 TryHackMe - Startup CTF Write-up

by Rakib Nadir

Introduction This is a beginner-level CTF challenge from TryHackMe called Startup. It is designed to test skills in enumeration, exploitation, and privilege escalation. The challenge revolves around a misconfigured FTP service that allows anonymous access, leading to a reverse shell exploitation and, ultimately, root access through a scheduled cron job.

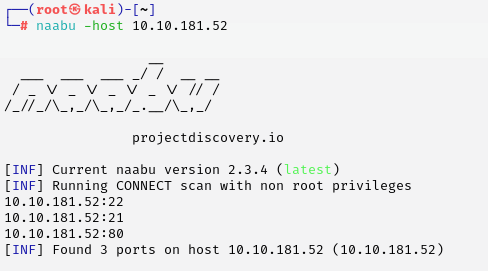
Challenge Questions:

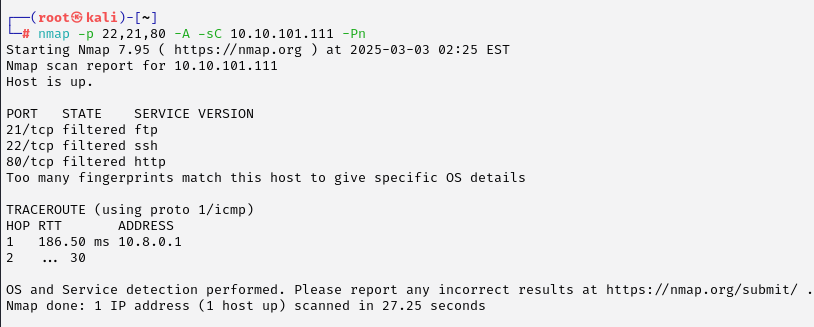
1. What is the secret spicy soup recipe
2. What are the contents of user.txt
3. What are the contents of root.txt

**Enumeration and Initial Access**

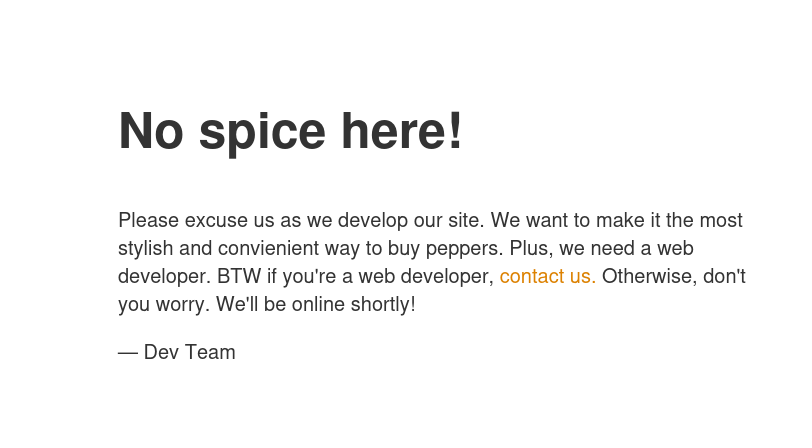
Scanning for Open Ports: After obtaining the target IP address, I scanned for running services using my favorite tool Naabu. It revealed three open ports. To get more details, I ran Nmap, which confirmed the presence of:

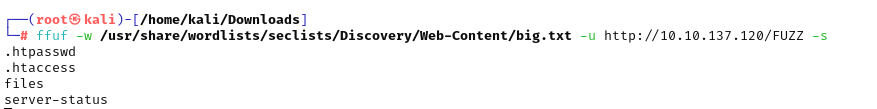
* FTP
* SSH
* HTTP

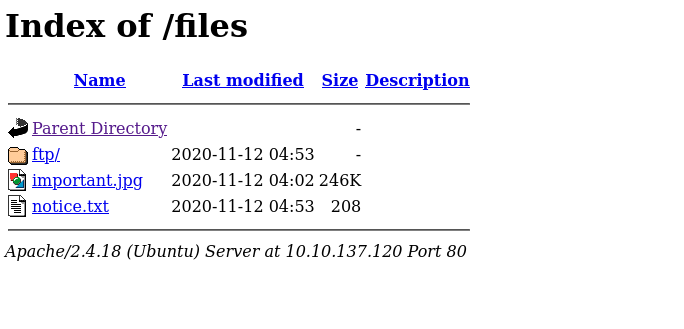




Exploring HTTP : I visited the web page hosted on port 80 but didn’t find anything interesting.

