NAME: ALEXANDER CHAIT

T.Z: [redacted]

EDUCATOR: BORIS FRENKEL

DATE: 03.27.2024

קורס למתחילים בסייבר

CSPP

מבואות סייבר

ניהול רשתות ובקר SOC

Linux Essentials Project

Linux Essentials

Final Work

Case Scenario:

- There has been suspicious activity in the system. In this case it will be necessary to create a snapshot of your system with all necessary information to send it to the technical support which can help you with the issue.
- You should create a script which should help you to get all the relevant information from your system, create the text files with this information, get the current log files from your system and create the archive file which contains all this data.

Steps for the script:

- Create the temporary directory which names **_support** in your current placement
- Copy the log files to the created directory. You should copy all the ***.log** files which are in the directory **/var/log**.
- Get all the relevant information about your hardware and store it in the text files. You should retrieve the info about your CPU, memory, storage, peripheral devices etc.
- Get all the relevant information about your operating system and its current state: kernel version, distribution info, users list, processes etc.
- Get all the relevant information about your network: network interfaces, routing table, DNS information, results of the network checking by ping, traceroute etc.
- Create the archive file, which will contain all the files/directories which you placed in the **_support** directory. The filename of the archive should be by like **support-<current-date-time>.tar.gz** where **<current-date-time>** should be provided by next format: **YYYY-MM-DD_HHMMSS**.

Deliverables:

- Provide the partial results for main of operations in the script (getting the information, creating the archive, etc.) and screenshots for its results (without script).
- Provide the final version of the script and screenshots with successful completion.

Notes:

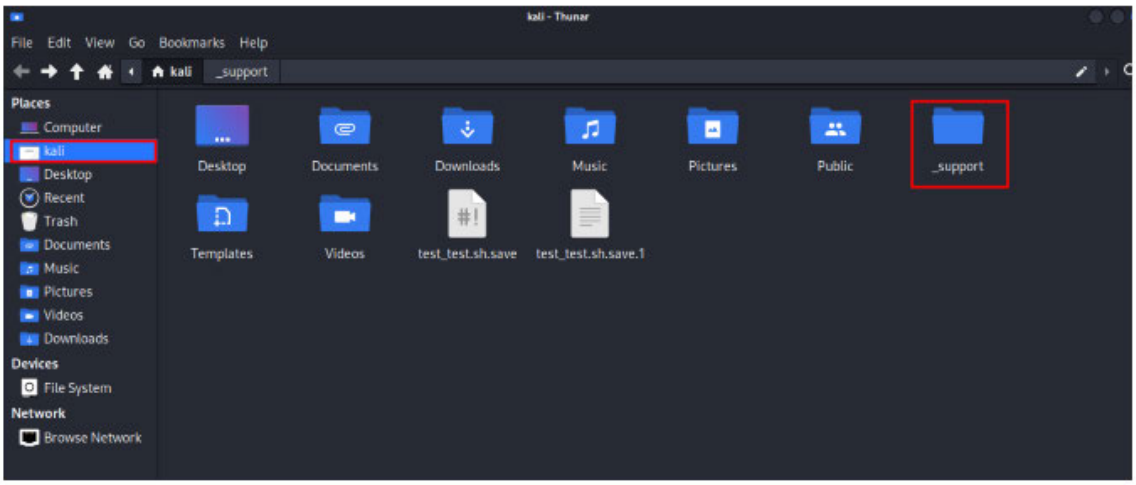
- Use the grep command to provide the data which relates to the informative sources. For example, getting the data only about the current user from the files **/etc/passwd** and **/etc/shadow**.
- Use the commands parameters to filter/expand the system information. For example, data only from active network interfaces.

Step 1: Creating a temporary directory

I'll start by utilizing this following command

```
mkdir _support
```

This command creates a directory named _support in the current location.



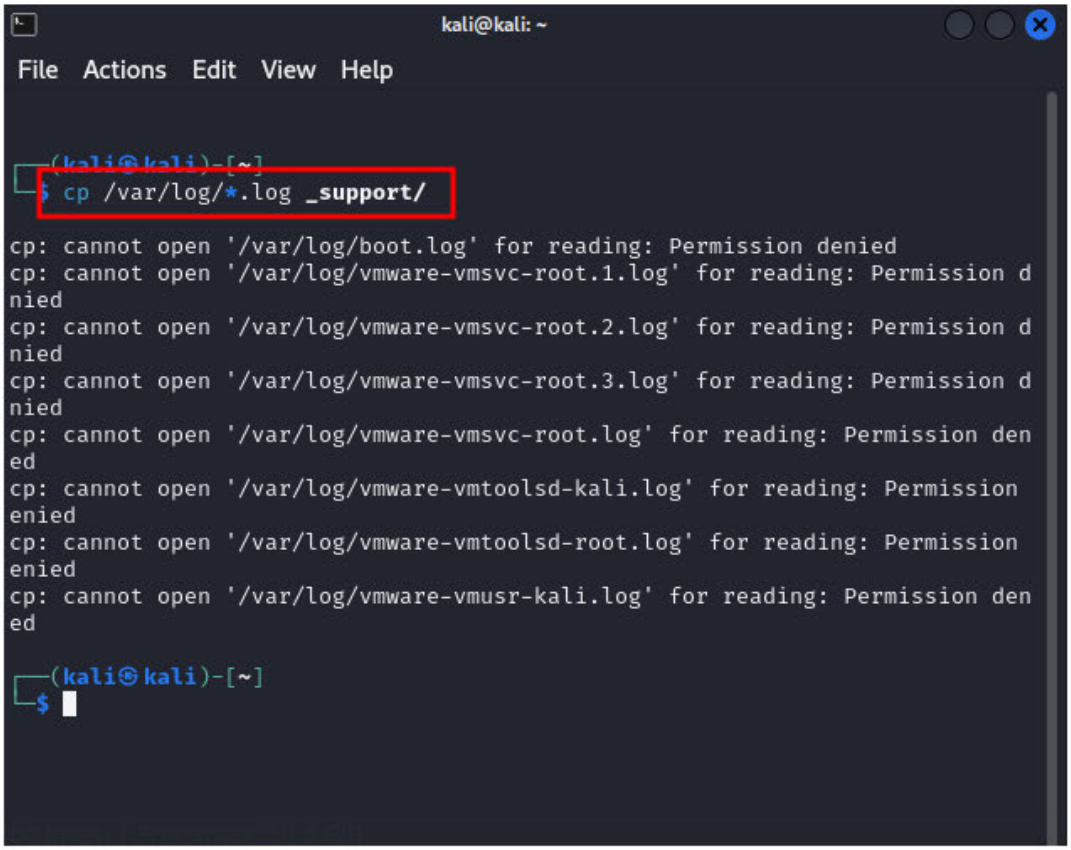
We can see that the _support directory has been created

