

MALWARE REMEDIATION SYSTEM: WALL GARDEN

MUHAMMAD IZZUDIN BIN ROZALI



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This report is submitted in partial fulfilment of the requirements for the Bachelor of
Computer Science (Computer Network)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

BORANG PENGESAHAN STATUS TESIS*

JUDUL: MALWARE REMEDIATION SYSTEM: WALL GARDEN

SESI PENGAJIAN: SESI 2016/2017

Saya MUHAMMAD IZZUDIN BIN ROZALI

(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan
Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)



(Mengandungi maklumat yang berdarjah
keselamatan atau kepentingan Malaysia
seperti yang termaktub di dalam AKTA
RAHSIA RASMI 1972)

(Mengandungi maklumat TERHAD yang
telah ditentukan oleh organisasi/badan di
mana penyediaan dijalankan)

/

TIDAK TERHAD

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



(TANDATANGAN PENULIS)

Jb622 KM14 JALAN JASA,
KAMPUNG UMBAI ,
77300 MERLIMAU,
MELAKA

Tarikh: 22 Ogos 2017



(TANDATANGAN PENYELIA)

DR. Robiah bt Yusof

Tarikh: 22/8/2017

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak
Berkuasa.

DECLARATION

I hereby declare that this project report entitled

MALWARE REMEDIATION SYSTEM: WALL GARDEN



UTeM

is written by me and is my own effort and that no part has been

plagiarized without citations.

اونيورسيتي تيكنيكل ماليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

STUDENT:  Date: 22/8/2017
(MUHAMMAD IZZUDIN BIN ROZALI)

SUPERVISOR:  Date: 22/8/2017
(DR. ROBIAH BINTI YUSOF)

DEDICATION

This thesis is dedicated to my parents Rozali bin Hamid and Anisah bt Haji Wahid. Also to all 3 BITC student cohort 2014 who are struggling to finish their last semester with a flying colour result.



ACKNOWLEDGEMENTS

First and foremost I would like express my gratitude to the Lord Almighty, whom without His guidance, for keeping me in the path of righteousness I would not have been able to be as I am today. I would also like to express an endless words of appreciation to Prophet Muhammad (pbuh) for his teachings to be a meaningful human and as a servant of God. Next, I would like to express millions of thank you to my supervisor Dr Robiah bt Yusof for guiding me throughout this project, thanks for being an open book and the inspiration for me to keep on going. Also, not forgetting my teammate in this project Mohamad Faiz bin Husin, Puteri Nur Ira Fatimah bt Rosdi and Nurfateha bt Nasir help me and give some advice to develop this project. In addition, also a lot of thank you to my ex classmate diploma Maizatul Akma bt Ya'ani and Mohd Aiman Afnan bin Mohd Yusoff for guide me to complete my thesis report and some guide to implement this system also give a brilliant idea especially in malware prevention process also in programming.

ABSTRACT

This research will be focus on the remediation system by wall garden that will be controls the user's access to Web content and services. It is recommended that wall garden be seriously considered as a method of notification as they are easy to implement and proven to be effective as a means of getting end user attention. But there will have restraint to develop this system that is coordinated malware remediation system through wall garden is developed to blocking the IP that infected by the malware from any DNS and allow the user to access web content and services after the remediation process clean the detection. However, the effective of wall garden to remediate the infected IP is unknown. There have three main objectives to ensure the project will be running smoothly such as to identify parameter involve in wall garden, to develop Malware Remediation System : Wall Garden and to test the effectiveness of the wall garden to prevent infected IP from entering the internet. Furthermore, rapid application development (RAD) will be the methodology in this research to develop the system. RAD is the process which accelerates the cycle of development of an application. RAD makes it possible to develop quality products faster, thus valuable resources can be saved. There have three significant contribution in this research that is to proposed analyse the parameter of the wall garden, to proposed the developing remediation system and develop wall garden and to proposed the effectiveness of wall garden to prevent infected IP by wall garden and remove the malware. Lastly, there are lot of work to be done in the future to be implement for Cyber Security in CMERP framework especially for the wall garden such as automatic malware removal tools, detection infected PC by mac address and fully recovery without interrupt the system.

ABSTRAK

Kajian ini akan memberi tumpuan kepada sistem pemulihan oleh 'wall garden' yang akan menjadi kawalan akses pengguna kepada kandungan dan perkhidmatan Web. Ia amat disyorkan bahawa 'wall garden' diberikan perhatian sebagai kaedah pemberitahuan sebagaimana mereka mudah untuk melaksanakan dan terbukti berkesan sebagai satu cara untuk mendapat perhatian pengguna. Tetapi, hal ini akan mempunyai kekangan untuk membangunkan sistem ini yang diselaraskan dengan sistem pemulihan 'malware' melalui 'wall garden' yang dibangunkan untuk menyekat IP yang dijangkiti 'malware' dari mana-mana DNS dan membolehkan pengguna untuk mengakses kandungan dan perkhidmatan web selepas proses pemulihan membersihkan 'malware' yang dikesan. Walau bagaimanapun, keberkesanan 'wall garden' untuk mengatasi IP yang dijangkiti tidak diketahui. Projek ini mempunyai tiga objektif utama bagi memastikan projek itu akan berjalan dengan lancar seperti untuk mengenalpasti parameter yang terlibat dalam 'wall garden', untuk membangunkan 'Malware Remediation System: Wall Garden' dan untuk menguji keberkesanan 'wall garden' untuk mengelakkan IP yang dijangkiti daripada memasuki internet. Selain itu, pembangunan aplikasi pantas (RAD) akan menjadi metodologi yang digunakan dalam kajian ini untuk membangunkan sistem. RAD adalah proses yang mempercepatkan kitaran pembangunan aplikasi. RAD berfungsi untuk membangunkan produk-produk berkualiti lebih cepat, oleh itu sumber-sumber berharga boleh disimpan. Sistem ini akan mempunyai tiga sumbangan penting dalam kajian ini iaitu menganalisis parameter 'wall garden', cadangan sistem pemulihan pembangunan dan membangunkan 'wall garden' dan cadangan keberkesanan 'wall garden' untuk mengelakkan IP dijangkiti oleh 'wall garden' dan membersihkan malware. Akhir sekali, terdapat banyak kerja yang perlu dilakukan pada masa depan yang akan dilaksanakan oleh CyberSecurity Malaysia dalam rangka kerja CMERP terutamanya untuk 'wall garden' seperti alat pembersihan 'malware' secara automatik, pengesan PC yang dijangkiti melalui alamat 'mac' dan pemulihan sepenuhnya tanpa mengganggu sistem.

TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	BORANG PENGESAHAN STATUS TESIS	1
	DECLARATION	2
	DEDICATION	3
	ACKNOWLEDGMENTS	4
	ABSTRACT	5
	ABSTRAK	6
	TABLE OF CONTENTS	7
	TABLE OF FIGURES	10
	LIST OF TABLES	12
1	INTRODUCTION	
	1.1 Introduction	13
	1.2 Problem Statement	14
	1.3 Project Question	15
	1.4 Project Objective	15
	1.5 Project Scope	16
	1.6 Project Contribution	16
	1.7 Thesis Organization	16
	1.8 Conclusion	18
2	LITERATURE REVIEW	
	2.1 Introduction	19
	2.2 Introduction of Malware	20
	2.2.1 Malware Classification	20
	2.2.2 Method of Malware Detection	23
	2.2.2.1 Classification of Malware Detection Techniques	23
	2.2.2.2 Classification of Computer Attack	24

2.3	Introduction of Remediation	25
2.3.1	Remediation	25
2.3.2	Framework of Malware Remediation System	26
2.4	Proxy	28
2.5	Effectiveness	29
2.6	Dataset of Project	30
2.6.1	Tools Used	31
2.6.2	Parameters / Attributes	33
2.6.3	Method Testing	33
2.7	Conclusion	34

3 METHODOLOGY

3.1	Introduction	35
3.2	Methodology	36
3.2.1	Phase 1 : Business Modelling	37
3.2.2	Phase 2 : Data Modelling	37
3.2.3	Phase 3 : Process Modelling	37
3.2.4	Phase 4 : Application Generation	38
3.2.5	Phase 5 : Testing and Turnover	38
3.3	Project Milestones and Gantt Chart	39
3.4	Conclusion	40

4 DESIGN

4.1	Introduction	41
4.2	Project Requirement	42
4.2.1	Hardware Requirement	43
4.2.2	Software Requirements	43
4.3	Design	45
4.3.1	Physical Network Design	45
4.3.2	Logical Network Design	47
4.3.3	Framework Malware Remediation System: Wall Garden	47
4.3.4	Flowchart	49

	4.3.5 Interface Design	53
	4.4 Conclusion	54
5	IMPLEMENTATION	
	5.1 Introduction	55
	5.2 Software Development Environment Setup	55
	5.3 Software Configuration Management	56
	5.3.1 Configuration Environment Setup	57
	5.4 Implementation Status	69
	5.5 Conclusion	70
6	TESTING AND ANALYSIS	
	6.1 Introduction	71
	6.2 Test Plan	72
	6.2.1 Test Organization	72
	6.2.2 Test Environment	72
	6.2.3 Test Schedule	72
	6.3 Test Strategy	73
	6.4 Test Design	74
	6.5 Test Result and Analysis	74
	6.6 Conclusion	76
7	PROJECT CONCLUSION	
	7.1 Introduction	77
	7.2 Project Summarization	77
	7.3 Project Contribution	78
	7.4 Project Limitation	78
	7.5 Future Works	78
	7.6 Conclusion	78
	REFERENCES	79

TABLE OF FIGURES

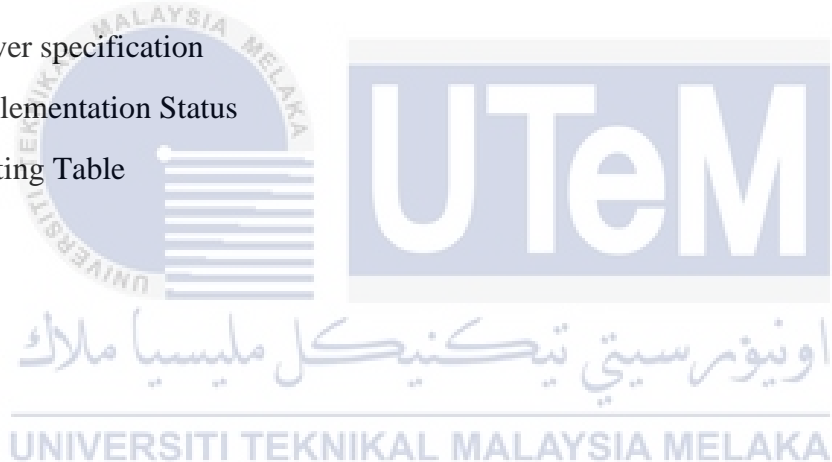
Figure 1.1: Overview of Coordinate Malware Eradication and Remediation Framework	14
Figure 2.1: Structure Overview of Chapter II	19
Figure 2.2: Existing Framework of Coordinated Malware Eradication and Remediation Project	25
Figure 2.3: Framework of Malware Remediation System: Wall Garden.	26
Figure 2.4: Web Proxy Acceleration	28
Figure 2.5: The relationship between the Efficiency and the Effectiveness	29
Figure 3.1: Rapid Application Development Model	34
Figure 4.1: Structure Overview of Chapter IV	42
Figure 4.2: Physical Design of Prevention Module	46
Figure 4.3: Logical Design of Prevention Module	46
Figure 4.4: Framework of Malware Remediation System: Wall Garden.	47
Figure 4.5: Flowchart of DNS Sinkhole directly user into Wall Garden	49
Figure 4.6: Flowchart of user download malware removal tools	50
Figure 4.7: Flowchart of upload log files	51
Figure 4.8: Flowchart of upload checking status	52
Figure 4.9: Proposed design for quarantine infected PC	53
Figure 4.10: Proposed design for malware removal tools	53
Figure 4.11: Proposed design after the malware clean	54
Figure 5.1: Architecture of Wall Garden	56
Figure 5.2: Flow Chart of Apache2 Installation and Configuration	57
Figure 5.3: Flow Chart of MySQL Installation and Configuration	58
Figure 5.4: Flow Chart of phpmyadmin Installation and Configuration	59
Figure 5.5: Flow Chart of PHP Installation and Configuration	60
Figure 5.6: Testing Wall Garden	61
Figure 5.7: Malware removal tools download pages	62
Figure 5.8: COMODO antivirus home	63
Figure 5.9: Malware Cleaning Process	64

Figure 5.10: Export log files	65
Figure 5.11: Upload log files	65
Figure 5.12: Choose log files	66
Figure 5.13: Log files uploaded alert	67
Figure 5.14: Checking Status	68
Figure 6.1: Top-down Testing Strategy (Weißleder 2013)	73
Figure 6.2: Facebook.com testing	73
Figure 6.3: Youtube.com testing	73
Figure 6.4: Testing upload log files	74
Figure 6.5: Data success upload	75
Figure 6.6: Checking Status Testing	76



LIST OF TABLES

Table 1.1: Summary of Problem Statement	15
Table 1.2: Summary of Project Question	15
Table 1.3: Summary of Project Objective	16
Table 1.4: Summary of Project Contribution	16
Table 2.1: Details in one dataset	29
Table 2.2: Dataset of the project	29
Table 3.1: Server specification	37
Table 3.2: Gantt Chart of Project	38
Table 3.3: Project Milestone	40
Table 4.1: Server specification	43
Table 5.1: Implementation Status	68
Table 6.1: Testing Table	74



CHAPTER I

INTRODUCTION

1.1 Introduction

Nowadays we need internet in our life especially student. But we did not know the risk that we face it later when we did not have any precautions and prevention from malware attack (“malicious software”). Malware will be expose in our device and will be infect by malware when we our device run slowly and frequently crash. We can recognize it by installing any other software to scan the malware to see that the malware are inside our device or not. With a powerful software, malware can be found easily with large of volume. Nowadays, malware detection just aiming on certain components based on the required functionality. But, most of the latest technology are focusing on administering ratings. Because of that, the IDS (Intrusion Detection System) was implement to monitor any malicious activity in our network and system thoroughly. It also can process a large amount of malicious activity and give an alert to us about the attack and the detection of the attack. Because of this attack, it will interrupt our system and network and also our device. Hence, this research will be focus on the remediation system by wall garden that will be controls user access into the web contents and services. In effect, the wall garden lead user navigation in certain areas, to allow access to the choice of materials, or deny access to other substances. Although the wall garden in fact does not prevent users from browsing outside wall, it makes it more harder than to live in an enchanting context. ISPs want to fence in users for several reasons. If the wall garden is used, a list of addresses that are known to both the operating system vendors and security vendors need to be established and maintained within the white list that allows access to the sites.. This can be essential to enable access away from

the garden wall by end users in finding the operating system and application patches. It is recommended that the garden wall is given attention as a method of notification as they are easy to perform and proved to be effective as a way to get the attention of end users Figure 1.1 below shows an overview of Coordinate Malware Eradication and Remediation (CMERP) framework with the division with eradication component and remediation component.



Figure 1.1 : Overview of Coordinate Malware Eradication and Remediation Framework

1.2 Problem Statement (PS)

This research will be focus on remediation malware system through wall garden. Even though the coordinate malware eradication and remediation framework are still new proposed by cyber security, this research will find the solution to prevent the malware attack by containment and malware removal by wall garden. Hence, two phase will be investigate to confirm the integration feasibility. Table 1.1 below shows the summary of problem statement that this project will embark upon.

Table 1.1: Summary of Problem Statement

PS	Problem Statement
PS1	Coordinated malware remediation system through wall garden is developed to quarantine the IP that infected by the malware from any DNS and allow the user to access web content and services after the remediation process clean the detection. However, the deployment of wall garden to remediate the infected IP is unknown.

1.3 Project Question (PQ)

How possible we can generate the remediation system by wall garden and test the effectiveness? A Forecast by study of the framework should be conducted on before any integration work will take over. Once the framework is enhanced, then how to validate the accuracy of the remediation component? Table 1.2 below shows the summary of project questions that this project will embark upon

Table 1.2: Summary of Project Question

PS	PQ	Project Question
PS1	PQ1	How study the parameter in malware remediation?
	PQ2	How to develop the Malware Remediation System: Wall Garden?
	PQ3	How to deploy of the wall garden to prevent infected IP from entering the internet?

1.4 Project Objective (PO)

Based on the questions of the project, the project will bring two major objectives to ensure that project will be proceed smoothly. We need to develop the remediation system by wall garden first before we can test the effectiveness of wall garden to prevent infected IPs. Table 1.3 below shows the summary of project objectives that this project will based upon.

Table 1.3: Summary of Project Objective

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To study the parameter involve in wall garden.
	PQ2	PO2	To develop Malware Remediation System: Wall Garden.
	PQ3	PO3	To deploy the wall garden to prevent infected IP from entering the internet.

1.5 Project Scope

Scope of the project is going to be handle as follows:

- i. Focus on develop malware remediation system by wall garden.
- ii. Test the wall garden to prevent infected IP from entering the internet.

1.6 Project Contribution

Table 1.4: Summary of Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Proposed the parameter of the wall garden.
	PQ2	PO2	PC2	Proposed the remediation system and develop wall garden.
	PQ3	PO3	PC3	Proposed the wall garden to prevent infected IP by wall garden and remove the malware.

1.7 Thesis Organization

Chapter I : Introduction

This chapter will cover on introduction, project background, project problem statement, project questions, project objectives, project scope, project contributions and thesis organization.

Chapter II : Literature Review

This chapter will focus more on reading materials and conference paper that will be explain more of this project and because of it, it will be support by literature reading. In this section, architecture of framework, malware containment and removal malware, example of tools and others will be included in this chapter.

Chapter III : Project Methodology

This chapter show the method that will be use to make some analysis process project briefly. Rapid application development will be use in this methodology. The implementing and organizing the project will be easy with the methodology.

Chapter IV : Analysis and Design

In this chapter, software and hardware have strong related each other to implement this project. The enhancement of the design will be carried out.

