**Pen-test Proposal**

How will your team work together?

* We will independently attempt to hack in to the box in the goal of reaching “root”
* We will share information of our attempted exploits
* We will share our observations of scans and information collected during initial and in depth attempts

Proof of Concept

* We have performed, in class and as a class, multiple facets of the penetration process
  + I.e. establishing connections, version scans, packet transfers, etc.
* We have a fond concept of security assessment through the use of SEIMS and other threat ranking/ monitoring tools
  + Through these concepts we will suggest, in our best opinions, possible solutions (by priority) to the exploits we used

SCOPE

Perform penetration test of virtual machine hosted on HackTheBox.com

* Perform the pen-test on “Poison”
* List forms of access (ssh, http, etc.)
* List strategies formed from initial information (information provided on hosting website)
* Attempt a multitude of exploits in an effort to perform/locate privilege escalation, script injection, backdoor access, etc.
* Document any payloads used that are unsuccessful/successful
* Create walk through on steps taken to gain access to the box and get to root

Give an overview of steps taken to exploit the machine

* Use of openvpn command in conjunction with download file provided from HackTheBox.com
* Transfer of VPN file to kali to designate it as the attack machine

Give possible patches, updates, or changes that can be made to prevent found exploits

Presentation outline:

* Outline of our mission in the project
* What technical tools were used
* What information was found by these tools
* Summary of findings made in our pen-test report
* (will give each person one of these to do for our presentation when we get there to who’s ever most comfortable in each section)

**Introduction**

In this exercise we will be demonstrating the aspects of a penetration test of a box on the Hack The Box website. Through the use of Active and Passive reconnaissance we will gain valuable information to use the tools we have learned through this course and along the way of cracking this box. This is where red turns purple and we use our red team findings to formulate a blue team response to the type of vulnerabilities that were found. Next you will see the first step into “Poison” a Hack The Box machine.

