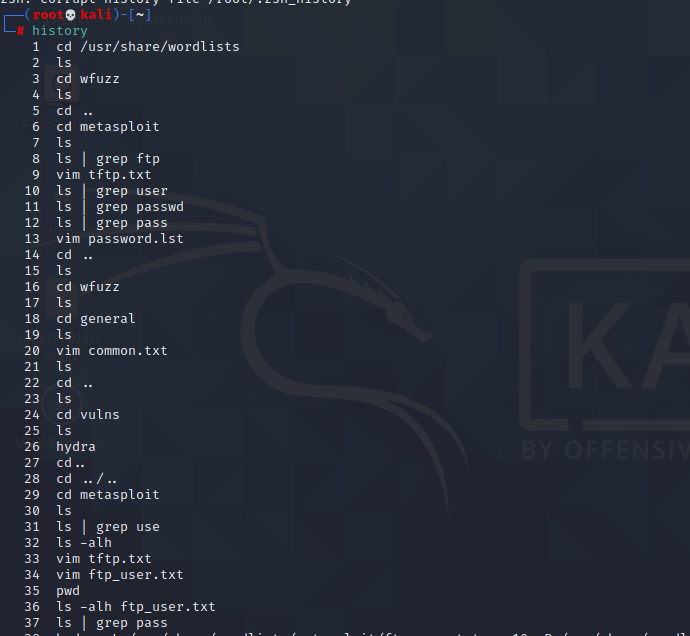
**chapter 3**

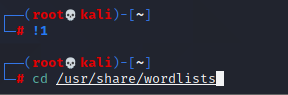
**3.1.3.2 Exercises**

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

use history

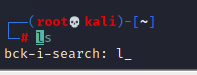


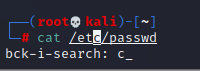
and select command

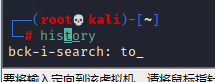


2. Execute different commands of your choice and experiment browsing the history through

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



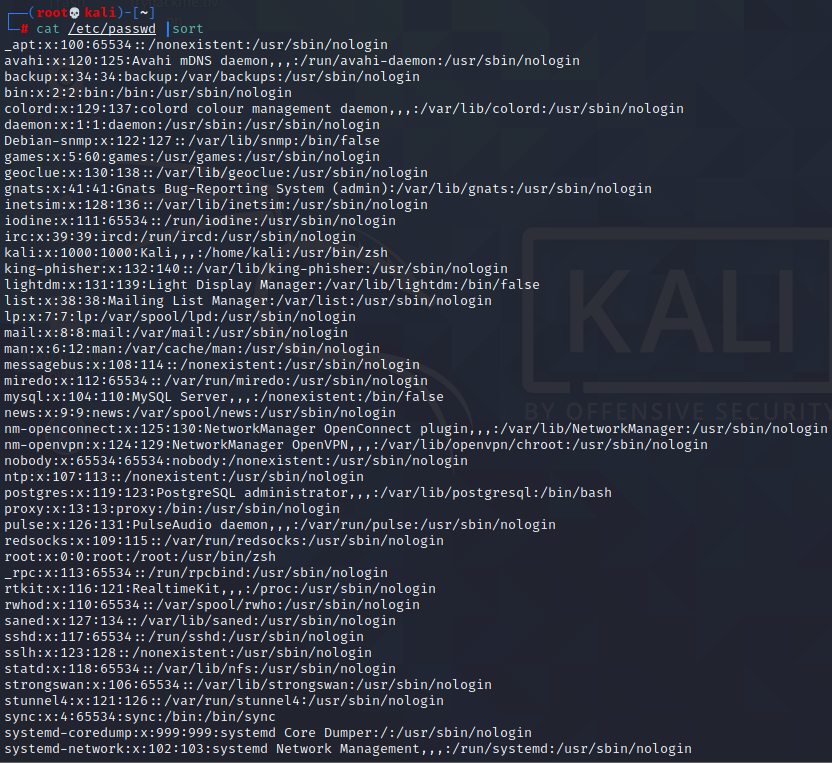




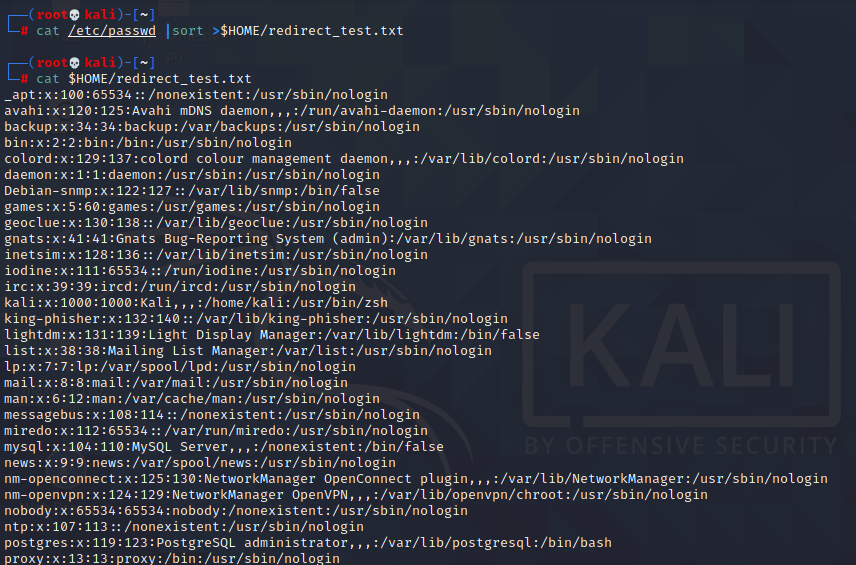
**3.2.5.1 Exercises**

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...

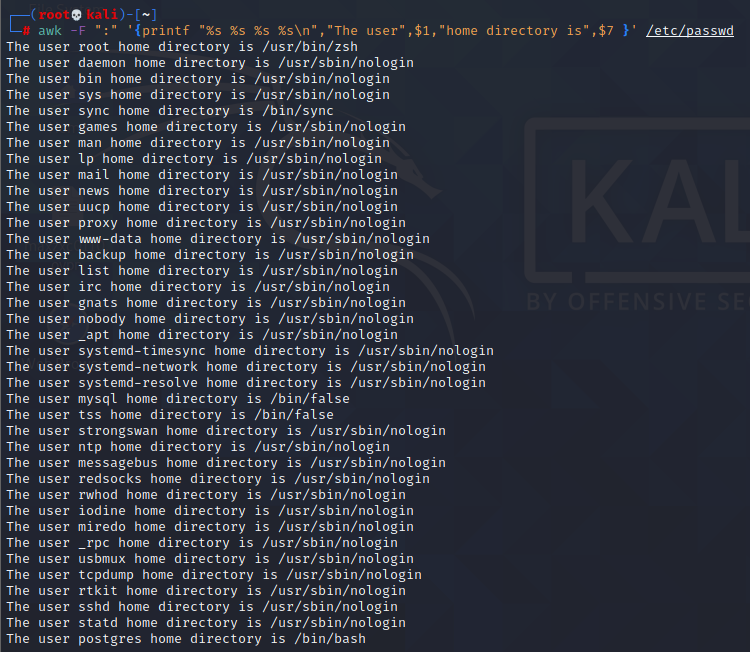
The user mysql home directory is /nonexistent

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

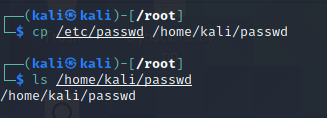
The user speech-dispatcher home directory is /var/run/speech-dispatcher