Chapter V : Implementation

This chapter will show what and how the installation of the software to develop malware remediation system by wall garden and tools that we used to remove malware and prevent it.

Chapter VI : Testing and Validation

This chapter will test and verify the modules and establish the wall garden to quarantine the infected IP address into wall garden and remove the malware.

Chapter VII : Project Conclusion

In this chapter, the summary of the project, project contribution and project limitation will be explained. The system that being develop in this project will be briefly explain all the steps that being done. In this phase also will be explain on additional creation can be done in the future.

1.8 Conclusion

As a conclusion, this chapter will guide to comprehend the project background, aiming that should be done exactly with the idea and problem happened before the project begin. This project will make the existing CMERP framework is enhanced on malware remediation which is for containment the malware and removal malware. On the next chapter will be cover on literature review. It will explain about the immersing model and related work with the framework and improve the framework.



CHAPTER II

LITERATURE REVIEW

2.1 Introduction

A new coordinated malware eradication and remediation framework is a framework which is used to provide the effective measure to safeguard. Throughout this chapter, the project will conduct an in-depth discussion about the literature review on Malware Remediation System by Wall Garden and other related project that is similar on the techniques that shall be used. In this subchapter is going to discuss about the architecture of this project. The research is conduct an in-depth about the history of malware, evolution of malware, remediation system and wall garden.

Figure 2.1 below shows the structure of this chapter that we will briefly discuss about.

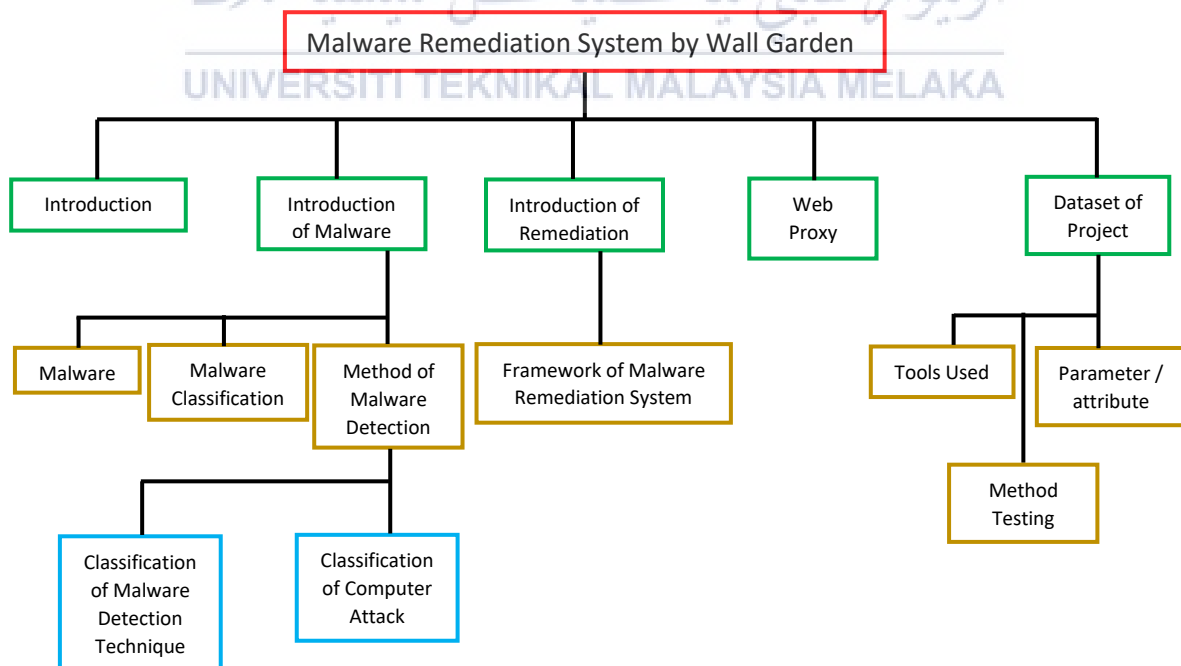


Figure 2.1 : Structure Overview of Chapter II

From the Figure 2.1 above shows that the structure overview for this chapter that will be explain briefly for each respectively. So, the explanation in this project is depends on this structure. There are several terms that will be used in this project and shall serve the purpose as the keywords throughout the research of the project.

2.2 Introduction of Malware

Based on article by (Milosevic, 2011) it can show the algorithm of code, the scripts, active content and another package. 'Malware' is a general term used to refer to a variety of forms of hostile or disruptive package. Malware includes virus, trojan horse, ransomware and other malicious programs; the most of active malware threats are usually worms or trojans compare to the viruses.

The combination of two words malicious and software made a term a malware nowadays, and to show the unwanted program and software. It was defined, generally by ((McGraw, 2000)) as “any code added, changed, or removed from a software system in order to intentionally cause harm or subvert the intended function of the system”.

However according to (E. Passerini, July 2009) Recent research indicate that a commercial anti-malware software fails to return even though the highest impact of all the actions performed by the malware during the infection.

2.2.1 Malware Classification

The different parameter are decided the category of malware such as how it change the system, meaning of the program or the function, deploy a mechanism and be either these programs requested authorize or consent of the user before executing a specific operations. The malware can be detected if these program are run one of this activities:

- Modifies another program.
- Replicates itself through a network or a file system without users's consent.
- Allows an authorized person to take control over a remote system.

- Sends personal or confidential information to a remote system without user consent.
- Sends data to a system in order to disrupt normal functioning.
- Opens a port for listening in on a local machine to accept commands from a control server.
- Records keystrokes and sends this information to remote servers.
- Connects to suspicious remote servers.
- Downloads and executes files from suspicious remote servers.
- Copies itself to multiple locations.
- Injects code into another program.
- Makes unauthorized changes to the system.
- Modifies a protected system setting.
- Modifies a registry setting used for launching program upon startup.

Now we will look into specific malware categories based on distinguished malicious features of the sample.

1. Virus

Virus is first category worms to emerge on the horizon of malware computer security. It is parasite infector because of self-replicate behaviour. It does not have a separate existence instead it inserts its code into existing files on the system.

2. Worm

Worms are also replicate themselves, however they are standalone malware strains. Worms do not change the system to deploy the infection but they make duplicate of themselves over same network or on the other systems. Worms deploy longer classified accordance to the mechanism are used as email, P2P, IRC, etc.

3. Trojan

A Trojan is always masquerading as useful software and seductive a user to install it and it is also bundles of undercover malicious features. Trojan does not replicate itself in nature. It does not spread in a similar manner as viruses or worms.

4. Backdoor

Backdoor give the authorize access to a compromised system by opening the port on the system. It will create a path to the system and change the system with some commands of malicious nature.

5. Hacktool

Hacktool always being used by a hacker to assault and exploit a system to get illegal access inside the system. It attempts to have any information about the system after trespassing the security in the system that are inherent to the system. Netcat is an example of hacker use to hacking.

6. Spyware

Spyware is software that collect the personal data and information from user system without them noticed. It includes monitoring the system without their notice it. It will include to monitoring the system of user to collect confidential data such as habits of browsing, latest browsing, passwords, credit card data and other confidential information.

7. Rootkit

Rootkits use to steal techniques to hidden their coming by concealing their components such as files, registry keys, running processes and other objects. All of these techniques are be used to hide their behaviour from users and to intercept a tracking of security applications.

2.2.2 Method of Malware Detection

2.2.2.1 Classification of Malware Detection Techniques

Signature-based and anomaly-based techniques

According to (Idika, 2007) basically all of the malware scanner, utilize signature-based and oddity based methods for distinguishing personalities of projects. In any case, there are strategies utilizing these procedure: dynamic techniques that utilization during the movement of data from malware, when it is implemented in the memory; static method is done by removing the characteristics of static malware, when it is in the disc and a hybrid method used is a combination of dynamic and static methods.

Based on (Ye, 2009) to recognize malignance of a record utilizing mark based strategies, scanner programming assesses its data to a vocabulary of infection marks in a database to see whether a mark found there. The benefit of such methods is its viability. In any case, the principle impediment with mark based strategies is that they can't safeguard against obscure malware.

Heuristic based techniques

Based on book (Chandola, 2009) was utilize with mark based and irregularity based methods to improve their effectiveness. neural systems (NNs) have been embraced for their versatility to natural changes and their capacity of expectation.

As indicated by in light of (Professor S. F., 2000) Fuzzy logic is a counterfeit consciousness approach got from fuzzy hypothesis, which utilize guess for rationale instead of exact traditional rationale. Hereditary calculation is another machine learning-based system utilized as a part of malware recognition prepare for determining grouping rules and choosing suitable components or ideal parameters for ideal arrangement. It applies standards of transformative science, for example, legacy, change, choice and mix. The principle favorable position of this procedure is the inference of arrangements

from numerous headings with no requirement for earlier information about framework conduct.

There has same opinion from two researcher (Kevadia Kaushal, 2012. 2(3)) and (Chandola, 2009) that said Measurable and numerical strategies are utilized as a part of malware location by applying factual and scientific models on the data of framework exercises, for example, organize associations, transmission capacity, memory utilization, framework call utilized by items and so on.

Conclusion

This exploration introduced a nitty gritty survey of the best in class for malware, malware discovery strategies and advancements. Specifically, it gives an exceptional relative review for the clear majority of malware families and additionally it condenses various malware discovery frameworks. Although the creating procedures of malware and their identification frameworks are quickly developing, this review can be considered as the primary reference for the designers in the field.

2.2.2.2 Classification of Computer Attack

Introduction

According to (Carmona, 2005) achievement of the assault methods is exceptionally dependant on the inquiry of data on the objective machine and organize exercises, for example, utilized operational framework, opened administration ports, introduced powerless programming and client accounts, in unique, with the get to secret word. By methods for the objective system acknowledgment, named examining, utilizing programming or social building procedures, it is conceivable to investigate the vulnerabilities of the operational frameworks or correspondence arrange conventions.

The investigation of frameworks and system assets vulnerabilities can be distinguished through various strategies, for example, checking of framework occasion logs, and examination of payload information conveyed by the system

bundles. Based on the book (Northcutt, 2002) the investigation of frameworks and system assets vulnerabilities can be distinguished through various strategies, for example, checking of framework occasion logs, and examination of payload information conveyed by the system bundles.

2.3 Introduction of Remediation

2.3.1 Remediation

According to (E. Passerini, July 2009) the safest way to handle it is to reformat the permanent storage and reinstalling the operating system from the beginning. Although it is effective, this approach can also increase the cost of expensive maintaining and usually results in loss of valuable personal data without any backup, especially when making copies of data that are incomplete or non-existent. Instead, end users and administrators can choose to make a copy of data that only resources left by malware or remove it and leaving the rest of the system that have a strong security. But, the latest anti-malware products work well on this task. Figure 2.2 below show the CMERP framework.

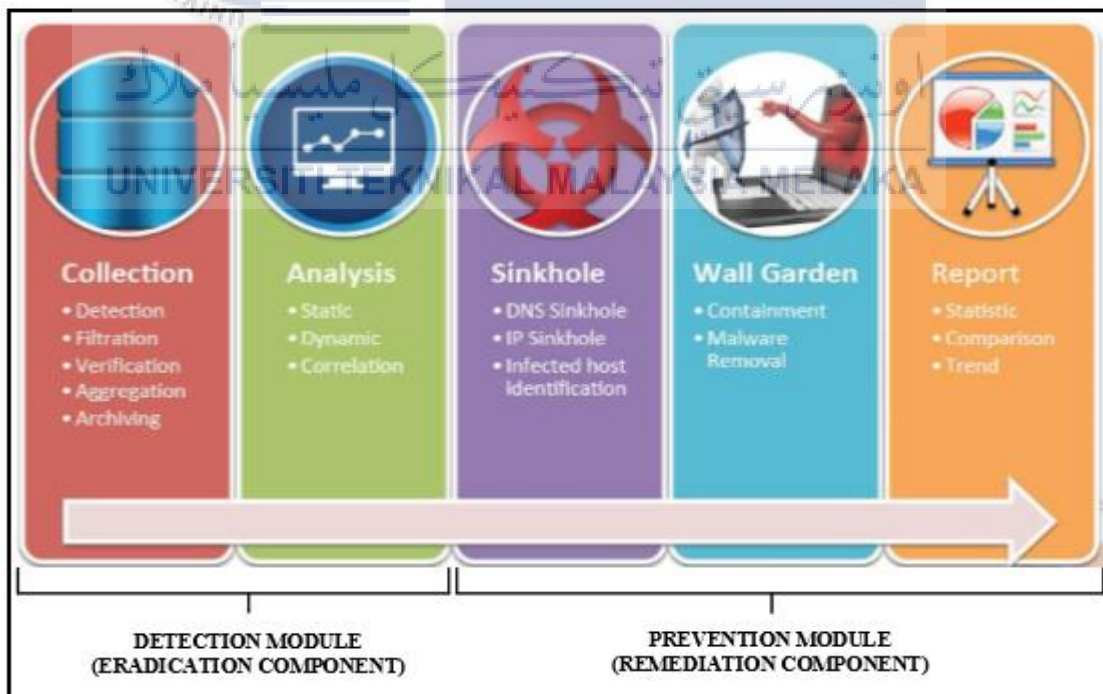


Figure 2.2: Existing Framework of Coordinated Malware Eradication and Remediation Project

The existing framework of Coordinated Malware Eradication and Remediation Project (CMERP) is illustrated in Figure 2.2 above. There are five phases included in this framework which is Collection phase, Analysis phase, Sinkhole phase, Wall Garden phase and Report phase. But, for this project, it only focused on one phases that involved in remediation component which are Wall Garden phase.

2.3.2 Framework of Malware Remediation System

Framework consists of five phases which include Collection, Analysis, Sinkhole, Wall Garden and Report. The main phases which focus on are Wall Garden phase. These phases are including in remediation component which for containment and malware removal. The framework of Malware Remediation System: Wall Garden as in Figure 2.3 below.

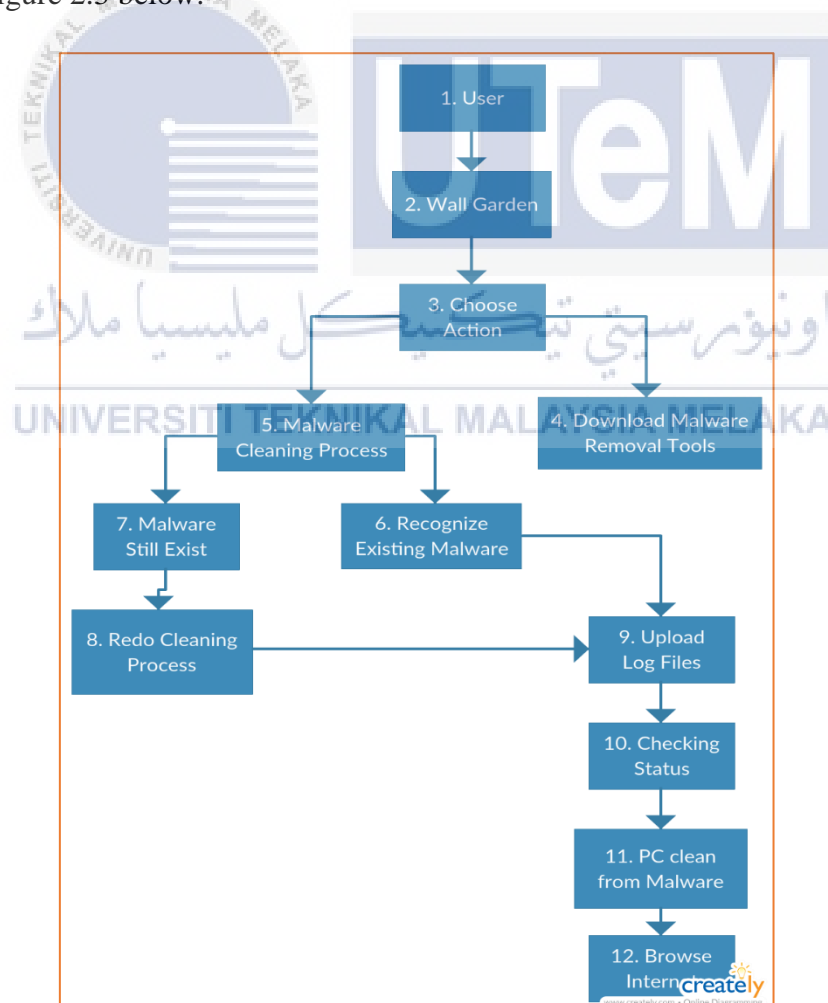


Figure 2.3: Framework of Malware Remediation System: Wall Garden.

1. **User:** User from DNS sinkhole will be recognize and quarantine inside wall garden.
2. **Wall Garden:** The area that will quarantine the infected user to access into the internet.
3. **Choose action:** To choose whether user should download first the malware removal tools or clean the malware by existing malware removal tools that had been install in their PC.
4. **Download malware removal tools:** User must download the removal tools that wall garden provided to them.
5. **Malware Cleaning Process:** User will clean the malware inside the PC using the malware removal tools that wall garden provided or not.
6. **Recognize Existing Malware:** To ensure the malware were remove from the PC and cleaned.
7. **Malware Still Exist:** There have some malware inside the PC.
8. **Redo Cleaning Process:** If the malware still exist, the cleaning process will be redo.
9. **Upload Log Files:** User will upload log files into the system to update the data about cleaning process.
10. **Checking Status:** To check the user PC are completely clean from malware.
11. **PC Clean From Malware:** There have no malware inside the PC.
12. **Browse internet:** After the cleaning session and the IP was update, user can browse the internet.

2.4 Web Proxy

Based on (B. M. Duska, (1997)), (A. Feldmann, (1999).) and (Gribble, (1997)) proxy cache for Internet Service Provider that are recognized for people receive a high number of requests. There are many a review showing that the ratio frequently missed at least 40% -50%, even large cache proxy is used. Therefore, reduce of overhead cache missed can cause a large increase in throughput for a proxy cache for applications experiencing a high number of. Figure 2.4 below show the web proxy acceleration from IBM.

