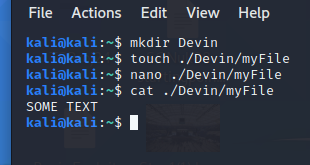
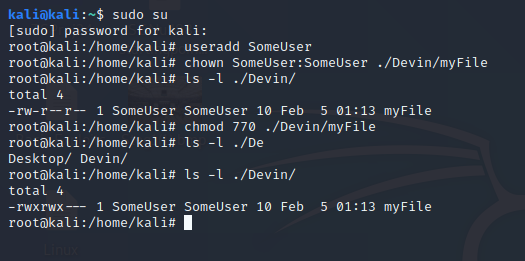
Using the instructions in the lesson/lecture in the Module, accomplish the following tasks. Use your Kali VM to do this. Submit a report including the commands/screenshots, and your script. It is okay to research online but please do not submit copy and pasted code from online.

**Task 1: In your Kali VM (25 points)**

1. Create folder titled as your name.
   * Create a file inside this folder
   * Add text to the file

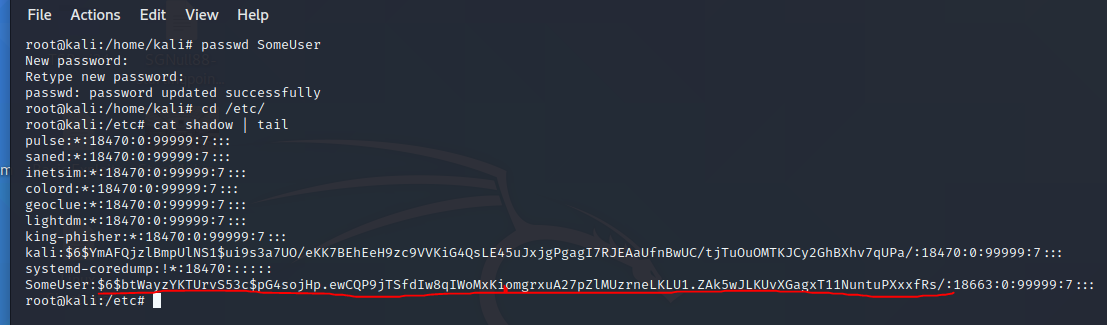


1. Create a user.
   * Make that user the owner of that file in part 1.
   * Change the file permissions to user and the group have full rights.

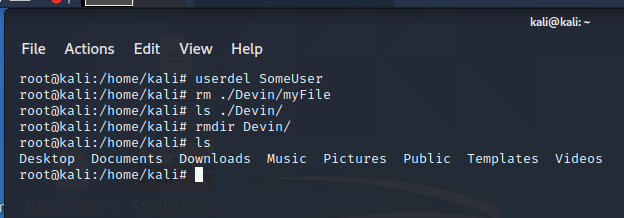


* + What is the hash value of the user’s password you created?
  + What does the $6$ at the beginning of the hashed password means?