Step 2: Copy Log Files

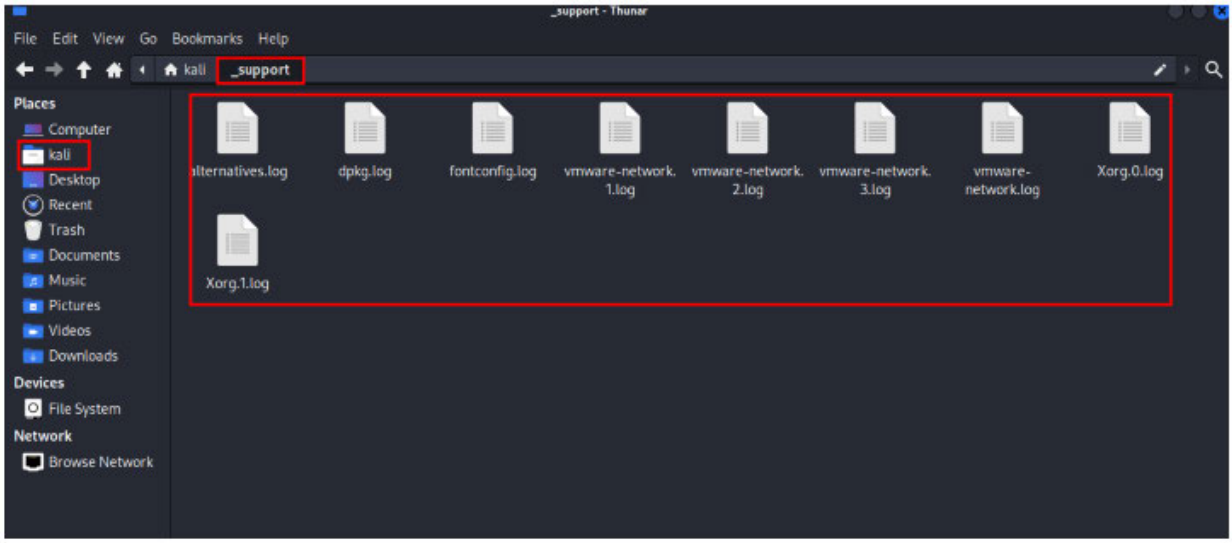
This command copies all files with the extension .log from the /var/log/ directory to the _support directory.

```
cp /var/log/*.log _support/
```

I'll use the command in the terminal emulator



Now we can a see a copy of all the .log files from /var/log/ being copied to the _support directory

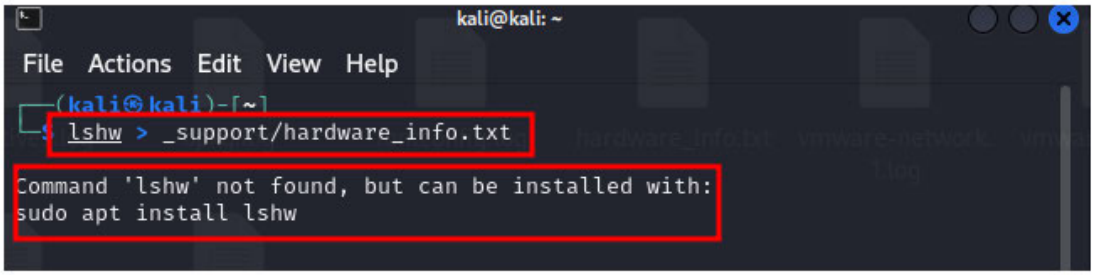


Step 3: Get Hardware Information

This command uses the lshw command to retrieve hardware information and redirects the output to a text file named hardware_info.txt in the _support directory.

```
lshw > _support/hardware_info.txt
```

After entering the command I can see the lshw is not yet installed, next step will be to install it using the sudo apt install lshw command