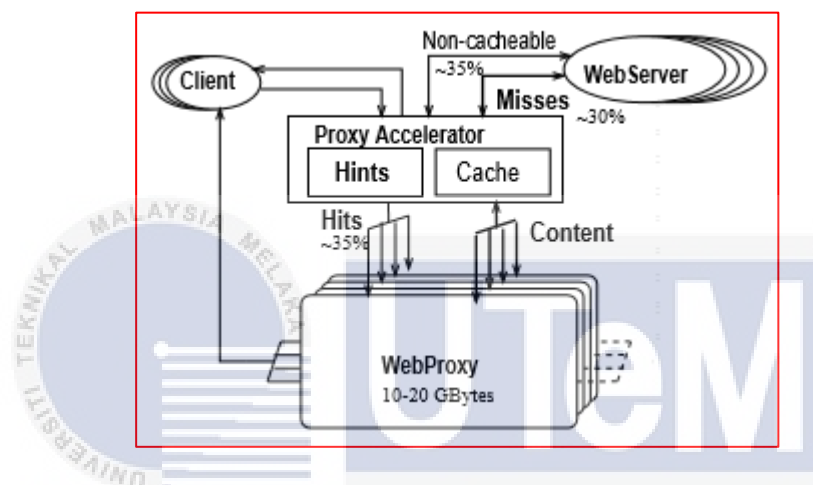


Figure 2.4: Web Proxy Acceleration (E. Levy, (1999))

According to (E. Levy, (1999)) HTTP accelerator was developed under the server running on embedded operating system optimized for communication, such as a Web server accelerator described, but based on (Netfinity, (2000)) under the general cover to an operating system in kernel-mode will slow down the services at a level, such as Netfinity Web Server Accelerator.

Accelerated Web Proxy

Web proxy systems that developed of high-performance development made up of a group of nodes that implement in a node application proxy and the web front-end circulated the request to the node of application. However according to (Johnson, (1999)) a content-based the scheme that distinguishes Among the cacheable and the content noncacheable been declared would lead to significant performance enhancements to the System the Web proxy forms when interfaced with off-loading the processing of demand for non-cacheable content from proxy nodes to the front end.

2.6 Dataset of Project

Based on (Milan) the dataset has been interlinked and published according to the Linked Data principles. In this project, there are several data that will be fetch on to quarantine the infected user. Data that will be fetch which is malware infection status, type of infection, source IP and mac address of the infected user. Table 2.1 below show the details about the data that will be use in this project.

Parameter	Details
Malware Infection	Yes / No
Type of Infection	Worm, Trojan Horse,
Source IP address	192.168.50.70/24
MAC Address	Ex: E0-B9-A5-FC-6F-98

Table 2.1: Details in one dataset

Table 2.1 above show use the detail of one data from one source when the dashboard analyse it. Table 2.2 below show all the dataset that will be use in this project.

1	Month	Malware Infection?	Date	Time	Type of infection	Source IP	Destination IP
2	January	Yes	2-May-17	9.00 AM	Worm	192.168.50.70	103.198.52.23
3	January	No					
4	January	Yes	2-May-17	11.30 AM	Worm	192.168.50.70	103.198.52.23
5	January	Yes	2-May-17	12.15 PM	Worm	192.168.50.70	103.198.52.23
6	January	No					
7	January	Yes	2-May-17	2.00 PM	Trojan Horse	192.168.50.70	103.198.52.23
8	January	Yes	2-May-17	3.10 PM	Trojan Horse	192.168.50.70	103.198.52.23
9	January	Yes	2-May-17	4.00 PM	Trojan Horse	192.168.50.70	103.198.52.23
10	January	No					
11	January	Yes	3-May-17	8.30 AM	Trojan Horse	192.168.50.70	103.198.52.23
12	January	No					
13	January	No					
14	January	No					
15	January	Yes	3-May-17	12.00 PM	Worm	192.168.50.70	103.198.52.23
16	January	Yes	3-May-17	1.00 PM	Worm, Trojan Horse	192.168.50.70	103.198.52.23
17	January	Yes	3-May-17	2.00 PM	Worm	192.168.50.70	103.198.52.23
18	January	Yes	3-May-17	3.30 PM	Worm, Trojan Horse	192.168.50.70	103.198.52.23
19	January	No					
20	January	No					
21	January	No					
22	January	Yes	4-May-17	9.15 AM	Worm	192.168.50.70	103.198.52.23
23	January	Yes	4-May-17	10.30 AM	Worm	192.168.50.70	103.198.52.23
24	January	Yes	4-May-17	11.00 AM	Worm	192.168.50.70	103.198.52.23
25	January	Yes	4-May-17	12.15 PM	Worm	192.168.50.70	103.198.52.23

Table 2.2: Dataset of the project

Table 2.2 above show the dataset that will be used in this project. This dataset maybe will be change if there have any problem occur when the system will be implemented.

2.6.1 Tools Used

In order to create the wall garden, there are several tools that will be use in this project. There are including the tools that will be generate the interface for wall garden.

1. PHP

According to (Doyle, (2010)). beginning PHP 5.3. HP is different programming languages to develop a dynamic, interactive site. As a general guideline, any PHP program runs on a Web server, and produce Web pages to visitors during the request. One of the main features of PHP is that you able to embed the PHP code in an HTML Web page, making it extremely simple for you to compose dynamic content faster.

2. Python

According to (Rossum G. V., (2012)) There are some alternative execution that have a particular interest to the different audience. There are including:

- **CPython** is written in C language that had being maintained always in python.
- **Jython Python** was executed in java language. Enactment of this may be used as a scripting language for Java applications, or applicable for make applications using libraries of java class. The testing always used the for the java libraries.
- **Pythonfor.NET** execution actually uses the CPython execution, but is being managed .NET application and to make .NET libraries available. It was created by Brian Lloyd.
- **IronPython** Python is an alternative to the .NET. Nothing like Python.NET, currently is the complete Python execution are generating a IL, and compiled Python code be directed to the assemblies. It was created by Jim Hugunin, the original creator of Jython.
- **PyPy** The execution Python was written completely in Python. It supports some of advanced features was not found in implementation tasks like assistance stackless and Just in Time compiler. Major goal of

this project is to promote an experiment with language itself by creating it much easier to customize the interpreter (since it is written in Python).

3. Ubuntu

Ubuntu will go about as the working framework to direct the venture. The form of Ubuntu that is utilized for this venture is Ubuntu Server LTS 16.04 "Trusty" with i386 engineering. Ubuntu is an open source working framework.

4. Oracle VM VirtualBox

Oracle VirtualBox will be utilized for the virtualization programming of this venture. The variant that is utilized for this venture is VirtualBox 5.0.20. VirtualBox is an open source virtualization programming in this manner clarifying why utilizing it for this venture.

5. PhpMyAdmin

This software will be used as database to store the identity of the malware, IP address, type of malware to be recognize them. When it being store inside the database, when the same PC are infected it will recognize the IP address and type of malware that always attack the user. So, the data can be analyzed after the remediation process end.

The software requirements that will involve in this project consist of Ubuntu Server LTS 16.04, Oracle VM VirtualBox and PhpMyAdmin. Every one of these sorts of software will be utilized to make the identification of malware and investigation the malware effectively.

6. PHP

PHP or hypertext preprocessor is a language that will be used to implement the HTML view of the garden wall interface. PHP version 5.6 will be used in the development of this system so that it can be adapted in a web page.

7. HTML

Hypertext Markup Language is a standard markup language to build web based and web applications. It will very friendly user when we put (CSS) cascading Style Sheets and JavaScript to make certain function will use

javascript. This language will be used to build web pages for wall garden to make easier for user with the system.

8. Apache2

Apache2 is commonly used for developed web server on Linux systems. Apache2 will be install in the server to develop the web pages through HTML language

9. MySQL

MySQL will be install inside the server in ubuntu. This software is an open source relational database management system (RDBMS). Apache web servers will be combine with MySQL to develop the wall garden

2.6.2 Parameters / Attributes

Based on (Shen Li et al, (2015)) a method of data flow modelling for a product design process oriented to data parameter is proposed. Malware that infected user will be trace and detected by DNS sinkhole first before wall garden take over the remediation process. After the infected IP were quarantine, wall garden page will guide user to remove the malware.

- **Analysis:** Details of the analysis, ip address, protocol, date, time, malware and log files.

2.6.3 Method Testing

Based on (Cleve & and Zeller, 15-21 May 2005,) top-Down Integration Approach is an incremental integration testing that begins by testing the top level module. After this, it adds in lower level modules one by one. It uses stubs to simulate lower level modules. A Stub is a special code arrangement that can stimulate the behavior of a well designed and existing module which is not yet constructed or developed.

2.5 Conclusion

As a conclusion, this chapter are representing the literature review of this project as a guideline to implement this system. The methodology will be explain more on the next chapter. On the next chapter will be covered the method that will be use for this project. From the literature review, it will cover all the project objectives (PO1, PO2 and PO3) to identify parameter involve in wall garden, develop Malware Remediation System : Wall Garden and to deploy the wall garden to prevent infected IP from entering the internet.



CHAPTER III

PROJECT METHODOLOGY

3.1 Introduction

In this chapter will be explain on details concerning the issues that are related to the methodology that will be used throughout this project. The main purpose of specifying a methodology is that to make sure this project running in time and conducted in a correct sequence. Figure 3.1 below shows the methodology that is implies for this project which is the Rapid Application Development model.

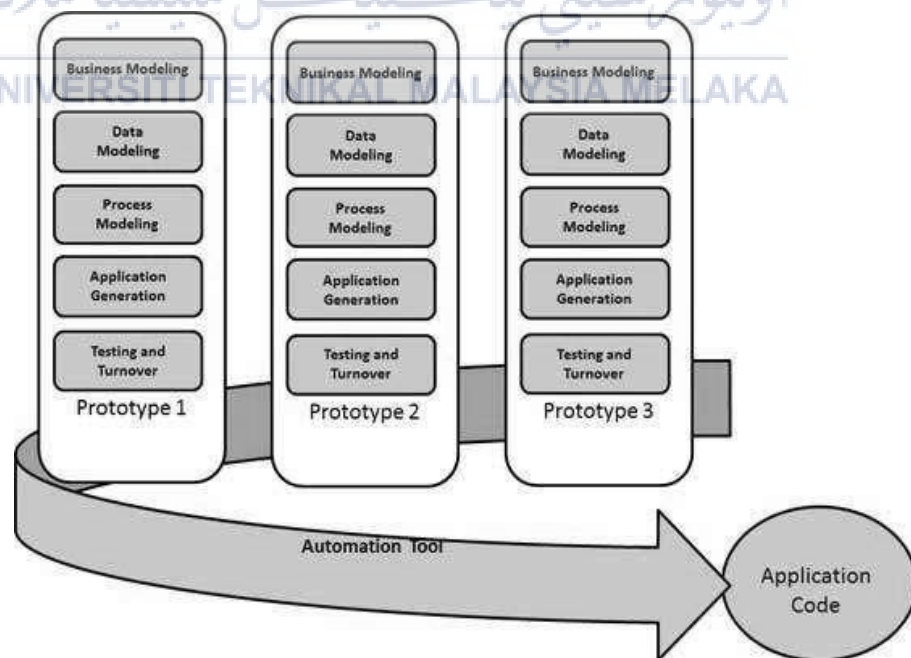


Figure 3.1: Rapid Application Development Model (tutorialspoint.com, 2017)

RAD display rapid empower conveyance as it decreases the future general improvement because of the applicability of the segments and parallel progress. RAD if only the functionality impressive high talented architects are is also accessible and customer-specific to accomplished the focus on model within the given time span. If there is a lack of responsibility on either side of the model concerned fizzle.

The upsides of the RAD Model are as per the following:

- Switching preconditions able to appropriate.
- Advance measurable.
- The cycle time able to short and intense use RAD devices.
- Benefits with low a target in a short period.
- An increase in reduced time.
- Build applicability segment.
- Audit snappy identification applied.
- Supports customer comments.
- Participate of absolute discretion the starting point illuminates a lot of issues of reconciliation.

3.2 Methodology

According to James Martin says, RAD is used for the life cycle of developing of the software that give more rapid development and also provide high quality software, then using the traditional software development life cycle.. RAD facilitating organization of developing software much faster and it also assist to reduce the cost of development and maintaining quality of the software. This technique is accomplished through chain certified the methods for develop applications in different lines. RAD methodology is different from an orderly development, but has a shorter construction period of traditional construction methodology or structured development methodology has the development period. In addition, the technique later is hard to acknowledg. In a word, RAD is processes that speed up application development cycle. RAD enables to develop good quality products much faster, thus precious resources could be saved.

3.2.1 Phase 1 : Business Modelling

For the most part, to get an enough data and assembled every one of the prerequisites for the framework, a talking of the customers are made. In this venture, every related issue with respect to the CMERP framework concentrating on eradication part which is introduction of malware, type of malware, malware classification, method of malware detection, classification of malware detection technique, classification of computer attack and numerous more are accumulated and talked about. All these data and hypotheses will then utilized for the following stages which is the plan stage.

3.2.2 Phase 2 : Data Modelling

Once the necessities and data are assembled, the framework is starts for snappy plan which is comprises of the fundamental and huge perspectives that will give a thought regarding the framework to the customer. This stage helps in procedure of building up the model and guarantee that the venture will advance effectively and easily. The physical and legitimate plan of the reproduction condition will be delivered. These outlines will help the execution of the venture.

3.2.3 Phase 3 : Process Modelling

To make the first prototype, all the accumulated information was altered from the phase 2 (data modelling). The product and equipment that are utilized all through this venture are recorded. The recorded necessity was assembled to direct and guarantee that the targets of this venture can be satisfied.

i. Setup environment

With respect to this venture, Ubuntu Server 15.04 was picked as the server to be introduced into physical host. Virtualbox is chosen and installed within the Ubuntu Server. The virtualbox will run Windows 7 Starter as the virtual visitor. For the hardware part of the project, the

ASUS A455L series are chosen to be a server. Table 3.1 below is the details of the specification of the product.

Specifications	Details
Processor	Intel Core i5-5200, up to 2.7Ghz
RAM	4GB DDR3L 1600 MHz SDRAM
Storage	1 TB HDD 5400 RPM
Networking	Realtek Ethernet, Fast Ethernet (PCI Slot 2 (PCI bus 2, device 0, function 0) 802.11 b/g/n
Graphics	NVIDIA GeForce 930M with 2GB DDR3 VRAM

Table 3.1: Server specification

3.2.4 Phase 4 : Application Generation

At that point, this stage has a contribution of the customer keeping in mind the end goal to make an assessment of the model when the framework that proposed is introduced to them. A coordination procedure which has been done is exhibited to client so as to identify the most grounded and the shortcomings of the framework like the things should be included or evacuated by any means. The client likewise gives the recommendations and remarks about the model and those things are gathered to go to the engineer.

3.2.5 Phase 5 : Testing and Turnover

After totally satisfied the client necessities, the last model is acknowledging by the client. The last framework is tried and assessed to guarantee the coordinating framework is working effectively in the wake of being altered and stay away from the huge size of disappointments

3.3 Project Milestones and Gantt Chart

Project milestone of reference is undertaking exercises over the different dates that should finished in order of this project. The Gantt diagram is the graphical point of reference projects, it empowers the venture engineer to monitoring deadlines and ensure undertaking attempt shall be made in course of events which have been set. The motivation rear the Gantt diagram is to guarantee that the joint are generated from the starting point the beginning until the end, each procedure in Gantt graph requested other procedures. Which implies, if not all; a portion of the procedure subject to the past procedure to be finished before it can be executed. Figure 3.2 underneath demonstrates the gantt diagram of the venture and Table 3.2 demonstrates the turning point of the venture.

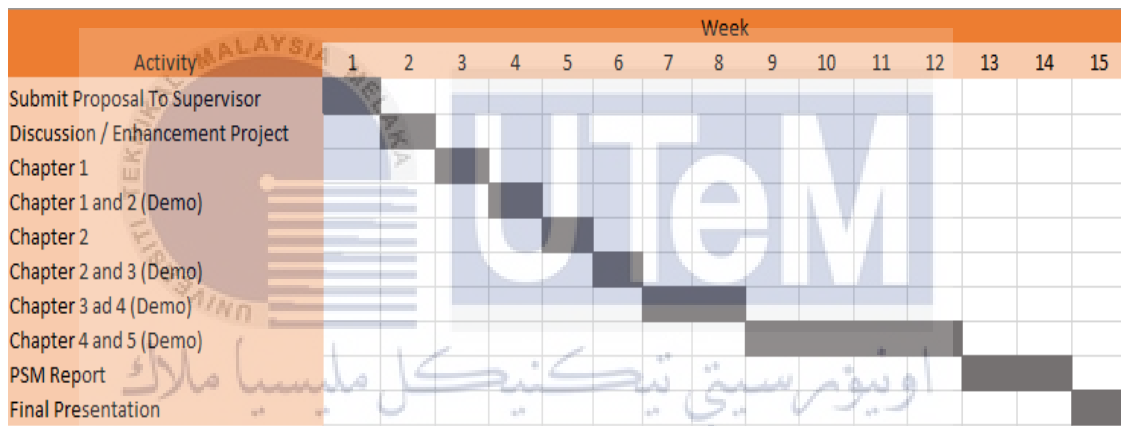


Table 3.2 : Gantt Chart of Project

Activity	Responsibility	Date Start	Date End
Submission proposal	AJK	Week 1	Week 2
Prepare chapter 1 and chapter 2	Student and Supervisor	Week 1	Week 2
Submission of chapter 1 and chapter 2 and discussion	Student and Supervisor	Week 2	Week 3

Prepare the analysis chapter 3 and project progress	Student and Supervisor	Week 3	Week 3
Submission of chapter 3 and discussion	Student and Supervisor	Week 4	Week 4
Design and Implementation	Student	Week 5	Week 10
Prepare chapter 4	Student and Supervisor	Week 8	Week 8
Progress evaluation	Student	Week 10	Week 10
Improvement of chapter 4 and prepare for PSM 1 presentation	Student	Week 11	Week 11
Presentation PSM 1	Student, SV and Evaluator	Week 12/13	Week 12/13
Discussion for PSM 2	Student and Supervisor	Week 1	Week 1
Prepare testing phase chapter 6 and project progress	Student	Week 2	Week 2
Progress evaluation	Student and SV	Week 5	Week 5
Submission of full report (draft)	Student and Supervisor	Week 6	Week 6
Demo Project	Student and SV	Week 7	Week 7

Presentation and evaluation of PSM 2	Student, Supervisor and evaluator	Week 8	Week 8
Submission full report	Student and Admin	Week 8	Week 8

Table 3.3: Project Milestone

3.4 Conclusion

This Chapters have been gives an overview of five phase current will bring this project throughout its implementation. This section also defines the methodology used to this project which is rapid application development methodology because all the details data which related to the requirement of input and output of the system is currently known. This model will allow the clients to communicate or interact and observed a model of working of the system which known as rapid prototype. The clients can feel the real environment of the system with the rapid prototype since the clients get the better understanding for the requirement of the whole system. The five phases are business modelling, data modelling, process modelling, application generations and testing and turnover. In the next topic, the design and implementation of the project will be reviewed.

