Final Project – Linux Essentials

Name: Samuel Kim

Educator: Boris Frenkel

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.

Table of content:

Step #1: _support Folder

First, let's create a folder (_support) using the command (mkdir) in which the required files will be stored

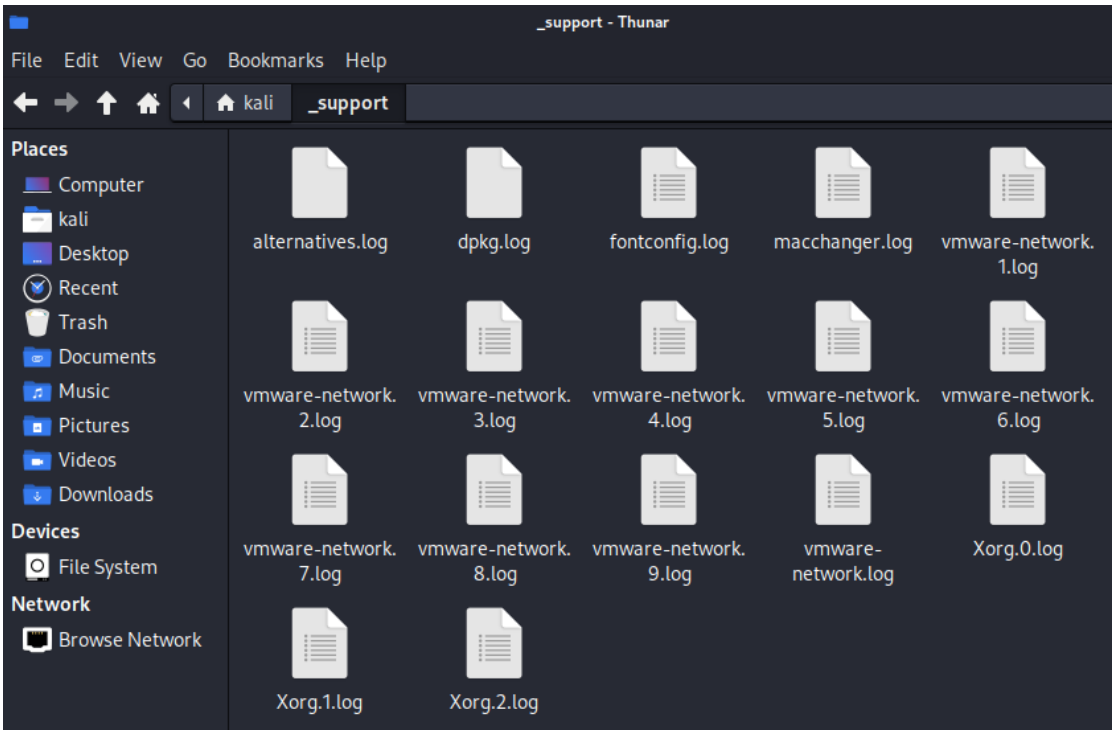
```
File  Actions  Edit  View  Help
┌──(oxy㉿OS)-[~]
└─$ mkdir _support

┌──(oxy㉿OS)-[~]
└─$ ls
Desktop    Downloads  Pictures  _support  Videos
Documents  Music      Public    Templates
```

Step #2: Log Files

Now we will transfer all records or Logs to a new folder. As we can see, there are Logs that we cannot transfer because they require root user permission.

In Linux, the /var/log directory is where system and application logs are stored. It contains various log files that record information about system events, user activities, and application behavior. These logs are essential for troubleshooting issues, monitoring system performance, and maintaining security.

```
┌──(oxy㉿OS)-[~]
└─$ cp /var/log/*.log _support
cp: cannot open '/var/log/boot.log' for reading: Permiss
ion denied
cp: cannot open '/var/log/vmware-vmsvc-root.1.log' for r
eading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.2.log' for r
eading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.3.log' for r
eading: Permission denied
cp: cannot open '/var/log/vmware-vmsvc-root.log' for rea
ding: Permission denied
cp: cannot open '/var/log/vmware-vmtoolsd-kali.log' for
reading: Permission denied
cp: cannot open '/var/log/vmware-vmtoolsd-oxy.log' for r
eading: 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 rea
```



Step #3: Hardware Information

The next thing we must do is to give the current information about our device, information about the Processor, RAM, Storage, Peripheral Devises etc. This info is good for monitoring system performance and troubleshooting of course.

