INFO EXTRACTOR

Author: Yvette K

CONTENTS

Introduction

The purpose of this report is to examine how system information can be gathered by creating a smart script to pull out key details from the system in a neat and efficient manner. Details include network information like IP and MAC addresses, CPU and memory usage to inspect how the system is running, and which files are largest.

Methodologies

A bash script was first created with geany and language was set (*#!/bin/bash*).

curl command was used to fetch the external IP address.

ifconfig command was used with *grep* and *awk* to determine the internal IP and MAC addresses.

top command was used to find the CPU usage of processes and *head* and *tail* were used to narrow down the results to top 5.

ps command was used to display memory usage with *--sort* and *%mem* used to further refine the results.

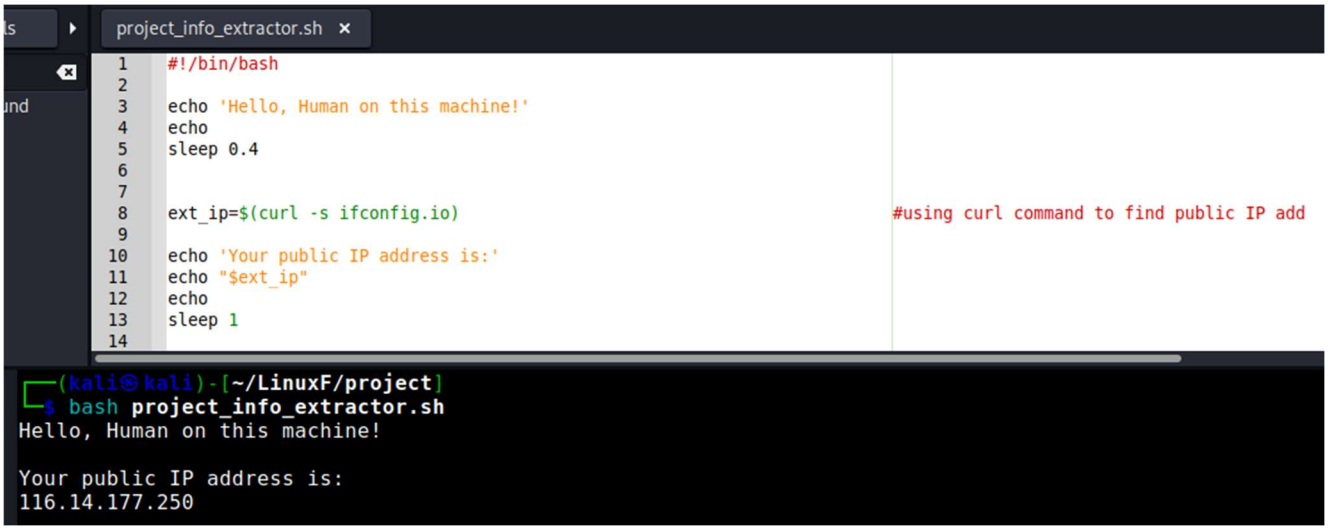
free command gave the memory and swap stats, which were then filtered with *head*, *tail* and *awk* to display specific details.

Command *systemctl --type=service* was used to display services in the system and flag "*--state=running*" was used to refine results to only running services.

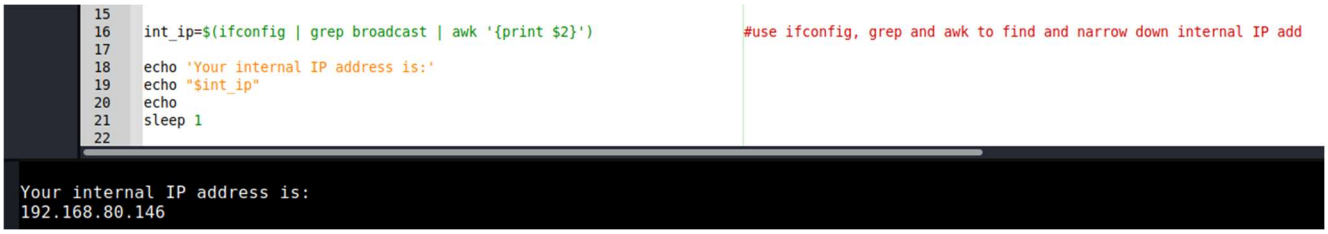
find and *du* commands were used to search files from /home, using flags *-type f*, *-h* and filtered by *sort -rh* and *head* to display the top 10 files.

Discussion

curl command used to retrieve public IP address from external web services. *-s* flag was used to silence any progress or error messages. *sleep* command is added after every information output to increase readability.



ifconfig command used to find current configuration for the network and narrowing down relevant information using *grep* and *awk*.