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

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

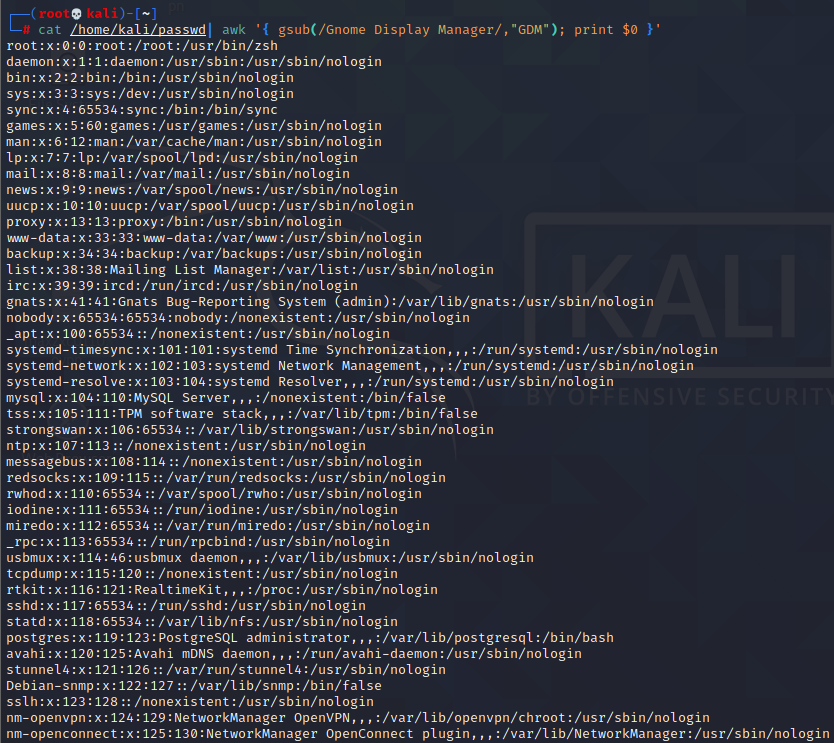


1. Copy the /etc/passwd file to your home directory ( /home/kali ).



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

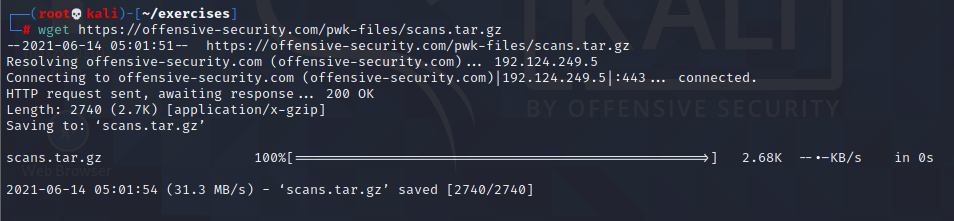
“Gnome Display Manager” string with “GDM”.



**3.5.3.1 Exercises**

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

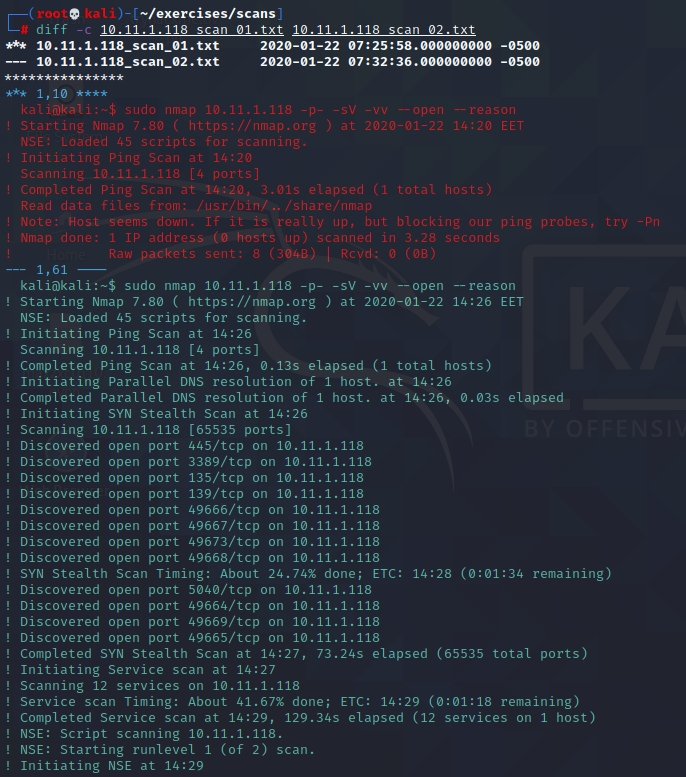
files/scans.tar.gz



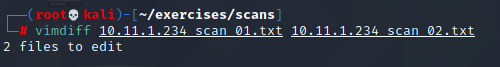
2. This archive contains the results of scanning the same target machine at different times.

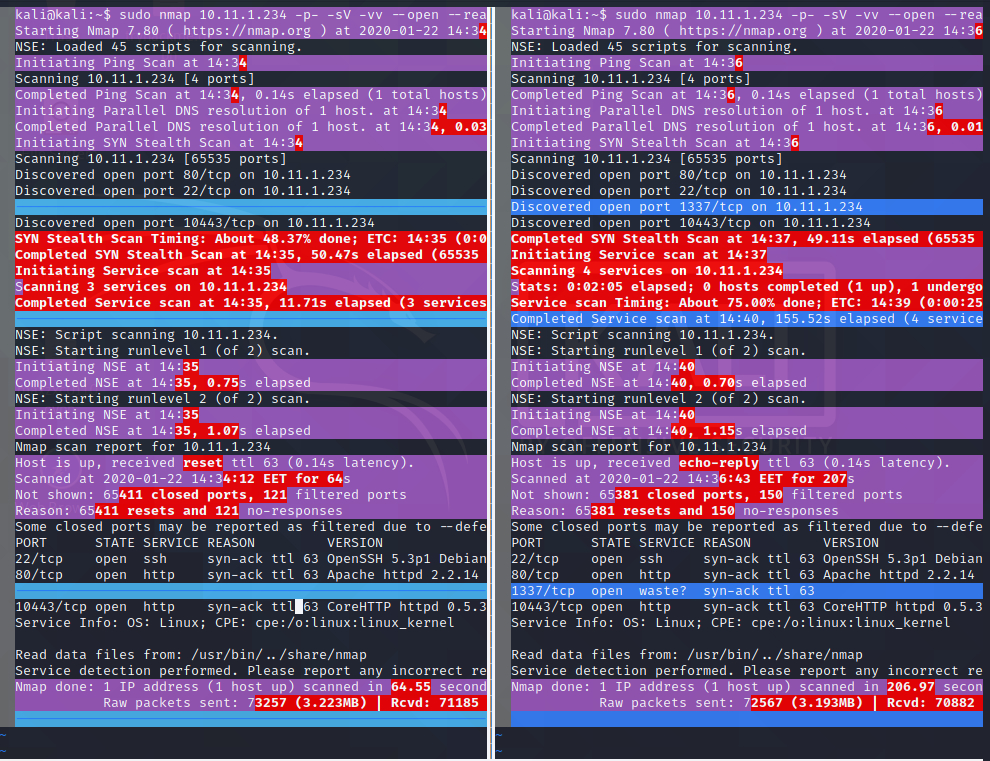
Extract the archive and see if you can spot the differences by diffing the scans.