CHAPTER IV

DESIGN

4.1 Introduction

In this section, the outline of the project that will be utilized for the implementation will be examined thoroughly. In the past section have clarify briefly about the rapid application development (RAD) that will be adapted with the project for the development of this project. Hence, in this section will be illustrate this project into something that individuals can easily comprehend with the project and furthermore will clarify about how the malware remediation process occur inside the wall garden. Figure 4.1 below represent the structure diagram of this part.

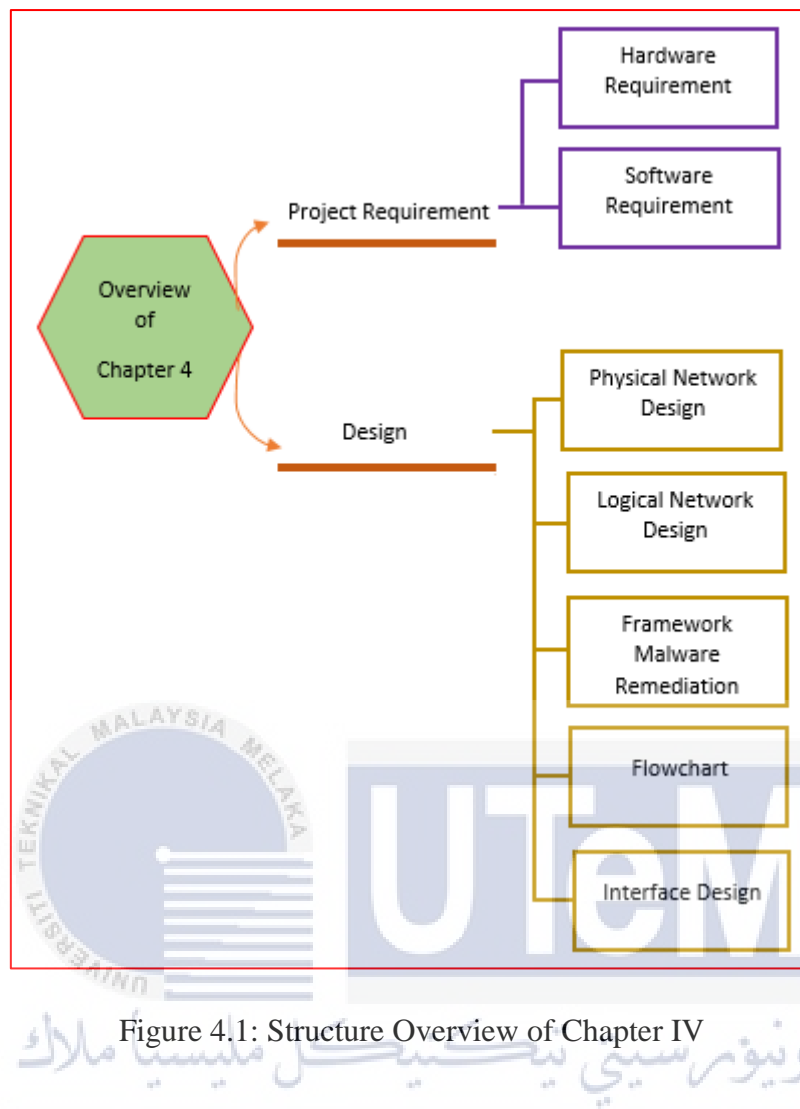


Figure 4.1: Structure Overview of Chapter IV

From the Figure 4.1 demonstrates the structure outline of this section. All the sub-parts that incorporated into this section depend on the structure above.

4.2 Project Requirement

Each project needs a projects necessity to help the improvement of the projects and to guarantee that the projects will run smoothly all through its construction. This project has gathered the most convenient hardware and software necessities so that can guarantee this project will run smoothly. As the following are the hardware and software prerequisite of this project.

4.2.1 Hardware Requirement

There is the hardware necessity that required in this project to guarantee that the project will run smoothly. The following below is the list of the equipment prerequisite that will utilized as a part of this venture.

i. Server Hardware

In this project, the hardware that will be use is ASUS A455L series as the server to accommodate the execution of the working environment of the malware remediation. Table 4.1 below shows the specifications of the server.

Specifications	Details
Processor	Intel Core i5-5200, up to 2.7Ghz
RAM	4GB DDR3L 1600 MHz SDRAM
Storage	1 TB HDD 5400 RPM
Networking	Realtek Ethernet, Fast Ethernet (PCI Slot 2 (PCI bus 2, device 0, function 0) 802.11 b/g/n
Graphics	NVIDIA GeForce 930M with 2GB DDR3 VRAM

Table 4.1: Server specification

The server will be utilized as a part of this project with a specific end goal to distinguish and remediation the malware effectively.

ii. Access Point Hardware

In this project, we will use Access Point as a network medium to connect the DNS Sinkhole, Wall Garden and client. Access point that we used is Huawei Y6II. This access point consists of 2GB RAM, 16GB memory and WAP connection.

4.2.2 Software Requirements

There is a few of software prerequisites that required for this venture with a specific end goal to make the project running proficiently. The following is the list of the product necessities that will be utilized as a part of this venture.

i. Ubuntu

Ubuntu will go about as the working framework to direct the venture. The form of Ubuntu that is utilized for this venture is Ubuntu Server LTS 16.04 "Trusty" with i386 engineering. Ubuntu is an open source working framework.

ii. Oracle VM VirtualBox

Oracle VirtualBox will be utilized for the virtualization programming of this venture. The variant that is utilized for this venture is VirtualBox 5.0.20. VirtualBox is an open source virtualization programming in this manner clarifying why utilizing it for this venture.

iii. PhpMyAdmin

This software will be used as database to store the identity of the malware, IP address, type of malware to be recognize them. When it being store inside the database, when the same PC are infected it will recognize the IP address and type of malware that always attack the user. So, the data can be analyzed after the remediation process end.

The software requirements that will involve in this project consist of Ubuntu Server LTS 16.04, Oracle VM VirtualBox and PhpMyAdmin. Every one of these sorts of software will be utilized to make the identification of malware and investigation the malware effectively.

iv. PHP

PHP or hypertext preprocessor is a language that will be used to implement the HTML view of the garden wall interface. PHP version 5.6 will

be used in the development of this system so that it can be adapted in a web page.

v. HTML

Hypertext Markup Language is a standard markup language to build web based and web applications. It will very friendly user when we put (CSS) cascading Style Sheets and JavaScript to make certain function will use javascript. This language will be used to build web pages for wall garden to make easier for user with the system.

vi. Apache2

Apache2 is commonly used for developed web server on Linux systems. Apache2 will be install in the server to develop the web pages through HTML language

vii. MySQL

MySQL will be install inside the server in ubuntu. This software is an open source relational database management system (RDBMS). Apache web servers will be combine with MySQL to develop the wall garden.

4.3 Design

Each project needs a project configuration to help the perspective of the improvement of the venture and to guarantee that the venture will run easily all through its advancement. There are three different of plan that will be clarify through this part which is logical network design, assault design outline and scripting outline.

4.3.1 Physical Network Design

This is a physical design for the prevention module from the infected PC with malware and the wall garden. It is consisted of switch which is connected with two server which is wall garden server and DNS sinkhole server. This switch also connected to the client to be in one network. Figure 4.2 below is the physical design of prevention module.

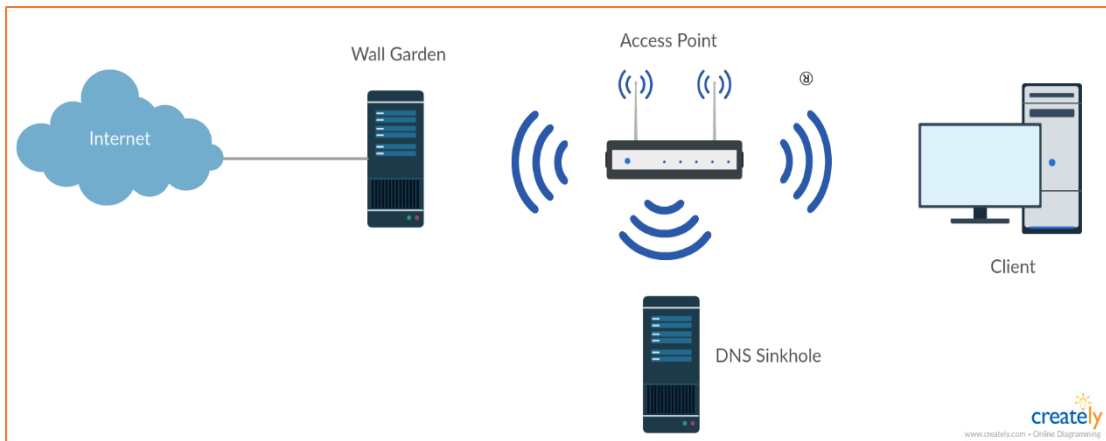


Figure 4.2: Physical Design of Prevention Module

The physical network design of prevention module is set up as Figure 4.2 above so that the project will run easily. The design includes the switch that connected to both of the server and the client PC. When the infected PC run the browser, it will be quarantine by the wall garden before it is clean. Before that DNS sinkhole will redirect traffic to wall garden. After the malware clean, client can browse the internet.

4.3.2 Logical Network Design

This is a physical design for the prevention module from the infected PC with malware and the wall garden. All the procedure occurs through the virtual system and information and logs are spared in the server. Figure 4.3 below is the logical design of prevention module.

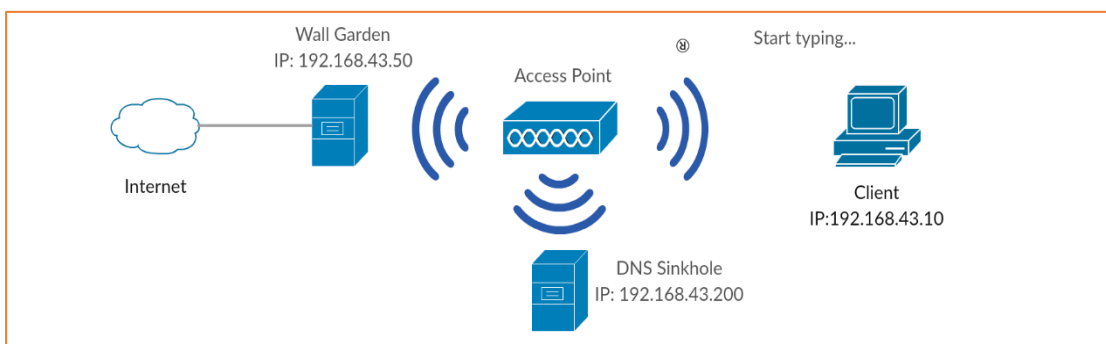


Figure 4.3: Logical Design of Prevention Module

The logical network design in Figure 4.3 above shows that the wall garden will quarantine the infected PC first before it will redirect to the internet. So, DNS sinkhole is responsible to redirect the traffic to the wall garden before the remediation process occur in wall garden.

4.3.3 Framework Malware Remediation System: Wall Garden

Based on Figure 4.4 below, this is the flow process of malware remediation by wall garden. The flow is shows the general process that will be go through in order to remove the malware and remediation process. The process of remediation will through the process of removing the malware. From the remediation will update the IP address and release it.

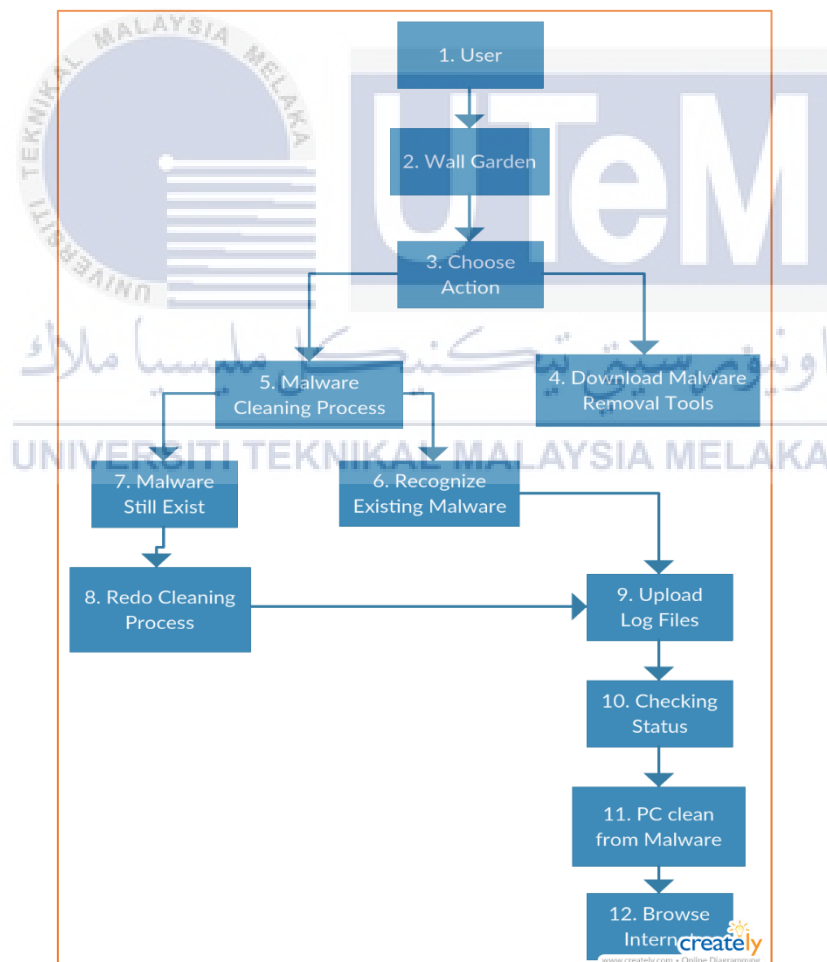


Figure 4.4: Framework of Malware Remediation System: Wall Garden.

1. **User:** User from DNS sinkhole will be recognized and quarantine inside wall garden.
2. **Wall Garden:** The area that will quarantine the infected user to access into the internet.
3. **Choose action:** To choose whether user should download first the malware removal tools or clean the malware by existing malware removal tools that had been install in their PC.
4. **Download malware removal tools:** User must download the removal tools that wall garden provided to them.
5. **Malware Cleaning Process:** User will clean the malware inside the PC using the malware removal tools that wall garden provided or not.
6. **Recognize Existing Malware:** To ensure the malware were remove from the PC and cleaned.
7. **Malware Still Exist:** There have some malware inside the PC.
8. **Redo Cleaning Process:** If the malware still exists, the cleaning process will be redo.
9. **Upload Log Files:** User will upload log files into the system to update the data about cleaning process.
10. **Checking Status:** To check the user PC are completely clean from malware.
11. **PC Clean from Malware:** There have no malware inside the PC.
12. **Browse internet:** After the cleaning session and the IP was update, user can browse the internet.

4.3.4 Flowchart

Flow chart or flow diagram is a progression of images that are utilized to depict the well-ordered flow of a procedure, framework, association or others. Figure 4.5 show the flow diagram that is describing the flow of how wall garden functions

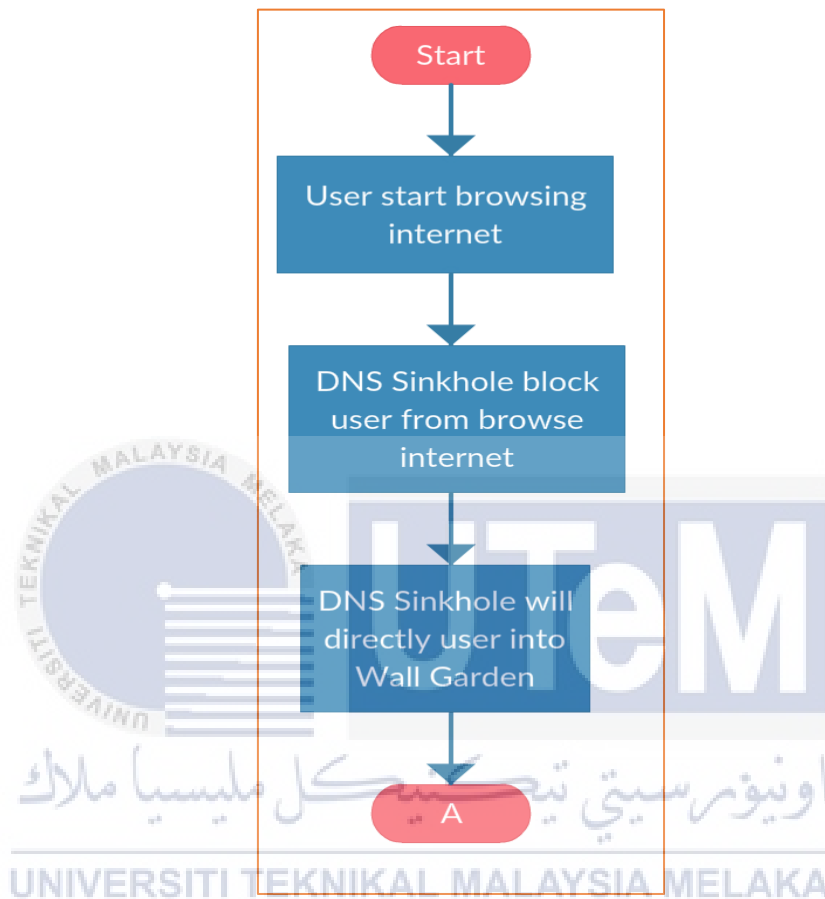


Figure 4.5: Flowchart of DNS Sinkhole directly user into Wall Garden

Figure 4.5 above shows the working of DNS sinkhole will be directly the user into Wall Garden. User will start browsing the internet and will be block by DNS Sinkhole. Then DNS Sinkhole will directly the user into the Wall Garden to be quarantine for a while.

