Joshua Lane
CYSE450 Section 23190
12/01/2022

Laboratory Exercise L4 – Bash Scripting: Robots

1. Overview

In this lab exercise, students will learn how to create a Bash script that takes a domain's robots.txt file, parses the path locations, then loops the site paths and opens each in a new Firefox browser tab.

2. Resources required

This exercise requires the Kali Linux with Metasploitable (2020.09) running in the Cyber Range.

3. Initial Setup

For this exercise, you will log in to your Cyber Range account, select the Kali Linux with Metasploitable (2020.09) environment, click "start" to start your environment, and then "join" to get to your Linux desktop.

4. Tasks

Task 1: Scripting Web Reconnaissance

- Open a text editor and set up the Bash shebang as shown in the screenshot below.
- On line 3, add a command to change the directory to our scripts folder.
- Name the script as **robots.sh** and save it to the scripts folder.

```
1 #! /bin/bash
2
3 cd /home/student/Desktop/scripts
4
```

- In the terminal, cd to the scripts folder.
- Change **robots.sh** to an executable file.

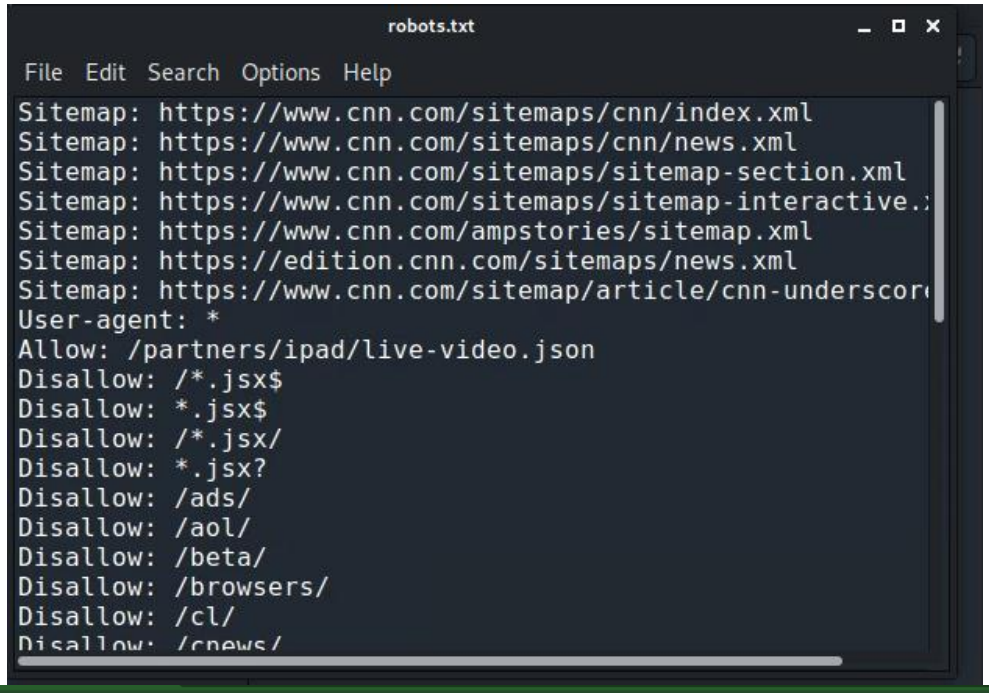
```
student@kali:/$ cd /home/student/Desktop/scripts/
student@kali:~/Desktop/scripts$ sudo chmod +x robots.sh
```