use diff

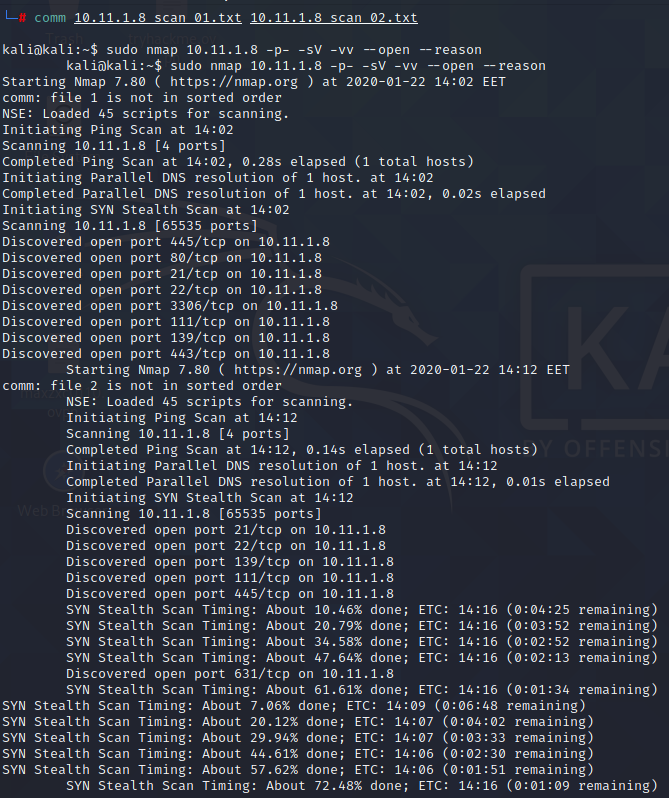


use vimdiff





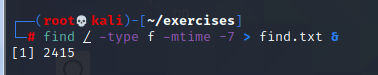
use comm



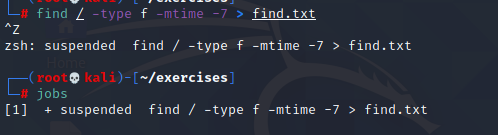
**3.6.3.1 Exercises**

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

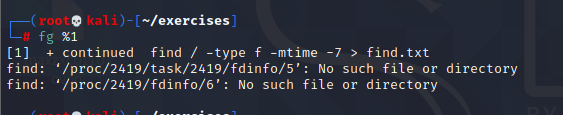
specific command in the background.



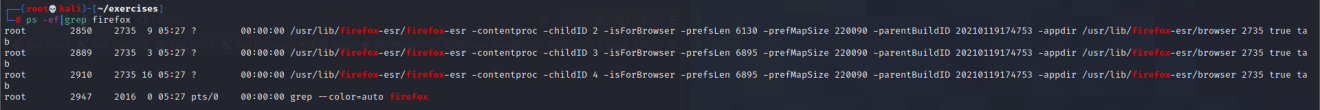
1. Re-run the previous command and suspend it; once suspended, background it.



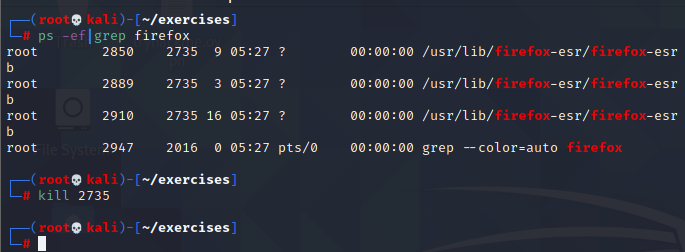
1. Bring the previous background job into the foreground.



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



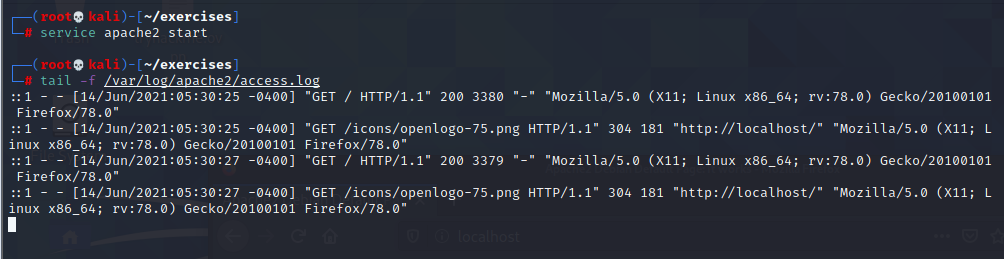
1. Terminate Firefox from the command line using its PID.



**3.7.2.1 Exercises**

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

real-time.

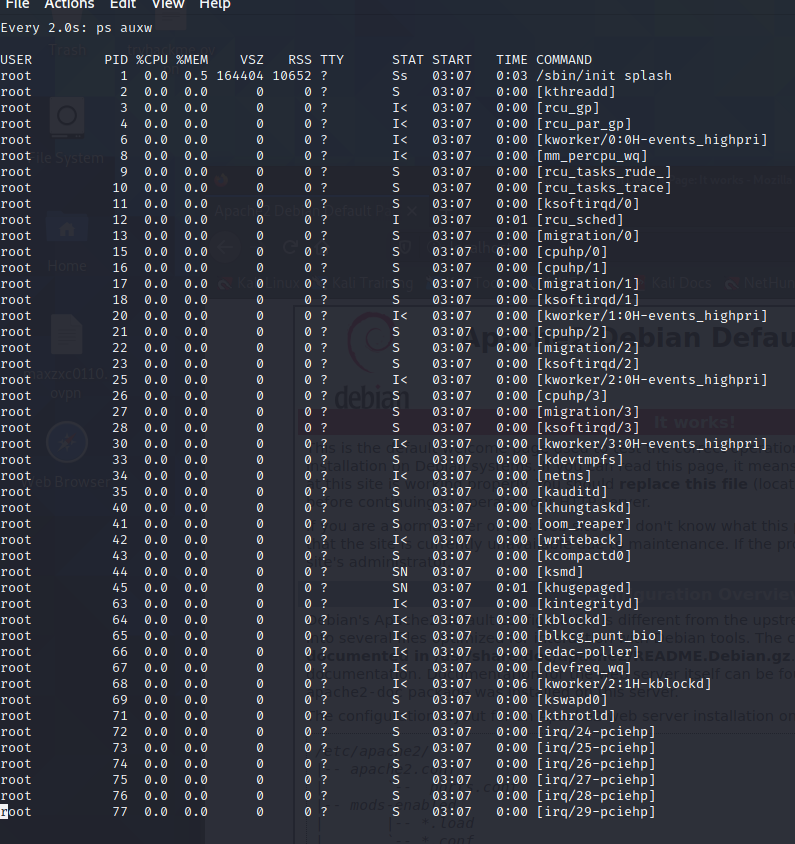


2. Use a combination of watch and ps to monitor the most CPU-intensive processes on your

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

in real time.



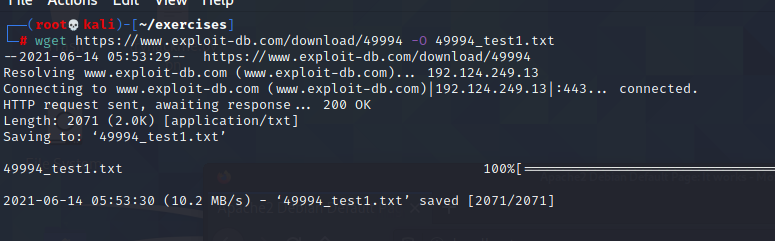


**3.8.3.1 Exercise**

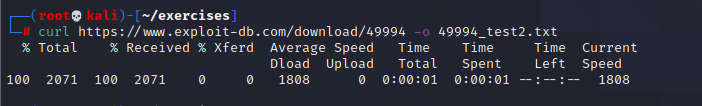
1. Download the PoC code for an exploit from https://www.exploit-db.com using curl , wget ,

and axel , saving each download with a different name.

wget



curl



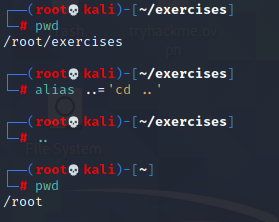
axel



**3.9.3.1 Exercises**

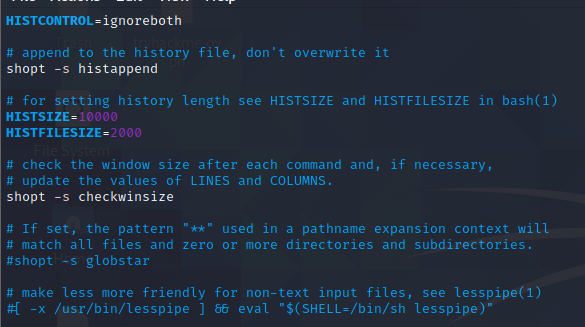
1. Create an alias named “..” to change to the parent directory and make it persistent across

terminal sessions.



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

in its output.