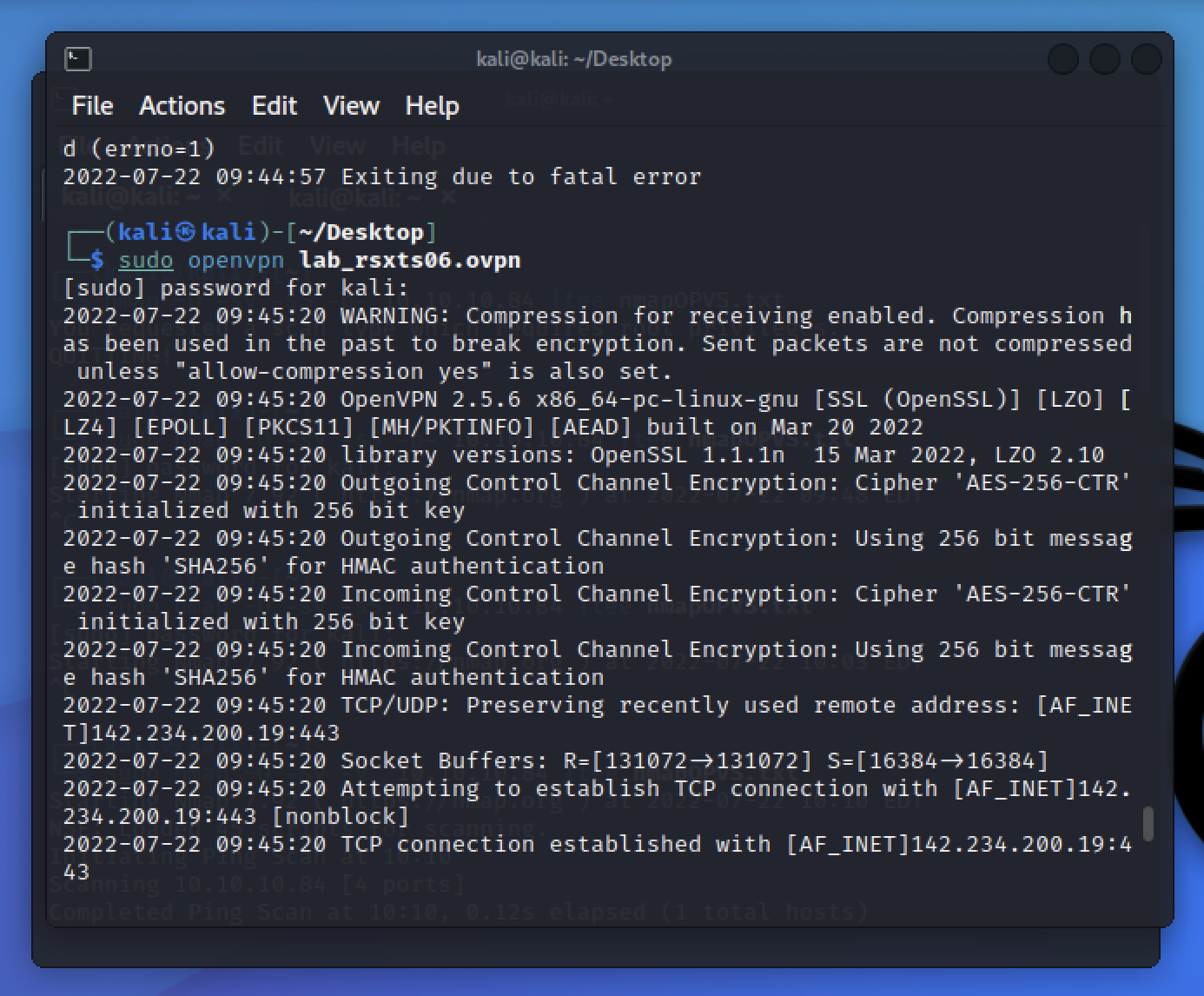
**Methodology**

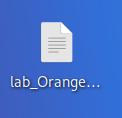
1. Port scanning
2. Using LFI to find the username
3. Gaining encrypted password file using LFI
4. Decrypting password file
5. Logging in to SSH using a decrypted password
