Linux Essentials – Final Work

Shaked Uriel Brami – 213164379

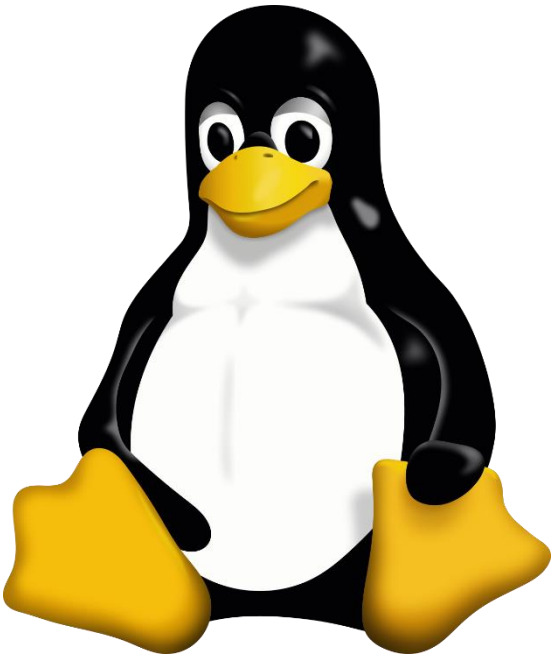
CSPP86

Instructor - Boris Frenkel

Date of Submission: 15/3/24

Table of Contents

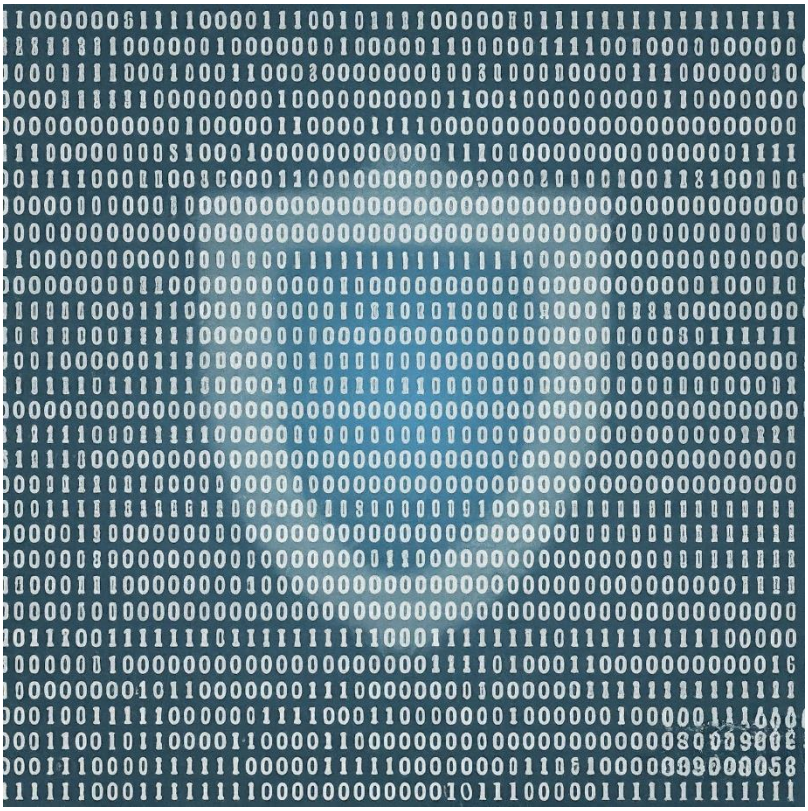
Table of Contents



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:

For this project, we got instructions, and steps for creating the script.

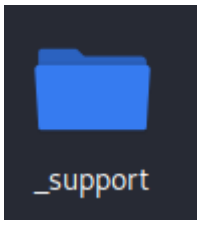
We will have to break it down step-by-step.

So let's start with the first step.

Create the temporary directory which names _support in your current placement

In order to create a folder, we will have to run "mkdir _support" command in the terminal.

Proof Of Work:



Copy the log files to the created directory. You should copy all the *.log files which are in the directory /var/log.