Installing lshw

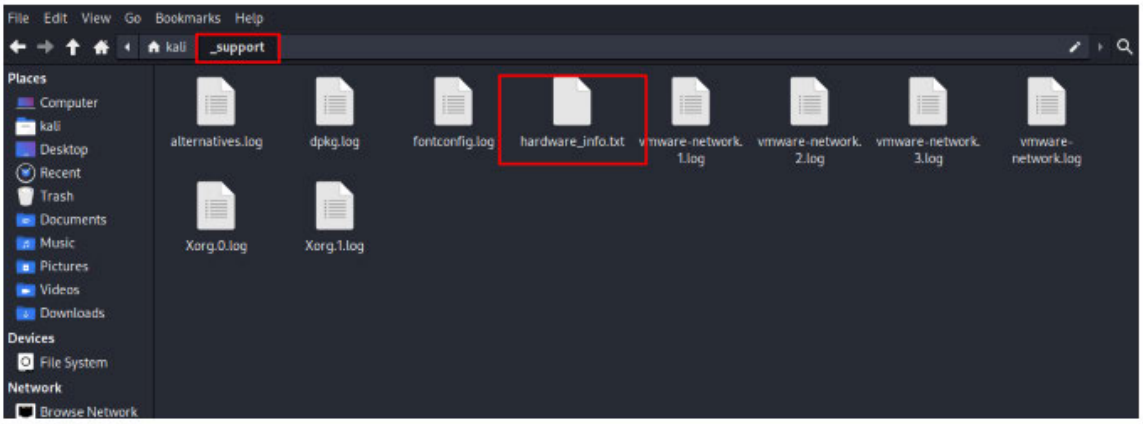
```
┌──(kali㉿kali)-[~]
└─$ sudo apt install lshw
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
  lshw
0 upgraded, 1 newly installed, 0 to remove and 1525 not upgraded.
Need to get 300 kB of archives.
After this operation, 941 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 lshw amd64 02.19.git.
2021.06.19.996aaad9c7-2+b1 [300 kB]
Fetched 300 kB in 2s (149 kB/s)
Selecting previously unselected package lshw.
(Reading database ... 399698 files and directories currently installed.)
Preparing to unpack .../lshw_02.19.git.2021.06.19.996aaad9c7-2+b1_amd64.deb .
..
Unpacking lshw (02.19.git.2021.06.19.996aaad9c7-2+b1) ...
Setting up lshw (02.19.git.2021.06.19.996aaad9c7-2+b1) ...
Processing triggers for kali-menu (2023.4.6) ...
Processing triggers for man-db (2.12.0-1) ...
```

Installation is complete, back to lshw command to retrieve hardware information and redirects the output to a text file named hardware_info.txt in the _support directory.

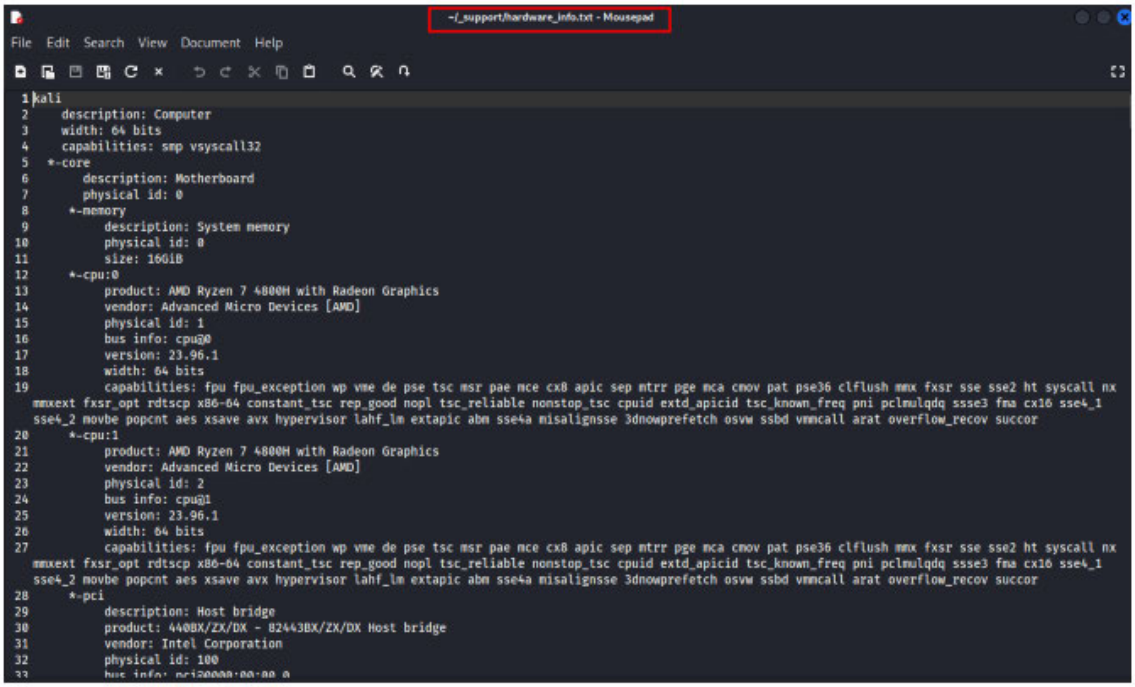
```
lshw > _support/hardware_info.txt
```

Using the command in the terminal:

We can see that the command worked and we have the hardware_info.txt file that provides us with all the hardware information into a simple txt file