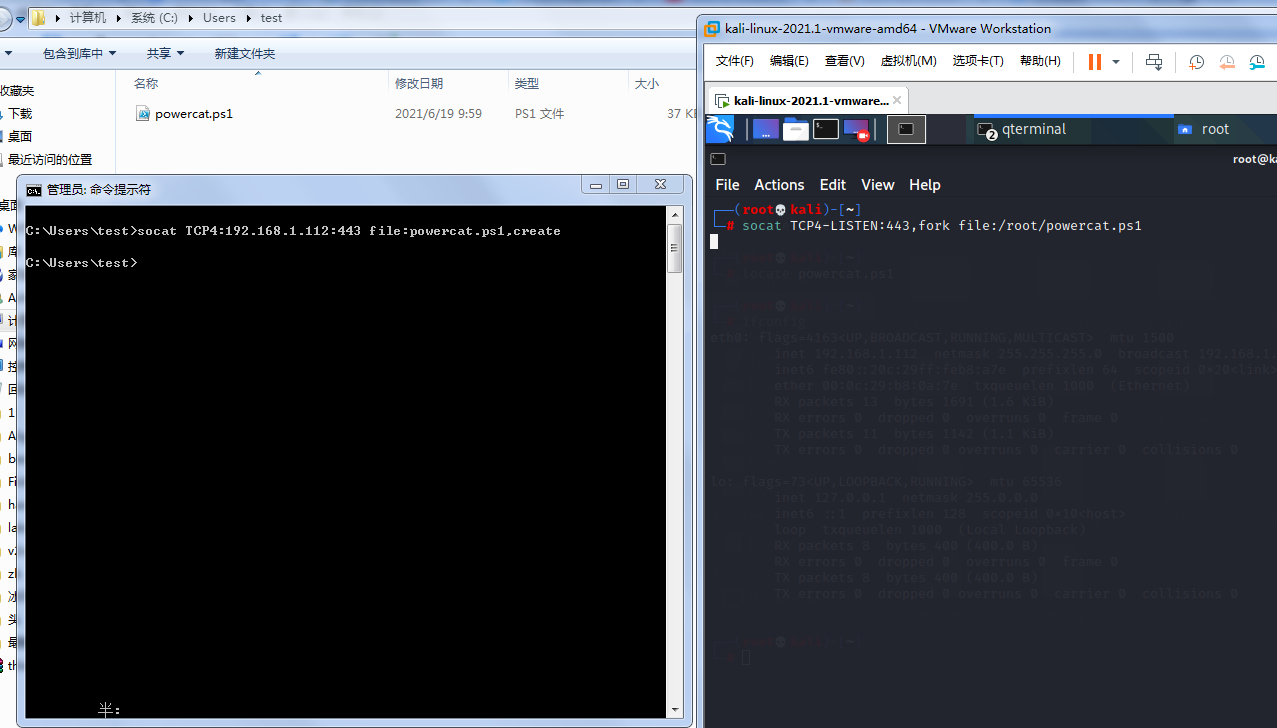


chapter 4

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

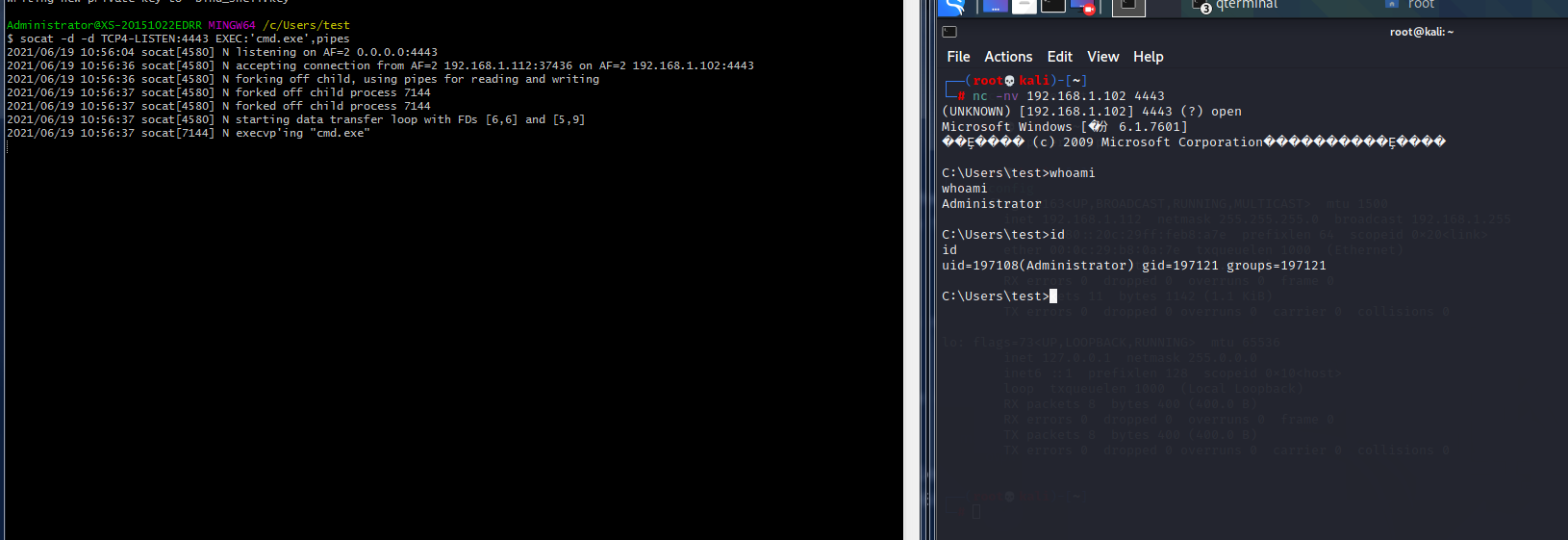
Netcat. Does it work?

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

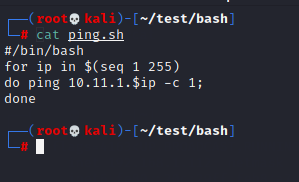
PSH packets.

chapter 5

5.7.3.1 Exercises

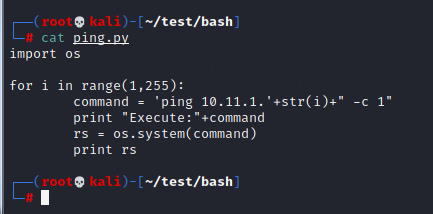
1. Research Bash loops and write a short script to perform a ping sweep of your target IP range

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

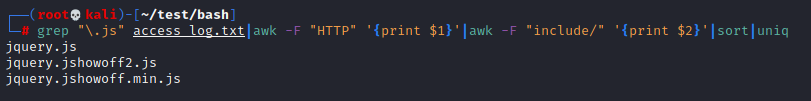
JavaScript files from the access\_log.txt file (http://www.offensive-security.com/pwk-

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

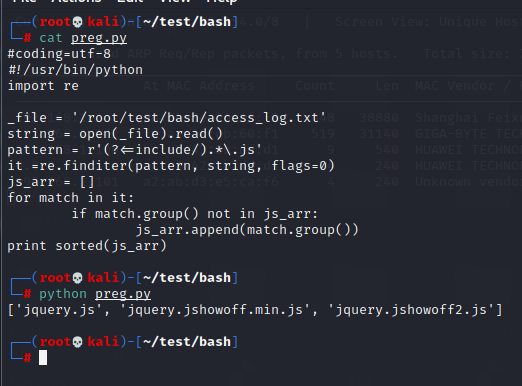
are sorted.

command:

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



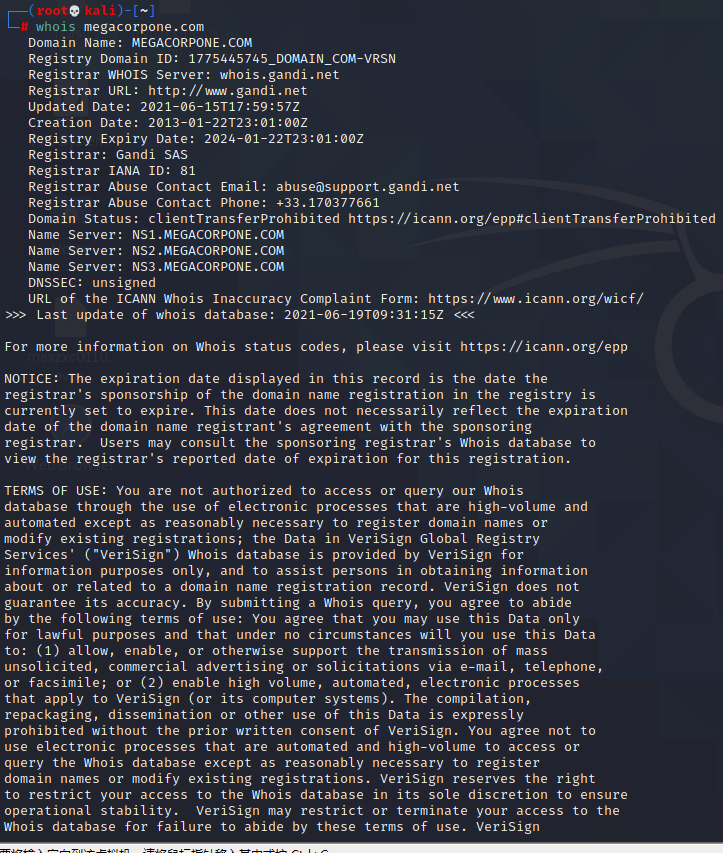
1. Re-write the previous exercise in another language such as Python, Perl, or Ruby.



chapter 6

6.3.1.1 Exercise

1. Use the whois tool in Kali to identify the name servers of MegaCorp One.



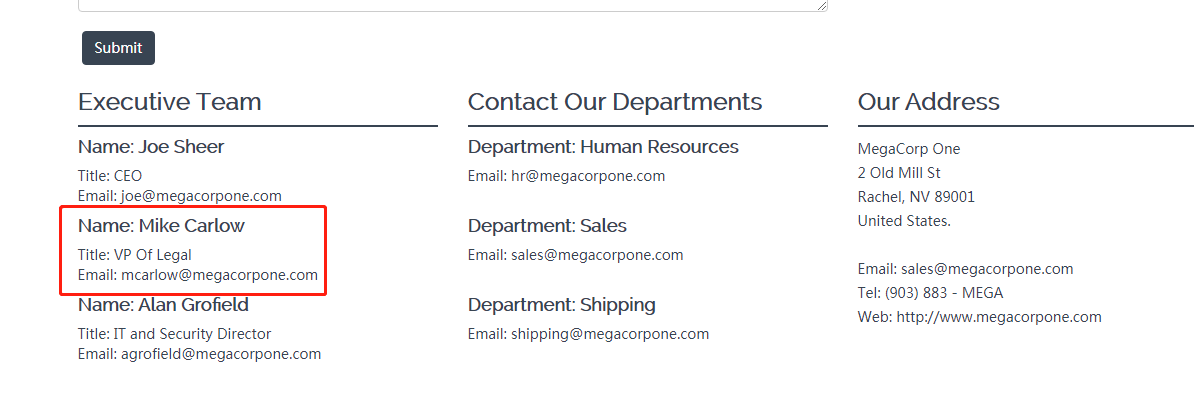
6.4.1.1 Exercises

1. Who is the VP of Legal for MegaCorp One and what is their email address?

#### **Name:Mike Carlow**

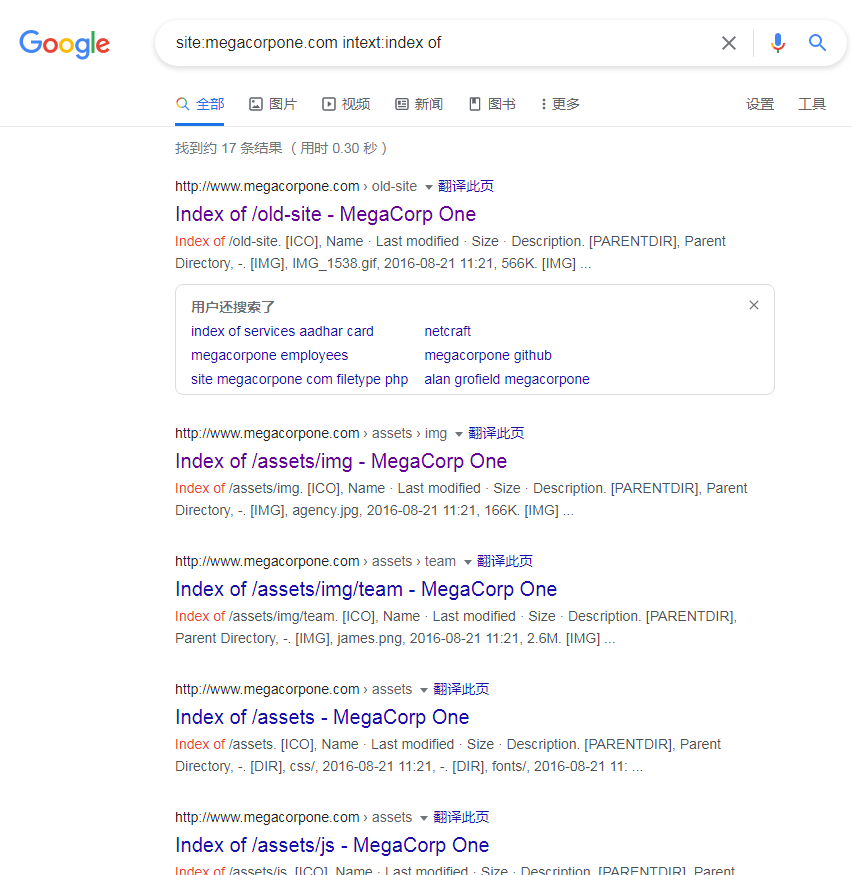
**Email:Email: mcarlow@megacorpone.com**





2. Use Google dorks (either your own or any from the GHDB) to search

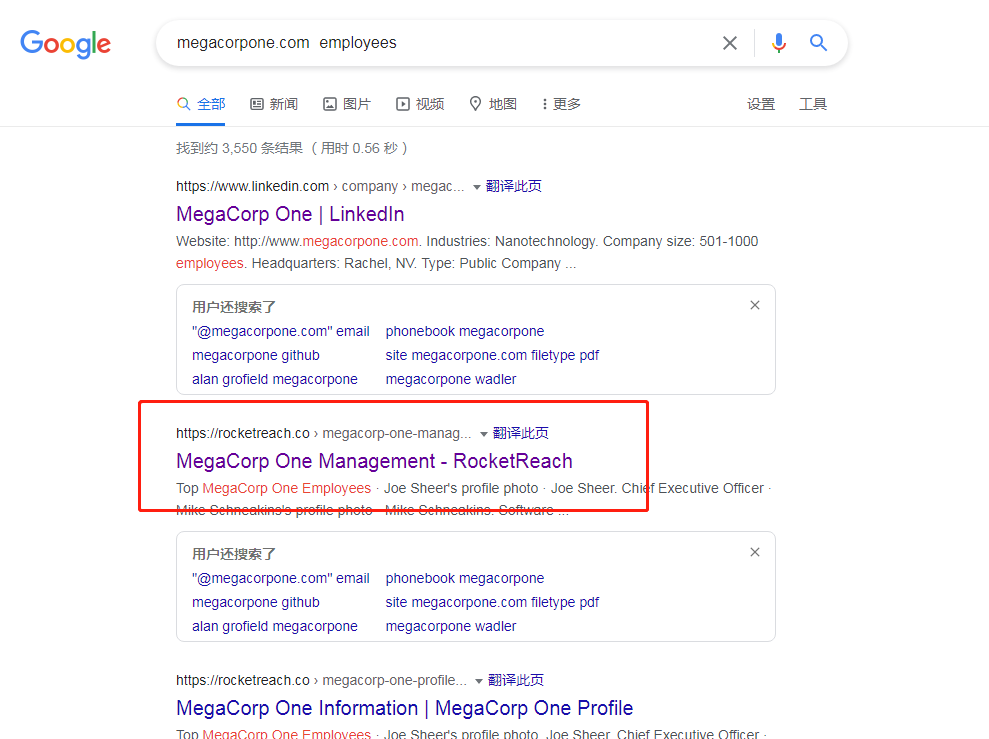
www.megacorpone.com for interesting documents.