Same *ifconfig*, *grep* and *awk* commands used to narrow down different parts of the configuration information to display the MAC address of the machine, censoring the first 24 bits for privacy.

top command, essential for monitoring CPU and memory usage on per-process basis. It is a valuable utility that provides critical information about the system performances and running processes and can be customised to display only information relevant to the use. In this script, the top 5 were listed using *head* and *tail* commands. *top* command also supports various keyboard shortcuts and optional

features such as colour change, highlighting and graphs to enhance readability. Flags *b* and *n1* were utilised to print in batch mode and single snapshot respectively. To display only the relevant information, *awk* with *column -t* were used to extract and print neatly. Given subsequent information to be displayed may be substantial, a while loop is used to slow down output, printing them line by line.

```
34  echo "CPU usage (%) for the top 5 processes are:"                                    #using command top to show processes' CPU usage % with -b for batch mode and -n 1 output single snapshot
35  sleep 1
36  top -b -n 1 | head -12 | tail -6 | awk '{print $1, $2, $9, $12}' | column -t  | while read line; do      #using awk to only output PID, user, CPU usage % and command
37      echo "$line"                                                                    #using while loop and sleep to slow down output with multiple lines
38      sleep 0.4
39  done
40  echo
41  sleep 2
42
```

```
CPU usage (%) for the top 5 processes are:
PID    USER  %CPU  COMMAND
66795  kali  12.5  top
1      root  0.0   systemd
2      root  0.0   kthreadd
3      root  0.0   pool_wo+
4      root  0.0   kworker+
```

ps aux command was used to list all running processes in Linux and filtered by *--sort* and *%mem* to sort by memory usage.

```
44  echo 'Memory usage statistics:'                                                    #using ps and free commands to display memory usage
45  sleep 1
46
47  ps aux --sort -%mem | while read line; do                                          #using ps aux to list all running processes with detailed information for each, and --sort -%mem to sort and list processes consuming the most memo
48      echo "$line"                                                                   #using while loop and sleep to slow down output with multiple lines
49      sleep 0.1
50  done
51  echo
52  sleep 1
```

```
Memory usage statistics:
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1022 15.8  6.7 422252 136840 tty7     Ssl+ 03:01  20:01 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
kali        1302  1.9  5.9 1264024 118976 ?       Sl   03:03   2:25 xfwm4
kali        3869  0.0  4.9 457664 100276 ?        Sl   03:07   0:04 /usr/bin/qterminal
kali        1347  0.0  3.3 485604 66820 ?         Sl   03:03   0:06 xfdesktop
kali        4103  1.8  3.1 493276 63404 pts/0     Sl+  03:07   2:16 geany project_info_extractor.sh
kali        1355  1.3  2.9 300248 60276 ?         Sl   03:03   1:42 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libcpugraph.so 1
3 16777228 cpugraph CPU Graph Graphical representation of the CPU load
kali        1488  1.2  2.5 331664 52228 ?         Sl   03:03   1:36 /usr/bin/vmtoolsd -n vmusr --blockFd 3
kali        1494  0.0  2.5 515564 51948 ?         Sl   03:03   0:02 /usr/bin/python3 /usr/bin/blueman-applet
kali        1341  0.0  2.3 537404 46808 ?         Sl   03:03   0:05 Thunar --daemon
kali        1417  0.0  2.2 618912 45240 ?         Sl   03:03   0:01 nm-applet
kali        1336  0.0  2.1 532980 42908 ?         Sl   03:03   0:05 xfce4-panel
kali        1348  0.0  2.0 457620 41828 ?         Sl   03:03   0:02 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libwhiskermenu.s
o 1 16777223 whiskermenu Whisker Menu Show a menu to easily access installed applications
kali        1400  0.0  2.0 458788 40704 ?         Ssl  03:03   0:01 /usr/lib/x86_64-linux-gnu/xfce4/notifyd/xfce4-notifyd
kali        1358  0.0  1.9 456460 40052 ?         Sl   03:03   0:01 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libpulseaudio-pl
ugin.so 16 16777231 pulseaudio PulseAudio Plugin Adjust the audio volume of the PulseAudio sound system
kali        1360  0.0  1.9 383220 38860 ?         Sl   03:03   0:04 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libxfce4powerman
ager.so 18 16777233 power-manager-plugin Power Manager Plugin Display the battery levels of your devices and control the brightness of your display
```

Using *free* command, memory statistics can be seen and flag *h* displays the information in auto-selected human readable format.

```
53
54  tm=$(free -h | head -2 | tail -1 | awk '{print $2}')                               #using free and -h to output system's memory usage in human-readable format with auto selection of units and awk to display required column
55  echo "Total Memory: $tm"
56  sleep 1
57
58  um=$(free -h | head -2 | tail -1 | awk '{print $3}')
59  echo "Used Memory: $um"
60  sleep 1
61
62  am=$(free -h | head -2 | tail -1 | awk '{print $NF}')
63  echo "Available Memory: $am"
64  echo
65  sleep 1
66
```

```
Total Memory: 1.9Gi
Used Memory: 801Mi
Available Memory: 1.1Gi
```