This means it is a SHA-512 hash



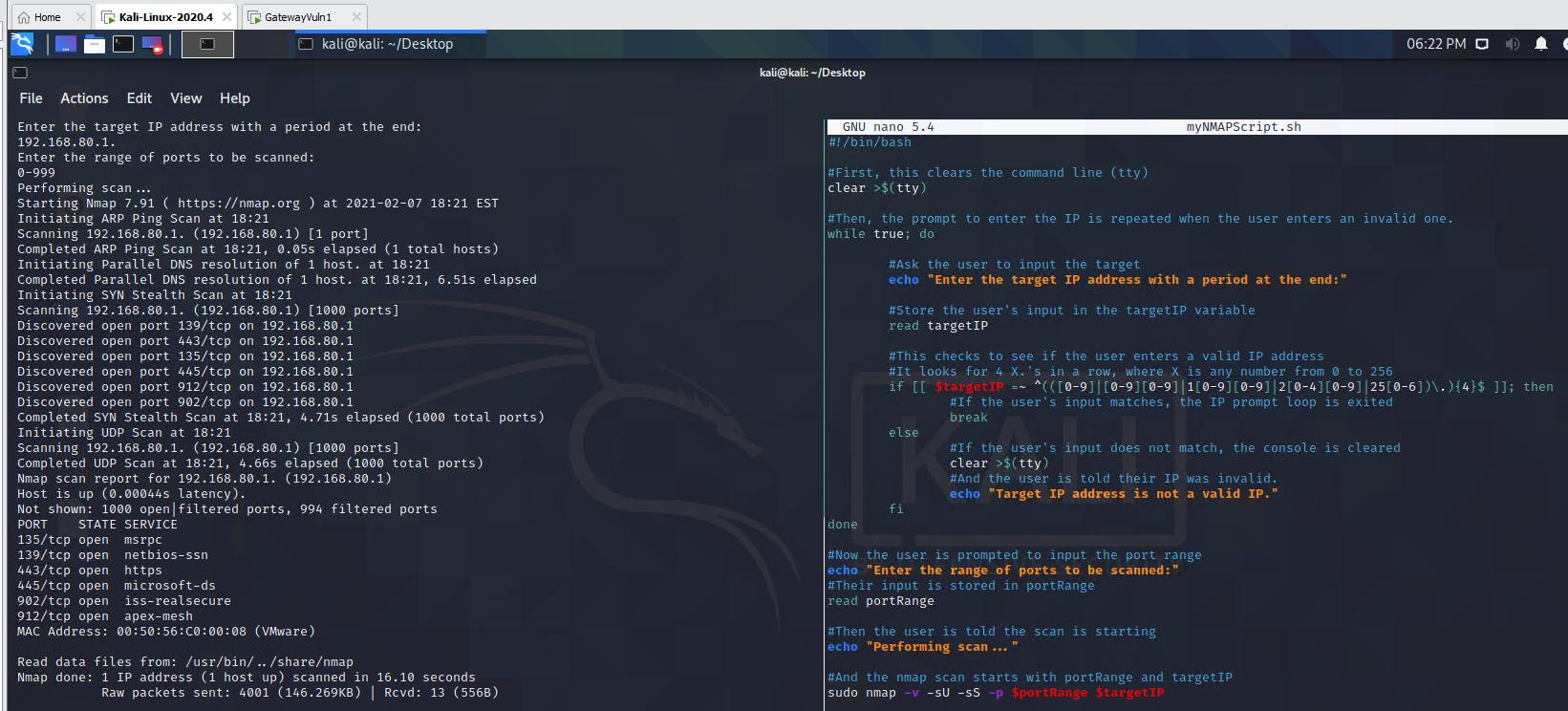
1. Remove the user.
2. Remove the file

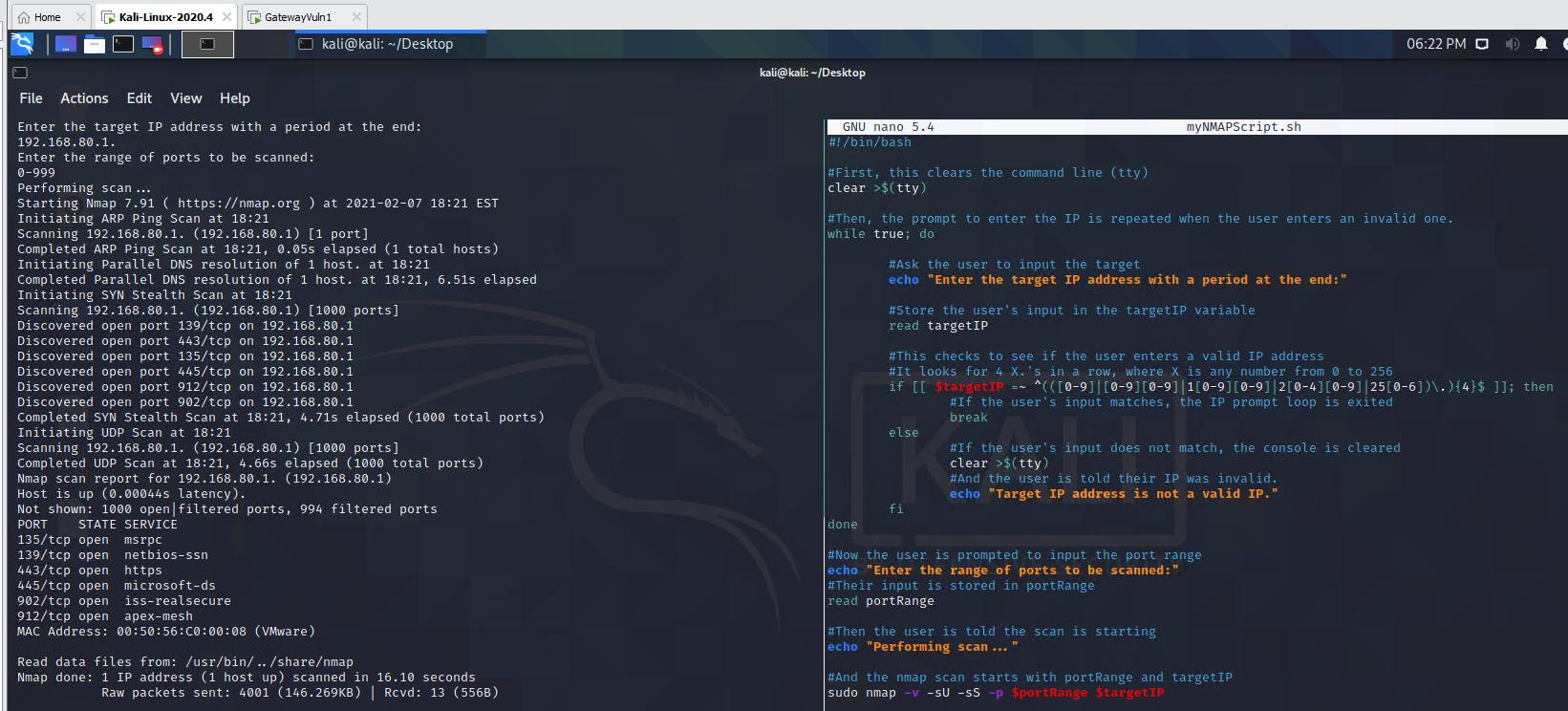


**Task 2: Creating BASH Script for Port Scanning (75 points)**

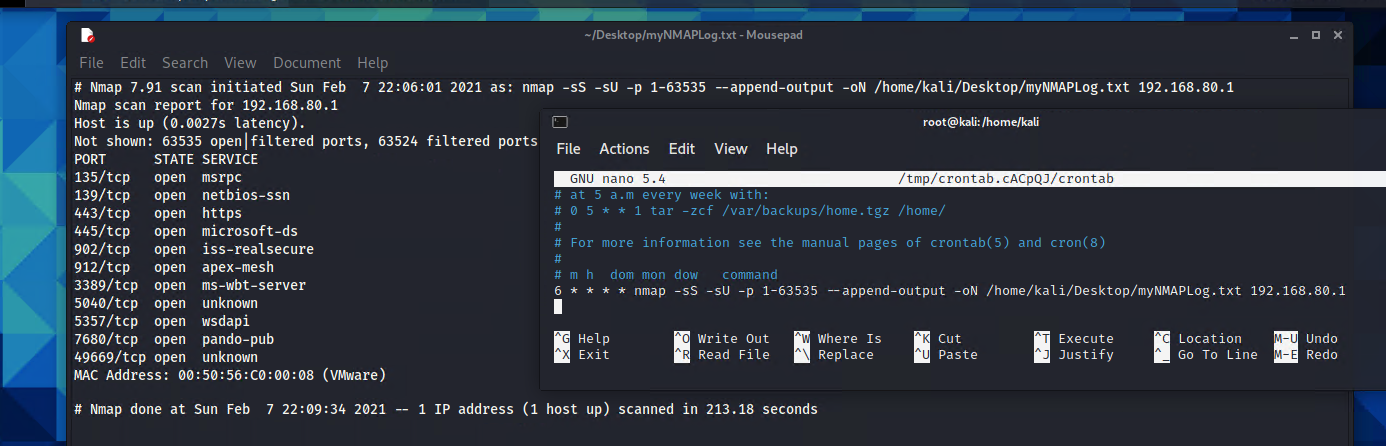
The optional reading in the module can help you on this assignment. You need to use nmap command in your script. [Here is a quick guide](https://nmap.org/book/port-scanning-tutorial.html) to understand nmap and multiple switches you can use. You can also use [netcat](https://www.cyberciti.biz/faq/linux-port-scanning/) **nc** command.

1. Create a Bash script to scan the GatewayVul virtual machine in your lab computer. (make sure this VM is up and running in VMware in the Lab)
2. Your script should include the following.
   1. Ask the user for an IP address. (by now you should know how to find the IP of the GatewayVuln machine)
   2. Verify if it is a valid IP address
   3. Ask the user for a range of ports to be scanned. Example port 1-63535. You can pass the input to a nmap using **$** sign.
   4. Do a UDP and a TCP scan. You can use nmap or nc as a tool to scan inside your script or write your own socket program.
   5. Include your final script and screenshot of your results from the scan. Make sure to explain each line of code. Not commenting will deduct points for each line.





1. Schedule your script to run every day at midnight using a cronjob. (ps: use a time that you can test and then change it to noon). Show your work.

Cronjobs are only ran in the background, so the user interaction portion will not work and the program wouldn’t do anything. However, I can set up a cronjob to run the scan and output the result to a log. Using --append-output and -oN, I was able to send the results to a file on my desktop and store the results for each time it ran.

I just changed the time to 0 0 \* \* \* instead of 6 \* \* \* \* so that it would run at midnight.