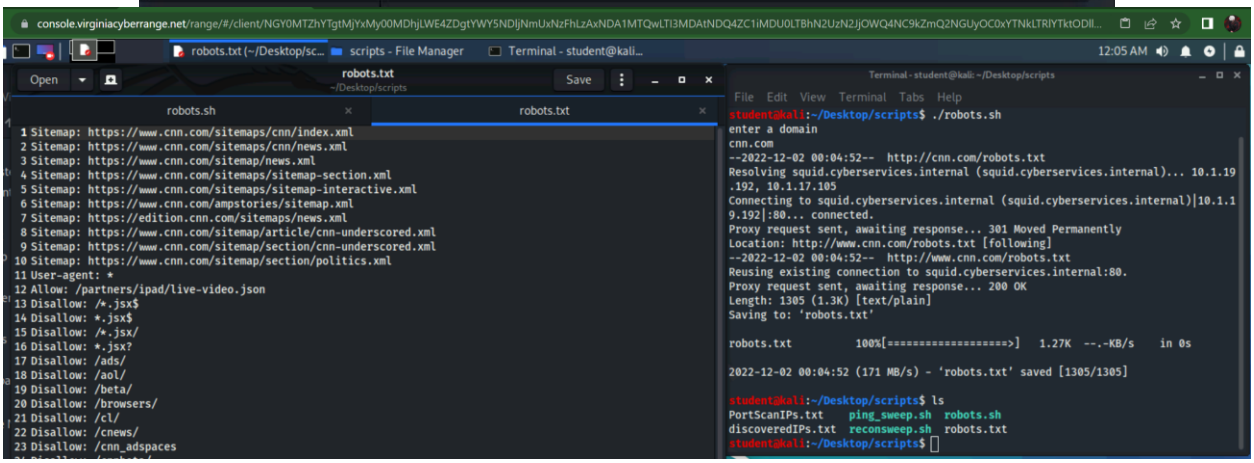
What we want to do is enter a domain and have the script check the robots.txt directory of that domain for directories that are denied web crawler access. For the script, we will save those denied directories to a .txt file.

- In the text editor, add the lines shown in the screenshot below.
- Save the file.
- Run the script as intended. Try using cnn.com as I have in the below screenshot.
- Open the robots.txt file (in the scripts folder) and review the information.

[NOTE: If you copy anything outside of the Range and then paste it into a .sh script, you will have to run the **dos2unix** command on the script as completed in the past.]

```
1 #!/bin/bash
2
3 cd /home/student/Desktop/scripts
4
5 echo "enter a domain"
6 read domain
7
8 wget $domain/robots.txt
```