I'll open the txt file to see the hardware information:




```
kali
    description: Computer
    width: 64 bits
    capabilities: smp vsyscall32
  *-core
       description: Motherboard
       physical id: 0
     *-memory
          description: System memory
          physical id: 0
          size: 16GiB
     *-cpu:0
          product: AMD Ryzen 7 4800H with Radeon Graphics
          vendor: Advanced Micro Devices [AMD]
          physical id: 1
          bus info: cpu@0
          version: 23.96.1
          width: 64 bits
          capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx
  mmxext fxsr_opt rdtscp x86-64 constant_tsc rep_good nopl tsc_reliable nonstop_tsc cpuid extd_apicid tsc_known_freq pni pclmulqdq ssse3 fma cx16 sse4_1
  sse4_2 movbe popcnt aes xsave avx hypervisor lahf_lm extapic abm sse4a misalignsse 3dnowprefetch osvw ssbd vmmcall arat overflow_recov succor
     *-cpu:1
          product: AMD Ryzen 7 4800H with Radeon Graphics
          vendor: Advanced Micro Devices [AMD]
          physical id: 2
          bus info: cpu@1
          version: 23.96.1
          width: 64 bits
          capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx
  mmxext fxsr_opt rdtscp x86-64 constant_tsc rep_good nopl tsc_reliable nonstop_tsc cpuid extd_apicid tsc_known_freq pni pclmulqdq ssse3 fma cx16 sse4_1
  sse4_2 movbe popcnt aes xsave avx hypervisor lahf_lm extapic abm sse4a misalignsse 3dnowprefetch osvw ssbd vmmcall arat overflow_recov succor
     *-pci
          description: Host bridge
          product: 440BX/ZX/DX - 82443BX/ZX/DX Host bridge
          vendor: Intel Corporation
          physical id: 100
```

We can see that the .txt file has all the hardware information in a simple easy to read format now inside the _support directory

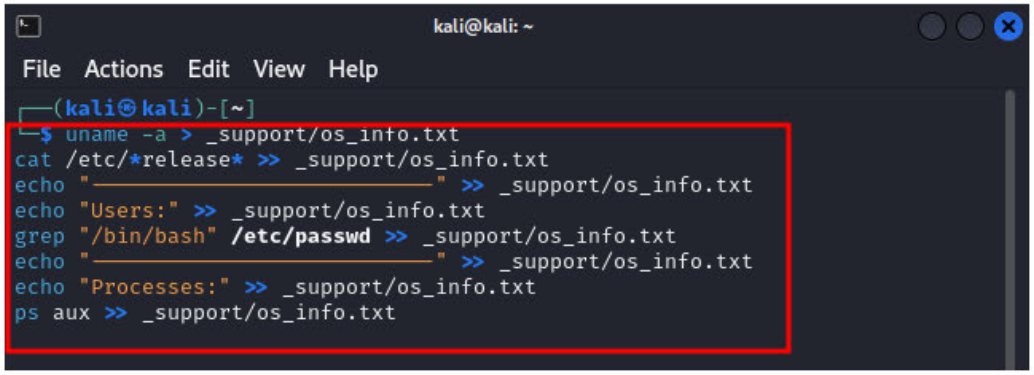
Step 4: Get Operation System Information

This series of commands collects operating system information such as kernel version, distribution info, users list, and processes.

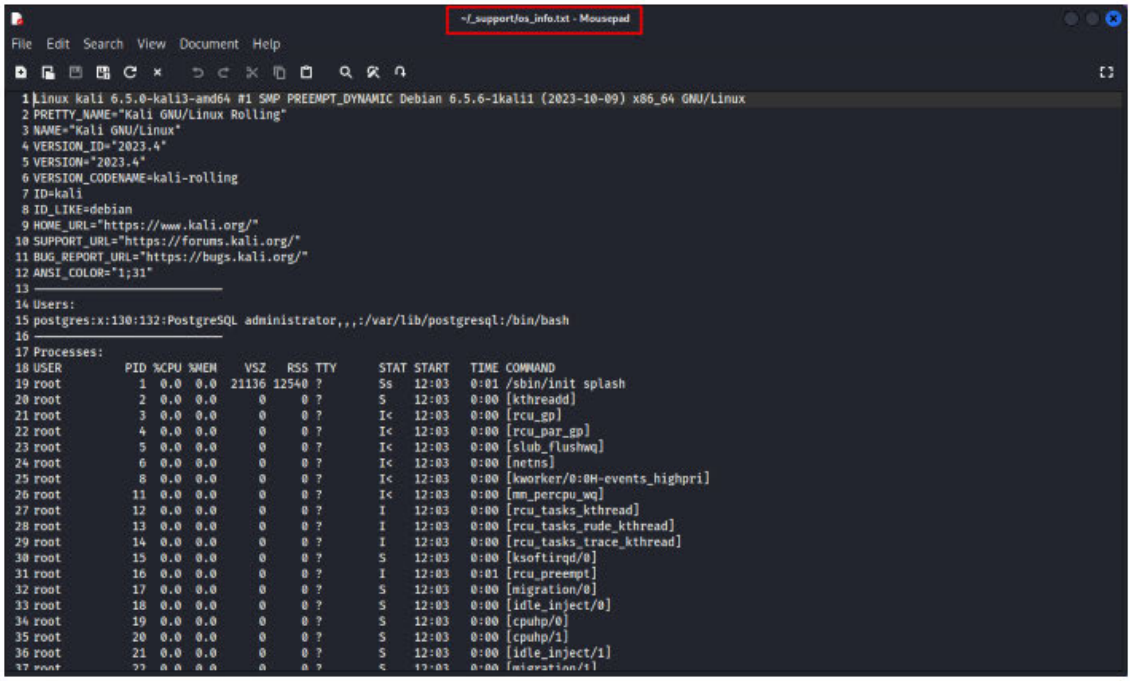
It appends each piece of information to a text file named os_info.txt in the _support directory.

```
uname -a > _support/os_info.txt
cat /etc/*release* >> _support/os_info.txt
echo "---------------------------" >> _support/os_info.txt
echo "Users:" >> _support/os_info.txt
grep "/bin/bash" /etc/passwd >> _support/os_info.txt
echo "---------------------------" >> _support/os_info.txt
echo "Processes:" >> _support/os_info.txt
ps aux >> _support/os_info.txt
```