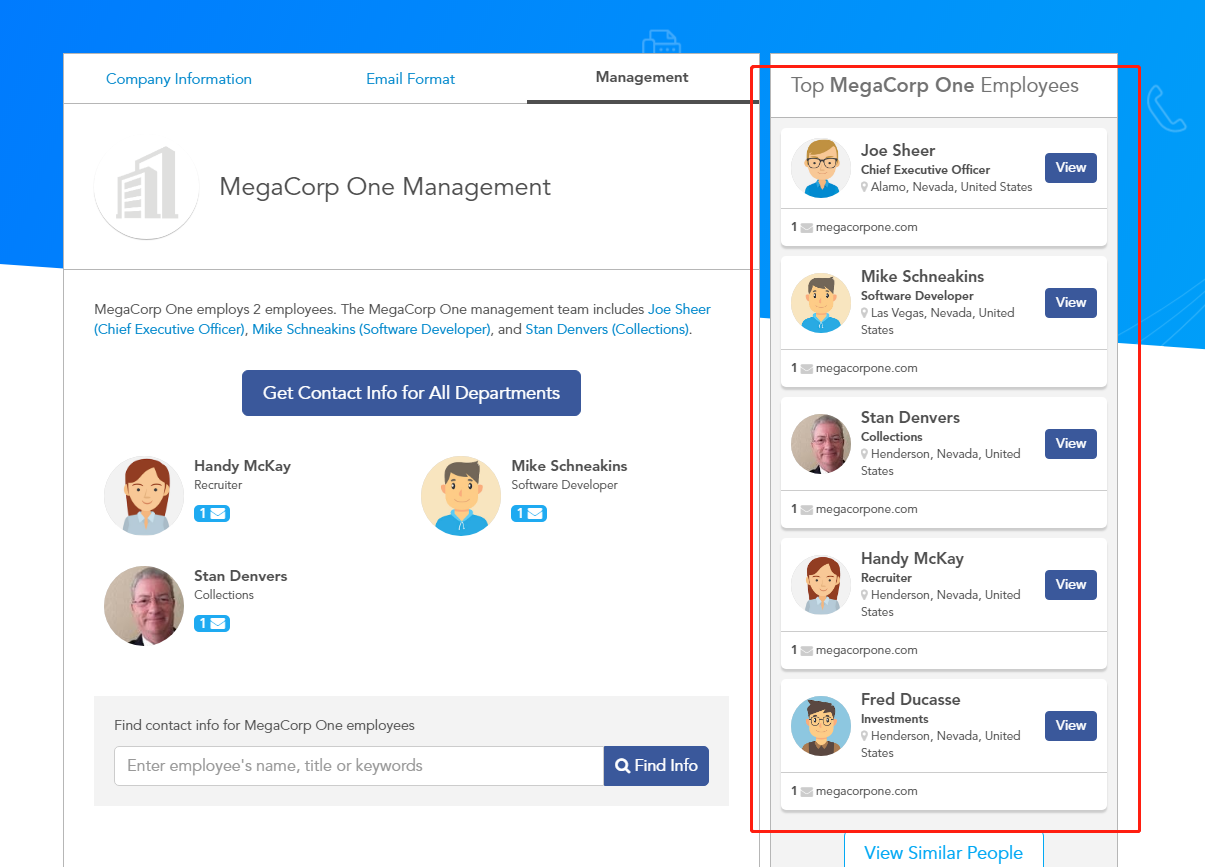




3. What other MegaCorp One employees can you identify that are not listed on

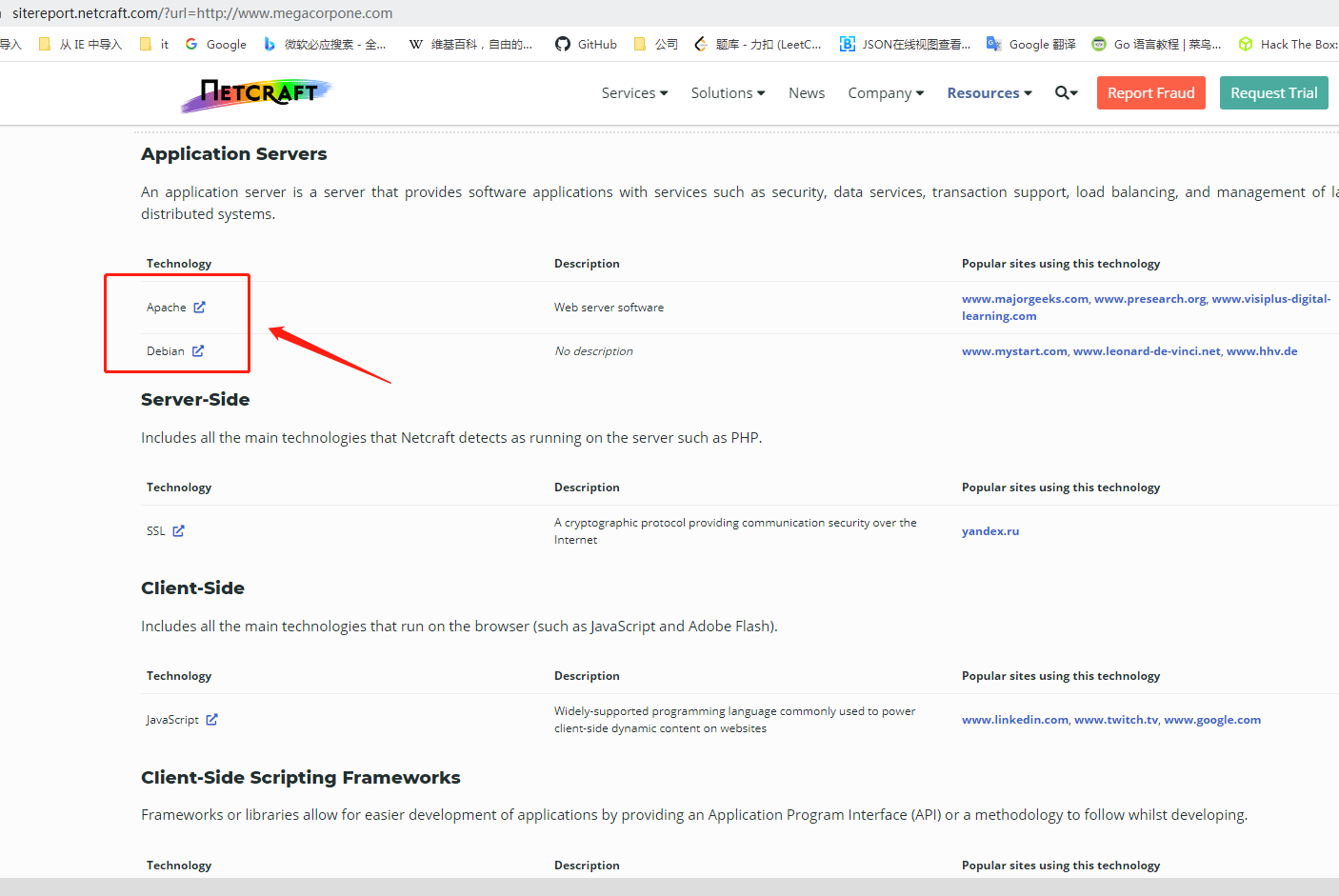
[www.megacorpone.com?](http://www.megacorpone.com?)