6. Transferring ZIP file and extracting to find a secret file
7. Discovery of VNC on the machine
8. VNC tunneling over SSH to get a root shell
9. Grabbing flag

**Rooting the Machine - Walkthrough:**

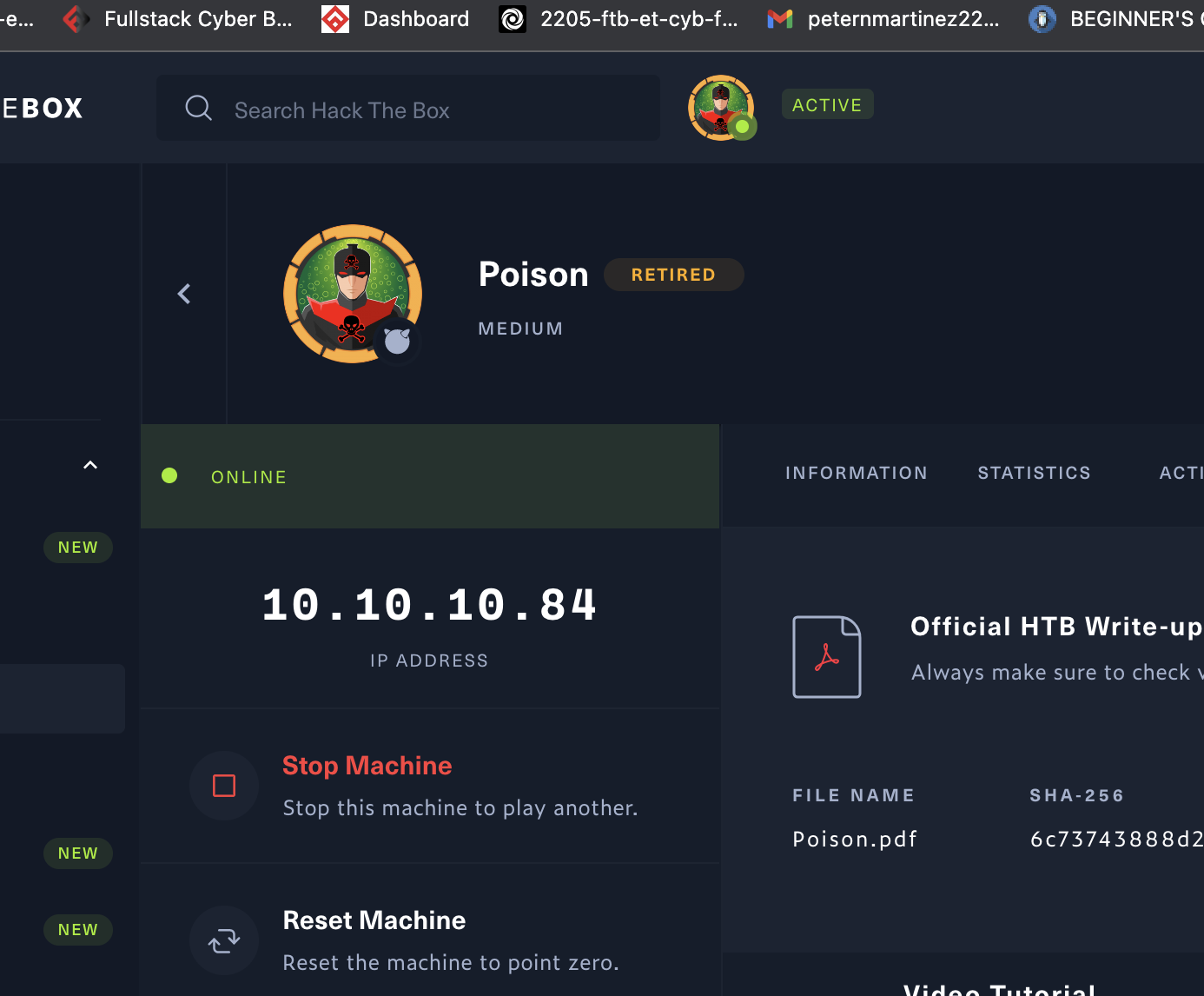
**Step 1:** Connecting to Target VM (openvpn route to Guest kali VM)