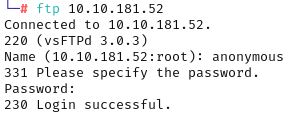
 To dig deeper, I performed directory brute-forcing using FFUF and discovered endpoint files.



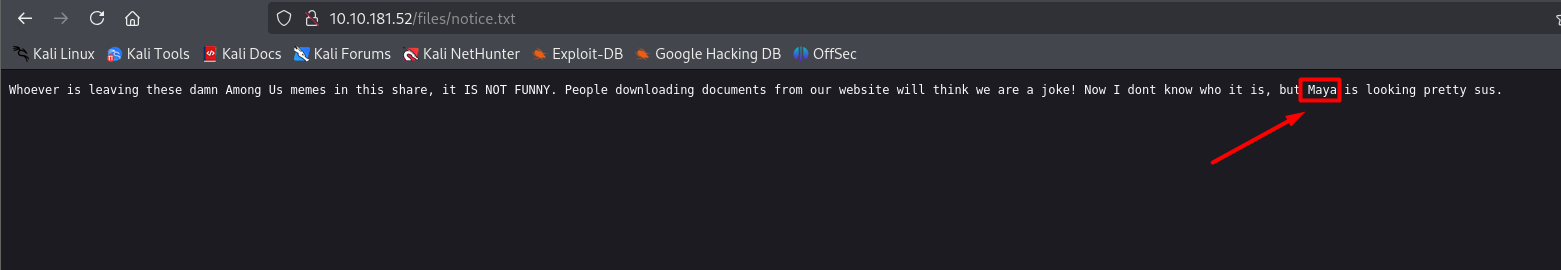
then i went to the url and found this 

But I suddenly realized that I could do that simply by playing with the ftp server. So I went directly to play with the ftp server of the system.

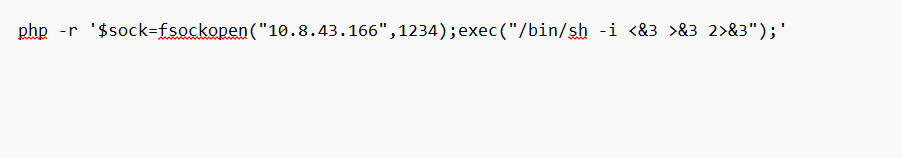
**Exploiting the FTP Service**: Instead of manually checking files, I decided to interact directly with the FTP server. Since anonymous login was enabled, I connected and browsed the available files.



Upon navigating to files, I found some files, but they didn’t seem useful at first glance. However, I noted a name mentioned in the files Maya.

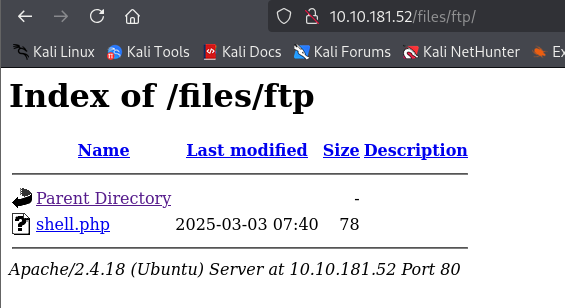


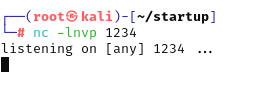
At this point, I had an idea What if I upload a reverse shell through FTP and execute it





