# SQUASHFS FILESYSTEM

Boot2root writeups

On dispose du fichier **iso** utilisé pour boot la machine (`BornToSecHackMe-v1.1.iso`). Cela signifie que ce fichier contient le **filesystem** de notre cible, qu'on peut alors examiner (sans le modifier puisqu'il est indiqué dans le sujet qu'il est interdit de modifier l'image **iso**).

Pour cela, on va commencer par **mount** l'image **iso** pour avoir accès aux fichiers qui la compose :

```
mkdir /mnt/iso
mount -o loop ~/Downloads/BornToSecHackMe-v1.1.iso /mnt/iso
```

On se rend dans le dossier de mount **/mnt/iso**. Différents fichiers nécessaires à la construction du système cible sont présents.

```
→ /mnt mkdir /mnt/iso
→ /mnt mount -o loop ~/Downloads/BornToSecHackMe-v1.1.iso /mnt/iso
mount: /mnt/iso: WARNING: source write-protected, mounted read-only.
→ /mnt cd iso
→ iso ls -la
total 18
dr-xr-xr-x 7 root root 2048 Jun 16  2017 .
drwxr-xr-x 3 root root 4096 Jan  7 07:06 ..
dr-xr-xr-x 2 root root 2048 Jun 16  2017 casper
dr-xr-xr-x 2 root root 2048 Jun 16  2017 .disk
dr-xr-xr-x 2 root root 2048 Jun 16  2017 install
dr-xr-xr-x 2 root root 2048 Jun 16  2017 isolinux
-r--r--r-- 1 root root  844 Jun 16  2017 md5sum.txt
dr-xr-xr-x 2 root root 2048 Jun 16  2017 preseed
-r--r--r-- 1 root root  201 Jun 16  2017 README.diskdefines
-r--r--r-- 1 root root    0 Jun 16  2017 ubuntu
→ iso
```

Le **filesystem** (ce qui nous intéresse) est disponible dans ce dossier sous la forme d'un **fichier squashfs** :
https://en.wikipedia.org/wiki/SquashFS

Un fichier squashfs contient une version hautement compressée d'un **filesystem Linux**, en **read-only**. C'est à partir de ce fichier que le véritable système est ensuite construit à partir de l'ISO. On peut trouver ce fichier squashfs au path **casper/filesystem.squashfs** dans le dossier où on a mount notre iso :

```
→ iso cd casper
→ casper ls -la
total 416385
dr-xr-xr-x 2 root root      2048 Jun 16  2017 .
dr-xr-xr-x 7 root root      2048 Jun 16  2017 ..
-r--r--r-- 1 root root     15188 Jun 16  2017 filesystem.manifest
-r--r--r-- 1 root root     15154 Jun 16  2017 filesystem.manifest-desktop
-r--r--r-- 1 root root        11 Jun 16  2017 filesystem.size
-r--r--r-- 1 root root 404209664 Jun 16  2017 filesystem.squashfs
-r--r--r-- 1 root root  17086307 Jun 16  2017 initrd.gz
-r--r--r-- 1 root root       201 Jun 16  2017 README.diskdefines
-r--r--r-- 1 root root   5045536 Jun 16  2017 vmlinuz
→ casper file filesystem.squashfs
filesystem.squashfs: Squashfs filesystem, little endian, version 4.0, zlib compressed, 404208299 bytes
→ casper
```

On va mount ce filesystem **squashfs** dans le dossier **/mnt/rofs** :

```
mkdir /mnt/rofs
mount -t squashfs iso/casper/filesystem.squashfs rofs
```

Et on dispose désormais d'un accès au **filesystem** de notre système cible (**en read-only**) :

```
→  /mnt mkdir rofs
→  /mnt mount -t squashfs iso/casper/filesystem.squashfs rofs
→  /mnt cd rofs
→  rofs ls -la
total 4
drwxrwxrwx  21 root     root  352 Jun 16  2017 .
drwxr-xr-x   4 root     root 4096 Jan  7 07:08 ..
drwxr-xr-x   2 root     root 2734 Oct 13  2015 bin
drwxr-xr-x   3 root     root  274 Oct 13  2015 boot
drwxr-xr-x   2 root     root   30 Jun 16  2017 dev
drwxr-xr-x 100 root     root 3295 Jun 16  2017 etc
drwxrwx--x   9 www-data root  126 Oct 13  2015 home
lrwxrwxrwx   1 root     root   37 Oct  7  2015 initrd.img → /boot/initrd.img-3.2.0-91-generic-pae
drwxr-xr-x  22 root     root 1420 Oct 13  2015 lib
drwxr-xr-x   3 root     root   28 Jun 16  2017 media
drwxr-xr-x   2 root     root    3 Jun 16  2017 mnt
drwxr-xr-x   2 root     root    3 Oct  7  2015 opt
drwxr-xr-x   2 root     root    3 Jun 16  2017 proc
drwx------   5 root     root  150 Oct 15  2015 root
drwxr-xr-x   2 root     root    3 Jun 16  2017 run
drwxr-xr-x   2 root     root 3446 Oct 13  2015 sbin
drwxr-xr-x   2 root     root    3 Mar  5  2012 selinux
drwxr-xr-x   3 root     root   26 Oct  7  2015 srv
drwxr-xr-x   2 root     root    3 Jun 16  2017 sys
drwxrwxrwt   2 root     root    3 Jun 16  2017 tmp
drwxr-xr-x  10 root     root  174 Oct  7  2015 usr
drwxr-xr-x  13 root     root  196 Jun 16  2017 var
lrwxrwxrwx   1 root     root   33 Oct  7  2015 vmlinuz → boot/vmlinuz-3.2.0-91-generic-pae
```