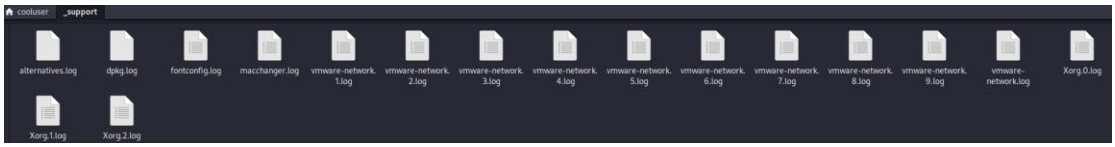
To copy all the .log files from the /var/log directory to the previously created "_support" directory, we can use the cp command along with the wildcard (*) to match all .log files.

This is the command which will transfer us the log files.

cp /var/log/*.log _support/

Proof Of Work:

```
┌──(cooluser㉿kali)-[~]
└─$ cp /var/log/*.log _support/
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-cooluser.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-cooluser.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-kali.log' for reading: Permission denied
```



that's our script meanwhile

```
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
```

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.

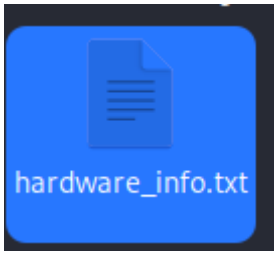
The command (lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt retrieves information about the CPU, memory, storage, USB devices, and PCI devices and stores it in a file named "hardware_info.txt" within the "_support" directory.

So we will run this command

(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt

Proof Of Work

```
┌──(cooluser㉿kali)-[~]
└─$ (lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
```



```
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Address sizes:                   48 bits physical, 48 bits virtual
Byte Order:                      Little Endian
CPU(s):                          4
On-line CPU(s) list:             0-3
Vendor ID:                       GenuineIntel
Model name:                      11th Gen Intel(R) Core(TM) i9-13900HX
CPU family:                      6
Model:                           183
Thread(s) per core:              1
Core(s) per socket:              2
Socket(s):                       2
Stepping:                        1
BogoMIPS:                        4038.40
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon nopl tsc_reliable nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 sse4_1 sse4_2
  movbe popcnt aes xsave avx hypervisor lahf_lm 3dnowprefetch pti arat
Hypervisor vendor:               VMware
Virtualization type:             full
L1d cache:                       192 KiB (4 instances)
L1i cache:                       128 KiB (4 instances)
L2 cache:                        8 MiB (4 instances)
L3 cache:                        72 MiB (2 instances)
NUMA node(s):                    1
NUMA node0 CPU(s):               0-3
Vulnerability Gather data sampling: Not affected
Vulnerability Itlb multihit:     KVM: Mitigation: VMX unsupported
Vulnerability L1tf:              Mitigation; PTE Inversion
Vulnerability Mds:               Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Vulnerability Meltdown:          Mitigation; PTI
Vulnerability Mmio stale data:   Unknown: No mitigations
Vulnerability Retbleed:          Not affected
Vulnerability Spec rstack overflow: Not affected
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Retpolines, STIBP disabled, RSB filling, PBRSB-eIBRS Not affected
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Not affected
              total        used        free      shared  buff/cache   available
Mem:          6.2Gi       1.1Gi       4.7Gi       19Mi       738Mi       5.2Gi
Swap:         1.0Gi          0B       1.0Gi
Filesystem      Size  Used Avail Use% Mounted on
udev            3.3G     0  3.3G   0% /dev
tmpfs           642M  1.4M  641M   1% /run
/dev/sda1        79G   15G   60G  21% /
tmpfs           3.2G     0  3.2G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           642M  128K  642M   1% /run/user/1001
tmpfs           642M  120K  642M   1% /run/user/1000
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 0e0f:0008 VMware, Inc. Virtual Bluetooth Adapter
Bus 001 Device 004: ID 0e0f:0006 VMware, Inc. Virtual Keyboard
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: Broadcom / LSI 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)
```

So that's our script meanwhile

```
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
```

Get all the relevant information about your operating system and its current state: kernel version, distribution info, users list, processes etc.