Processor (CPU):

Command: lscpu

Description: Displays CPU architecture information.

Example command and redirection to a text file: lscpu > _support/cpu_info.txt

Memory (RAM):

Command: free -h

Description: Shows information about total, used, and free memory.

Example command and redirection to a text file: free -h > _support/memory_info.txt

Storage (Disk):

Command: df -h

Description: Displays disk space usage.

Example command and redirection to a text file: df -h > _support/disk_info.txt

Peripheral Devices:

Command: lsusb (for USB devices), lspci (for PCI devices), lsblk (for block devices)

Description: Lists USB devices, PCI devices, and block devices respectively.

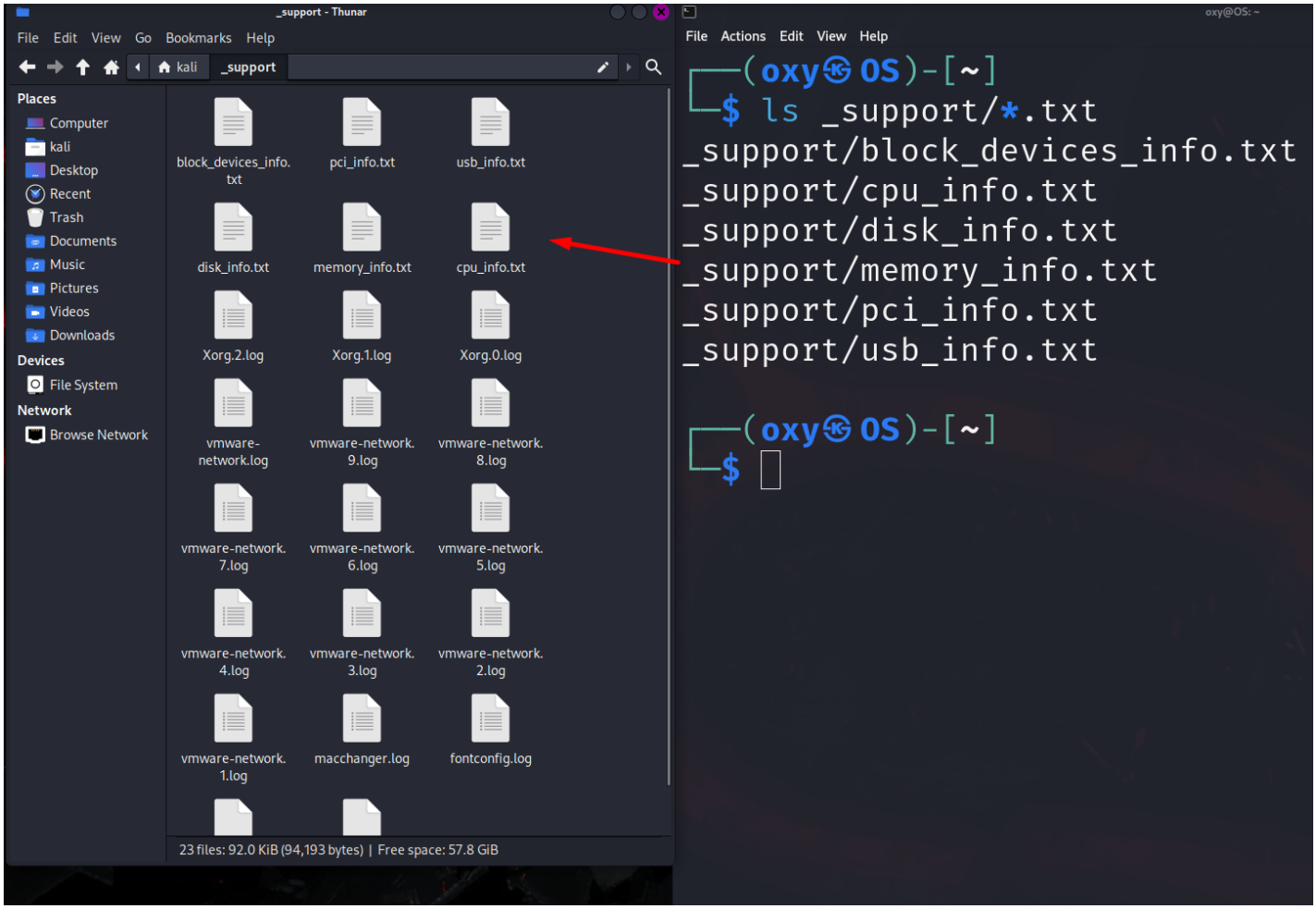
Example command and redirection to a text file:

lsusb > _support/usb_info.txt

lspci > _support/pci_info.txt

lsblk > _support/block_devices_info.txt

```
┌──(oxy㉿OS)-[~]
└─$ lscpu > _support/cpu_info.txt

┌──(oxy㉿OS)-[~]
└─$ free -h > _support/memory_info.txt

┌──(oxy㉿OS)-[~]
└─$ df -h > _support/disk_info.txt

┌──(oxy㉿OS)-[~]
└─$ lsusb > _support/usb_info.txt

┌──(oxy㉿OS)-[~]
└─$ lspci > _support/pci_info.txt

┌──(oxy㉿OS)-[~]
└─$ lsblk > _support/block_devices_info.txt
```



Step #4: OS Info

Relevant information about the operating system is crucial for system administration, troubleshooting, security, compliance, and resource allocation. It helps administrators manage, maintain, secure, and optimize system performance effectively.

Kernel Version:

Command: uname -a

Description: Displays kernel version information.

Example command and redirection to a text file: uname -a > _support/kernel_info.txt

Distribution Info:

Command: cat /etc/*release*

Description: Shows distribution-specific information.

Example command and redirection to a text file: cat /etc/*release* > _support/distribution_info.txt

Users List:

Command: cat /etc/passwd

Description: Lists all users on the system.

Example command and redirection to a text file: cat /etc/passwd > _support/users_list.txt

Processes:

Command: ps aux

Description: Displays information about running processes.

Example command and redirection to a text file: ps aux > _support/processes_info.txt

```
File  Actions  Edit  View  Help
┌──(oxy㉿OS)-[~]
└─$ uname -a > _support/kernel_info.txt

┌──(oxy㉿OS)-[~]
└─$ cat /etc/*release* > _support/distribution_info.txt

┌──(oxy㉿OS)-[~]
└─$ cat /etc/passwd > _support/users_list.txt

┌──(oxy㉿OS)-[~]
└─$ ps aux > _support/processes_info.txt
```

Step #5: Network

Gathering comprehensive details about the network is vital for diagnosing problems, fine-tuning performance, safeguarding against threats, planning future needs, meeting regulations, and managing network operations effectively.

Network Interfaces:

Command: ip addr show

Description: Displays information about network interfaces.

Example command and redirection to a text file: ip addr show > _support/network_interfaces.txt

Routing Table:

Command: ip route show

Description: Shows the routing table.