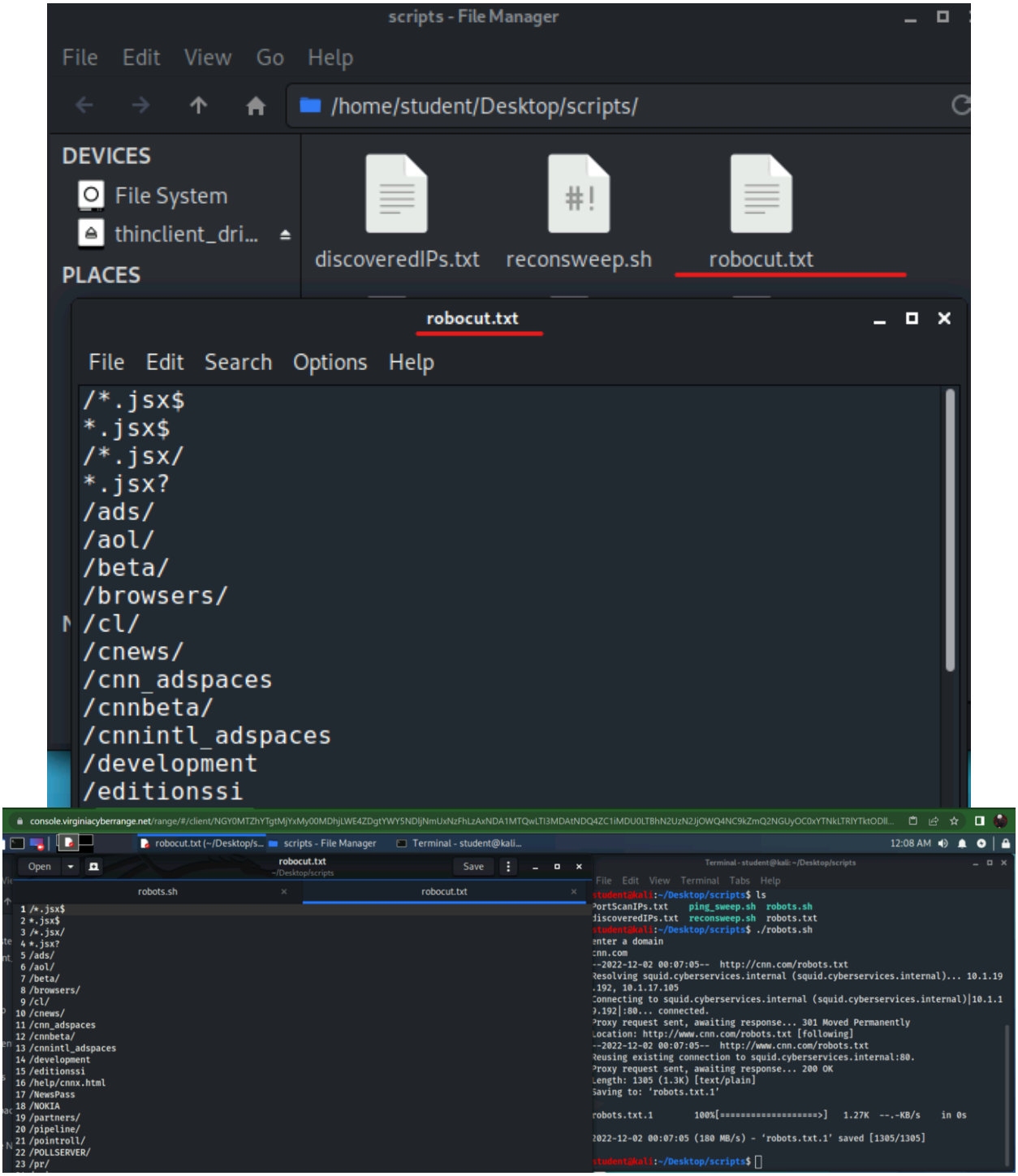


Here is the robots.txt file and me running the script on the right.

Now we want to cat the text and use the cut command to remove the fields in the robots.txt file that we do not want.

- On line 10, type **`cat robots.txt | grep 'Disallow' | cut -d ' ' -f2 > robocut.txt`** and save the file. Run the script.

- Open the **robocut.txt** file and review the information.



Here is my robocut.txt and the script run on the right.

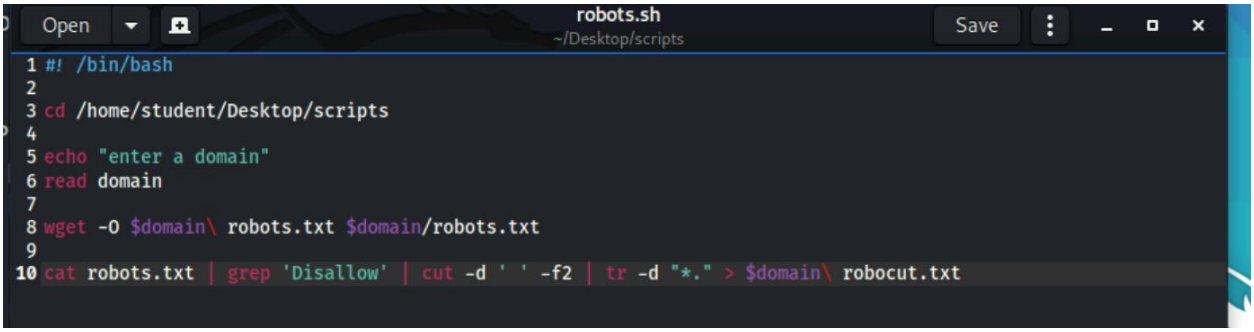
We know the /*. locations will not work, so we can delete characters using the **tr** command.

- On line 10, add **`| tr -d "*."`** after the **`f2 and`** before the **`>`**.
- Save the file.

If we were to use this script as is, we would save over each domain that we previously ran the script on. To prevent this, we are going to add a little variable magic and change the file name for each domain.

- Change line 8 to **`wget -O $domain\ robots.txt $domain/robots.txt`**
- On line 10 after the **`>`**, add **`$domain\ robocut.txt`**
- Save the file.

The **`-O`** allows us to save to a file. The **`\`** after **`$domain`** tells Bash to delete the space when saving the file. We need this because, without the space, Bash will include the robots.txt as a part of the variable. The space separates the variable from the filename. We mirror this technique with the **cat** command on line 10.



Here is my updated script.