/proc/meminfo command will also show the estimate of memory available for starting new applications without swapping (*MemAvailable*), amount of physical memory not being used by the system (*MemFree*), total usable memory (RAM) (*MemTotal*), total amount of buffer/page cache memory that is in active use (*Active*).

```
┌──(kali㉿kali)-[~]
└─$ grep MemAvailable /proc/meminfo
MemAvailable:    1276196 kB

┌──(kali㉿kali)-[~]
└─$ grep MemFree /proc/meminfo
MemFree:          449848 kB

┌──(kali㉿kali)-[~]
└─$ grep MemTotal /proc/meminfo
MemTotal:        2015132 kB

┌──(kali㉿kali)-[~]
└─$ grep Active /proc/meminfo
Active:           786628 kB
Active(anon):     341856 kB
Active(file):     444772 kB
```

The command *systemctl --type=service --state=running* shows all running services on a Linux system with system and provides details like name, load, sub-state, and description. These details can be informative when looking into a service or daemon that did not launch correctly.

```
68  echo 'Your Active System services are:'
69  sleep 1
70
71  systemctl --type=service --state=running | grep active | while read line; do                #using systemd to display list of all currently running services and while loop and sleep to slow down output with multiple
72      echo "$line"
73      sleep 0.4
74  done
75  echo
76  sleep 2
```

```
Your Active System services are:
accounts-daemon.service  loaded active running Accounts Service
colord.service           loaded active running Manage, Install and Generate Color Profiles
cron.service             loaded active running Regular background program processing daemon
dbus.service             loaded active running D-Bus System Message Bus
getty@tty1.service       loaded active running Getty on tty1
haveged.service          loaded active running Entropy Daemon based on the HAVEGE algorithm
lightdm.service          loaded active running Light Display Manager
ModemManager.service     loaded active running Modem Manager
NetworkManager.service   loaded active running Network Manager
open-vm-tools.service    loaded active running Service for virtual machines hosted on VMware
polkit.service           loaded active running Authorization Manager
rtkit-daemon.service     loaded active running RealtimeKit Scheduling Policy Service
systemd-journald.service loaded active running Journal Service
systemd-logind.service   loaded active running User Login Management
systemd-udevd.service    loaded active running Rule-based Manager for Device Events and Files
udisks2.service          loaded active running Disk Manager
upower.service           loaded active running Daemon for power management
user@1000.service        loaded active running User Manager for UID 1000
```

du command was used to analyse and report on disk usage within directories and files. This is useful in identifying space-hogging directories/files, managing disk space efficiently and gaining insights into storage consumption.

find command was used to search from /home, files filtered by flag *-type f*, based on size executed by command *du* with flag *-h* to display in human readable format and then sorted with *sort -rh* in descending order and filtered with *head* to display the top 10 largest files.

```
79  echo 'Your top 10 largest files are:'
80  sleep 1
81
82  sudo find /home -type f -exec du -h {} + | sort -rh | head -n 10 | while read line; do          #using superuser privileges to search all files in home directory and -exec runs the du -h command on each file, displaying
83      echo "$line"                                                                               #using while loop and sleep to slow down output with multiple lines
84      sleep 0.4
85  done
86  echo
87  sleep 2
88
```

```
Your top 10 largest files are:
[sudo] password for kali:
17M     /home/kali/.cache/mozilla/firefox/4tk7c499.default-esr/cache2/entries/E67870A0FB9AE0BE6430170A4B5BF2C73117274C
11M     /home/kali/.cache/mozilla/firefox/4tk7c499.default-esr/cache2/entries/14EE7BAEC485797709A95E9DCDFE3AA7E248D3A3
9.9M    /home/kali/.mozilla/firefox/4tk7c499.default-esr/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.sqlite
9.2M    /home/kali/.cache/mozilla/firefox/4tk7c499.default-esr/safebrowsing/google4/goog-phish-proto.vlpset
8.3M    /home/kali/.msf4/store/modules_metadata.json
8.1M    /home/kali/.cache/mozilla/firefox/4tk7c499.default-esr/startupCache/scriptCache-current.bin
5.0M    /home/kali/.mozilla/firefox/4tk7c499.default-esr/places.sqlite
5.0M    /home/kali/.mozilla/firefox/4tk7c499.default-esr/favicons.sqlite
3.9M    /home/kali/NR/mitm2.pcap
3.5M    /home/kali/NR/scripting/auth.log
```

Conclusion

There are many tools available to monitor CPU utilisation – the percentage of time the central processing unit (CPU) is actively processing tasks or carrying out instructions. It is essential in determining the efficiency and performance of a computer system. Monitoring CPU utilisation helps cybersecurity practitioners detect and prevent malicious activities by identifying bottlenecks, resource constraints and potential security threats in systems.