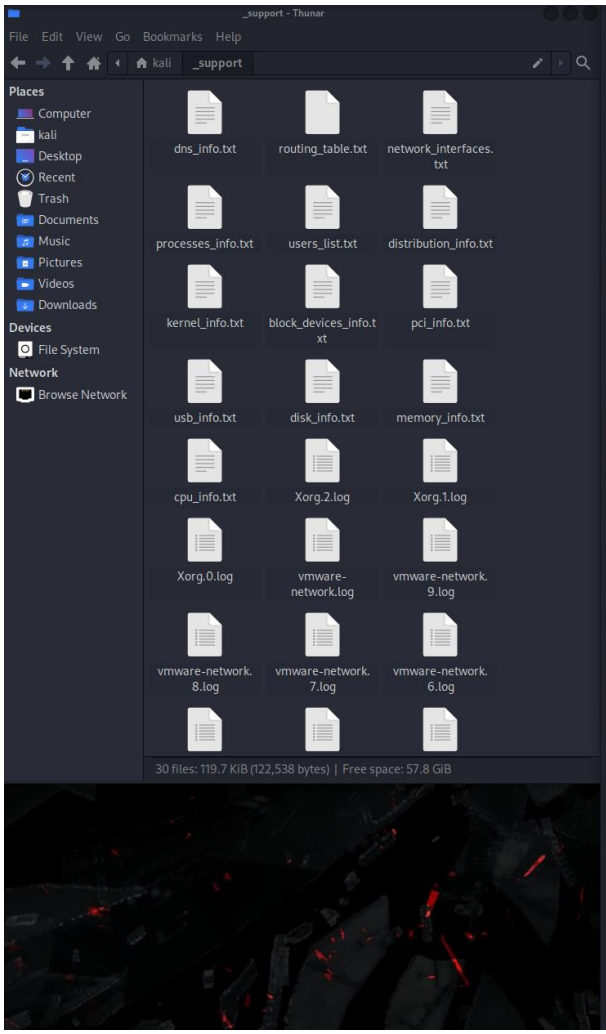
Example command and redirection to a text file: ip route show > _support/routing_table.txt

DNS Information:

Command: cat /etc/resolv.conf

Description: Displays DNS configuration.

Example command and redirection to a text file: cat /etc/resolv.conf > _support/dns_info.txt


_support - Thunar
File Edit View Go Bookmarks Help
kali
_support
Places
Computer
kali
Desktop
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
dns_info.txt
routing_table.txt
network_interfaces.txt
processes_info.txt
users_list.txt
distribution_info.txt
kernel_info.txt
block_devices_info.txt
pci_info.txt
usb_info.txt
disk_info.txt
memory_info.txt
cpu_info.txt
Xorg.2.log
Xorg.1.log
Xorg.0.log
vmware-network.log
vmware-network.9.log
vmware-network.8.log
vmware-network.7.log
vmware-network.6.log
30 files: 119.7 KiB (122,538 bytes) | Free space: 57.8 GiB


oxy@OS: ~
File Actions Edit View Help
┌──(oxy㉿OS)-[~]
└─$ ip addr show > _support/network_interfaces.txt
┌──(oxy㉿OS)-[~]
└─$ ip route show > _support/routing_table.txt
┌──(oxy㉿OS)-[~]
└─$ cat /etc/resolv.conf > _support/dns_info.txt
┌──(oxy㉿OS)-[~]
└─$ ls _support/*.txt
_support/block_devices_info.txt
_support/cpu_info.txt
_support/disk_info.txt
_support/distribution_info.txt
_support/dns_info.txt
_support/kernel_info.txt
_support/memory_info.txt
_support/network_interfaces.txt
_support/pci_info.txt
_support/processes_info.txt
_support/routing_table.txt
_support/usb_info.txt
_support/users_list.txt
┌──(oxy㉿OS)-[~]
└─$

Ping & Traceroute

Ping and Traceroute commands help diagnose network issues. Ping checks if a remote host is reachable and measures latency. Traceroute traces the route packets take, identifying network delays and failures.

Network Checking (Ping):

Command: ping -c 4 Google's public DNS servers.

Description: Tests connectivity to a target by sending ICMP echo requests.