Task 2: Looping through website paths

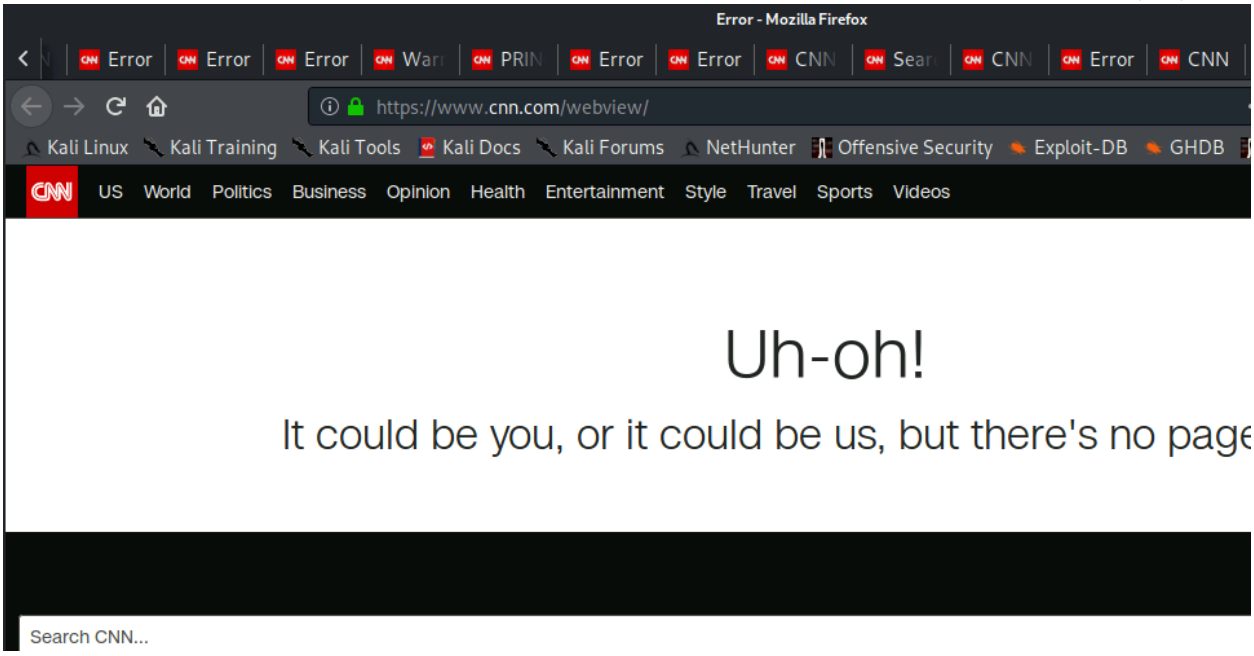
For this task, we want to set our script to open a Firefox tab for each page listed in the **<domain>robocut.txt** file.

Look at the screenshot below to complete the syntax and add it to your **robots.sh** script. Notice the same technique is used to **cat** the file and create the loop. We also **sleep** for 5 seconds to give the VM enough time to open Firefox.

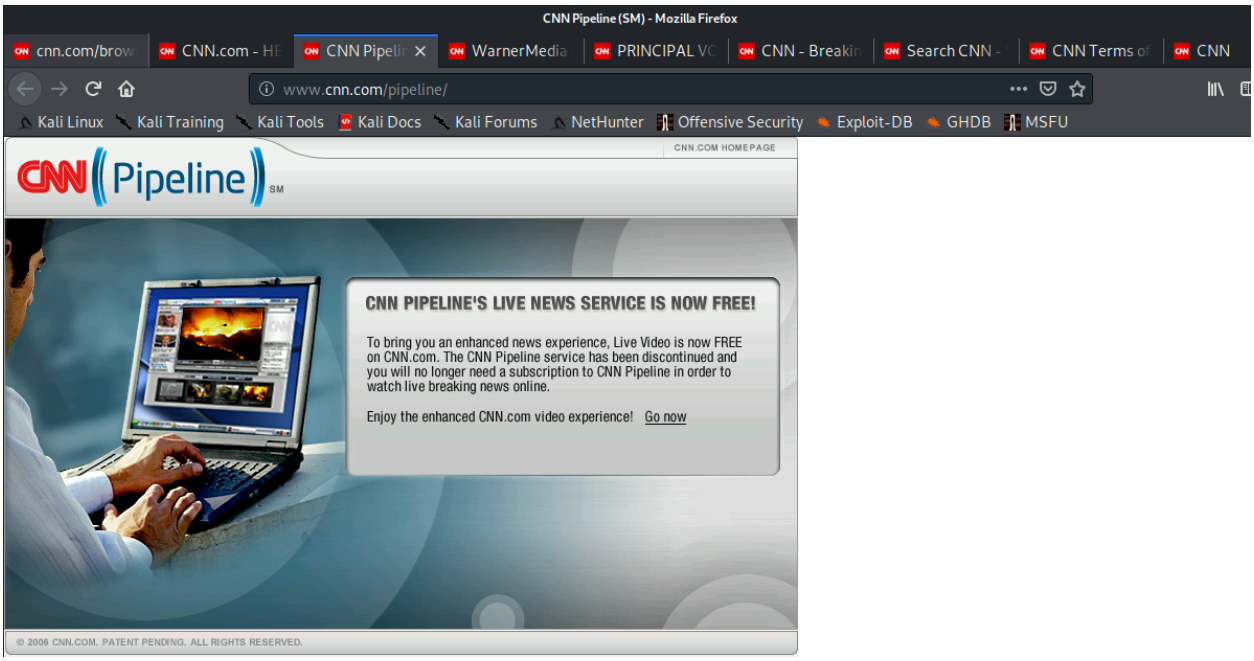
```
#!/bin/bash

cd /home/student/Desktop/scripts

echo "enter a domain"
read domain

wget -O $domain\ robots.txt $domain/robots.txt

cat robots.txt | grep 'Disallow' | cut -d ' ' -f2 | tr -d "*."  > $domain\ robocut.txt

firefox &
sleep 5

for i in $(cat $domain\ robocut.txt); do
        firefox -new-tab https://www.$domain$i &
        sleep 2
done
```

