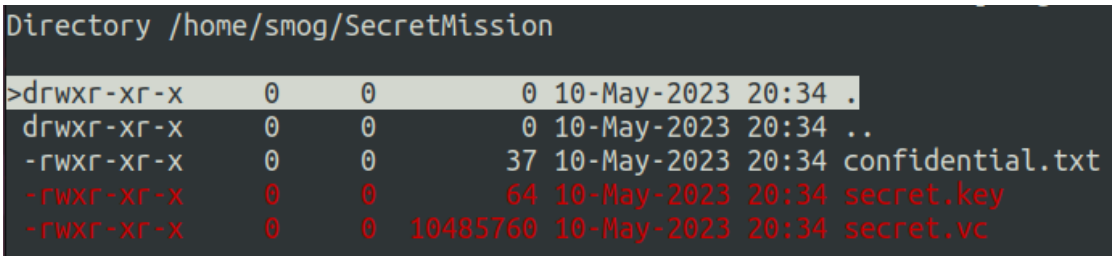
After selecting the partition and choosing “List”, I got the following results :

```
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org
   P FAT32                        0    4194303    4194304 [smog-deskto]
Directory /

>drwxr-xr-x     0     0         0 10-May-2023 20:33 home
 drwxr-xr-x     0     0         0 10-May-2023 20:32 _RASH-~1
 drwxr-xr-x     0     0         0 10-May-2023 20:38 etc
 drwxr-xr-x     0     0         0 10-May-2023 20:39 cdrom
 drwxr-xr-x     0     0         0 10-May-2023 20:41 boot
 drwxr-xr-x     0     0         0 10-May-2023 20:43 lost+found
 drwxr-xr-x     0     0         0 10-May-2023 20:43 opt
 drwxr-xr-x     0     0         0 10-May-2023 20:45 srv
 drwxr-xr-x     0     0         0 10-May-2023 21:02 sys
 drwxr-xr-x     0     0         0 10-May-2023 21:04 tmp
```

```
Directory /_RASH-~1/files

>drwxr-xr-x     0     0         0 10-May-2023 20:32 .
 drwxr-xr-x     0     0         0 10-May-2023 20:32 ..
 -rwxr-xr-x     0     0  10485760 10-May-2023 20:34 secret.vc
 -rwxr-xr-x     0     0        64 10-May-2023 20:34 secret.key
```



We notice the presence of 2 deleted files in the SecretMission folder :

- **secret.vc**
- **secret.key**

> `_RASH-~1` appears to be an additional residual folder. It represents the trash and also contains the same files as the SecretMission folder. So we proceed to the recovery of these files in order to analyze them.

Let's see what our two files represent.

- **secret.vc**
  - This file has an unusual extension. But as we discovered a Veracrypt documentation earlier, everything suggests that it is a container of this type.

> In the past with Truecrypt you could see **.tc** extensions, today for Veracrypt it is the **.hc** extension that seems to be officially recognized. However, I have also seen containers with the **.vc** extension maybe because it seems more logical to rename it like that for VeraCrypt.

- **secret.key**
  - As the name suggests, it is a secret key that may be used to open the container? Let's try it!