In order to get the ip for the target and begin the exercise, we need to connect via openvpn file (.ovpn) and Spawn our victim machine. First, search and find ‘Poison’ in the search field, it will be marked as a retired machine. Click on ‘Connect To HTB’ in the top right, select server, and download the ovpn file. Once you’ve imported/downloaded the file, input ‘sudo openvpn’ lab\_file.ovpn and allow connection to complete. Once the status is “Connected”, click on “Spawn Machine” to generate a victim machine and click on the ip to copy into clipboard.

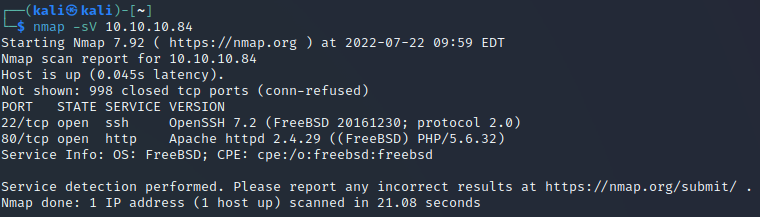
****

****

The “Poison” box is now ready to boot up and run for this penetration test.

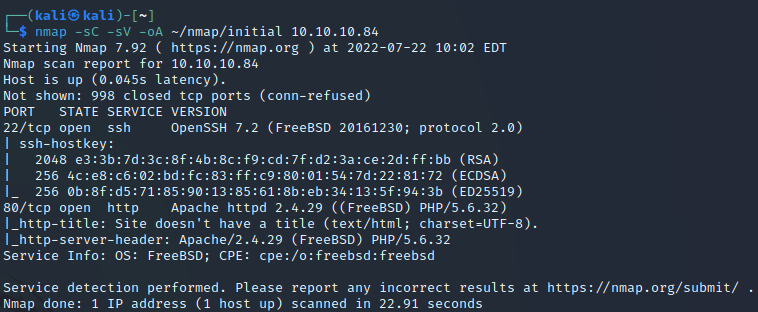
****

^ Target IP Address ^

****

**Step 2:** Enumeration

We’ve opened up with a normal port scan which showed that port 80 and 22 were open. Nmap -sV(a detailed study of ports to determine the version of services)

****

With step two here we dug a little deeper into the nmap scan, showing us more information including the ssh keys. With the nmap -sC -sV -oA ~/nmap/initial 10.10.10.84 command it also created a directory and folder with some nice information for us. **(See Below)**