To gather information about the operating system and its current state, including kernel version, distribution info, users list, processes, etc., and save it in the same folder, we can use the following command:

```
{
  uname -a
  grep '^ID=' /etc/os-release
  grep -v nologin /etc/passwd | cut -d: -f1
  ps aux
} > _support/os_info.txt
```

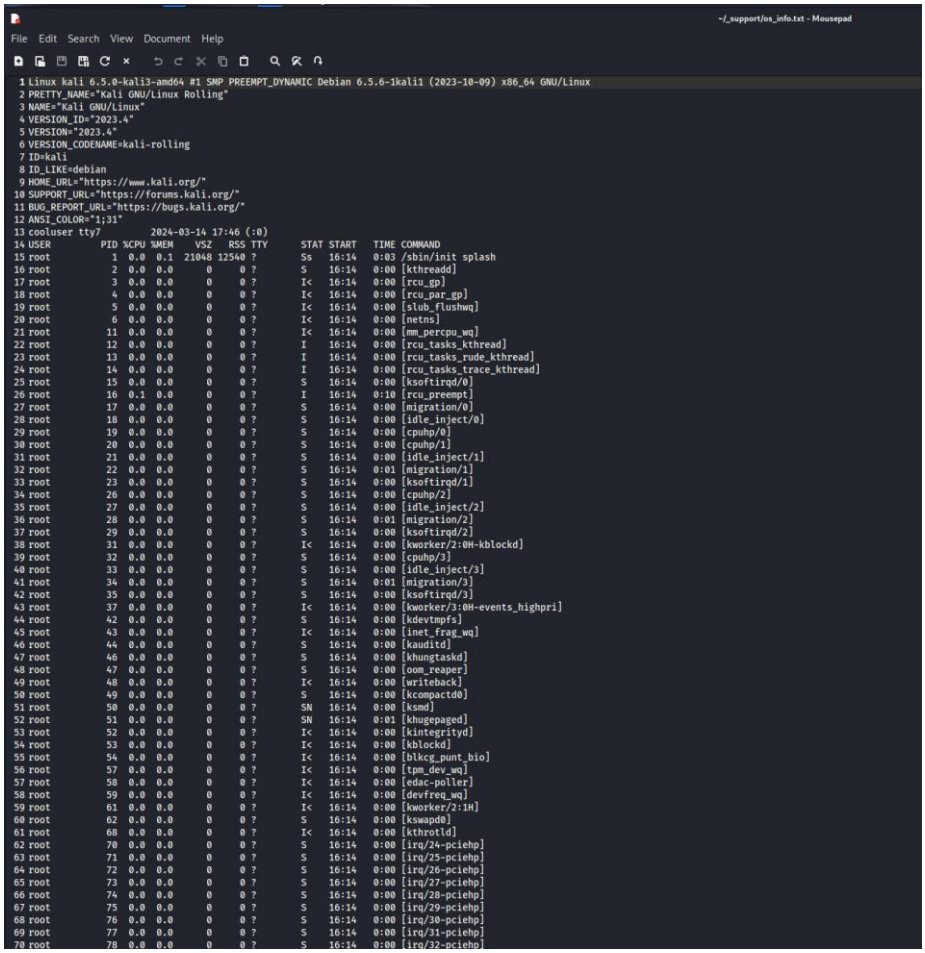
uname -a: Displays system information, including the kernel version.

ps aux: Provides a snapshot of the current processes running on the system.

> _support/os_info.txt: Redirects the combined output of all the commands to a file named "os_info.txt" within the "_support" directory.

Proof of work:

```
┌──(cooluser㉿kali)-[~]
└─$ {
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
```



So that's our script now

```
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
# Get all the relevant information about your operating system and its current state
{
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
```

Get all the relevant information about your network: network interfaces, routing table, DNS information, results of the network checking by ping, traceroute etc.

To gather information about the network, including network interfaces, routing table, DNS information, results of network checking by ping, traceroute, etc., and save it in the same folder, we can use the following command:

```
{
  ip addr
  ip route
  cat /etc/resolv.conf
  ping -c 5 www.google.com
  traceroute -n www.google.com
} > _support/network_info.txt
```

ip addr: This command displays information about network interfaces, including IP addresses assigned to them.

ip route: This command displays the kernel routing table, which shows how packets will be forwarded based on their destination IP addresses.