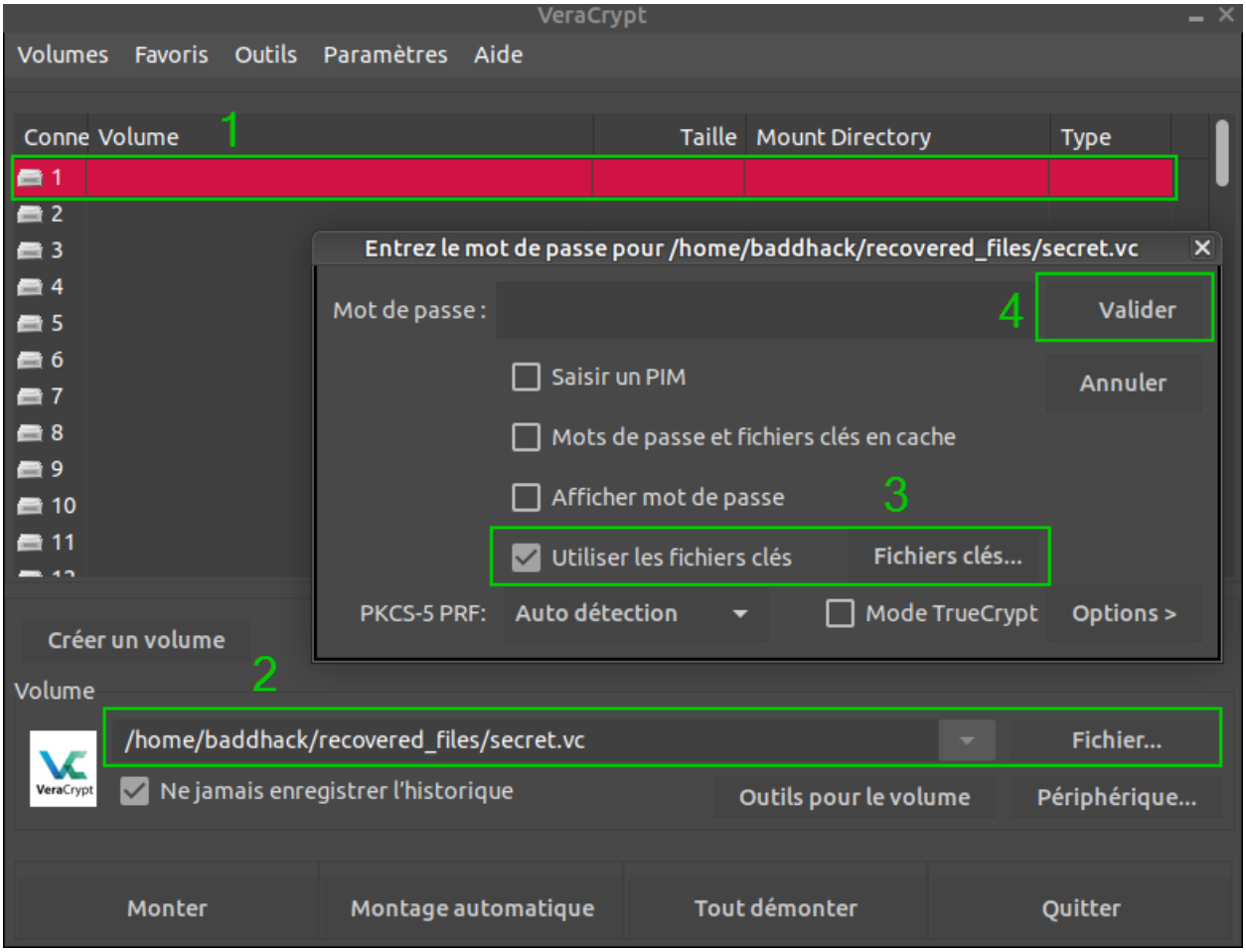
I am going to use the Veracrypt GUI version.

1 - Choose a drive where to mount the container

2 - Select the container and click on “Mount”

3 - Specify the secret key in the “Key files” parameter

4 - Click on “Validate” to mount the container



The Veracrypt container is now open and there is only 1 file inside it. The file is named `consignes.txt` and it contains the following text :

> **FR -**
>
> Salut J' ! J'espère que tu vas bien.
>
> Tu dois être le seul à lire ce message car tu es l'unique possesseur de la clé secrète, je l'ai supprimé de mon côté. Ce conteneur contient l'adresse du lieu où l'opération doit être menée. Tu connais déjà le mot de passe de mon ordinateur, j'ai utilisé le même ;) Pour accéder aux informations, tu n'as qu'à suivre ce que je t'ai appris. Bon courage pour la suite.
>
> Adieu.
>
> **EN -**
>
> Hi J'! I hope you are well.

> You must be the only one reading this message because you are the sole owner of the secret key, I deleted it on my end. This container contains the address of the place where the operation will be conducted. You already know the password of my computer, I used the same one ;) To access the information, you just have to follow what I taught you. Good luck for the next step.
>
> Goodbye.

According to the text file, there is more informations about the mission in this container. If you remember, smog saved a documentation in his personnal folder, dealing with the creation of a hidden Veracrypt container. You can read the documentation for details but to summarize, you can create a Veracrypt container with a password, file or other and add a hidden container inside the same file but with a different way to unlock it.

In this case, smog is telling to his friend that he used the same password as his computer. As it is a Linux system we should take a look at the `/etc/passwd` or `/etc/shadow` files.