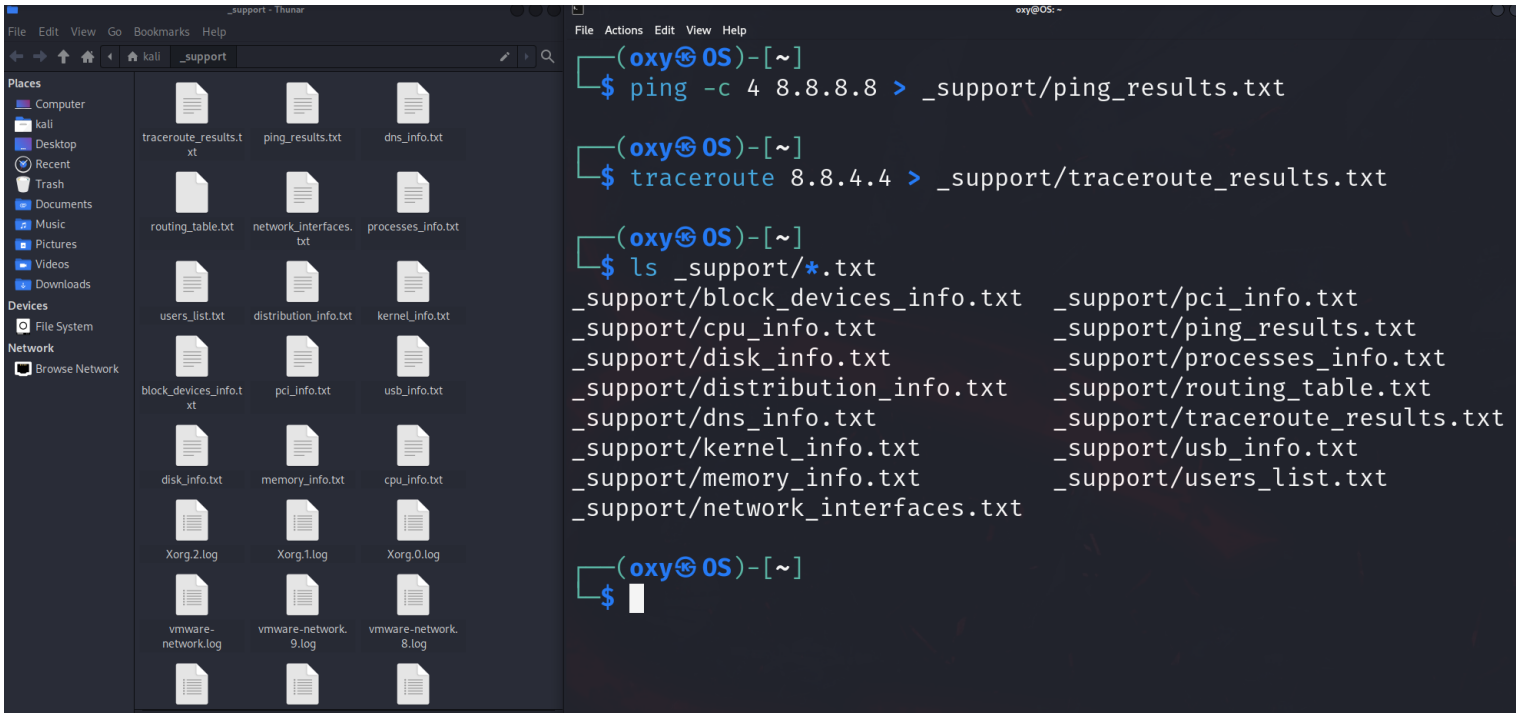
Example command and redirection to a text file: ping -c 4 8.8.8.8 > _support/ping_results.txt

Network Checking (Traceroute):

Command: traceroute Google's public DNS servers.

Description: Displays the route packets take to reach the target.