An abrupt and unexpected increase in CPU utilisation may suggest the presence of malware or unauthorised activities. A noticeable spike in CPU can be a strong indicator of a compromised system and an automated process actively monitoring CPU utilisation can help alert organisations when anomalies occur and necessary actions can be taken swiftly. Anomalies can also be identified and addressed to mitigate risks.

An example of such anomalies is unusual patterns in CPU and network utilisation across the affected network that are often the result of botnet attacks, where cybercriminals control a network or compromised machines to conduct coordinated attacks.

By closely monitoring CPU utilisation, organisations can potentially identify memory-resident cyber threats, such as advanced persistent threats (APTs), that pose a significant challenge to cybersecurity. APTs attempt to remain hidden by using minimal system resources while still affecting CPU utilisation differently. Being aware of how different systems and processes affect CPU utilisation will allow cybersecurity practitioners to better identify such incongruities and take appropriate timely actions to mitigate their impact.

Recommendations

Effective cybersecurity measures should optimise CPU utilisation and organisations can implement prevention tips such as using reputable security solutions that include antivirus and antimalware features that can detect and address malware, viruses and other malicious software that can significantly impact CPU utilisation.

It is essential to employ tools that continuously monitor CPU usage regularly. By setting alerts for abnormal spikes and patterns, organisations can quickly identify potential security threats and take swift action.

Systems should be kept up to date to maintain optimal CPU utilisation by regularly applying security patches and updates to operating systems, applications and security software. This would address vulnerabilities that cybercriminals could exploit.

References

Antil, Pradeep. “How to Find Public IP Address in Linux From Command Line” *LinuxBuzz*, 26 July 2023, https://www.linuxbuzz.com/how-to-find-public-ip-address-in-linux/#google_vignette

“How to Continue a Bash Script Post cURL Command” *Squash*, 21 October 2023, https://www.squash.io/continuing-bash-scripts-post-curl-command-in-linux/#:~:text=option%20tells%20curl%20to%20operate,command%20completed%E2%80%9D%20to%20the%20terminal

Jevtic, Goran. “How to Check CPU Utilization in Linux with Command Line” *phoenixNAP*, 6 March 2024, https://phoenixnap.com/kb/check-cpu-usage-load-linux

McKay, Dave. “How to Use the Linux top Command (and Understand Its Output)” *How To Geek*, 17 August 2023, https://www.howtogeek.com/668986/how-to-use-the-linux-top-command-and-understand-its-output/

Jethva, Hitesh. “How to Find the Top Memory Consuming Processes in Linux” a*tlantic.net*, 30 April 2022, https://www.atlantic.net/vps-hosting/find-top-10-running-processes-by-memory-and-cpu-usage/#:~:text=Use%20ps%20Command%20to%20Find%20Top%20Processes,-ps%20is%20a&text=You%20can%20use%20the%20ps%20command%20with%20%E2%80%93sort%20argument%20to,by%20memory%20and%20CPU%20usage

“Understanding ps Command in Linux” S*hiksha Online*, 20 September 2023, https://www.shiksha.com/online-courses/articles/understanding-ps-command-in-linux/#:~:text=Command%20in%20Linux-,What%20is%20the%20ps%20Command%20in%20Linux%3F,about%20the%20currently%20running%20processes

Kaplarevic, Vladimir. “How to Check Memory Usage in Linux via CLI and GUI” *phoenixNAP*, 28 March 2024, https://phoenixnap.com/kb/linux-commands-check-memory-usage

Ramuglia, Gabriel. “‘du’ Linux Command Guide: Disk Usage Made Easy” *I/O Flood*, 11 December 2023, https://ioflood.com/blog/du-linux-command/#:~:text=The%20'du'%20command%20in%20Linux%20is%20used%20to%20estimate%20file,path%2Fto%2Fdirectory%5D%20.&text=In%20this%20example%2C%20we%20use,the%20%2Fhome%2Fuser%20directory

McKay, Dave. " How to List Linux Services With systemctl" *How To Geek*, 19 August 2023, https://www.howtogeek.com/839285/how-to-list-linux-services-with-systemctl/#:~:text=Use%20%22systemctl%20%2D%2Dtype%3Dservice,in%20Linux%20that%20launches%20services

"du command in Linux with examples" *GeeksforGeeks*, 14 December 2023, https://www.geeksforgeeks.org/du-command-linux-examples/

"CPU Utilization: Enhancing the Definition and Understanding" *VPN Unlimited*, https://www.vpnunlimited.com/help/cybersecurity/cpu-utilization