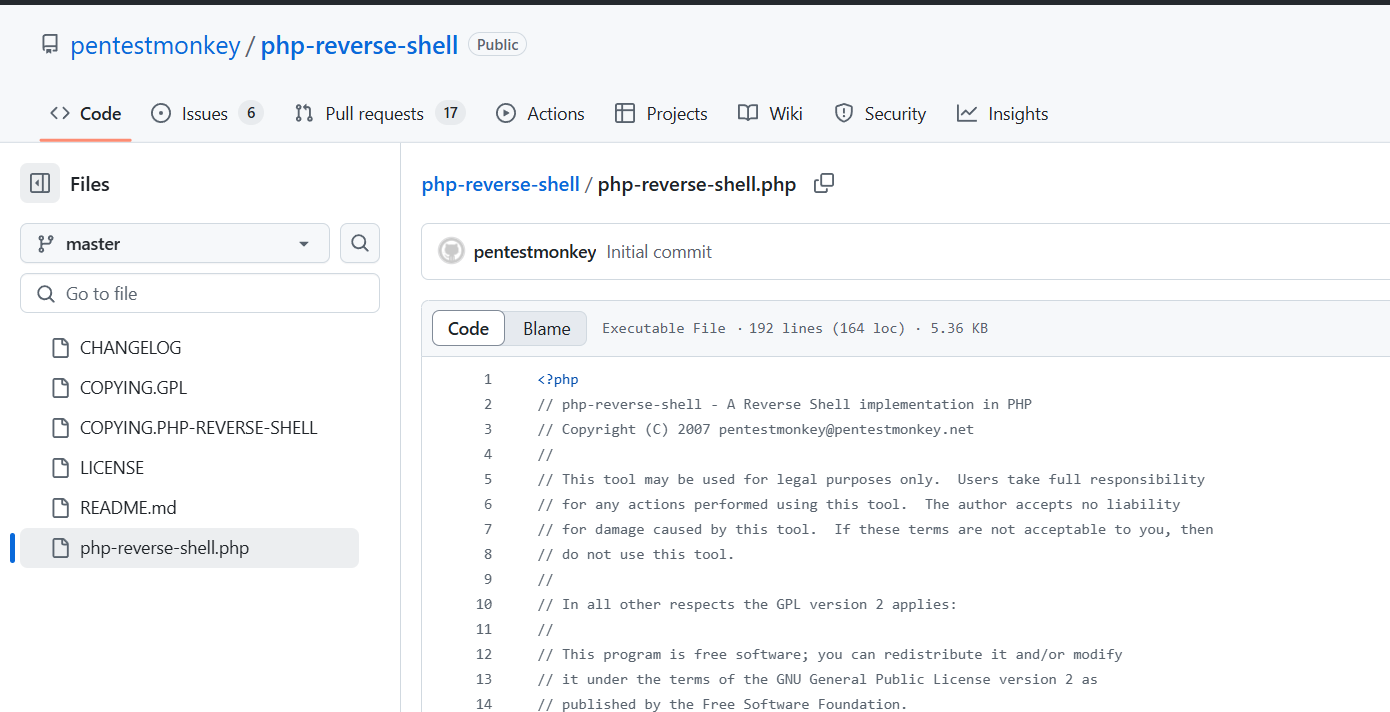
I set a listener and tried to execute it

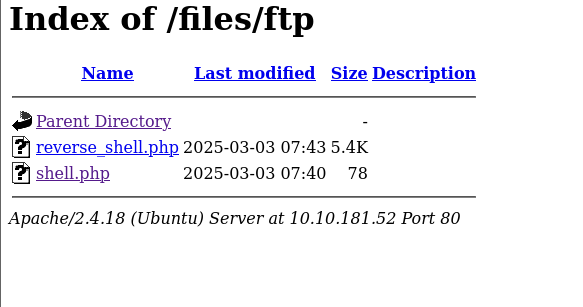


****

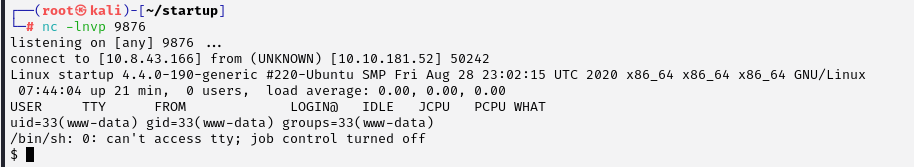
**But that did not work**

**I attempted to upload a PentestMonkey reverse shell but couldn't execute it. So, I tried a PHP reverse shell instead.**





**boom!! Our very desiring initial access**



From there, I found the secret spicy soup recipe, answering the first challenge question.

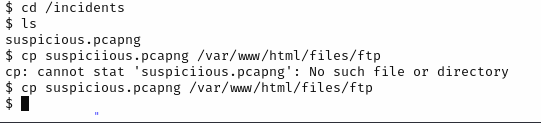


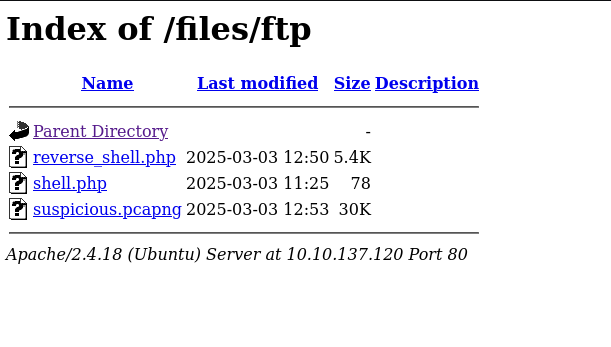
**Privilege Escalation to User**

Finding User Credentials I navigated to the home directory and found a user named lennie, but I couldn’t access their flag without their credentials. To escalate privileges, I transferred Linpeas.sh via a Python HTTP server and executed it to look for potential weaknesses.

