# Try Hack Me room : The Server from hell

# 

# Cccc

# First we deployed the room and made vpn connection.

Then using the provided IP address, we started solving the room

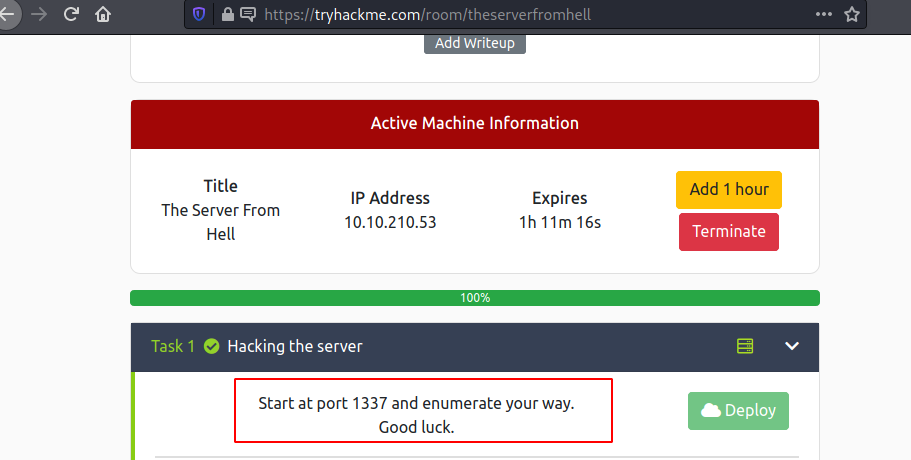
**Scanning**

Using n-map, we performed enumeration

**From the result of n-map scanning, we found many open ports available.**

Not shown: 94 closed ports  
PORT STATE SERVICE VERSION  
1/tcp open tcpmux?  
| fingerprint-strings:   
| NULL:   
|\_ 550 12345 0000000000000000000000000000000000000000000000000000000  
3/tcp open compressnet?  
| fingerprint-strings:   
| NULL:   
|\_ 550 12345 0000000000000000000000000000000000000000000000000000000  
4/tcp open unknown  
| fingerprint-strings:   
| NULL:   
|\_ 550 12345 0000000000000000000000000000000000000000000000000000000  
6/tcp open unknown

**Looking at the room description, there was a hint for us**

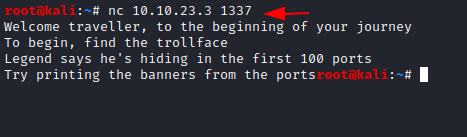
****

**Port Scanning**

Using the hint from the room description, next step was to perform port scanning using netscan.

**Netcat** is used for port scanning, **port redirection**, as a port listener (for incoming connections); it can also be used to open remote connections and so many other things.

**Command** nc ip\_address port\_number

****

This is the message we get when we connect to port 1337

Now, its time to make  a simple script to go over 100 ports and connect to it to grab banner.

i=1  
while [ $i -ne 100 ]  
do   
nc 10.10.173.96 $i  
i=$(( $i + 1 ))  
done

**NOTE**: To run the script , first we made it executable mode using chmod

chmod +x check\_port.sh

where check\_port.sh is the filename and chmod +x makes the file executable.

Then run the script using as: **./filename**

Result **:** It directed us to **go to port 12345**

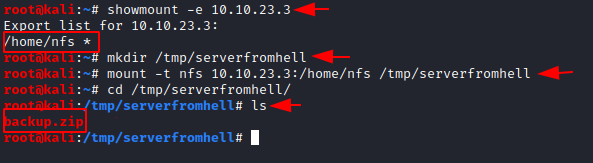
# PORT 12345

**C:\Users\500068672\AppData\Local\Temp\Temp1_The Server From Hell.zip\The Server From Hell\4.png**

The NFS Daemon runs only on NFS Servers (not on clients). It already runs on a static port, 2049 for both TCP and UDP. Firewalls should be configured to allow incoming packets to this port on both TCP and UDP. This firewall exception is only needed for packets incoming to a NFS Server**.**

**The default port of nfs share is 2049 so lets see if there are any shares that we can mount on our local machine**

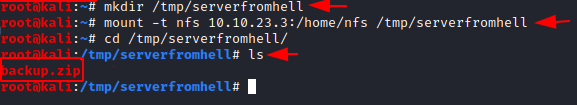
**Command showmount -e ip\_address(**where e prints the list of files that are shared or are exported.)

****

**Now let’s mount that share**

**Command mount –t nfs 10.10.23.3:/home/nfs /tmp/server/fromhell**

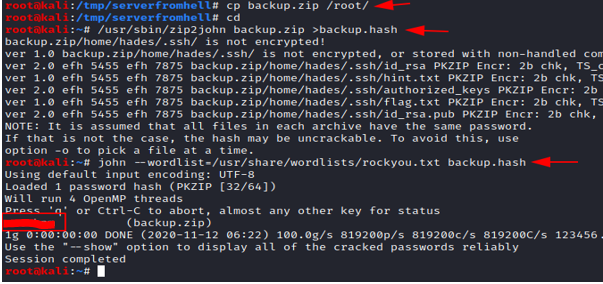
where 10.10.23.3 is the IP of the NFS server and /home/nfs is the the directory which the server is exporting and /tmp/server/fromhell is the local mount point



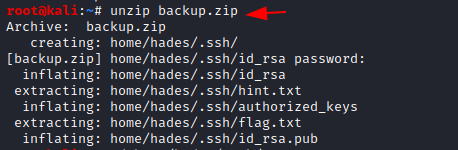
We find a backup.zip and it is asking for password to unzip the folder.

**Password cracking**

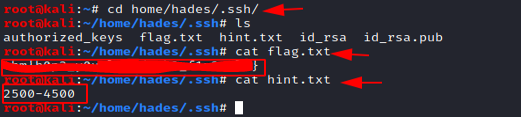
**We use zip2john password cracker and got the password: zxcvbnm**



Now unzip the backup.zip

****

Go to home/hades/.ssh/

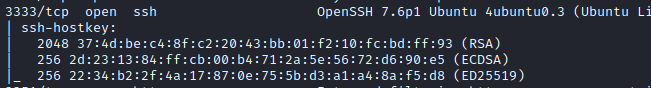


We got our flag along with hint and ssh private key

The hint says:

2500-4500

**From the results of the scan I searched for ssh with open-ssh client**

****

Now ssh login into the system using:

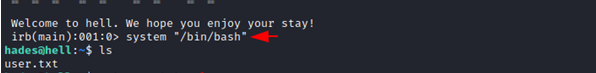
**ssh hades@ip –i id\_rsa –p 3333**

And was logged in ☺



This irb is an interactive ruby shell just like we get in python so in order to get a /bin/bash shell run

**system “/bin/bash”**



So here is our user flag:

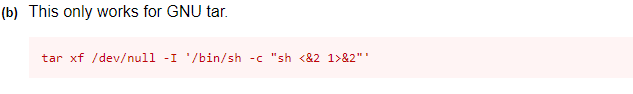


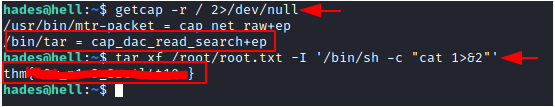
# Privilege Escalation

Now from the room hint, we got to know that we need to use getcap

Getcapcommand tells that which file or binary has capability to access almost anything on the system so run

**getcap –r / 2>/dev/null**(2>/dev/null ,here 2 just redirects Standard output error to null )

From gtfobins (<https://gtfobins.github.io/gtfobins/tar/>) , we get the script to run: 



Finally, we got our root flag ☺