Example command and redirection to a text file: traceroute 8.8.4.4 > _support/traceroute_results.txt



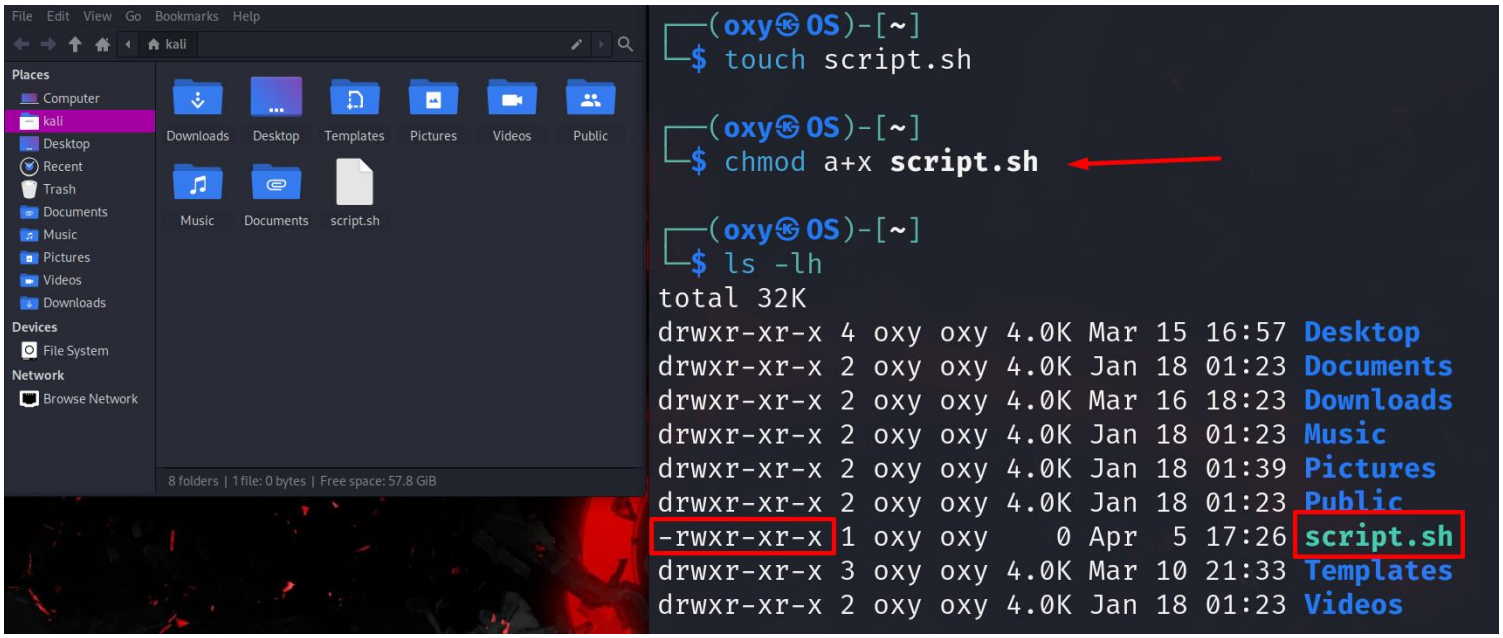
Step #6: Full Information - Script.sh

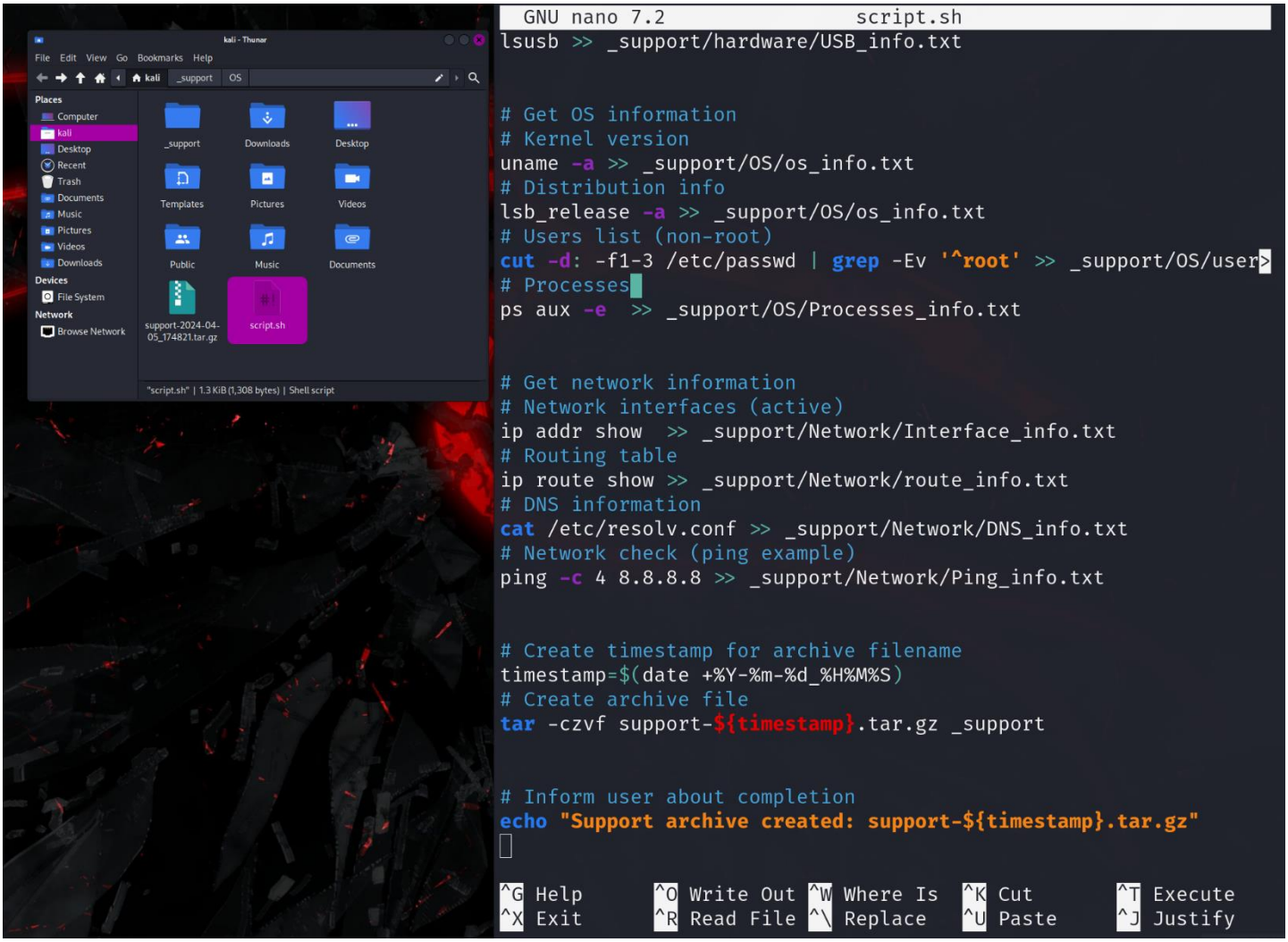
Creating scripts in Linux is essential for automating repetitive tasks, ensuring consistency, improving efficiency, offering flexibility, enabling repeatability, simplifying maintenance, and facilitating scalability.

Allow for the automation of repetitive tasks, reducing the need for manual intervention and saving time. By writing a script, you can execute a series of commands or operations with a single command, streamlining workflows and increasing efficiency.

By consolidating multiple commands or operations into a script, you can execute complex tasks more efficiently. Scripts enable you to streamline processes, eliminate redundant steps, and optimize resource utilization, ultimately improving productivity and reducing overhead.