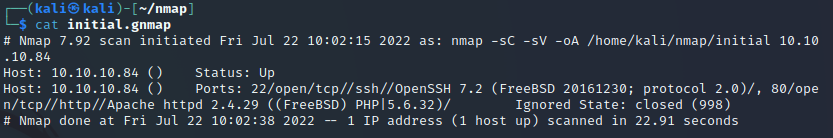
**Nmap**

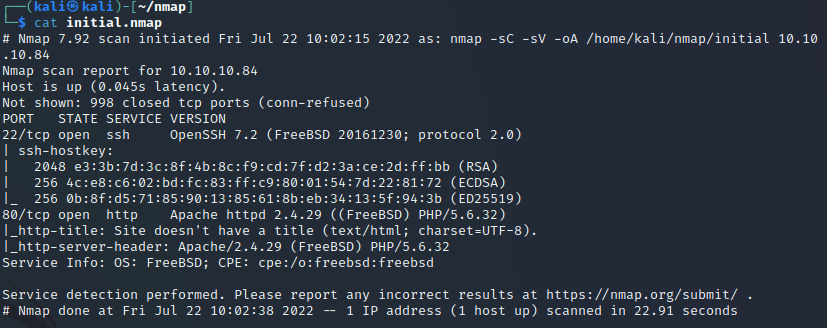
**-sC =** run the default script

**-sV =** a detailed study of ports to determine the version of services

**-oA =** a heavy scan into discovering the Operating System of the machine

****

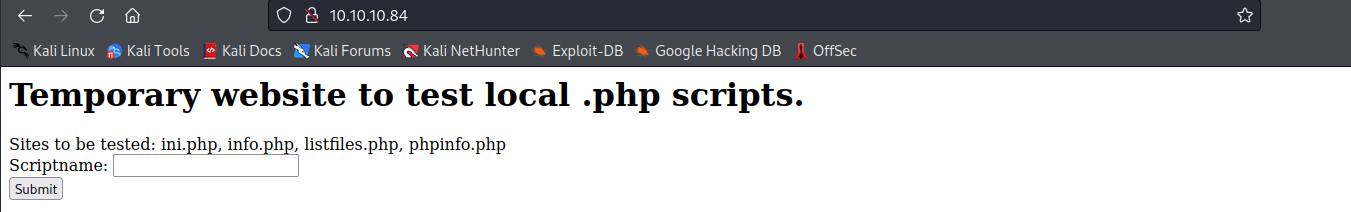
****

****

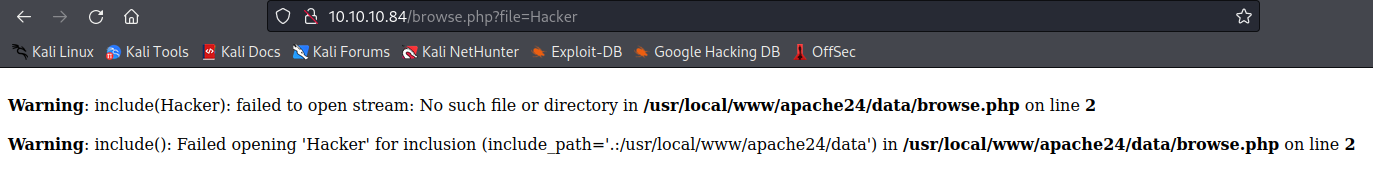
The initial.xml file was a verbose file filled with all of the ports that were scanned through the nmap scan.

**Step 3:** Local File-Inclusion (LFI)

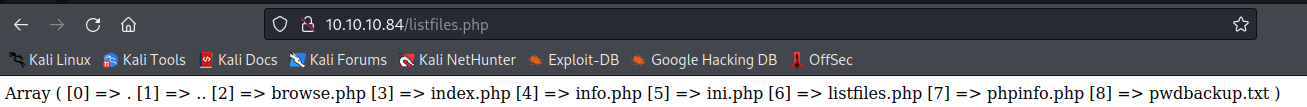
Next we checked out the website that this IP address has set up.

****

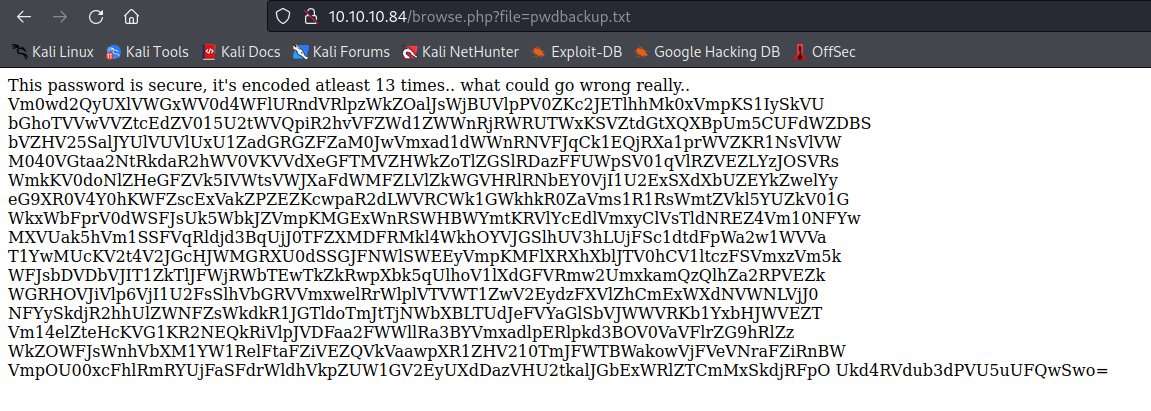
As you can see this site has testing sites that you can try and enter in the Scriptname bar

****

We entered a random term “Hacker” to see what type of results we were going to receive. There isn’t much on the actual site to go on but we do see an area of URL that could possibly be vulnerable for a file inclusion attack