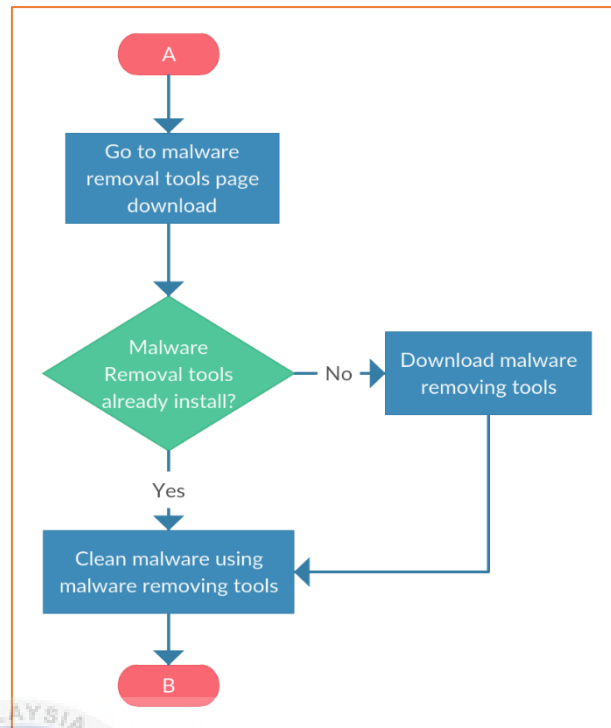


Figure 4.6: Flowchart of user download malware removal tools

Figure 4.6 above shows the process of user download the malware removal tools that being provided by wall garden. User will go to page that provide malware removal tools to download the tools. But, if user already have the malware removal tools they can start cleaned the malware and if not, they must download first the malware removal tools.

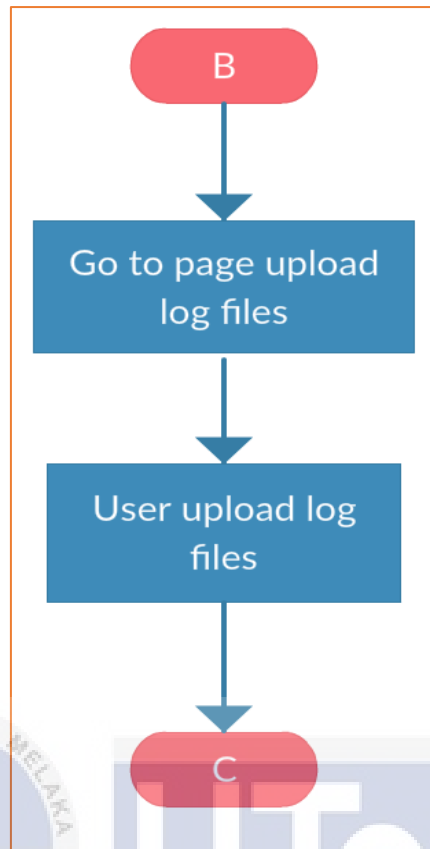


Figure 4.7: Flowchart of upload log files

Figure 4.7 show us the flowchart of user to upload the log files inside the system. User will generate log files from malware removal tools function and upload it inside the system. When the database has update the malware database, user can browse the internet if they have satisfied with the malware cleaning process.

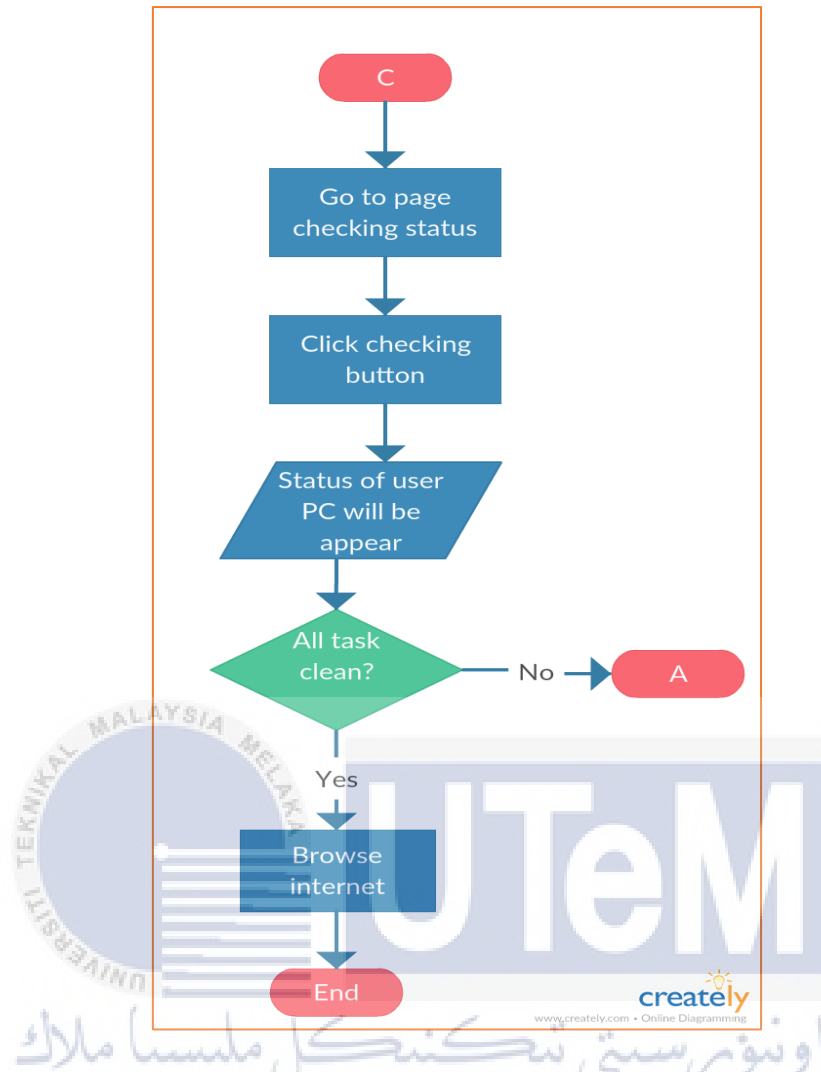


Figure 4.8: Flowchart of upload checking status

Figure 4.8 above shows the flowchart of checking status. After user upload the log files, the user will be checking the status of the PC are fully clean or not. After the checking status finish, it will show the result of the status. If the status shows the PC component are clean, the user can browse the internet. If the status shows there have one or more not clean, the user will try to clean again using the antivirus until the status show all component clean.

4.3.5 Interface Design

In this sub-part will give the proposed user interface to the remediation module with HTML page. The proposed design for this system are as follows:

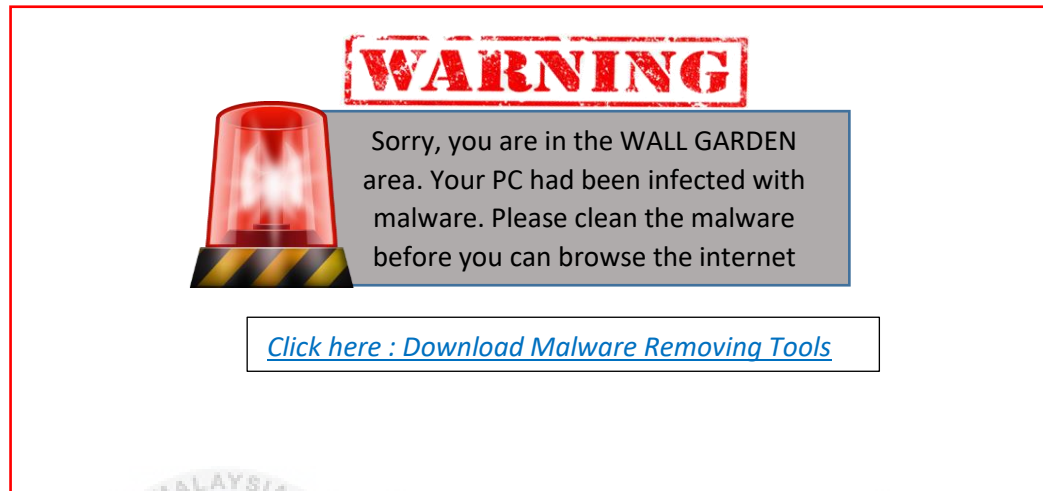


Figure 4.8: Proposed design for quarantine infected PC

Figure 4.8 show the page that will be appear after the infected PC are tried to browse the internet. The warning message will be appeared on the page and quarantine user inside this page until the malware was clean. The download link to download the malware removing tools appear to user download the removing tools to remove the malware.

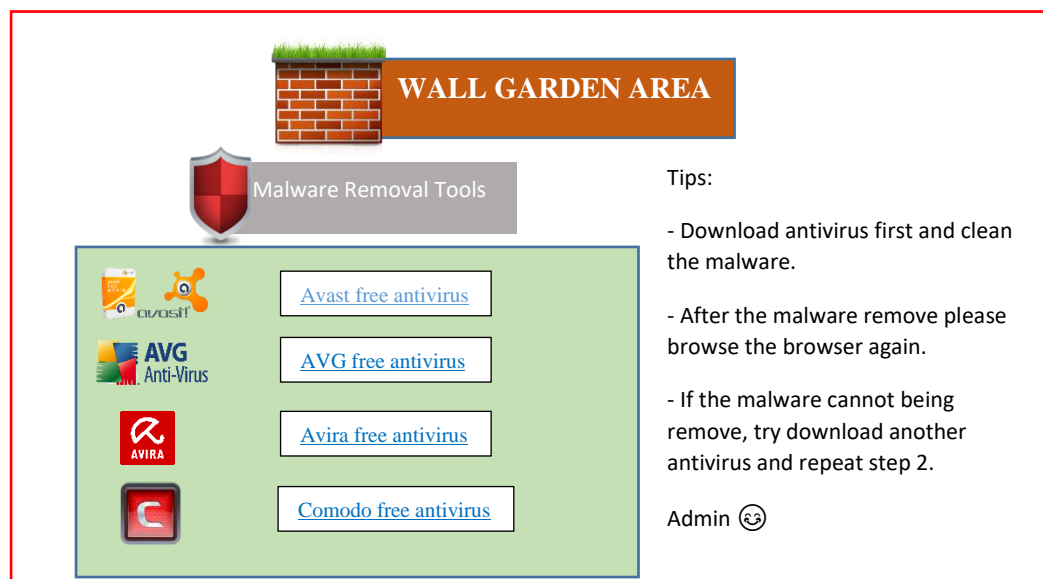


Figure 4.9: Proposed design for malware removal tools

Figure 4.9 above show the page for user to download the malware removal tools that had been suggest by the administrator. User must download the removal tools first if they have no any antivirus in their PC. There has instruction at the right side page to guide the user after remove the malware.

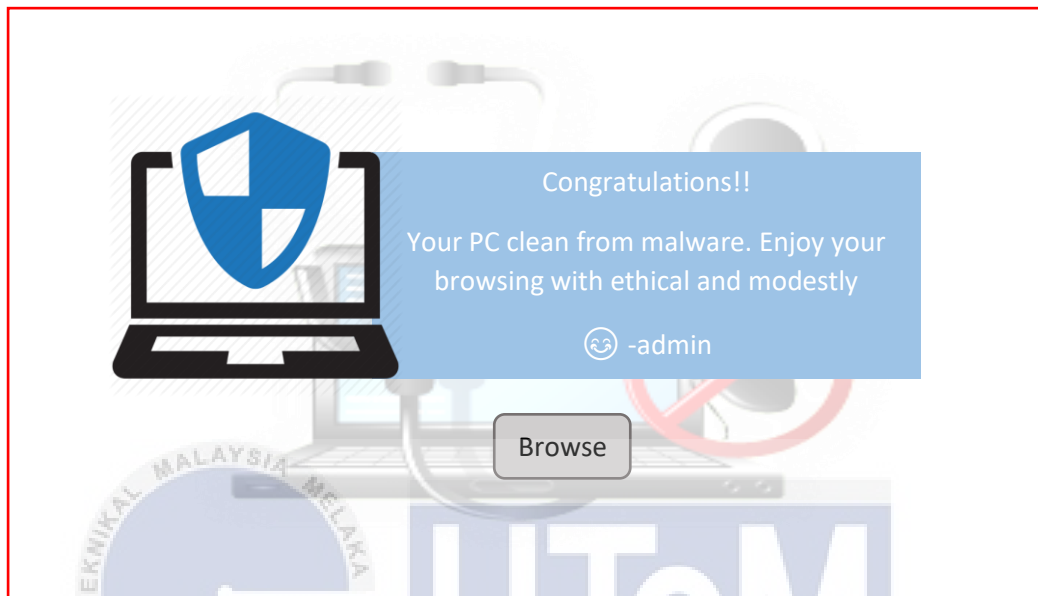


Figure 4.10: Proposed design after the malware clean

Figure 4.10 show the page that will be appear after user clean the malware using malware removal tools. After the malware clean, user can access the internet and the quarantine time is over.

4.4 Conclusion

To simplify this part, this chapter conclude describes overall of the design and flow of the project. Proposed design is important to ensure this project will run smoothly and the implementation will be done easily without to think about the biggest failure. This chapter also describes the blueprint of the proposed design of the project to make it as a reference. On the next chapter, implementation will take over and provide full documentation of the project which will comprised of the wall garden development and setting up the environment of the wall garden for remediation component from the CMERP framework.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

In the previous chapter we have explained briefly about the design of this project. In this chapter, we are going to cover about the implementation phase of the project. All of the software and hardware environment setup will be carried out and described thoroughly.

5.2 Software Development Environment Setup

In this part, we will briefly have explained the environment for the project. This project will require of apache2 installation, mysql installation, phpmyadmin installation for database, html configuration and php installation.

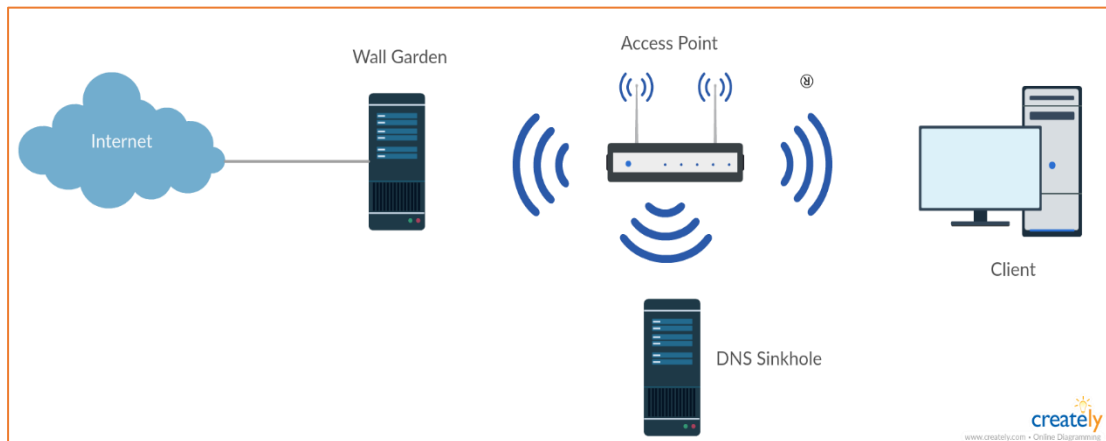


Figure 5.1: Architecture of Wall Garden

Figure 5.1 shows the architecture of wall garden environment. The DNS Sinkhole server will redirect affected user into Wall Garden before the user can access the internet. The Wall Garden server will provide the user with few options of malware removal tools to remove the malware in their device.

5.3 Software Configuration Management

In this part, we will thoroughly explain about the design and the setup of the configuration management of this project. We will also be explaining the software tools that are used to support our configuration control.