First, we need to create a file (Script.sh) and give it execute permission. The file format itself does not matter, and for now it is just a file that we will turn into a script using commands in the next step.





```bash
#!/bin/bash


mkdir _support
mkdir _support/Log
mkdir _support/hardware
mkdir _support/OS
mkdir _support/Network


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



# Get hardware information
# CPU
lscpu >> _support/hardware/CPU_info.txt
# Memory
free -m | grep 'Mem' >> _support/hardware/Memory_info.txt
# Storage (list disks)
lsblk -d >> _support/hardware/Storage_info.txt
# Peripheral devices (USB)
lsusb >> _support/hardware/USB_info.txt


# Get OS information
# Kernel version
uname -a >> _support/OS/os_info.txt
# Distribution info
lsb_release -a >> _support/OS/os_info.txt
# Users list (non-root)
cut -d: -f1-3 /etc/passwd | grep -Ev '^root' >> _support/OS/users_info.txt
# Processes
ps aux -e  >> _support/OS/Processes_info.txt


# Get network information
# Network interfaces (active)
ip addr show  >> _support/Network/Interface_info.txt
# Routing table
ip route show >> _support/Network/route_info.txt
# DNS information
cat /etc/resolv.conf >> _support/Network/DNS_info.txt
# Network check (ping example)
ping -c 4 8.8.8.8 >> _support/Network/Ping_info.txt


# Create timestamp for archive filename
timestamp=$(date +%Y-%m-%d_%H%M%S)
# Create archive file
tar -czvf support-${timestamp}.tar.gz _support


# Inform user about completion
echo "Support archive created: support-${timestamp}.tar.gz"
```

After writing the script, let's run it and see if it works. After writing the script, let's run it and see if it works, for this we use the command ./script.sh

Everything worked and a folder with the necessary files was created, as well as an archive with the time information!

```
┌──(oxy㉿OS)-[~]
└─$ tree _support
_support
├── hardware
│   ├── CPU_info.txt
│   ├── Memory_info.txt
│   ├── Storage_info.txt
│   └── USB_info.txt
├── Log
│   ├── alternatives.log
│   ├── dpkg.log
│   ├── fontconfig.log
│   ├── macchanger.log
│   ├── vmware-network.1.log
│   ├── vmware-network.2.log
│   ├── vmware-network.3.log
│   ├── vmware-network.4.log
│   ├── vmware-network.5.log
│   ├── vmware-network.6.log
│   ├── vmware-network.7.log
│   ├── vmware-network.8.log
│   ├── vmware-network.9.log
│   ├── vmware-network.log
│   ├── Xorg.0.log
│   ├── Xorg.1.log
│   └── Xorg.2.log
├── Network
│   ├── DNS_info.txt
│   ├── Interface_info.txt
│   ├── Ping_info.txt
│   └── route_info.txt
└── OS
    ├── os_info.txt
    ├── Processes_info.txt
    └── users_info.txt

5 directories, 28 files
```

```
┌──(oxy㉿OS)-[~]
└─$ ls -lh
total 64K
drwxr-xr-x 4 oxy oxy 4.0K Mar 15 16:57 Desktop
drwxr-xr-x 2 oxy oxy 4.0K Jan 18 01:23 Documents
drwxr-xr-x 2 oxy oxy 4.0K Mar 16 18:23 Downloads
drwxr-xr-x 2 oxy oxy 4.0K Jan 18 01:23 Music
drwxr-xr-x 2 oxy oxy 4.0K Apr  5 17:52 Pictures
drwxr-xr-x 2 oxy oxy 4.0K Jan 18 01:23 Public
-rwxr-xr-x 1 oxy oxy 1.3K Apr  5 17:48 script.sh
drwxr-xr-x 6 oxy oxy 4.0K Apr  5 17:48 _support
-rw-r--r-- 1 oxy oxy  22K Apr  5 17:48 support-2024-04-05_174821.
tar.gz
drwxr-xr-x 3 oxy oxy 4.0K Mar 10 21:33 Templates
drwxr-xr-x 2 oxy oxy 4.0K Jan 18 01:23 Videos
```