After running linpeas.sh i found something,

Linpeas highlighted an interesting binary. I transferred it back to my machine through FTP and analyzed it using the strings command.

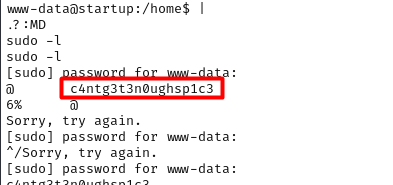




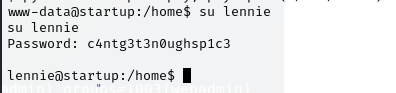
Now I can simply download it . I use the ‘strings’ command to see the file in human readable format



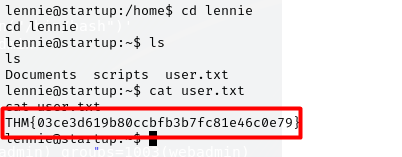
Guess what I found a password!!



I used the password for the user lennie and it worked!!!!!!!!!!!!!!!!



Then i found the user.txt

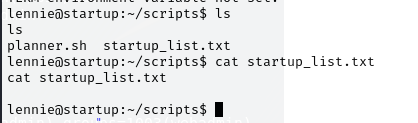


Now the Third task , Lets find how to be the root!!

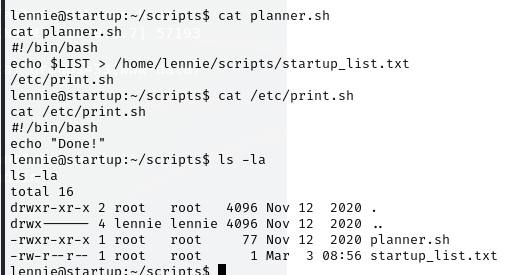
**Privilege Escalation to Root**

Identifying Root Privilege Escalation Vector Inside the scripts directory, I found two files:

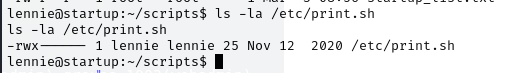
* planner.sh
* print.sh



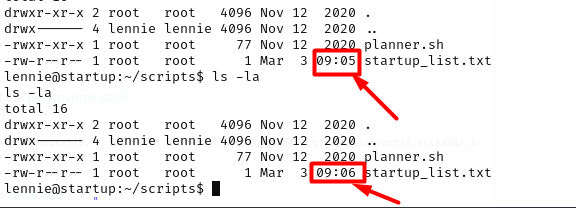
The planner.sh script was executing etc/print.sh, and it was running as root. This looked promising



I then checked print.sh and noticed that it had write permissions, meaning I could modify it.



I got stuck again until i saw this

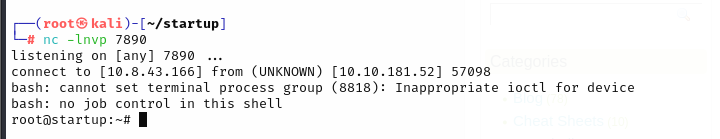


Something is updating the file every minute. As the print.sh file was being run by root I could get root access if the file updates as well . so i echo a reverse shell to the /etc/print.sh file

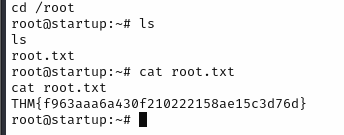
Since print.sh was being executed as root, I replaced its contents with a reverse shell payload and set up a Netcat listener.



and set up a listener to see what happens after a minutes.



**ROOOOOT**!!!



**Lessons Learned**: Always check write access on FTP services Look for automatically running scripts they could be part of a cron job Use Linpeas to quickly find privilege escalation vectors If stuck, monitor files for modifications over time

Things I Struggled With Privilege escalation through cron jobs took me a long time to figure out. I didn’t immediately notice that print.sh was updating every minute.

**Conclusion** :This was a great beginner-friendly CTF that reinforced enumeration, exploitation, and privilege escalation techniques. It also highlighted the importance of not overcomplicating things and focusing on basic misconfigurations first.

Thank you everyone