There is no hash in the `passwd` file but here is the `shadow` file :

```
root:!:19487:0:99999:7:::
daemon:*:19213:0:99999:7:::
bin:*:19213:0:99999:7:::
sys:*:19213:0:99999:7:::
sync:*:19213:0:99999:7:::
games:*:19213:0:99999:7:::
man:*:19213:0:99999:7:::
lp:*:19213:0:99999:7:::
mail:*:19213:0:99999:7:::
news:*:19213:0:99999:7:::
uucp:*:19213:0:99999:7:::
proxy:*:19213:0:99999:7:::
www-data:*:19213:0:99999:7:::
backup:*:19213:0:99999:7:::
list:*:19213:0:99999:7:::
irc:*:19213:0:99999:7:::
gnats:*:19213:0:99999:7:::
nobody:*:19213:0:99999:7:::
systemd-network:*:19213:0:99999:7:::
systemd-resolve:*:19213:0:99999:7:::
messagebus:*:19213:0:99999:7:::
systemd-timesync:*:19213:0:99999:7:::
syslog:*:19213:0:99999:7:::
_apt:*:19213:0:99999:7:::
tss:*:19213:0:99999:7:::
uuidd:*:19213:0:99999:7:::
systemd-oom:*:19213:0:99999:7:::
tcpdump:*:19213:0:99999:7:::
avahi-autoipd:*:19213:0:99999:7:::
usbmux:*:19213:0:99999:7:::
dnsmasq:*:19213:0:99999:7:::
kernoops:*:19213:0:99999:7:::
avahi:*:19213:0:99999:7:::
cups-pk-helper:*:19213:0:99999:7:::
rtkit:*:19213:0:99999:7:::
whoopsie:*:19213:0:99999:7:::
sssd:*:19213:0:99999:7:::
speech-dispatcher:!:19213:0:99999:7:::
```

```
nm-openvpn:*:19213:0:99999:7:::
saned:*:19213:0:99999:7:::
colord:*:19213:0:99999:7:::
geoclue:*:19213:0:99999:7:::
pulse:*:19213:0:99999:7:::
gnome-initial-setup:*:19213:0:99999:7:::
hplip:*:19213:0:99999:7:::
gdm:*:19213:0:99999:7:::
smog:$y$j9T$yyna0ApsoYAVxkSLrqwWF0$.hcmFMWWQ3EAZa0dqXA7aB6fSX1ujaGHaiKigAVjNt2:19487:0:99999:7:::
vboxadd:!:19487::::::
```

Now we have smog's password hash but it seems to be difficult to bruteforce. Indeed, the hash algorithm used is **yescrypt**. You can find that just by googling the prefix `$y$`. Depending on the password or your wordlist, it can take a very long time.

> Reference : https://manpages.debian.org/unstable/libcrypt-dev/crypt.5.en.html#Prefix

Fortunately, our best agents did a great job to collect many informations on smog and to provide us a dictionary with about 20k possible passwords.

To perform a dictionary attack we need to use John The Ripper because Hashcat doesn't support the **yescrypt** algorithm yet. So we need to give the **/etc/shadow** (or just the hash in a `hash.txt` file), specify the `crypt` format and our wordlist file.

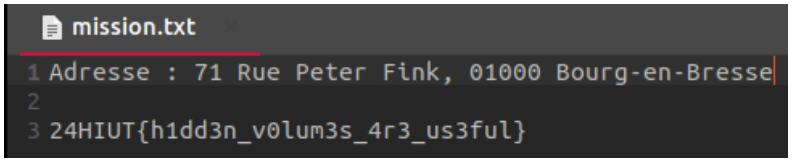
WARNING : In general you can omit the `-wordlist=` option and directly specify the wordlist. I don't know why but for this hash format, the dictionnary attack fails if you don't use the parameter.

```
┌──(baddhack㉿kali)-[~]
└─$ john /media/sf_Kali/shadow --format=crypt -wordlist=/media/sf_Kali/Files/wordlist.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
milflover        (smog)
1g 0:00:00:51 DONE (2023-05-11 18:55) 0.01926g/s 210.8p/s 210.8c/s 210.8C/s oscarteamo..kenya
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Well … we got our password ! 👀😅

All you have to do know is using it to access the hidden container. Instead of choosing the "Key files" option juste enter the password and click on "Validate".

Then, you will gain access to a `mission.txt` file with the adress of the place where the operation will be conducted.



Conclusion

This challenge is not very complicated and I left a lot of clues to guide the players on the steps to achieve. I wanted to create this challenge because I already faced a Veracypt container where I had to guess that there was a hidden container. To my knowledge, it is technically impossible to know that there is a hidden container. So I left clues to help the participants discover this. The dictionary attack part is quite classical in this kind of exercise but the yescrypt algorithm is rarely used because of the complexity to break it, that's why I provided a reduced version of rockyou.txt, so with a weak password it's still possible to do it in the given time. Finally, as this competition is also intended for beginners, I thought it was a good idea to add a deleted file recovery part at the beginning of the challenge.

I hope you enjoyed this challenge or at least discovered some things. Thanks for your participation.