5.3.1 Configuration Environment Setup

1. Flow Chart of Apache2 Installation and Configuration

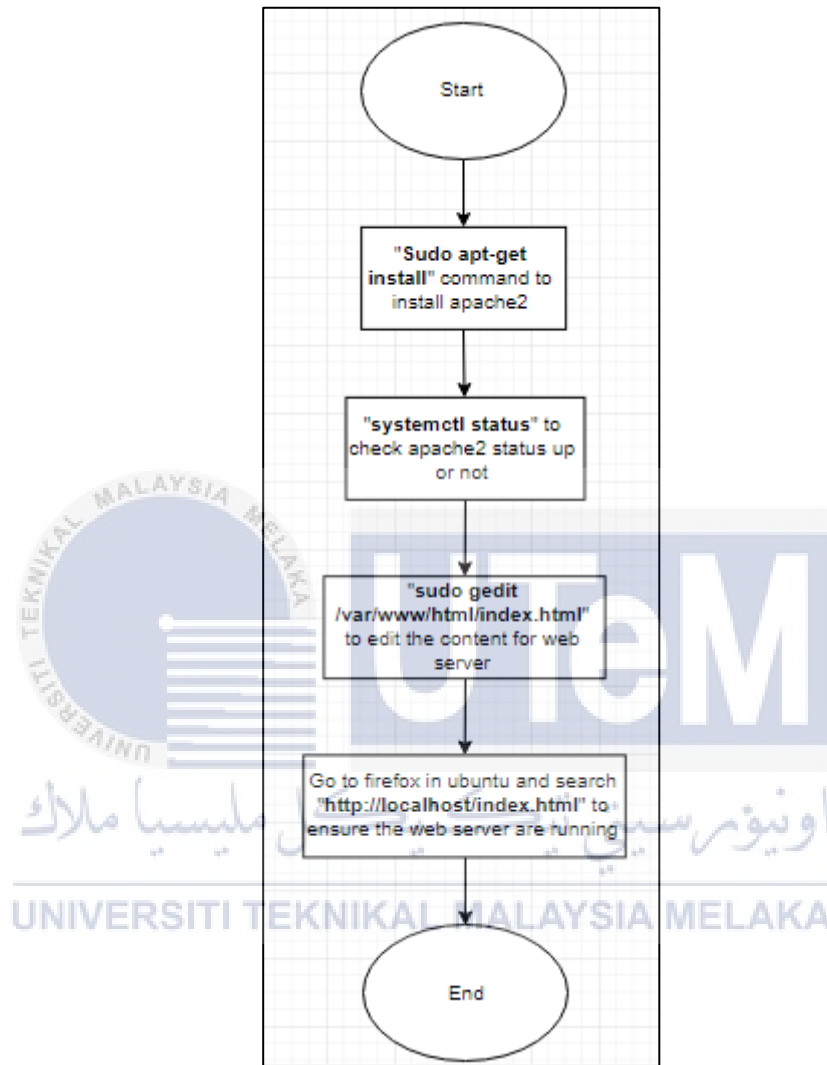


Figure 5.2: Flow Chart of Apache2 Installation and Configuration

Figure 5.2 shows the flow chart of how Apache2 is developed in this project. Firstly, Apache2 will be installed inside the Ubuntu server. After the installation is finished, we should check the status of Apache2 to see if it is up or not. When the status of Apache2 is up, we can start configuring the HTML file to create our own web server. After the configuration of the web server is complete, we can start browsing the web server that we configured in the file.

2. Flow Chart of MySQL Installation and Configuration

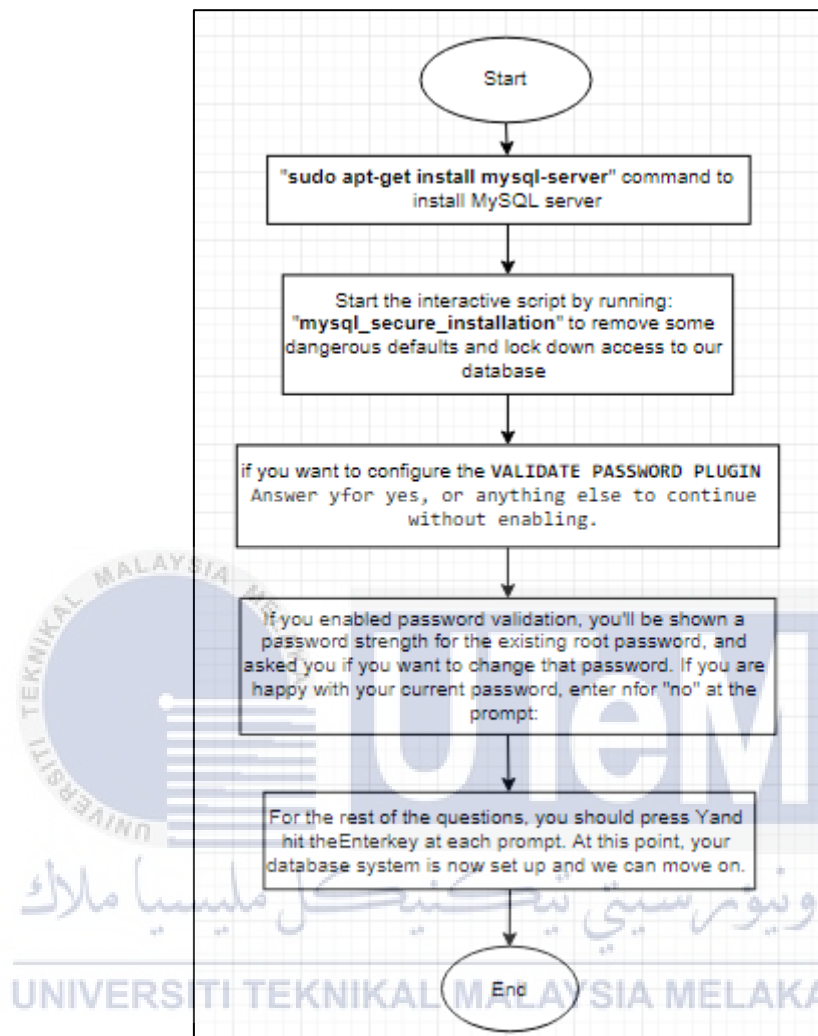


Figure 5.3: Flow Chart of MySQL Installation and Configuration

Figure 5.3 above shows the configuration and installation of MySQL server. Firstly, we need install the MySQL server. Next, to run a simple security script that will remove some dangerous defaults and lock down access to our database we should write **mysql_secure_installation**. Next, if you want to configure the validate password plugin answer Y for yes or N or another word for no. The password strength will be shown and if you do not want to change enter N for no. Lastly, for the next question answer Y for yes until finish.

3. Flow Chart of phpmyadmin Installation and Configuration

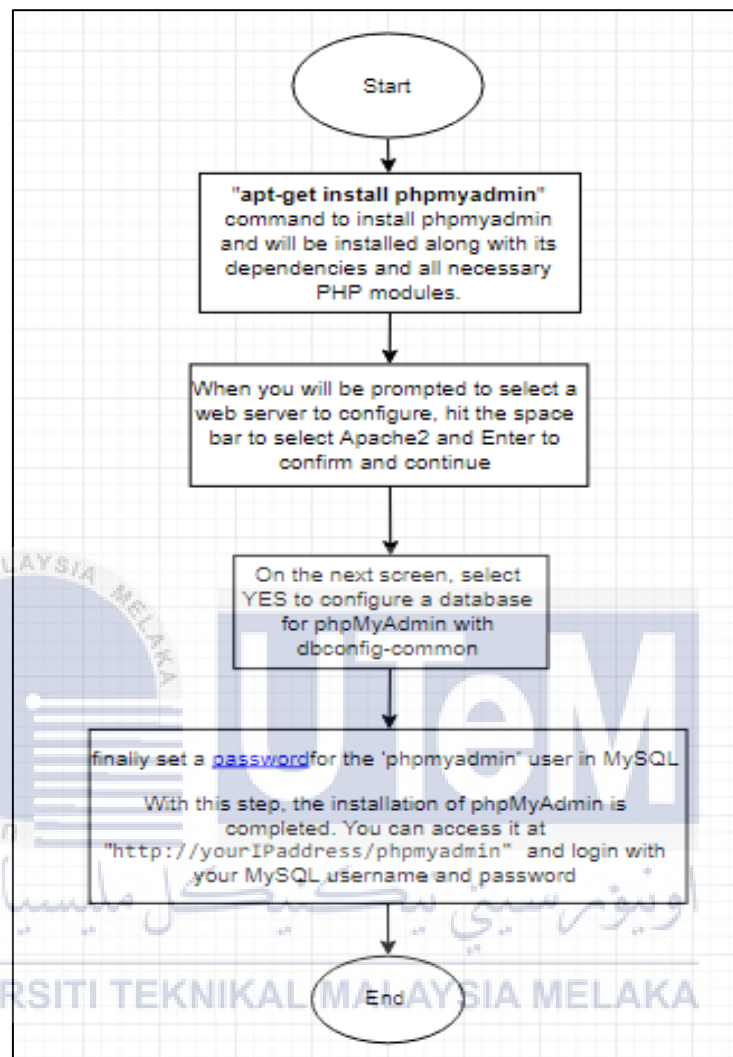


Figure 5.4: Flow Chart of phpmyadmin Installation and Configuration

Figure 5.4 above shows the flow chart on how we install and configure phpmyadmin. Firstly, we install from ubuntu terminal the phpmyadmin and necessary of PHP module will be install. Next, select Apache2 when the prompt is appear on the screen and continue the configuration. On the next screen, choose yes to configure a database with dbconfig-common to provide the database code/scripts to setup the data base, source the maintainer script libraries and launch dbconfig-common. Finally, set a password for the phpmyadmin for your username in MySQL. To complete the configuration please test the phpmyadmin in browser with <http://localhost/phpmyadmin> and login with your MySQL username and password.

4. Flow Chart of PHP Installation and Configuration

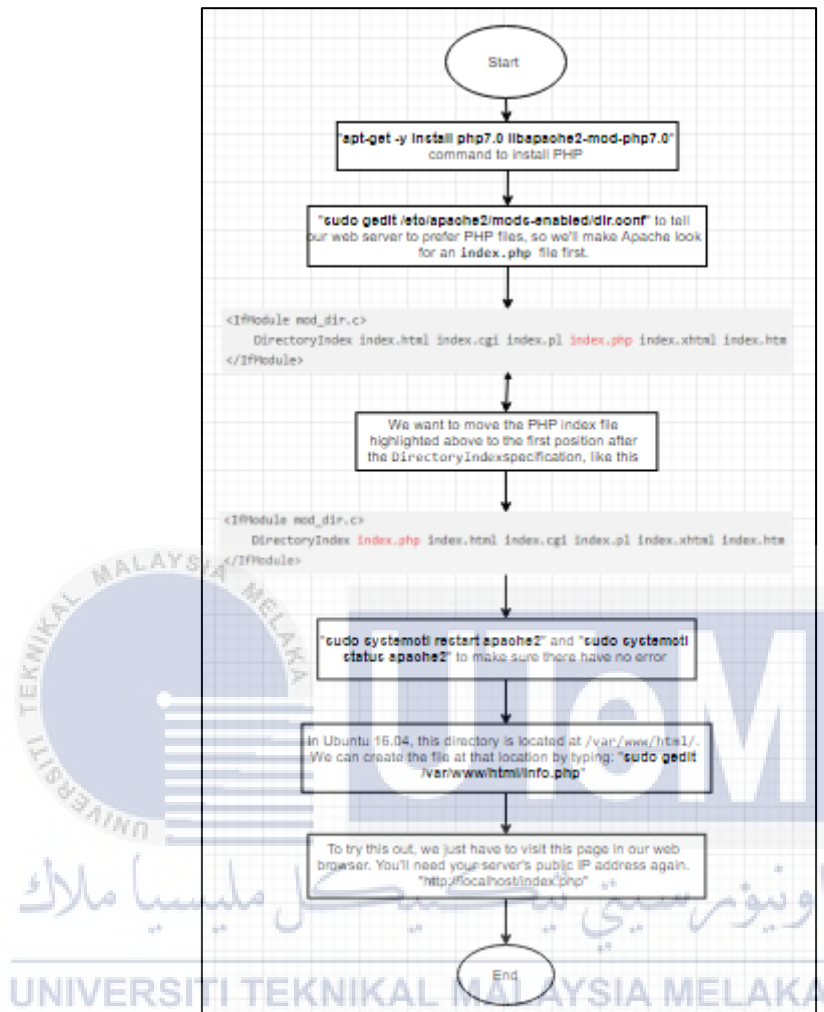


Figure 5.5: Flow Chart of PHP Installation and Configuration

Figure 5.5 above shows the installation and configuration of PHP inside ubuntu to ensure apache2 can use the PHP. Firstly, we will install the PHP through terminal in ubuntu. Next, we should configure the file by move the file in first position. Next, apache2 will be restart and will be check the status to ensure they have no error in configuration. Next, we will add the file named index.php to test the php are run or not. Lastly, open the browser and put you ip address or localhost/index.php to test the php.

5. Method Development

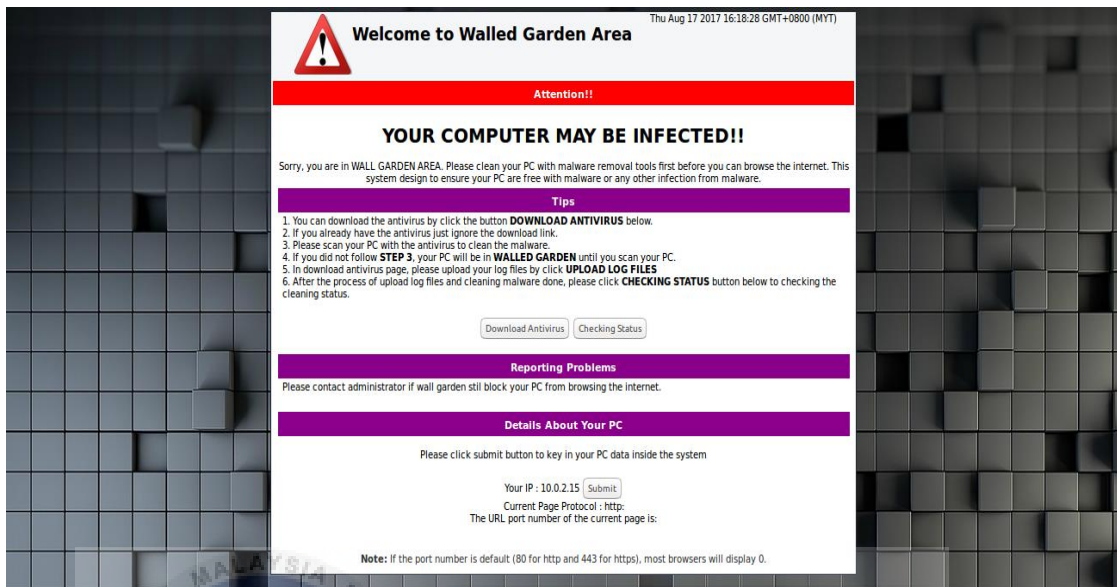






Figure 5.6: Testing Wall Garden

Figure 5.6 above shows that the wall garden had been quarantine user inside the wall garden. This is the main pages of the wall garden that will appear in browser when user being blocked by DNS Sinkhole. Here user can choose either they want to download the malware removal tools or start browsing and ignore the alert.

Removal Tools	Direct Links To Download
 COMODO Creating Trust Online®	Comodo Free Antivirus
 avast!	Avast Free Antivirus
 AVG Anti-Virus	AVG Free Antivirus
 AVIRA	Avira Free Antivirus

Note:Please upload your log files.
Hint:Log Files are available on antivirus. Please export log files as HTML files and upload by click the button below.

Figure 5.7: Malware removal tools download pages

Figure 5.7 above shows the pages that provide a few of malware removal tools that can be download by user. There have 4 malware removal tools that can be download by user to clean the malware.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

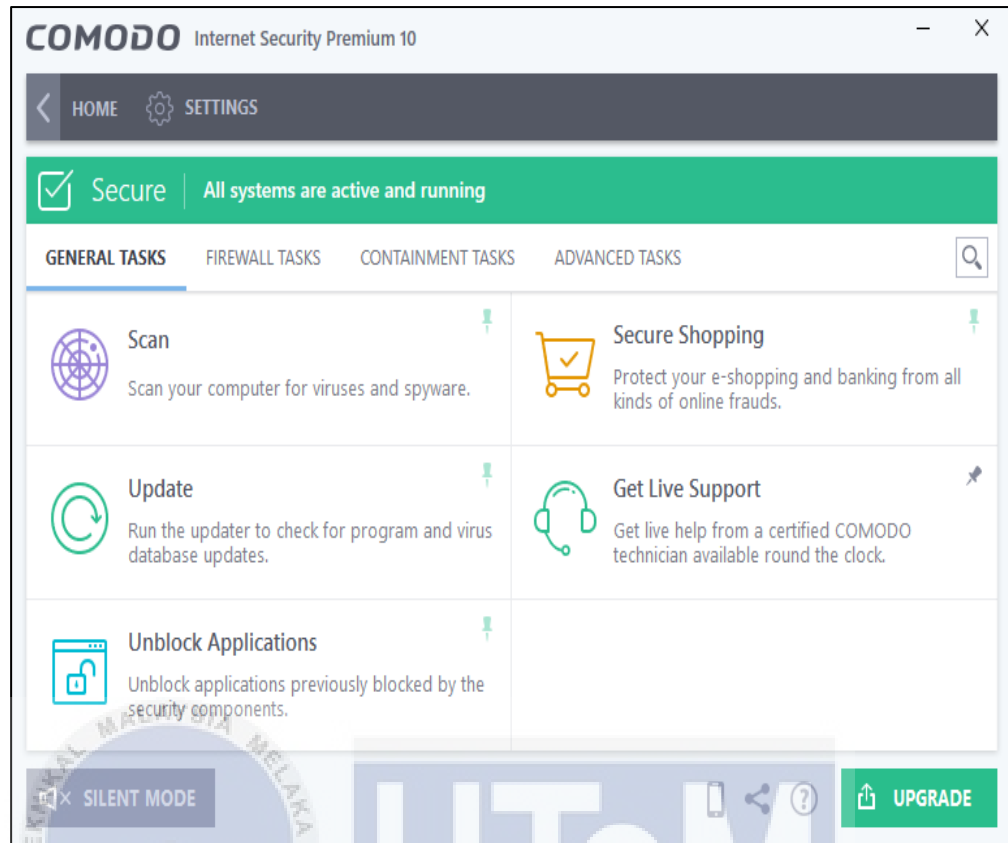


Figure 5.8: COMODO antivirus home

Figure 5.8 above shows the home of COMODO internet security premium 10. Here user can start scan by click the Scan and choose to 'Quick Scan', 'Full Scan', 'Rating Scan' and 'Custom Scan'. The user is remind to choose full scan for scanning for all entire folder in user PC. So, the cleaning progress can be done completely without skip any folder.



Figure 5.9: Malware Cleaning Process

Figure 5.9 above shows the malware cleaning process that had been done by COMODO antivirus. User should connect with the internet to ensure the antivirus will be updated with the malware signature and the database. When the process of the cleaning malware ended, it will show what file, destination and malware that will be detected and quarantine.