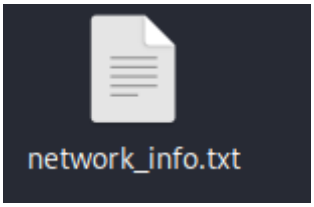
cat /etc/resolv.conf: This command displays the DNS resolver configuration file, which contains information about the DNS servers used by the system.

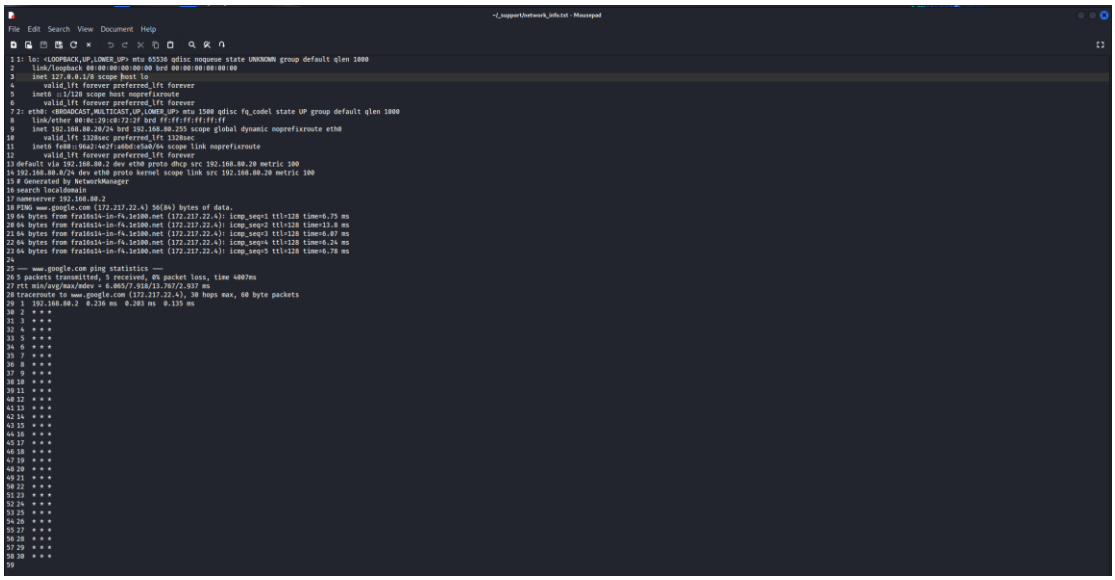
ping -c 5 www.google.com: This command sends ICMP echo requests to the specified host (www.google.com) and waits for responses. It sends 5 packets in total.

traceroute -n www.google.com: This command traces the route that packets take from the local machine to the specified host (www.google.com) by sending packets with increasing TTL values and recording the IP addresses of the routers along the way.

> _support/network_info.txt: Redirects the combined output of all the commands to a file named "network_info.txt" within the "_support" directory.

Proof Of Work:





So that’s our script now:

```
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
# Get all the relevant information about your operating system and its current state
{
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
# Get all the relevant information about your network
{
    ip addr
    ip route
    cat /etc/resolv.conf
    ping -c 5 www.google.com
    traceroute -n www.google.com
} > _support/network_info.txt
```

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.

We can create the archive file containing all the files and directories in the "_support" directory using the following command:

tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support

Explanation:

tar: This command is used to manipulate archives.

-czvf: Options used for creating a compressed archive:

c: Create a new archive.

z: Compress the archive using gzip.

v: Verbose mode to display the files being archived.

f: Specifies the filename of the archive.

"support-$(date +'%Y-%m-%d_%H%M%S').tar.gz": This constructs the filename of the archive using the current date and time in the specified format. $(date +'%Y-%m-%d_%H%M%S') is a command substitution that inserts the current date and time formatted as YYYY-MM-DD_HHMMSS.