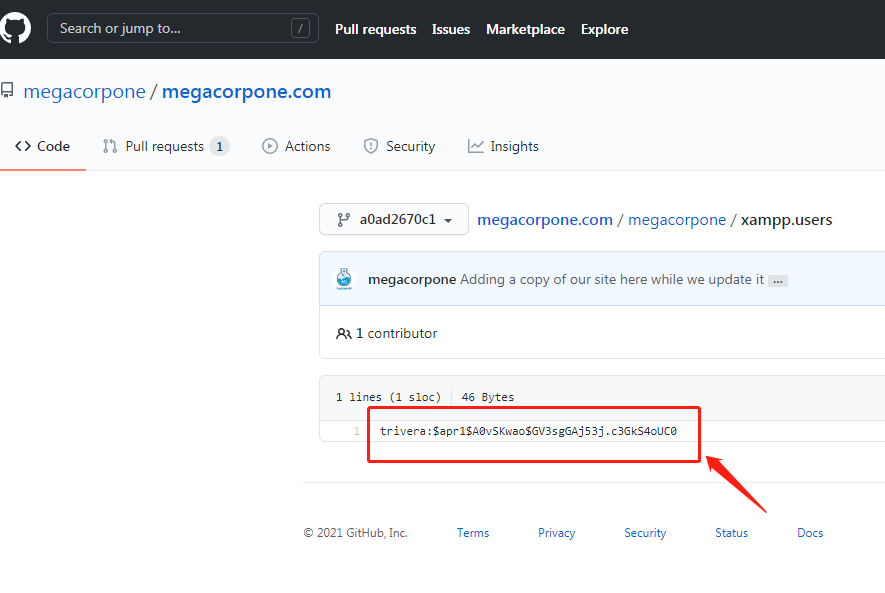
6.5.1.1 Exercise

1. Use Netcraft to determine what application server is running on www.megacorpone.com.



6.7.1.1 Exercise

1. Search Megacorpone’s GitHub repos for interesting or sensitive information.



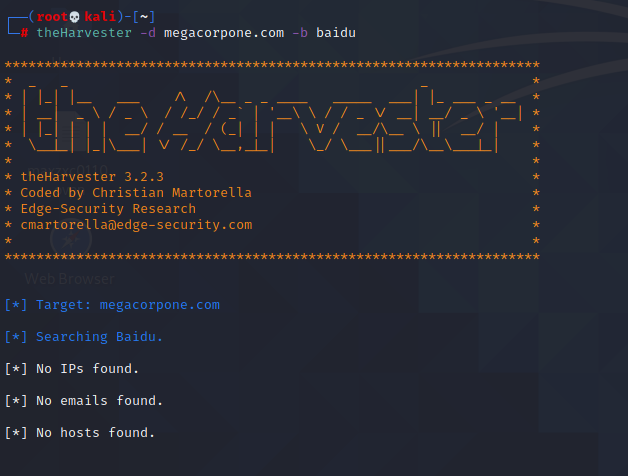
6.12.1.1 Exercises

1. Use theHarvester to enumerate emails addresses for megacorpone.com.



2. Experiment with different data sources (-b). Which ones work best for you?

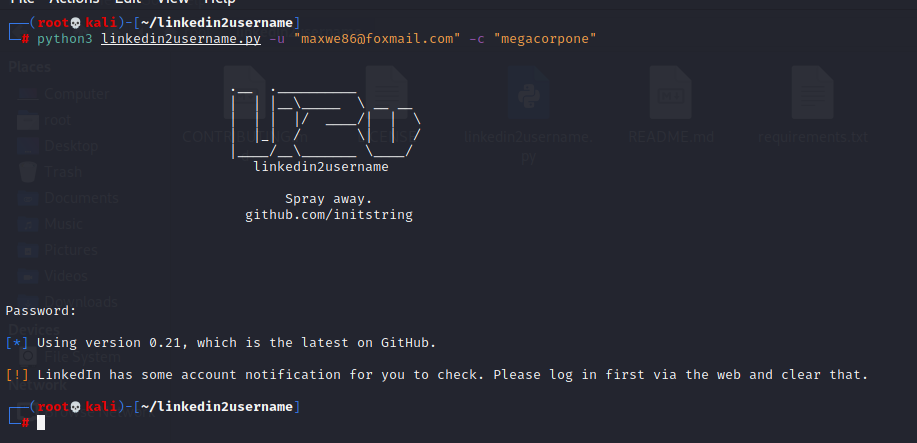




6.13.2.1 Exercise

1. Use any of the social media tools previously discussed to identify additional MegaCorp One

employees.