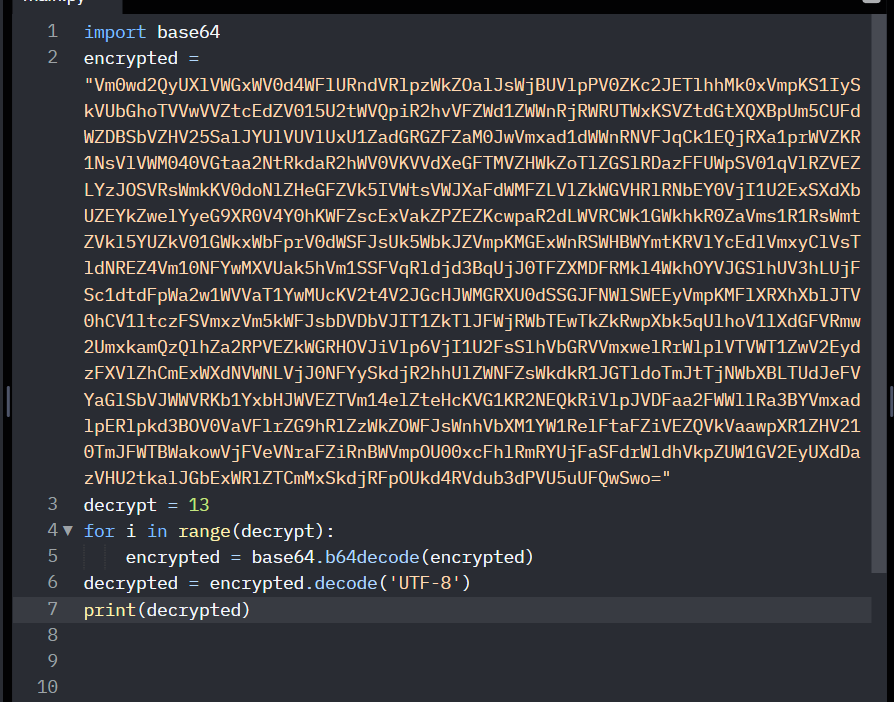
****

Using the “listfiles.php” directory that was given on the main page into the Scriptname bar it shows us a few more directories not shown on the main page. We see the pwdbackup.txt file at number 8 and use that file after the = in the URL and it gives us what appears to be a backup password that has been encoded at least 13 times.

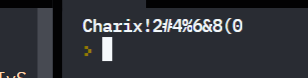
****

**(Bonus Step):** Python script to decode in Base64 (13 times)

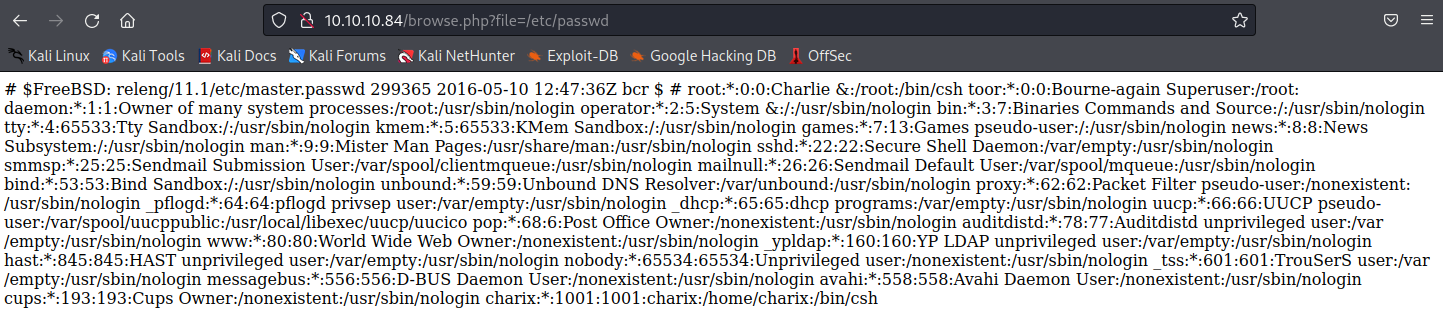
Using python we wrote a script to decrypt the base64 encryption, the message at the top of the passwd page tells us we need to run this function 13 times.

****

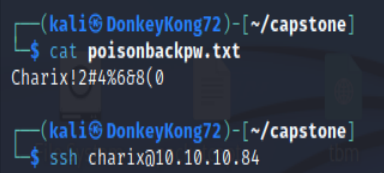
The output of this function is a short string of character that we can only assume to be a password

****

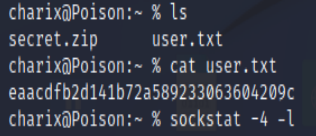
Now using **Local File-Inclusion** within the URL field, we’ve tried locating the etc/passwd file, and here it is and now have found the username for this machine.