Next step is running the command in the terminal emulator



Now I'll check if the os_info.txt has all the relevant information



Yes os_info.txt file has all the information about the OS

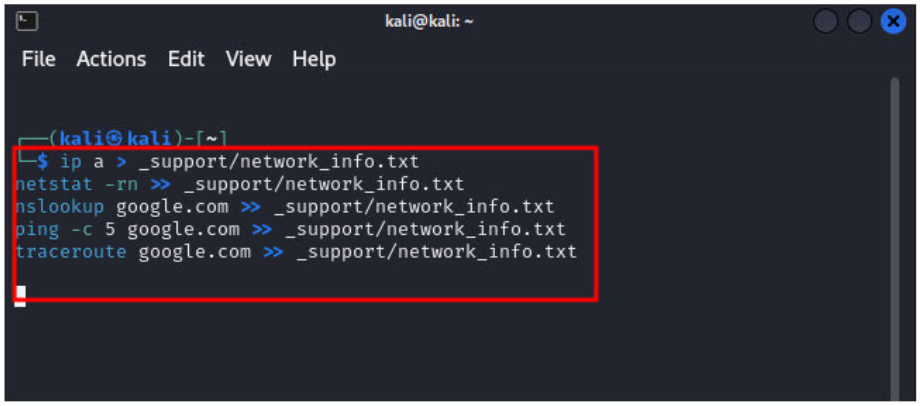
Step 5: Get Network Information

This set of commands retrieves network information including network interfaces, routing table, DNS information, ping results, and traceroute results.

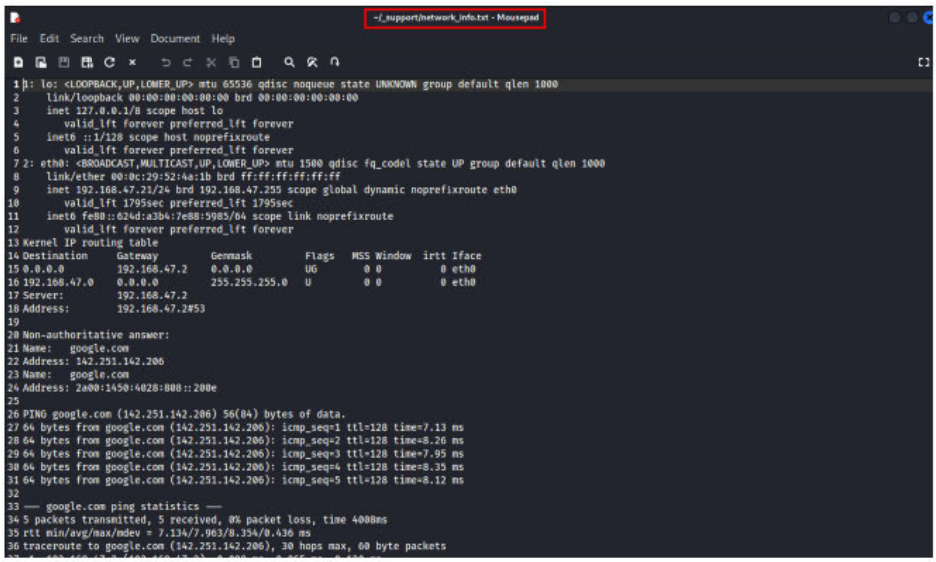
It writes the information to a text file named network_info.txt in the _support directory.

```
ip a > _support/network_info.txt
netstat -rn >> _support/network_info.txt
nslookup google.com >> _support/network_info.txt
ping -c 5 google.com >> _support/network_info.txt
traceroute google.com >> _support/network_info.txt
```

I'll run the commands in the terminal emulator to create an network_info.txt file with all the needed network information in it



Now let's see if it worked



The commands have worked, we have a network_info.txt file with all the relevant network information inside the _support directory