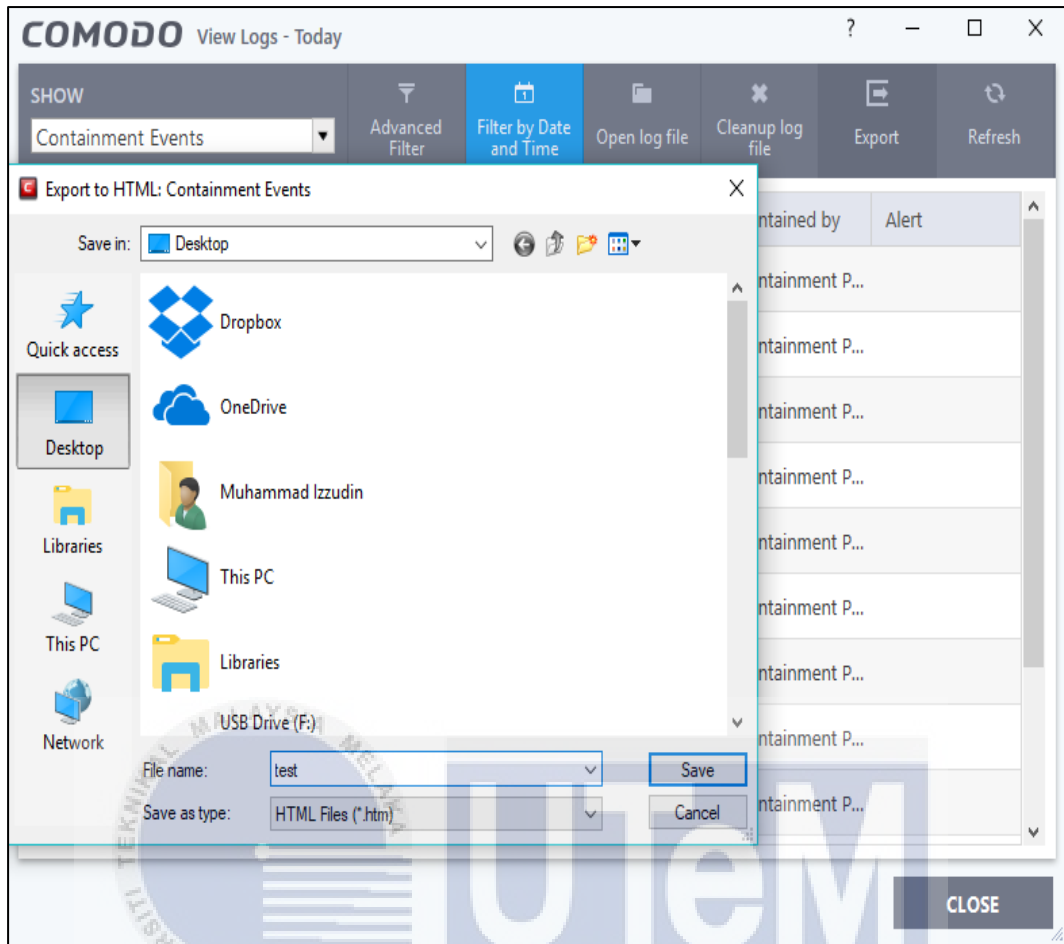


Figure 5.10: Export log files

Figure 5.10 above shows the export log files from malware removal tools.

Here user should choose 'Antivirus Events' and click 'Export' to export the log files.

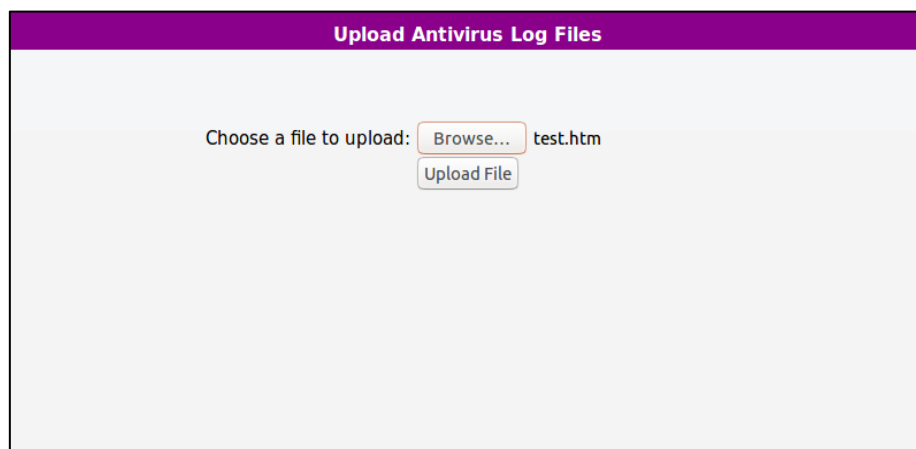


Figure 5.11: Upload log files

Figure 5.11 above shows the upload log files session into the system. Here the user should upload the log files into the wall garden system to update the database. User will choose the file that had been export from malware removal tools and click 'Submit' to send into the database.

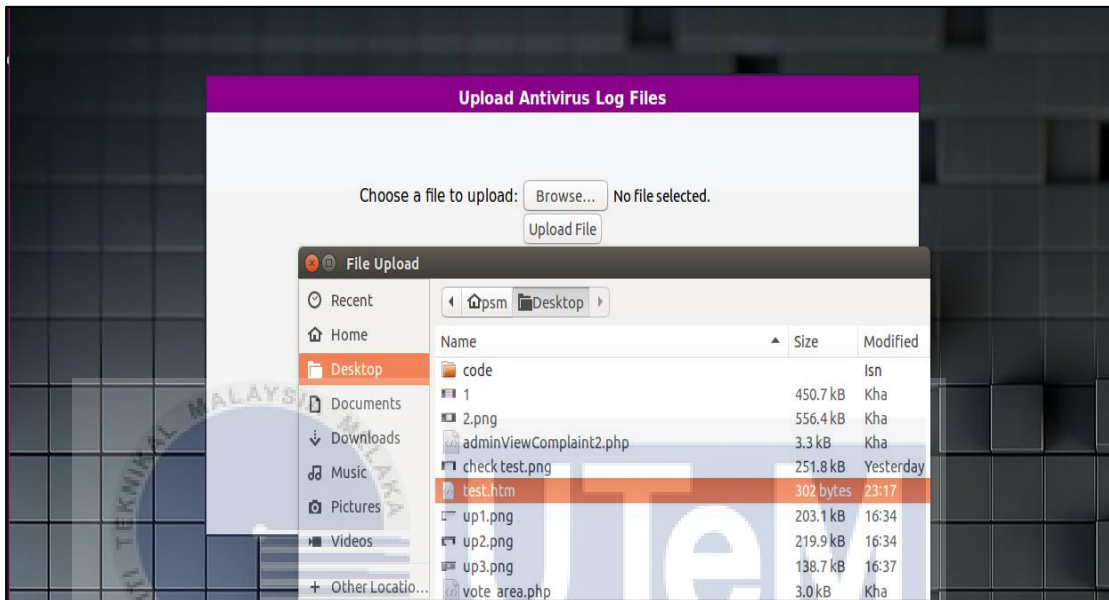


Figure 5.12: Choose log files

Figure 5.12 above shows how the user choose the log files. The malware removal tools will provide html file as a log files. User should save it inside the PC and upload it into the system.

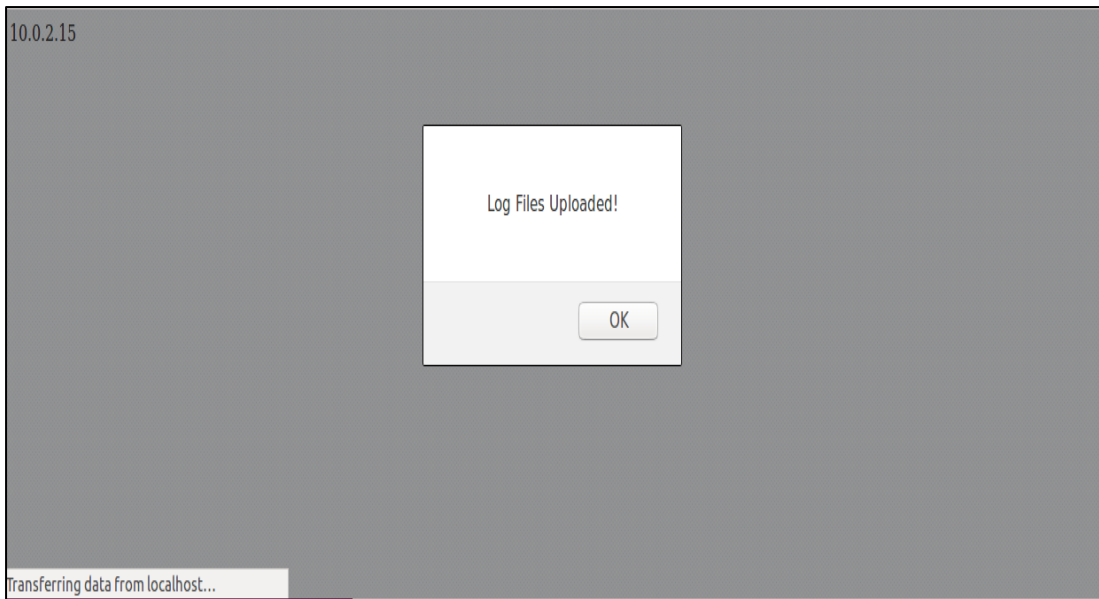


Figure 5.13: Log files uploaded alert

Figure 5.13 above shows the log files uploaded alert. This alert will appear when the log files has been success uploaded into the system.

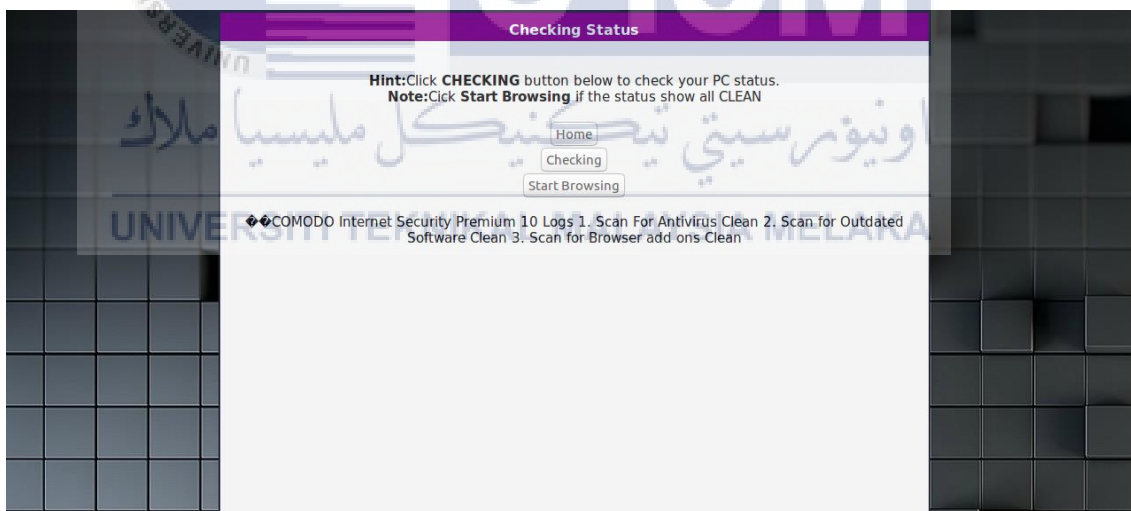


Figure 5.14: Checking Status

Figure 5.14 above shows the checking status of user PC after cleaning process. This status shows when the user clicks the **Checking** button and the page will display the checking status of the PC after cleaning process. When the status shows all the process are clean, the user can access the internet by click **Start Browsing** button.

5.4 Implementation Status

In this part, we will explain about the status of the development of each component or module. In this explanation, it will have comprised of module name, description, duration to complete and date completed. The progress is shown in Table 5.1 below.

Table 5.1 Implementation Status

No	Module Name	Description	Duration to Complete	Date Completed
1	Setting up server	Installation of Ubuntu server and setting up the physical environment for the project	1 Day	10 July 2017
2	Setting up Switch Layer 3	Configuration of D-Link switch layer 3 and setup the network ip address to be use	5 Days	17 July 2017
3	Setting up Web Server	Installation of apache2, php, phpmyadmin, MySQL server	14 Days	31 July 2017
4	Setting up Database	Connection between html pages and database	4 Days	4 August 2017
5	Setting up antivirus log files	Identify which log files will be use and how to get the log files	5 Days	11 August 2017
6	Testing the Wall Garden	Running user that contain malware to test the wall garden	5 Days	11 August 2017

5.5 Conclusion

This chapter explained about the network setup and the status of development of the project. Implementation refers to the process of turning strategies and plans into action in order to achieve the objective of the project. In Chapter 6, we will conduct the testing of the system to determine if the customization is working properly.



CHAPTER VI

TESTING AND ANALYSIS

6.1 Introduction

In the previous chapter, we have discussed briefly about the implementation of the project. Now that we have implement the project we will now continue to discuss and review the testing of this project. Testing of a project is crucial to ensure that the project is completed and met the requirements of the project, the result in testing the wall garden will be included as we go through this chapter.

6.2 Test Plan

This part will be explaining about the basis of each of the system testing. Testing scope and the activities that will be carried out throughout the testing phase will also be covered in this part.

6.2.1 Test Organization

Wall Garden can be managed by administrator n assistant of the system and user; therefore the administrator will be conducting the testing of the system. This is mainly because, the administrator will have the clear understanding of how the system works from the beginning to the end.

6.2.2 Test Environment

In the design phase, we have developed the Wall Garden environment for quarantine infected user inside it and make user clean the malware by antivirus provided by wall garden. Therefore, the administrator will be using the developed environment to conduct the testing phase of the system.

6.2.3 Test Schedule

The cycle of the testing will be as follows. Firstly, user will browse the internet by using browser and DNS sinkhole will block user and directly user into wall garden. Second, inside wall garden the user can choose between to download the malware removal tools or ignore it and start browsing. But, if user ignore the alert the malware will not be clean inside user PC. Next, when user had been used the malware removal tools and start cleaning the malware, the user should upload the log files from antivirus into the system to update the database about the malware had been clean. Lastly, after the cleaning malware process ended, the user can start browse the internet.

6.3 Test Strategy

For this project, we will be using top-down testing strategy. According to (Weißleder 2013) Top-down testing approach is conducted from the main module down to the sub module, the advantage of top-down testing strategy is that we can identify if major flaws happen during the execution of the main module and it also boost the morale of the developer as the skeletal part of the system are tested first. Figure 6.1 below shows the top-down testing strategy of this project.

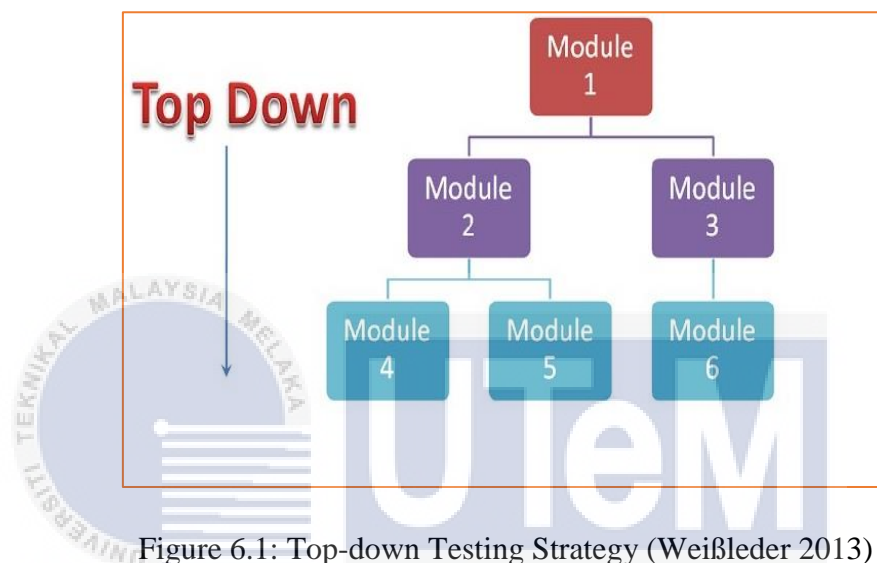


Figure 6.1: Top-down Testing Strategy (Weißleder 2013)

Firstly, the user will browser the internet by using browser to test the DNS sinkhole are blocked or not the domain that DNS sinkhole provided. Secondly, if the domain had been blocked the user PC will go to wall garden pages to be quarantine for a while. Next, the user can download the malware removal tools or start browsing the internet. Next, after the user had installed the malware removal tools, the user need to upload the log files that provided by malware removal tools into the system to update the database about malware that will be recognize. Lastly, after the process of cleaning malware ended, the user can browse the internet without worried about the malware infected their PC.

6.4 Test Design

Test design is a process that describe on how the system should be used. In these process, user will browse the internet to test the PC will be blocked or not from DNS sinkhole and will be directly user into Wall Garden. Then the user will have an option to download the malware removal tools or ignore the installation. Lastly, user should upload the log files form removal tools into the system and start browsing.

Table 6.1: Testing Table

No	Testing	Description / Data insert
1	Testing Wall Garden: Facebok.com Testing	This testing to test the Wall Garden effectiveness to block user from access the site. Data insert – www.facebook.com
2	Testing Wall Garden: Youtube.com Testing	This testing to test the Wall Garden effectiveness to block user from access the site. Data insert – www.youtube.com
3	Testing upload log files	To test upload log files into the system. Data insert – test.html
4	Testing upload log files: Data success upload	Data successful insert into database.
5	Testing checking status	The browser will fetch the data from log files and display the status of the PC after cleaning process.

6.5 Test Result and Analysis

After carrying out the testing strategy and testing plan that has been explained earlier, the project was successfully developed as mentioned in chapter 5.