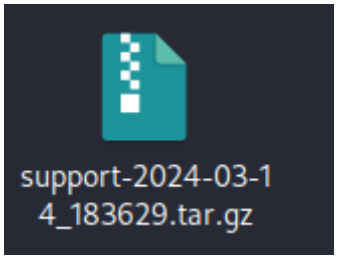
_support: Specifies the directory to be archived.

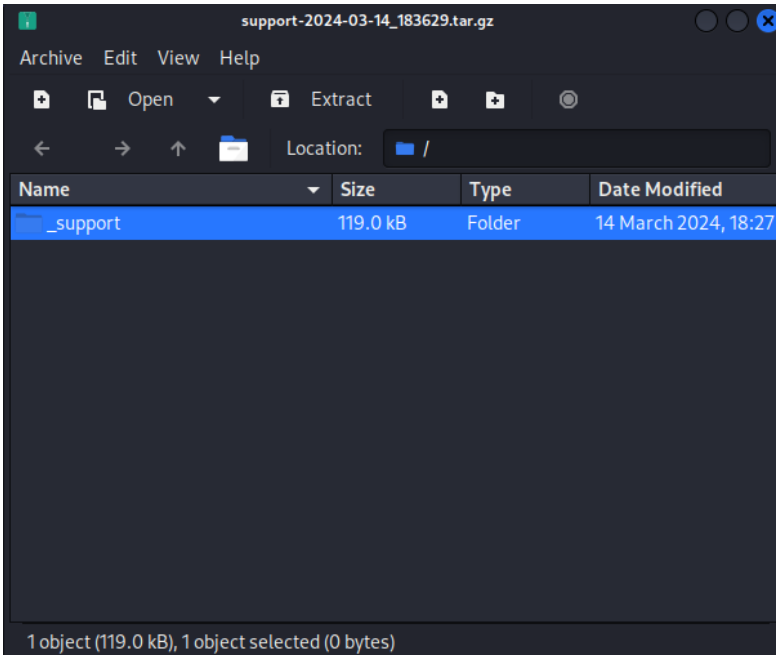
So that's our script now:

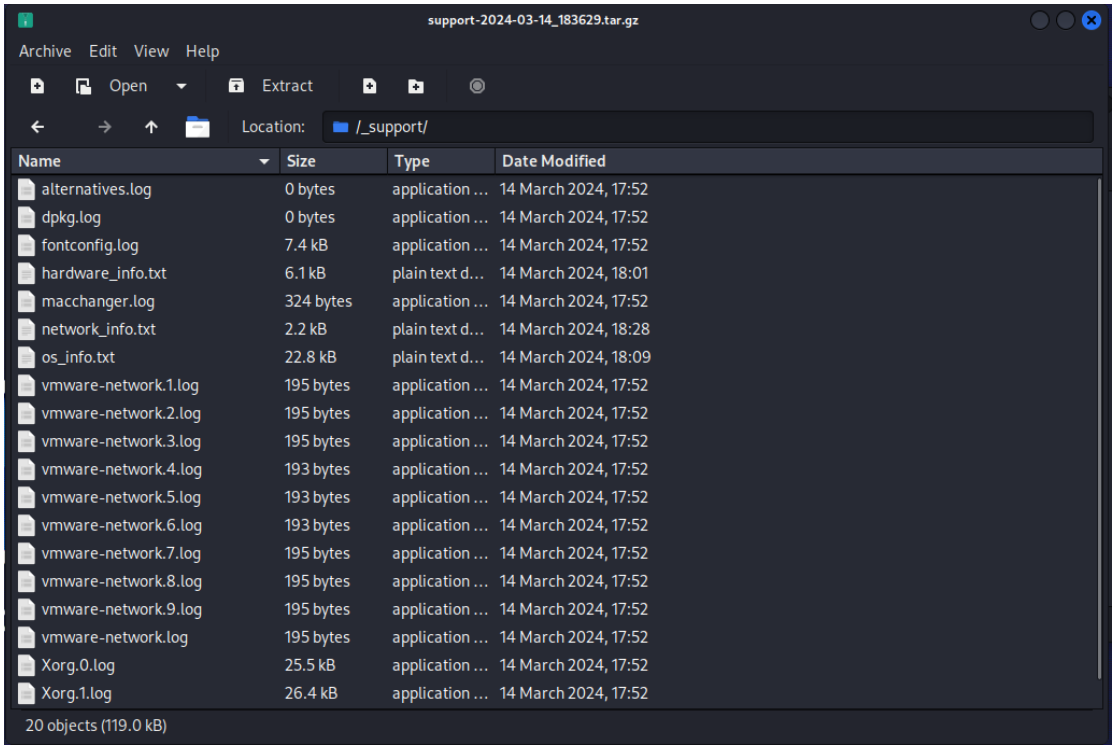
```
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
# Get all the relevant information about your operating system and its current state
{
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
# Get all the relevant information about your network
{
    ip addr
    ip route
    cat /etc/resolv.conf
    ping -c 5 www.google.com
    traceroute -n www.google.com
} > _support/network_info.txt
# Create the archive file
tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support
```