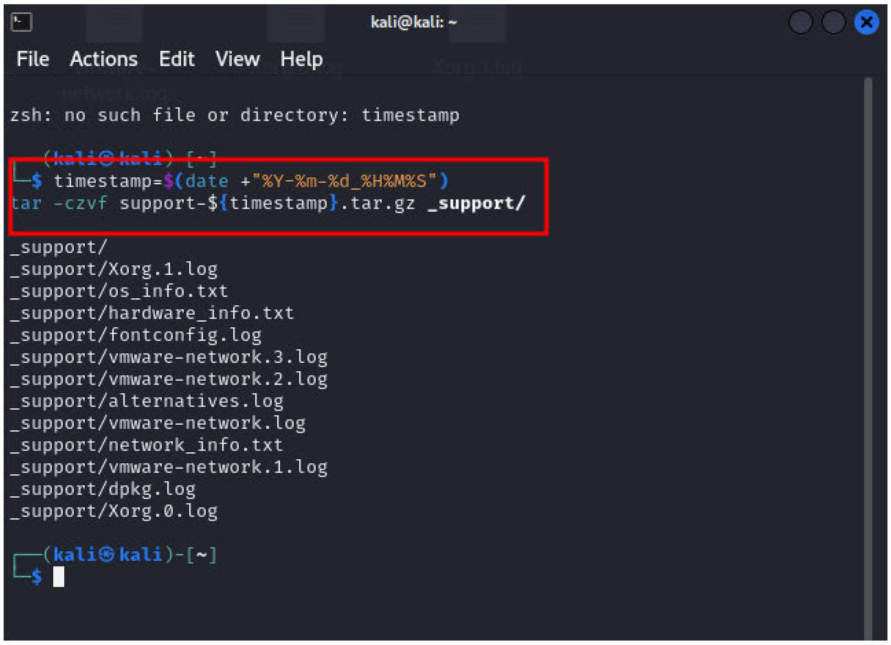
Step 6: Create The Archive

This command generates a timestamp in the format YYYY-MM-DD_HHMMSS

then creates a tar.gz archive file named support-<timestamp>.tar.gz containing all the files and directories within the _support directory.

```
timestamp=$(date +"%Y-%m-%d_%H%M%S")
tar -czvf support-${timestamp}.tar.gz _support/
```

I'll run the command in the terminal emulator



We can see that the command is successful and the archive of all our .log files and our .txt files has been created

Step 7: Creating The Script

In this next step I'll be creating an easy executable script that will allow any person to run it and get all the steps above in one simple quick click

First I'll be getting the script that does all the needed actions, Heres a script that adheres to the requirements

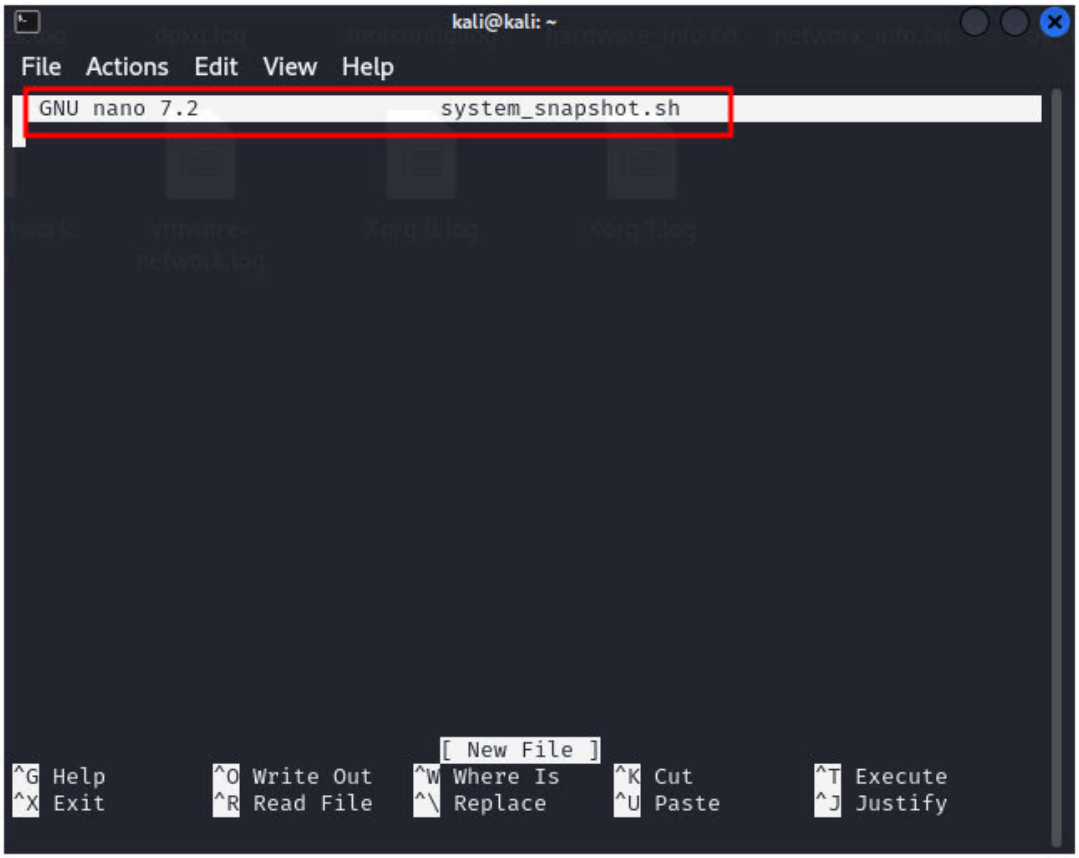
Bash

```bash
#!/bin/bash

# Create temporary directory
mkdir _support

# Copy log files
cp /var/log/*.log _support/

# Gather Hardware Information
# - CPU
echo "CPU Information:" >> _support/hardware_info.txt
lscpu | grep -E "model name|CPU cores|MHz" >> _support/hardware_info.tx
# - Memory
echo "Memory Information:" >> _support/hardware_info.txt
free -m | grep "Mem" >> _support/hardware_info.txt
# - Storage
echo "Disk Usage:" >> _support/hardware_info.txt
df -h | grep -vE "^Filesystem|tmpfs" >> _support/hardware_info.txt
# - Peripheral Devices (limited - adjust as needed)
echo "USB Devices:" >> _support/hardware_info.txt
lsusb | grep Bus >> _support/hardware_info.txt

# Gather OS Information
# - Kernel Version
echo "Kernel Version:" >> _support/os_info.txt
uname -r >> _support/os_info.txt
# - Distribution Info
echo "Distribution Information:" >> _support/os_info.txt
cat /etc/lsb-release | grep -E "DISTRIB_ID|DISTRIB_RELEASE" >> _support/
# - Users List (non-sensitive information)
echo "User Accounts:" >> _support/os_info.txt
cat /etc/passwd | grep -vE ":x:|/sbin/nologin" | cut -d: -f1 >> _suppor
# - Processes (basic info)
echo "Running Processes:" >> _support/os_info.txt
ps -eo pid,user,comm | head -n 10 >> _support/os_info.txt

# Gather Network Information
# - Network Interfaces (active)
echo "Network Interfaces:" >> _support/network_info.txt
ip addr show | grep -E "^link|^inet " >> _support/network_info.txt
# - Routing Table
echo "Routing Table:" >> _support/network_info.txt
ip route show | grep default >> _support/network_info.txt
# - DNS Information
echo "DNS Servers:" >> _support/network_info.txt
systemd-resolve --status | grep nameserver >> _support/network_info.txt
# - Network Check (ping example)
ping -c 3 8.8.8.8 >> _support/network_info.txt 2>&1  # Redirect standar

# Create Archive
current_time=$(date +%Y-%m-%d_%H%M%S)
tar -czvf support-${current_time}.tar.gz _support/

# Informative message
echo "System information snapshot created successfully!"
echo "Archive: support-${current_time}.tar.gz"

# Note: Script doesn't include traceroute (potentially time-consuming)
```