6.5.1 Testing Wall Garden

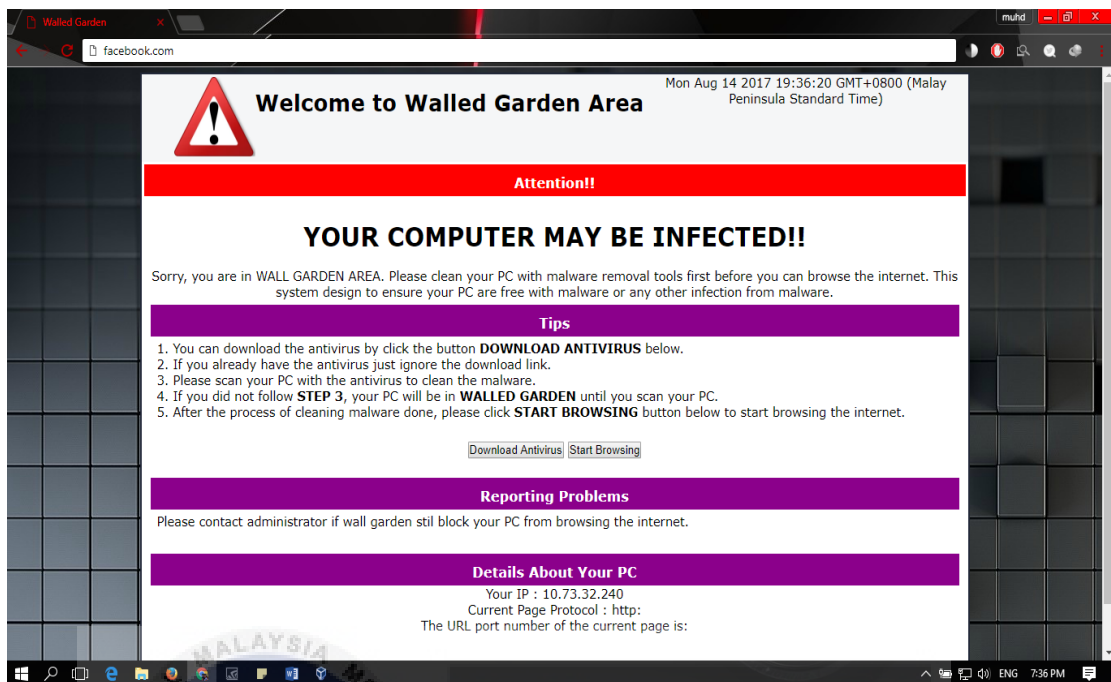


Figure 6.2: Facebook.com testing

Figure 6.2 above shows the testing of the wall garden by browse *facebook.com*. The *Wall Garden* will quarantine user from access *facebook.com*.

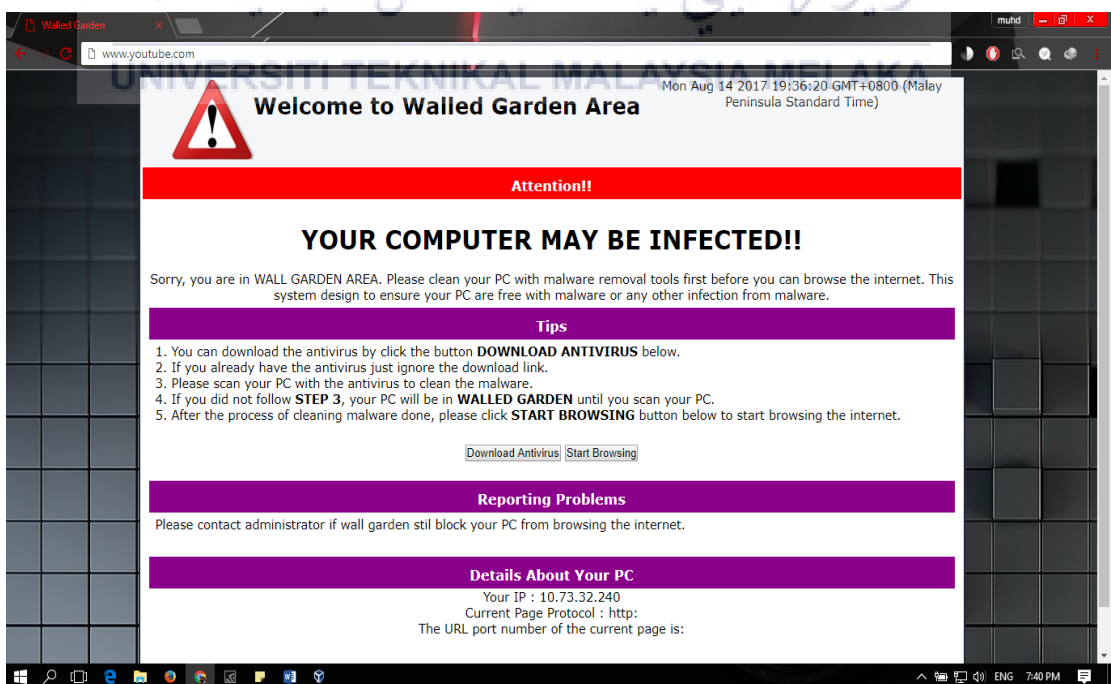


Figure 6.3: Youtube.com testing

Figure 6.3 above shows the testing of the wall garden by browse *youtube.com*. The *Wall Garden* will quarantine user from access *youtube.com*.

Conclusion for this testing is, the *Wall Garden* still quarantine user from access any site from user until they clean their PC with malware removal tools.

6.5.2 Testing upload the log files

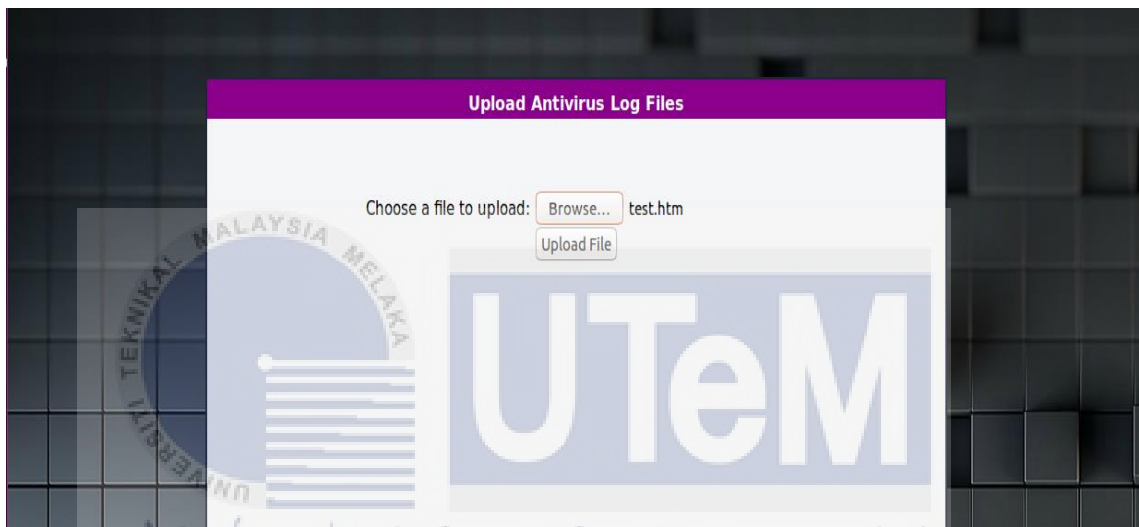


Figure 6.4: Testing upload log files

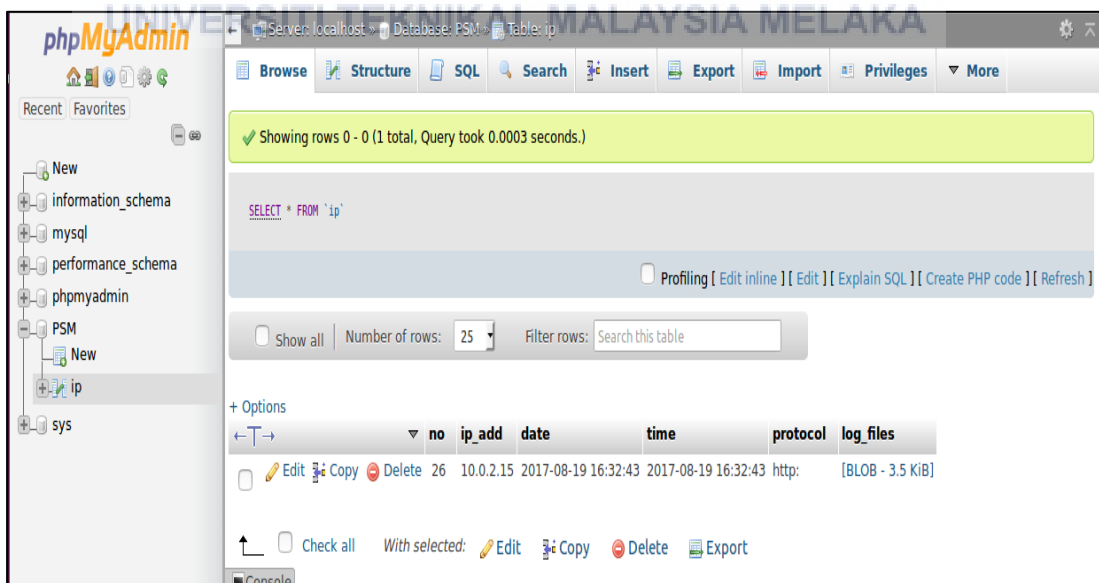


Figure 6.5: Data success upload

Figure 6.4 and 6.5 above shows the testing to upload log files inside the system. The testing had been successful upload inside the system and database. In this case, the file can be any type of file to upload inside the system because the antivirus will produce different types of log files such as .txt, .doc or .htm.

6.5.3 Testing of Checking Status

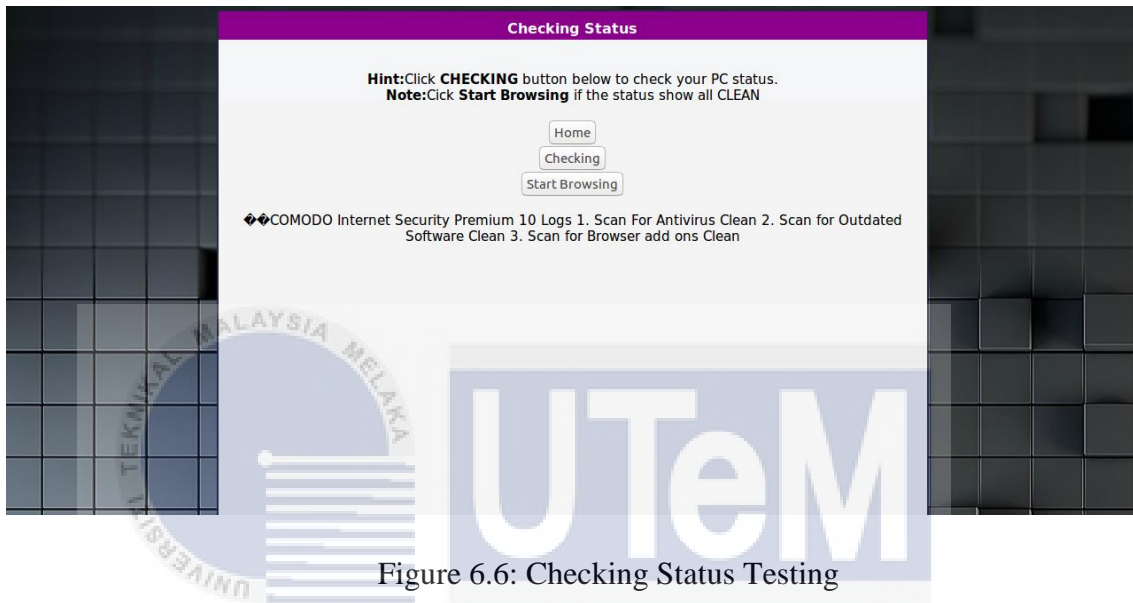


Figure 6.6: Checking Status Testing

Figure 6.6 above show the checking status testing. This test was successful test by user. The log files will show the data inside it in this page to update the status of the PC after cleaning process done.

6.6 Conclusion

As for conclusion, all of the objectives of this project had been achieved which that we have manage to study the process and architecture of a *Wall Garden*, we have also developed a system to ensure the user PC are clean from malware. There also guide the user to use the wall garden. In the next chapter, we will cover about the conclusion of the whole project development.

CHAPTER VII

PROJECT CONCLUSION

7.1 Introduction

In this chapter, we will discuss about the summarization of the project, project contribution, project limitation and future works of the project.

7.2 Project Summarization

In view of the early part of this project, we have set up to achieve three objectives which are to identify parameter involve in wall garden, to develop Malware Remediation System: Wall Garden and to test effectiveness of the wall garden to prevent infected IP from entering the internet. The first objective has been achieved in chapter 2, where we identify which parameter that will be used inside the wall garden. The second objective has been achieved in chapter 5, where we have started to implement the wall garden and provide removal tools inside wall garden. While the third objective has been achieved in chapter 6, where we have deployed the wall garden to prevent malware from separate inside internet.

7.3 Project Contribution

This project contributed to the public what is malware and type of malware that always attack our pc. This project also contributes to the implementation of the wall garden to prevent user from enter the internet directly before they remove the malware by malware removal tool that wall garden provided to user. Lastly, it also contributed how to deploy the wall garden. This would help the community especially the new users the important to secure our pc from malware attack.

7.4 Project Limitation

There are several limitations of this project. Firstly, the effectiveness of the wall garden to remove malware are unknown. We cannot measure the percentage of the malware remove and clean inside the pc. We cannot provide malware removal tools inside web server because of limitation of source. Next, the wall garden cannot take the log files from the malware removal tools and user should upload it manually. Lastly, there are constraints on equipment used to create a complete network.

7.5 Future Works

Further works need to be done to ensure the effectiveness of the wall garden can be measured and clean the malware below than high risk so the malware attack can be reduced. Furthermore, the works should be combined with one antivirus that have highly percentage to clean the malware. Besides that, error upload file for log files will be created to prevent user from upload wrong log files type. Lastly, the checking status will be improve by if one task of cleaning process is not clean, the user will be quarantine until all process are done with clean status.

7.6 Conclusion

Finally, we can conclude that all the research objectives of project have been achieved. We have successfully study the process and architecture of the wall garden, learn about malware classification and how the effectiveness of the wall garden to remove malware can be measure.

REFERENCES

- A. Feldmann, R. C. ((1999).). Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments in: Proc. IEEE INFOCOM'99 .
- B. M. Duska, D. M. ((1997)). The Measured Access Characteristics of WorldWide-Web Client Proxy Caches in: Proc. USENIX Symposium on Internet Technologies and Systems).
- Brandl, G. ((2010)). Documenting Python. October, 13(1),. 1–54. .
- Carmona, T. (2005). Segredos da Espionagem Digital, Digerati Books, São Paulo, section 1, . pp.11-12.
- Chandola, V. a. (2009). *Anomaly detection: A survey*. ACM Computing Surveys.
- Cleve, H., & and Zeller, A. (15-21 May 2005.). Locating causes of program failures. Proc. 27th ACM Int. Conf. Software Eng. (ICSE),. pp. 342-51.
- Doyle, M. .. ((2010)). Beginning PHP 5.3. Beginning PHP 5.3, 21, 841. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/22996945>.
- Drucker, P. (2001). Eficienta factorului decizional ("The efficiency of the decision makers").
- E. Levy, A. I. ((1999)). Design and Performance of a Web Server Accelerator in: Proc. IEEE INFOCOM'99 .
- E. Passerini, R. P. (July 2009). How good are malware detectors at remediating infected systems? In 6th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA),.
- Gribble, S. D. ((1997)). System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace in: Proc. USENIX Symposium on Internet Technologies and Systems .
- Idika, N. a. (2007). A Survey of Malware Detection Technique.
- ISO 9241-11. ((1998)). *Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability* .
- Johnson, E. ((1999)). (ArrowPoint Communications) Increasing the Performance of Transparent Caching with Content-Aware Cache Bypass.
- Kevadia Kaushal, P. N. (2012. 2(3)). Metamorphic Malware Detection Using Statistical Analysis. International Journal of Soft Computing and Engineering (IJSCE), .
- Lundh, F. ((2001)). Python Standard Library. Python Standard Library. <https://doi.org/978-0596000967>.
- Mandl U., D. A. (2008). The effectiveness and efficiency of public spending, . p.3. .
- McGraw, G. M. (2000). Attacking Malicious Code: a report to the Infosec Reserach Council. . : *a report to the Infosec Reserach Council* .

- Milan Dojchinovski, T. V. (06/24/2016). Linked Web APIs Dataset: Web APIs meet Linked Data.
- Milan. (n.d.). Linked Web APIs Dataset: Web APIs meet Linked Data.
- Milosevic, N. (2011). History of Malware.
- Netfinity, I. ((2000)). *IBM Netfinity Web Server Accelerator V2.0*. Retrieved from [http://www.pc.ibm.com/us/solutions/netfinity /server_accelerator.html](http://www.pc.ibm.com/us/solutions/netfinity/server_accelerator.html).
- Northcutt, S. a. (2002). Network Intrusion Detection, Third Ed., New Riders, . chapter 12, pp. 234 , chapter 13, pp. 237-243.
- Professor, S. (2000). Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection. In Proceedings of the National Information Systems Security Conference (NISSC), .
- Professor, S. F. (2000). *Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection*. In Proceedings of the National Information Systems Security Conference (NISSC),.
- Rossum, G. V. ((2010)). The Python Library Reference. October, 1–1144. Retrieved from <http://scholar.google.com/scholar?q=intitle:Python+Library+Reference#0>.
- Rossum, G. V. ((2012)). The Python Language Reference. History.
- Shen Li et al, I. J. ((2015)). Bifurcation Chaos 25, 1540037. [13 pages] .
- tutorialspoint.com. (2017). *SDLC RAD Model*. Retrieved from [sdlc_rad_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm):
https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm
- Ye, Y. e. (2009). Intelligent file scoring system for malware detection from the gray list, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 1386-1394.