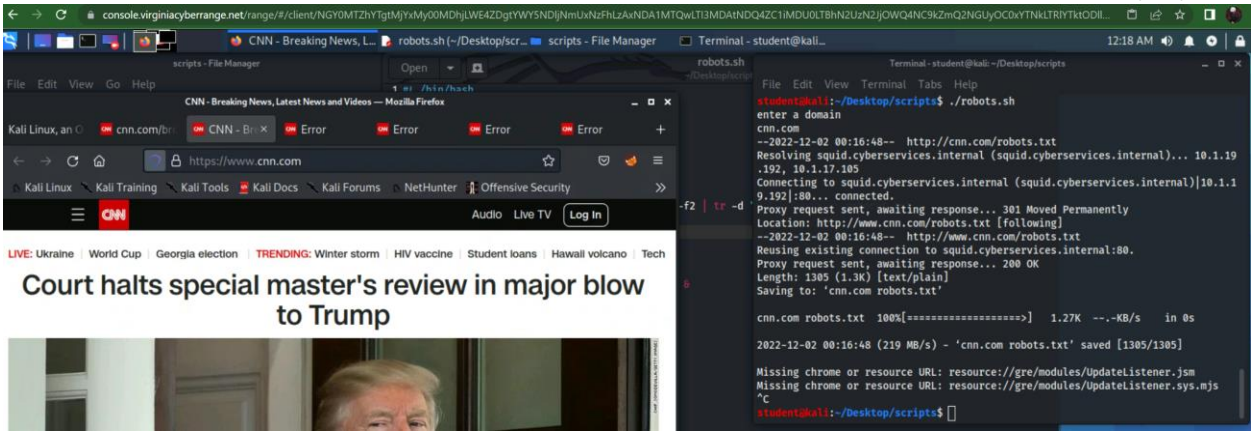
- Save the file.
- Run the script as intended for the domain **cnn.com**.

```
student@kali:~/Desktop/scripts$ ./robots.sh
enter a domain
cnn.com
--2021-11-16 18:56:05--  http://cnn.com/robots.txt
Resolving squid.cyberservices.internal (squid.cyberservices.internal)... 10.1.19
.230, 10.1.17.212
Connecting to squid.cyberservices.internal (squid.cyberservices.internal)|10.1.1
9.230|:80... connected.
Proxy request sent, awaiting response... 301 Moved Permanently
Location: http://www.cnn.com/robots.txt [following]
```



Several of the locations will show a pre-created error page; we will ignore these. Exit out of all of the tabs with errors. What you are left with are the pages that can be accessed.





Here is my screenshot for running the script and finding accessible tabs.

When on an engagement, you will be looking for cisco and other login pages that could give internal access. The following is an example:

In this lab exercise, we learned how to create a Bash script that takes a domain's robots.txt file, parses the path locations, loops the site paths and opens each in a new Firefox browser tab.