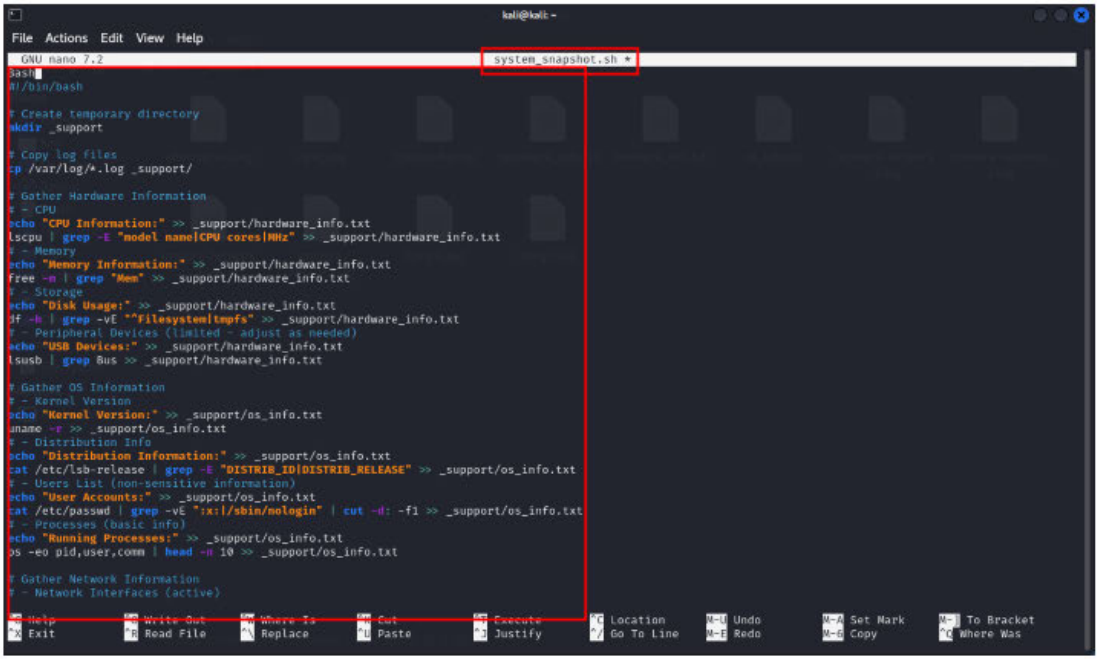
In the next step I'll be using the nano command to create a new file and open it in the nano text editor

```
nano system_snapshot.sh
```