L'idée à partir de là est d'examiner les fichiers afin de pouvoir trouver des credentials, des hash, des clés ssh etc... On récupère d'abord les hash de mot de passe utilisateurs dans le fichier **/etc/shadow**.

```
root:$6$P3HXAOsR$Lmz85I7RXUJLU8KR.C2okbToyNfq5QIDj6YOoWYilDWQ3e.dhXC/bamN4xLcAZVHHLFuszMaGD6nRa5HrFAls0:16723:0:99999:7:::
daemon:*:16715:0:99999:7:::
bin:*:16715:0:99999:7:::
sys:*:16715:0:99999:7:::
sync:*:16715:0:99999:7:::
games:*:16715:0:99999:7:::
man:*:16715:0:99999:7:::
lp:*:16715:0:99999:7:::
mail:*:16715:0:99999:7:::
news:*:16715:0:99999:7:::
uucp:*:16715:0:99999:7:::
proxy:*:16715:0:99999:7:::
www-data:*:16715:0:99999:7:::
backup:*:16715:0:99999:7:::
list:*:16715:0:99999:7:::
irc:*:16715:0:99999:7:::
gnats:*:16715:0:99999:7:::
nobody:*:16715:0:99999:7:::
libuuid:!:16715:0:99999:7:::
syslog:*:16715:0:99999:7:::
messagebus:*:16715:0:99999:7:::
whoopsie:*:16715:0:99999:7:::
landscape:*:16715:0:99999:7:::
sshd:*:16715:0:99999:7:::
ft_root:$6$lOBKtnYu$GBZ.hMzEtOsj02ngxGvMKZzFFBy2DEzil11s0G6a5eDUyEo88t1VZfdVqlfUc/jUG9748avR0.FhT4rJmPqmI1:16723:0:99999:7:::
mysql:!:16715:0:99999:7:::
ftp:*:16716:0:99999:7:::
lmezard:$6$GMSDNCKf$jY6x/LD/KP9rBEYlMucnU78Id5n.RjZ1B5qiuuFqYumqyzYDG5lwK9H1uO0M/NnOZ6N2ER.fSykFXOLPiRAAI/:16716:0:99999:7:::
laurie@borntosec.net:$6$CbcWogd7$NWvym3ZSzuBFOcALEMgBcrZD3KBFIyoJzVG7u8rAcleMnxDIwFSRyDJV2wolMlWpZ0CmS32lnnHWh.JGsR3CI0:16716:0:99999:7:::
laurie:$6$LWmR.4Nu$JduesZbtCf47RAQmHc5t4zCPHnTszTj2ukh4LjHBXXy8LqL7EO8VrUTrTLLUxNa4ZSW7T4dwkrxvaw.Y8xlV2/:16716:0:99999:7:::
thor:$6$VtZqKCe4$KGD40R8uofXRTqzeH4vJt.FjGSODzcCKTkzX6×6bXAeA/mkJbWR1ogTTZjZHbYQ8j309AZF4gZxfGTkufXxwt.:16716:0:99999:7:::
zaz:$6$K9KtjXnH$nsSKIiDTgEzg1NwQSE/anf3fy5wwl6Ybgho3AXgFk9Vou5xeYTj8r0JGdCOYDC47N9mATDPRZ8lsClY2dqJvS0:16716:0:99999:7:::
dovecot:*:16716:0:99999:7:::
dovenull:*:16716:0:99999:7:::
postfix:*:16716:0:99999:7:::
```

On a pas mal de hash, malheureusement on ne parvient pas à les bruteforce facilement avec une

wordlist de type rockyou.txt, et je suppose que l'idée n'est pas de mettre plusieurs jours à bruteforce du sha512.

On ne voit pas non plus de clés privées SSH (tout est en password). Cependant, le **.bash_history** de l'utilisateur **root** nous révèle des credentials pour l'utilisateur **zaz** :


Ce qui nous permet de directement nous connecter en tant que l'utilisateur **zaz**. Il ne nous reste plus ensuite qu'à résoudre **exploit_me** comme dans le writeup1, pour **obtenir root**.