This Local File-Inclusion technique gives us all of the usernames that are on this machine from the etc/passwd file.



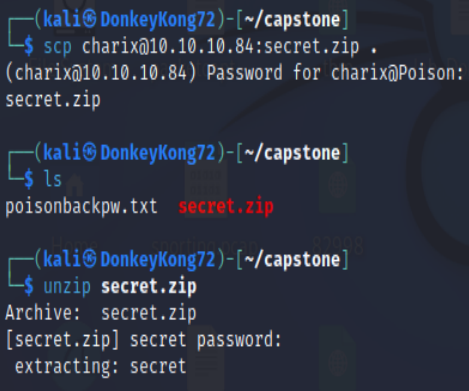
We then touched the file “poisonbackpw.txt” and entered that password to use later.



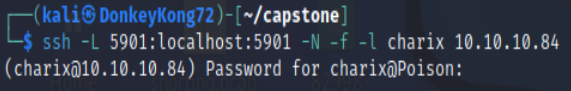


**Step 4:** SSH into VM and Privilege Escalation via SCP.

By learning the password and username from the file-inclusion attack performed on the Victim Machine’s web server, it was then possible to ssh into the machine. Here file secret.zip was found by a simple “ls” command. Now we want to have a list of open ports to see if any can be exploited. This was accomplished by running sockstat -4 -l which gave the list of all IPv4 open ports.



In order to gain access to the secret.zip file from a remote IP. A secure copy was made on the file. It is now able to be unzip on our machine.



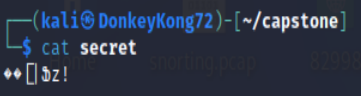
By binding the port, host and hostport together to a username and ip address, it allowed us to run commands specifically for that port with the localhost.

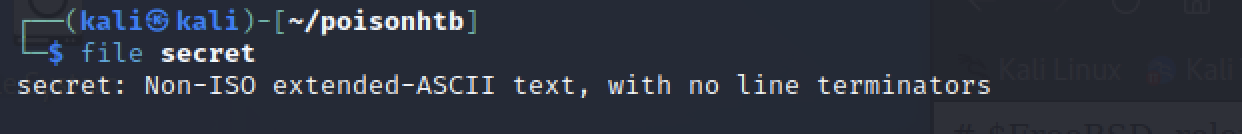
**-L**: Binds the port, host, and hostport together (5901:localhost:5901) to the IP address and username. ”Specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side“

**-N**: Doesn’t execute remote command, just forwards the port

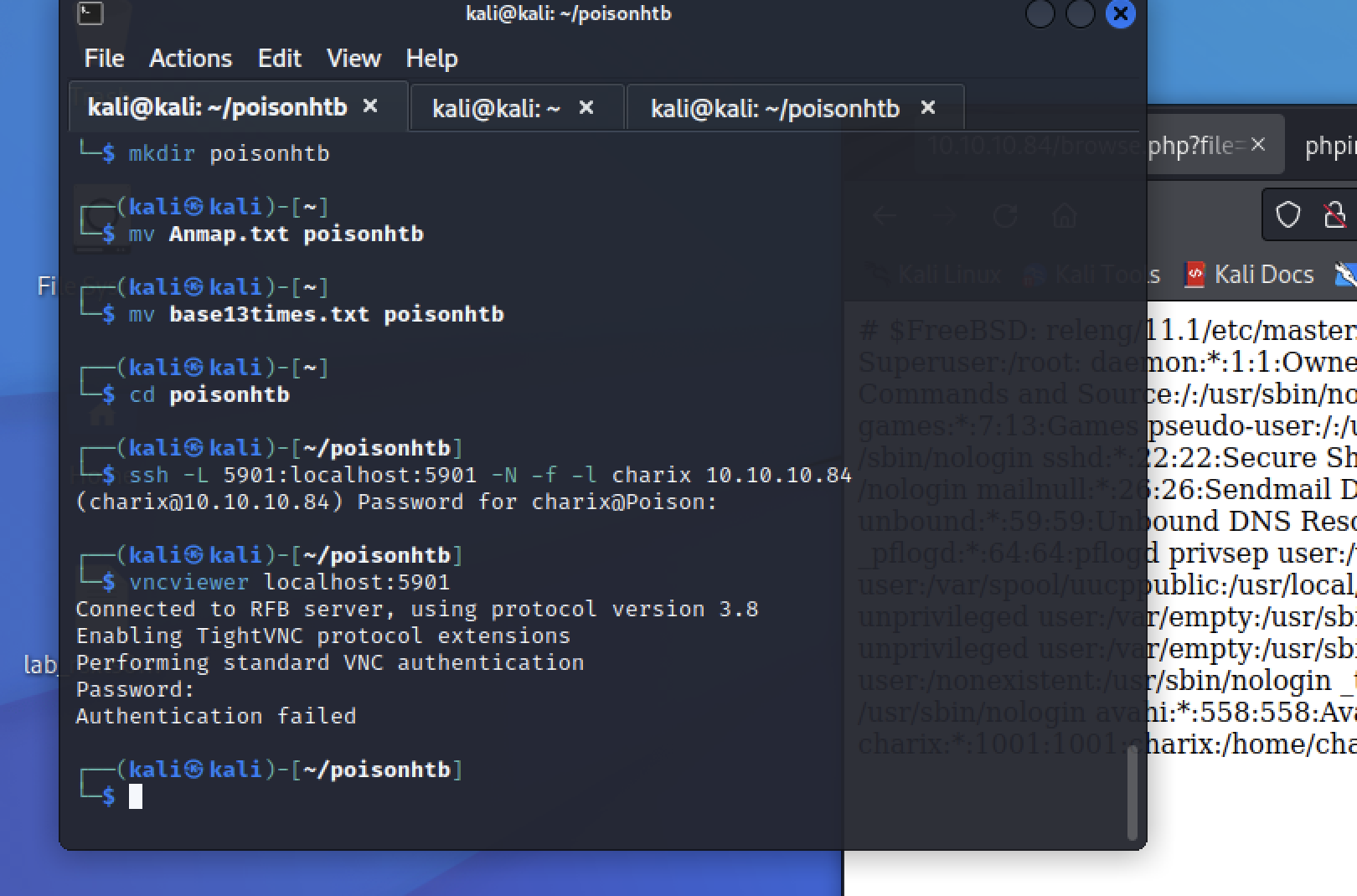
**-f**: Request ssh to go to background when all remote port forwards