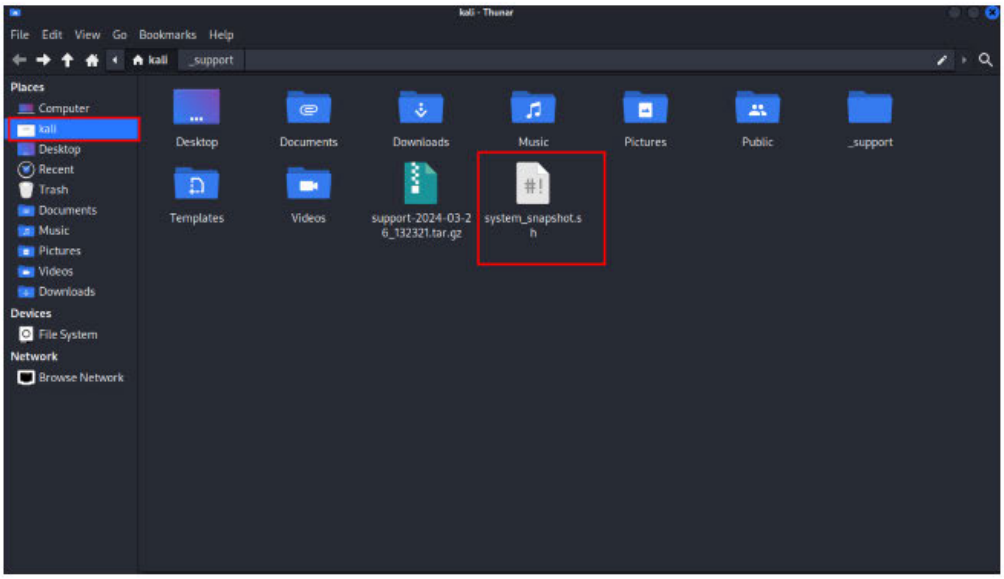
Run the command in the terminal emulator



Now were using the nano 7.2 txt editor to create the sh script
I'll copy the script the system_snapshot.sh file & I'll save it



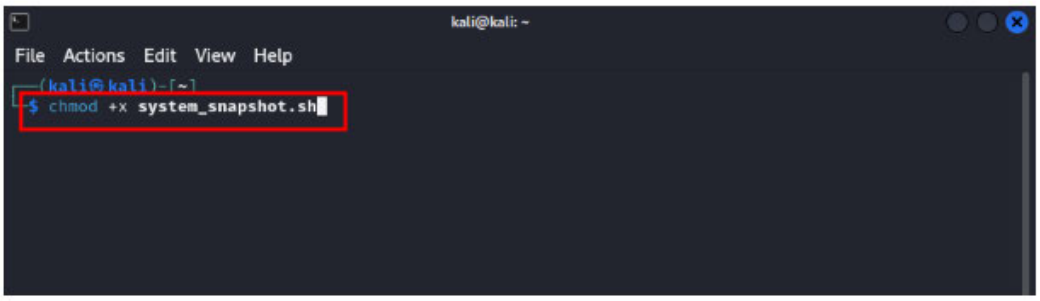
We can see that the script has been created using the nano editor



Next I'll make the script executable by running the following command Replacing the name to system_snapshot.sh

```
chmod +x script_name.sh
```

I'll run the command

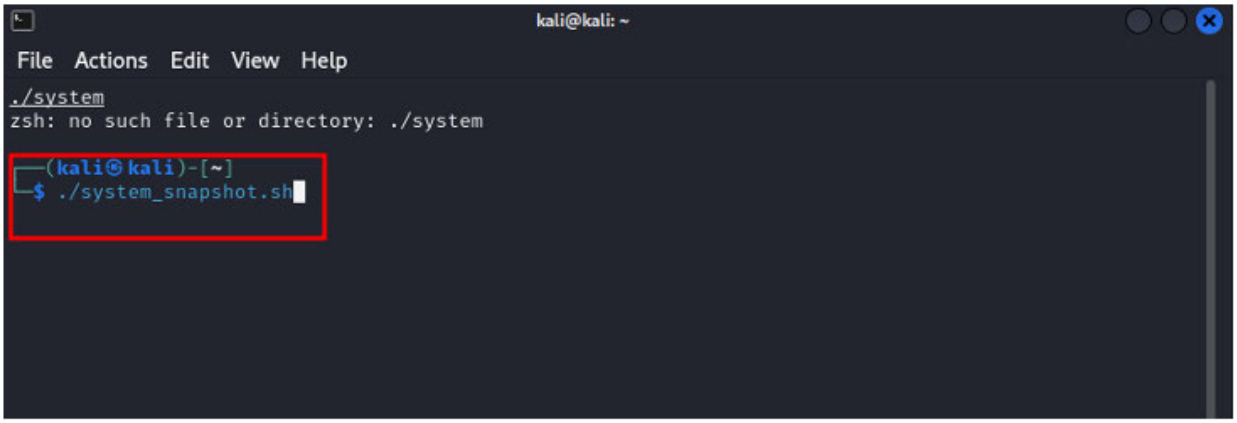


Now the script should be executable

This next command will execute the script, creating the system snapshot as described in the task.

```
./script_name.sh
```

I'll run it now



```
kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ ./system_snapshot.sh
./system_snapshot.sh: 1: Bash: not found
mkdir: cannot create directory '_support': File exists
cp: cannot open '/var/log/boot.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.1.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.2.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.3.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmtoolsd-kali.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmtoolsd-root.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-kali.log' for reading: Permission denied
cat: /etc/lsb-release: No such file or directory
./system_snapshot.sh: 47: systemd-resolve: not found
_support/
_support/Xorg.1.log
_support/os_info.txt
_support/hardware_info.txt
_support/fontconfig.log
_support/vmware-network.3.log
_support/vmware-network.2.log
_support/alternatives.log
_support/vmware-network.log
_support/network_info.txt
_support/vmware-network.1.log
_support/dpkg.log
_support/Xorg.0.log
System information snapshot created successfully!
Archive: support-2024-03-26_134328.tar.gz

┌──(kali㉿kali)-[~]
└─$
```

We can see that the script is working correctly!