Proof Of Work:

```
┌──(cooluser㉿kali)-[~]
└─$ tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support
_support/
_support/Xorg.1.log
_support/macchanger.log
_support/vmware-network.4.log
_support/os_info.txt
_support/hardware_info.txt
_support/fontconfig.log
_support/vmware-network.6.log
_support/vmware-network.3.log
_support/vmware-network.8.log
_support/vmware-network.7.log
_support/vmware-network.2.log
_support/vmware-network.5.log
_support/alternatives.log
_support/vmware-network.log
_support/network_info.txt
_support/vmware-network.9.log
_support/Xorg.2.log
_support/vmware-network.1.log
_support/dpkg.log
_support/Xorg.0.log
```







Provide the final version of the script and screenshots with successful completion.

After we did everything successfully, we will take everything into one script, and we will try it as one script.

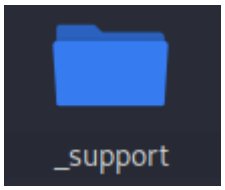
That's my ready script:

```
# Create the temporary directory which names _support in your current placement

mkdir _support

# Copy the log files to the created directory.

cp /var/log/*.log _support/

# Get all the relevant information about your hardware

(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt

# Get all the relevant information about your operating system and its current state

{

   uname -a

   grep '^ID=' /etc/os-release

   grep -v nologin /etc/passwd | cut -d: -f1

   ps aux

} > _support/os_info.txt

# Get all the relevant information about your network

{

   ip addr

   ip route

   cat /etc/resolv.conf

   ping -c 5 www.google.com

   traceroute -n www.google.com

} > _support/network_info.txt

# Create the archive file

tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support
```

now I will run this on my terminal

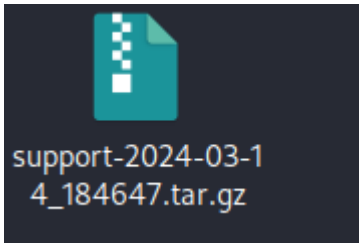
```
┌──(cooluser㉿kali)-[~]
└─$ # Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
# Get all the relevant information about your operating system and its current state
{
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
# Get all the relevant information about your network
{
    ip addr
    ip route
    cat /etc/resolv.conf
    ping -c 5 www.google.com
    traceroute -n www.google.com
} > _support/network_info.txt
# Create the archive file
tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support
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-cooluser.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-cooluser.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-kali.log' for reading: Permission denied
```

we will check that we have the _support folder:

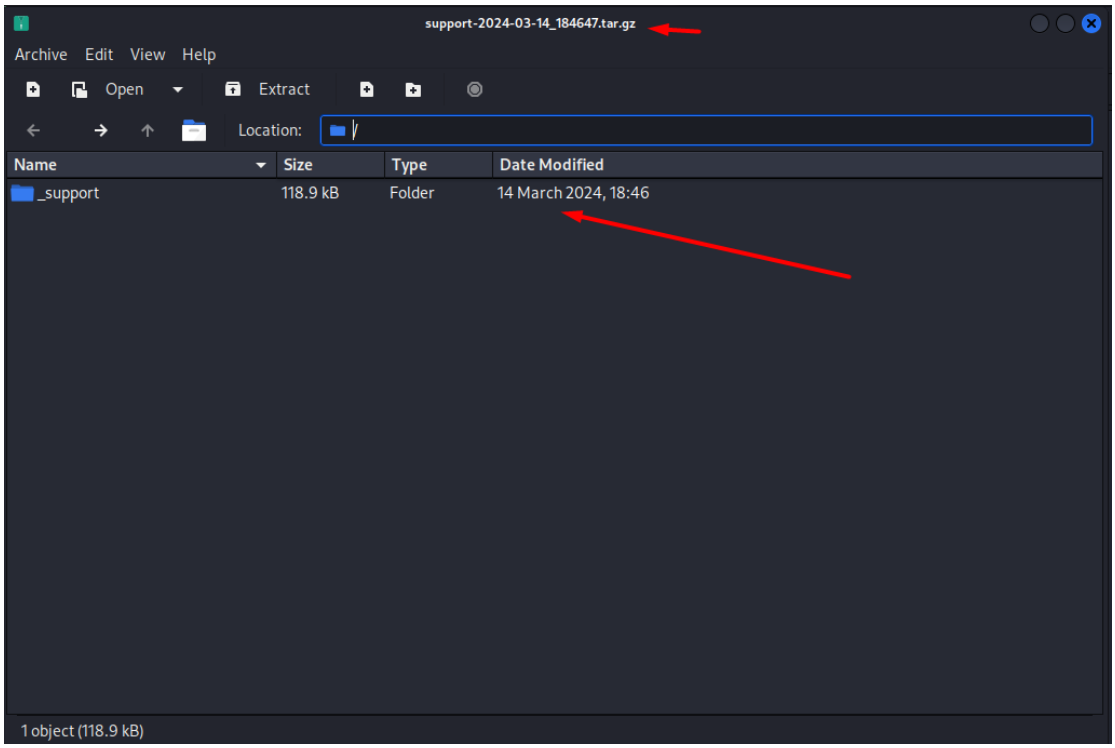


We will check the files there:

Now we will check if the .tar.gz file has been created:



Now we will check the files there too:



We would like to create a script file, that could be easily opened.

To create a script file we would need to write this command

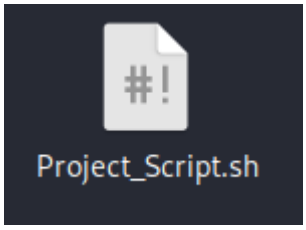
cat > Project_Script.sh

```
┌──(tester1㉿kali)-[~]
└─$ cat > Project_Script.sh
```

After we will enter this command, it will give us a blank space, which there we will need to insert our script.

```
┌──(tester1㉿kali)-[~]
└─$ cat > Project_Script.sh
#!/bin/bash
# Create the temporary directory which names _support in your current placement
mkdir _support
# Copy the log files to the created directory.
cp /var/log/*.log _support/
# Get all the relevant information about your hardware
(lscpu && free -h && df -h && lsusb && lspci) > _support/hardware_info.txt
# Get all the relevant information about your operating system and its current state
{
    uname -a
    grep '^ID=' /etc/os-release
    grep -v nologin /etc/passwd | cut -d: -f1
    ps aux
} > _support/os_info.txt
# Get all the relevant information about your network
{
    ip addr
    ip route
    cat /etc/resolv.conf
    ping -c 5 www.google.com
    traceroute -n www.google.com
} > _support/network_info.txt
# Create the archive file
tar -czvf "support-$(date +'%Y-%m-%d_%H%M%S').tar.gz" _support
```

After we inserted our script to there, we will get out new file.



In order to run it, we will need to open this file for executing for everyone.