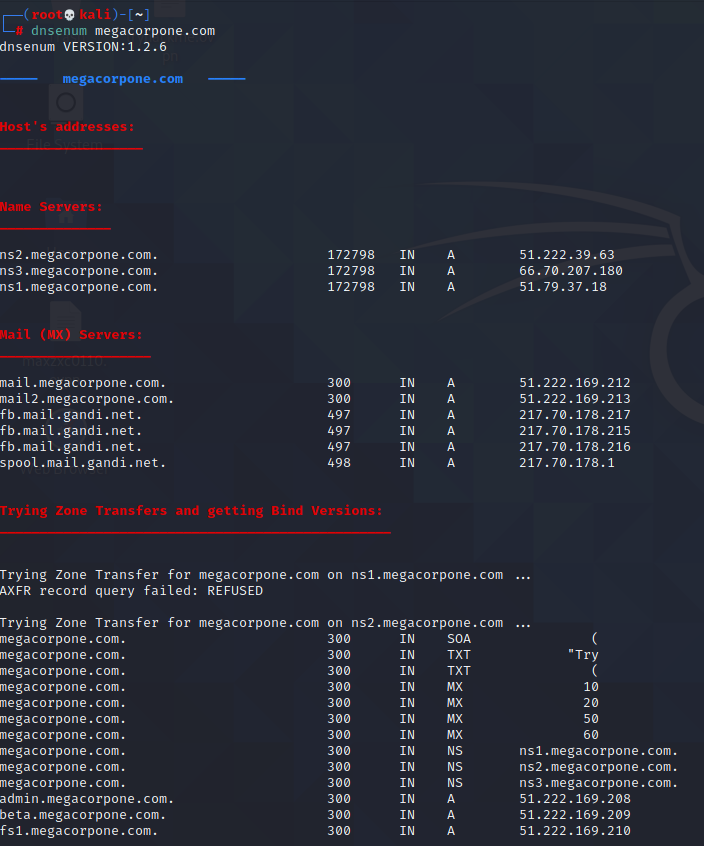


chapter 7

7.1.6.3 Exercises

1. Find the DNS servers for the megacorpone.com domain.

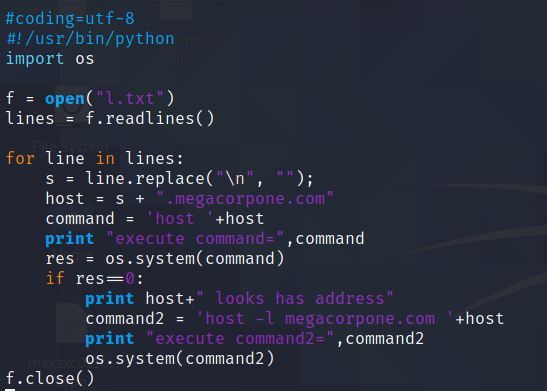
dnsenum megacorpone.com



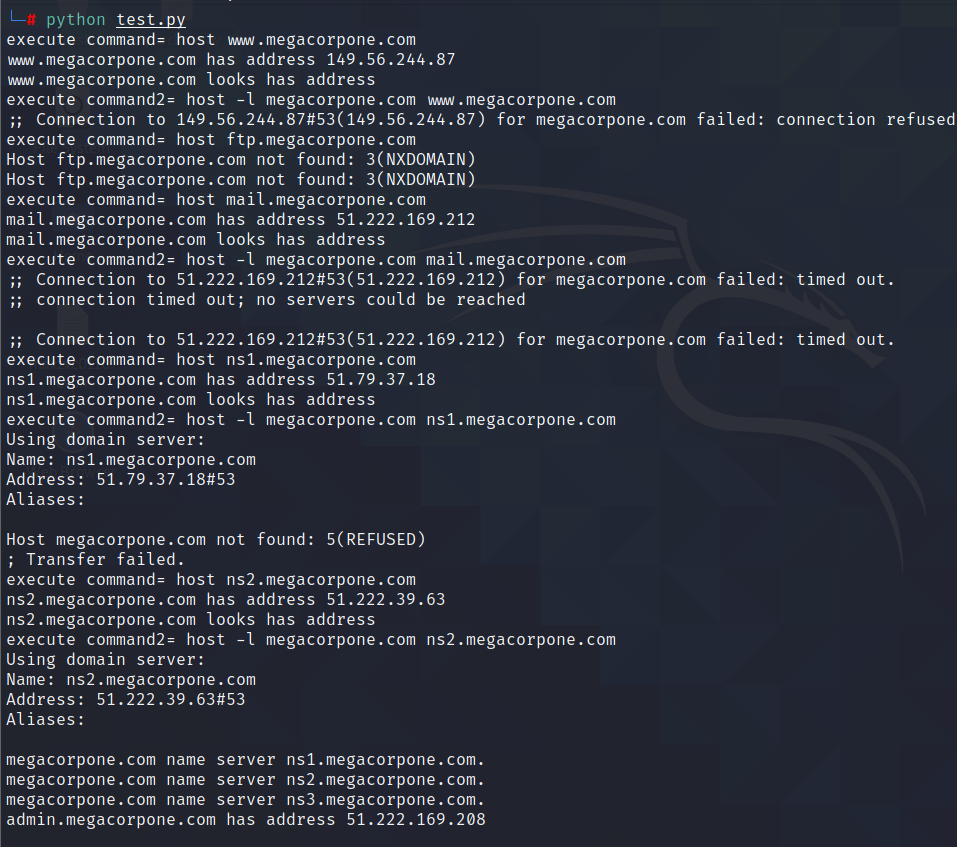
2. Write a small script to attempt a zone transfer from megacorpone.com using a higher-level

scripting language such as Python, Perl, or Ruby.

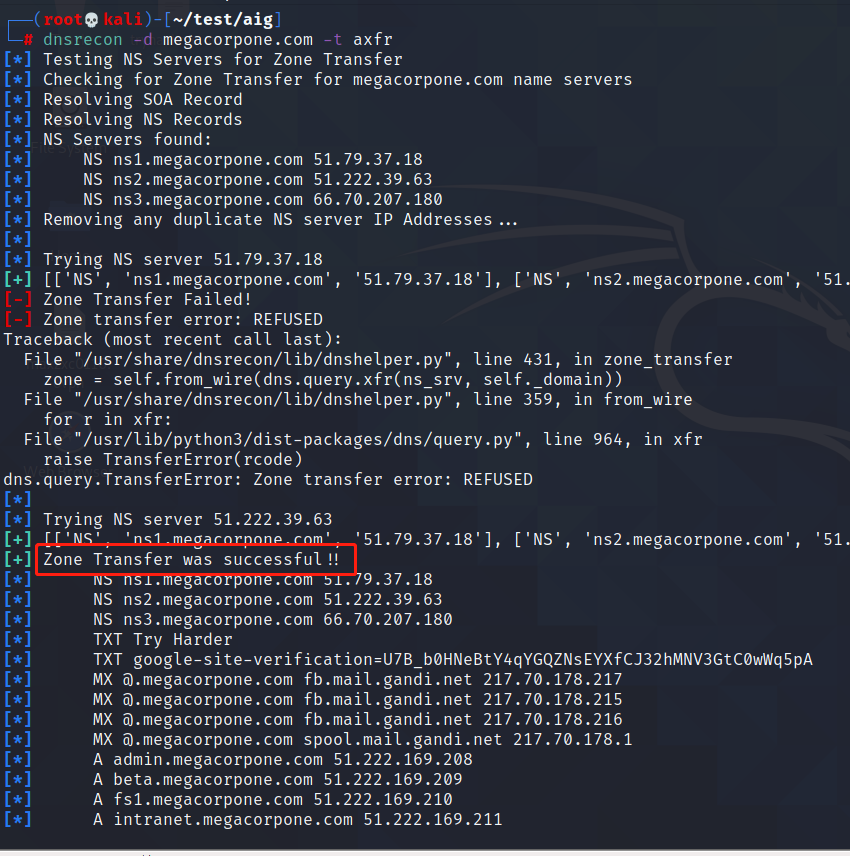
python2 source code



Execute:



1. Recreate the example above and use dnsrecon to attempt a zone transfer from megacorpone.com.



7.3.2.1 Exercises

1. Use Nmap to make a list of the SMB servers in the lab that are running Windows.

2. Use NSE scripts to scan these systems for SMB vulnerabilities.

3. Use nbtscan and enum4linux against these systems to identify the types of data you can

obtain from different versions of Windows.

7.4.2.1 Exercises

1. Use Nmap to make a list of machines running NFS in the labs.

2. Use NSE scripts to scan these systems and collect additional information about accessible

shares.

7.5.1.1 Exercises

1. Search your target network range to see if you can identify any systems that respond to the

SMTP VRFY command.

2. Try using this Python code to automate the process of username discovery using a text file

with usernames as input.

7.6.3.6 Exercises

1. Scan your target network with onesixtyone to identify any SNMP servers.

2. Use snmpwalk and snmp-check to gather information about the discovered targets.

chapter 8

8.2.4.2 Exercises

1. Follow the steps above to create your own unauthenticated scan of Gamma.

2. Run the scan with Wireshark open and identify the steps the scanner performed to

completed the scan.

1. Review the results of the scan.

8.2.5.2 Exercises

1. Follow the steps above to create your own authenticated scan of your Debian client.

2. Review the results of the scan.

8.2.6.1 Exercises

1. Follow the steps above to create your own individual scan of Beta.

2. Run Wireshark or tcpdump during the individual scan. What other ports does Nessus scan?

Why do you think Nessus scans other ports?

1. Review the results of the scan.

8.3.1.1 Exercise

1. Find an NSE script similar to the NFS Exported Share Information Disclosure that was

executed in the “Scanning with Individual Nessus Plugins” section. Once found, run the script

against Beta in the PWK labs.

chapter 9

9.3.4.1 Exercise

1. Spend some time reviewing the applications available under the Web Application Analysis menu in Kali Linux.



9.4.1.3 Exercises

1. Use Burp Intruder to gain access to the phpMyAdmin site running on your Windows 10 lab machine.

2. Insert a new user into the “users” table.

9.4.2.5 Exercises

1. Exploit the XSS vulnerability in the sample application to get the admin cookie and hijack the

session. Remember to use the PowerShell script on your Windows 10 lab machine to

simulate the admin login.

2. Consider what other ways an XSS vulnerability in this application might be used for attacks.

3. Does this exploit attack the server or clients of the site?

9.4.3.2 Exercise

1. Exploit the directory traversal vulnerability to read arbitrary files on your Windows 10 lab machine.

9.4.4.7 Exercises

1. Exploit the RFI vulnerability in the web application and get a shell.

2. Using /menu2.php?file=current\_menu as a starting point, use RFI to get a shell.

3. Use one of the webshells included with Kali to get a shell on the Windows 10 target.

9.4.4.10 Exercises

1. Exploit the LFI vulnerability using a PHP wrapper.

2. Use a PHP wrapper to get a shell on your Windows 10 lab machine.

9.4.5.4 Exercises

1. Interact with the MariaDB database and manually execute the commands required to

authenticate to the application. Understand the vulnerability.

2. SQL inject the username field to bypass the login process.

3. Why is the username displayed like it is in the web application once the authentication

process is bypassed?

4. Execute the SQL injection in the password field. Is the “LIMIT 1” necessary in the payload?

Why or why not?

9.4.5.9 Exercises

1. Enumerate the structure of the database using SQL injection.

2. Understand how and why you can pull data from your injected commands and have it

displayed on the screen.

1. Extract all users and associated passwords from the database.

9.4.5.11 Exercises

1. Exploit the SQL injection along with the MariaDB INTO OUTFILE function to obtain code

execution.

1. Turn the simple code execution into a full shell.

9.4.5.13 Exercises

1. Use sqlmap to obtain a full dump of the database.

2. Use sqlmap to obtain an interactive shell.