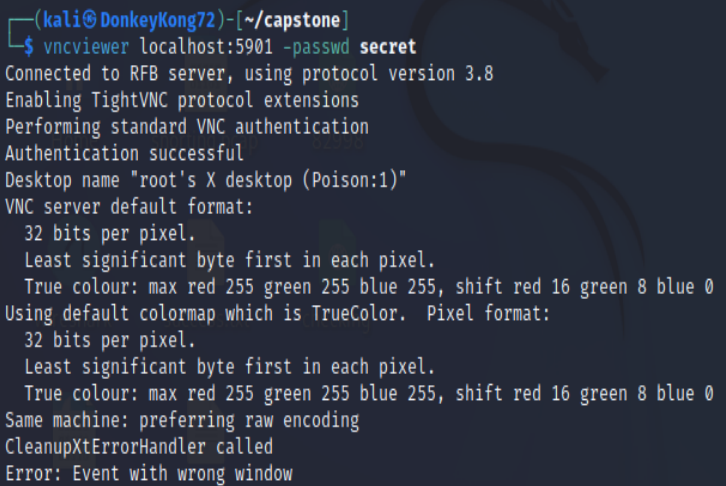
**-l**: Specifies the user to log in as on the remote machine





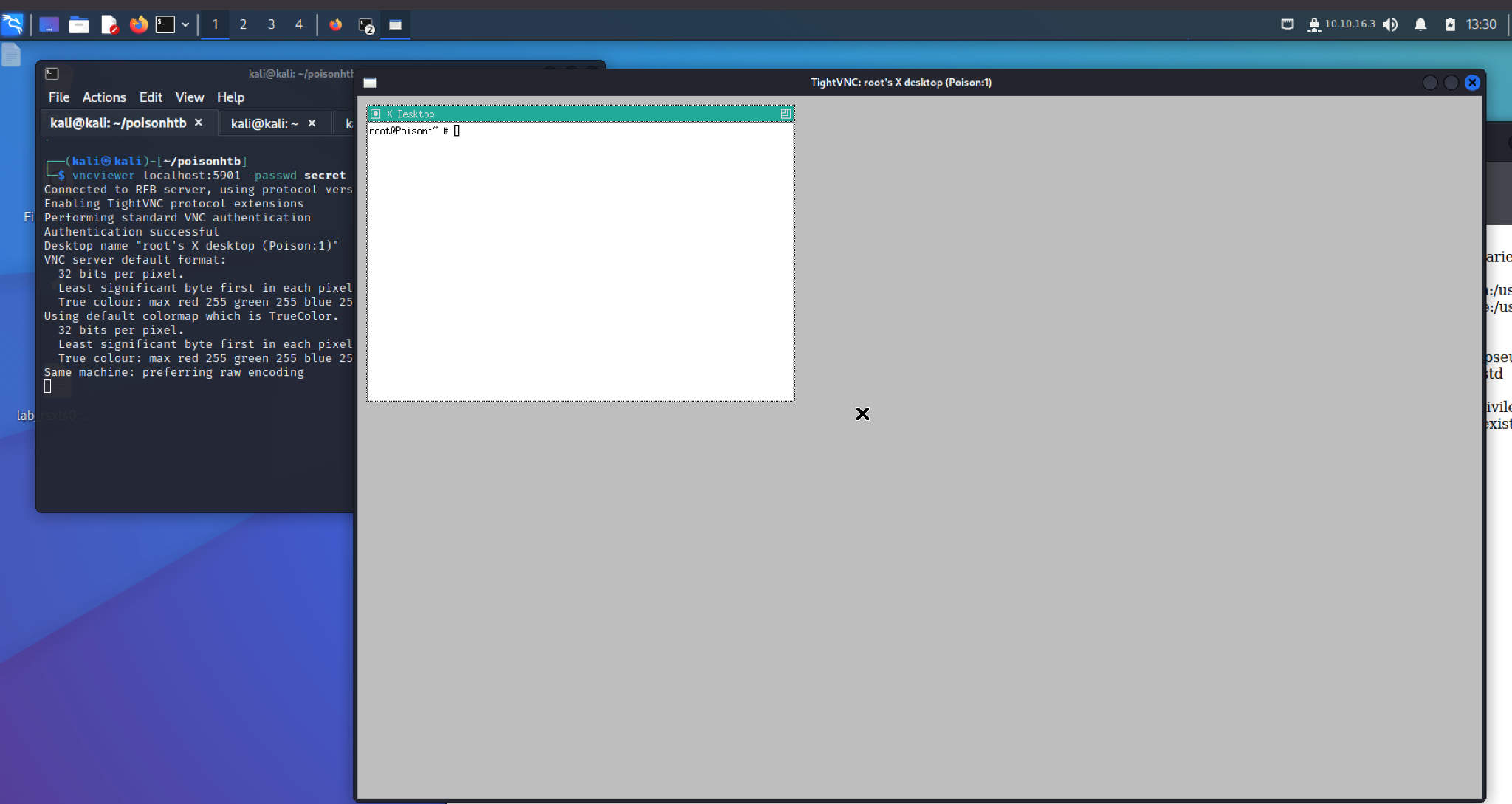
It appears that the “secret” file is in a non-readable format.





After finding that root has port 5901 open which is generally used for Virtual Network Computing (VNC). Run ‘vncviewer’ to allow root-access on the localhost of the Victim Machine through a graphical interface.

Knowing that the password was in the secret file, adding -passwd secret to the command, it allowed the file to be used to authenticate access in order to Escalate our Privileges to root.



**Root** access has now been established.

**BLUE TEAM REMEDIATION**

* **.**

**SSH Server**

* **Disable Root Login Over SSH Server:** This lowers the vulnerability of having the root password exposed and giving hackers access to all services in the SSH server. To disable the remote root login, it can be accomplished by opening the ssh\_config file. Then running “*sudo vim /etc/ssh/sshd\_config*” then setting **PermitRootLogin** to no.
* **Disable SSH Login Authentication with Password:** It is recommended to use ssh public key authentication for remote access to the server. This will better secure brute force attacks on the password. This can be done by setting the value of the “PasswordAuthenication” to no.
* **Limit User Login Access Using SSH:** By default, SSH server allows all users to access remotely. Changing this will allow who can access or be denied remote access. This can be done in **ssh\_config** and setting AllowUsers/DenyUsers rules.
* **Use Stronger Passwords:** Setting the minimum for all users to15 Characters, Number, Symbol, Uppercase/Lowercase will make brute force attacks much more difficult.

**.PHP Prevention**

* **Authentication access:** even for low level or base line web operations, when hosted on a main machine should require some level of password authentication before allowing input
* **Sanitize input fields within server:** the url, with no security, will act more as a command line than a website. Sanitizing the server will prevent unwanted execution and navigation of input in the server itself.
* **SSL (secure socket layer)**: the transfer of information from the webhost to the database is crucial and the encryption is just as important, in this case the SSL will help prevent those without proper authority to access important data
* **WAF (web application Firewall)**: the use of a firewall is crucial in many ways. Just as a general rule, this attack could have been almost entirely prevented with the use of a good firewall

**Firewalls**

* Block any http request (firewall-cmd - -permanent - -zone=public - -remove-service=http)

**Conclusion and Summary**

Poison, a HackTheBox CTF Target-Practice VM, introduced new and interesting concepts of exploitation, including using SCP to download files from Victim Machine, incorporating VNC Viewer to escalate to root, and the exploitation shows examples of possible threats and vulnerabilities there could be.