We will run this command

chmod +x Project_Script.sh

```
┌──(tester1㉿kali)-[~]
└─$ chmod +x Project_Script.sh
```

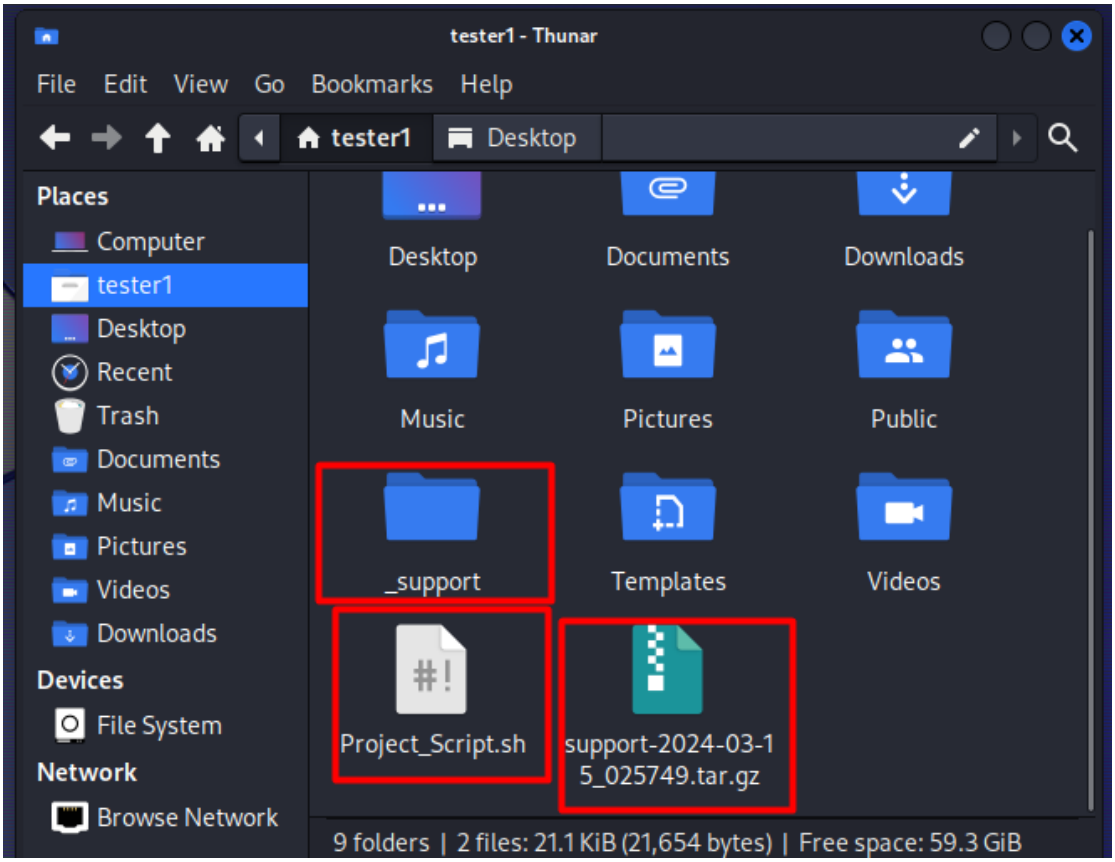
Now to run the script we will need to run this command

./Project_Script.sh

```
┌──(tester1㉿kali)-[~]
└─$ ./Project_Script.sh
```

And we can see its working

```
┌──(tester1㉿kali)-[~]
└─$ ./Project_Script.sh
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-cooluser.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-vmtoolsd-tester1.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-cooluser.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-kali.log' for reading: Permission denied
cp: cannot open '/var/log/vmware-vmusr-tester1.log' for reading: Permission denied
_support/
_support/Xorg.1.log
_support/macchanger.log
_support/vmware-network.4.log
_support/os_info.txt
_support/hardware_info.txt
_support/fontconfig.log
_support/vmware-network.6.log
_support/vmware-network.3.log
_support/vmware-network.8.log
_support/vmware-network.7.log
_support/vmware-network.2.log
_support/vmware-network.5.log
_support/alternatives.log
_support/vmware-network.log
_support/network_info.txt
_support/vmware-network.9.log
_support/Xorg.2.log
_support/vmware-network.1.log
_support/dpkg.log
_support/Xorg.0.log
```



The script file will be available in the teams assignment too