

TRACKING INTRUDERS USING HONEYPOT



SURYATI BINTI ABD KADIR ZAILANI

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Networking)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS*

JUDUL: TRACKING INTRUDERS USING HONEYPOT

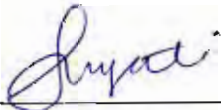
SESI PENGAJIAN: 2009/2010

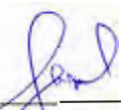
Saya SURYATI BINTI ABD KADIR JALANI
(HURUF BESAR)

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 (/)

<u> </u>	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
<u> </u>	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan dimana penyelidikan dijalankan)
<u> / </u>	TIDAK TERHAD	


(TANDATANGAN PENULIS)
Alamat tetap: LOT 1433 LORONG BS
KAMPUNG MUHIBBAH 95000, SRI AMANJUR SARAWAK
Tarikh: 29-08-2012


(TANDATANGAN PENYELIA)
DR. MOHD. FAIZAL ABDOLLAH
Nama Penyelia
Tarikh: 29-08-2012

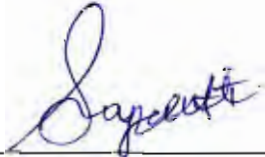
CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)


** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa

DECLARATION

I hereby declare that this project report entitled
IMPLEMENTATION OF SECURE ONLINE MUSIC STORE

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

STUDENT:  Date: 29-08-2012
(SURYATI BINTI ABD KADIR ZAILANI)

SUPERVISOR:  Date: 29-08-2012
(DR. MOHD FAIZAL ABDOLLAH)

DEDICATION

To my family, thank you for the continuous support during my study at UTeM. A deep appreciation and love for the encouragement, and guidance throughout everything that I love to do.

To all my lecturers, thank you for helping me until I can reach what I have today.

To my dear friends, thank you for the patient on being beside me to get through my study life here.

ACKNOWLEDGEMENTS

I would like to acknowledge Dr Mohd Faizal Abdollah for the assistance during the period to complete this project.

I would also like to thanks my beloved family that always giving me the support and courage that I need to complete this project.

To my friends, thank you for the help and support in during the process to complete this project.

ABSTRACT

This project is mainly focus on how Honeypot track intruders that try to probe a system in the network. Honeypot is a software that known as decoy system as it really act as decoy to protect the real system. Intruders that would like to attack the system, will not know that what is been attack is a decoy or a honeypot. Honeypot is not software to replace firewall, IDS, or anti-virus since it has its own job. This project will study on what the intruders want in a system, and how they do it. This system will be added an extra feature that is a Graphic User Interface that will make other user to use it easily. To conduct this project, equipment is set according to the early plan and should be suitable to run the software in it. A honeypot has the ability to listen on a port number which automatically will become the decoy system that is ready to be attacked. As an example, honeypot will listen to port 80 which it will listen to http service, and wait for the intruders to connect into it. Every IP address that connected to the system will be recorded and save into a log file. The data in the log file can be view for more details information. At the end, this project should meet the entire objective that has been planned earlier.

ABSTRAK

Projek ini tertumpu kepada bagaimana Honeypot boleh menjejaki penceroboh yang cuba untuk menceroboh sesuatu system dia dalam sesebuah rangkaian. Honeypot ada sejenis perisian yang dikenali sebagai system umpan kerana ia bertindak sebagai system yang khas dibangunkan untuk melindungi system sebenar. Penceroboh yang ingin menyerang system itu, tidak akan tahu bahawa system yang diserangnya adalah honeypot. Honeypot bukan merupakan perisian yang boleh menggantikan firewall, IDS, atau anti-virus kerana honeypot sendiri mempunyai tugas yang dikhaskan iaitu menjadi system umpan. Projek ini kan mengkaji kehendak seseorang penceroboh yang cuba untuk menyerang system yang sedia ada, dan bagaimana mereka melakukannya. Sistem ini akan ditambah satu cirri tambahan, iaitu mempunyai muka grafik pengguna yang membolehkan pengguna berinteraksi dengan perisian ini secara mudah. Untuk menjalankan projek ini, setiap peralatan yang sesuai telah disediakan agar projek ini boleh berjalan dengan lancar. Honeypot mempunyai keupayaan mendengar sesuatu port nombor dan secara automatiknya, ia akan membangunkan system umpan itu sendiri. Contohnya, jika honeypot diarahkan untuk mendengar pada port 80, ia secara tidak langsung telah membangunkan servis http dan bila penceroboh menemui system tersebut, mereka mungkin akan menyerang dan di situlah keupayaan honeypot untuk merekodkan segala data penting tentang penceroboh. Setiap data tersebut akan disimpan di dalam File log dan boleh diakses untuk mengetahui maklumat lebih terperinci tentang sesuatu pencerobohan terhadap system umpan itu.

TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENT	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS/SYMBOLS	xiv
CHAPTER I	INTRODUCTION	
	1.1 Project Background	1
	1.2 Problem Statements	3
	1.3 Objectives	4
	1.4 Scope	4
	1.5 Project Significance	5
	1.6 Expected Output	5
	1.7 Conclusion	5
CHAPTER II	LITERATURE REVIEW AND PROJECT METHODOLOGY	
	2.1 Introduction	6

2.1.1 History of Honeypots	6
2.2 Facts and Findings	7
2.2.1 Domain	8
2.2.2 Keyword	8
2.3 Project Methodology	10
2.3.1 Traditional life cycle method	10
2.4 Project Requirements	12
2.4.1 Previous project research	12
2.4.1.1 German HoneyNet Project – HoneyNet research at the Laboratory for Dependable Distributed Systems	12
2.4.1.1.1 Honeypot Background	13
2.4.1.1.2 Tools and technique Tools and Technique (German HoneyNets Project)	13
2.4.1.1.3 Findings (German HoneyNet Project)	14
2.4.1.1.4 Honeypots as a Detection Solution: (German HoneyNet Project)	14
2.4.1.2 Design of network security projects using honeypots	16
2.4.1.2.1 Lab Project	16
2.5 Proposed Project	17
2.6 Project schedule and milestone	18
2.6.1 Gantt Chart	18
2.7 Conclusion	19
CHAPTER III	ANALYSIS
3.1 Introduction	20
3.2 Facts and Findings	20

3.2.1 Problem analysis	20
3.2.2 Honeypot concept	22
3.2.2.1 Architecture	23
3.3 Requirements Analysis	27
3.3.1 Quality of data	27
3.3.1.1 Data Collection	27
3.3.1.2 Command to use honeypot	28
3.3.1.3 Data collection flow	31
3.3.1.4 Data captured format	32
3.3.1.5 Architecture on attacker flow	33
3.4 Conclusion	34
CHAPTER IV	
DESIGN	
4.1 Introduction	35
4.2 High-level design	35
4.2.1 Introduction	35
4.2.2 Scope	35
4.2.3 Definitions	36
4.2.4 Overview	36
4.3 General description	37
4.3.1 Product perspective	37
4.3.2 Tools used	37
4.3.3 General constraints	37
4.3.4 Assumptions	38
4.3.5 Special design aspects	38
4.4 Design details	38
4.4.1 Main design features	38
4.4.2 Applications architecture	39

4.4.3 Technology architecture	40
4.4.3.1 Web application architecture	40
4.4.3.2 Presentation layer	40
4.4.3.3 Data access layer	41
4.4.3.4 Tools Used	41
4.4.4 Functional requirements	41
4.4.5 Files	41
4.4.6 User interface	41
4.4.7 Reports	41
4.4.8 Error handlings	41
4.4.9 Interfaces	42
4.4.10 Help	43
4.4.11 Major class	43
4.5 Non-Functional requirements	43
4.5.1 Performances	43
4.5.2 Security	43
4.5.3 Reliability	43
4.5.4 Maintainability	44
4.6 Navigation design	44
4.7 Conclusion	47

CHAPTER V

IMPLEMENTATION

5.1 Introduction	48
5.2 Software development Environment setup	49
5.3 Software Configuration Management	53
5.3.1 Configuration environment setup	53

	5.3.2 Version Control Procedure	54
	5.4 Implementation status	55
	5.5 Conclusion	59
CHAPTER VI	TESTING	
	6.1 Introduction	60
	6.2 Test Plan	60
	6.2.1 Test Environment	60
	6.2.2 Test Schedule	62
	6.3 Test Design	63
	6.3.1 Test Description	63
	6.3.2 Test Data	65
	6.4 Test results and analysis	66
	6.4.1 Honeypot activity	66
	6.5 Honeypot source code	69
	6.6 Conclusion	70
CHAPTER VII	CONCLUSION	
	7.1 Introduction	71
	7.2 Project summarization	71
	7.2.1 Propositions for improvement	72
	7.2.2 Contribution	72
	7.3 Observation on weaknesses and strengths	72
	7.4 Conclusion	74
	REFERENCES	75
	APPENDIX A	77
	APPENDIX B	79

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1.1	The research problems in this project	3
Table 1.2	The research problems and research questions in this project	3
Table 1.3	The research problems, research questions, and research Objectives.	4
Table 3.1	Current scenario on Honeypot	21
Table 3.2	The IP address of the Honeypot	26
Table 5.1	Device configuration Details (Honeypot Network)	49
Table 5.2	Hardware setup	53
Table 5.3	Version control procedure	54
Table 6.1	Hardware and software involve	61
Table 6.2	Setup for setting	65

LIST OF FIGURES

DIAGRAM	TITLE	PAGE
Figure 2.1	Traditional life cycle method	10
Figure 3.1	Flow of honeypot on listening to decoy system port number	23
Figure 3.2	Honeypot working with software to detect malicious activity	24
Figure 3.3	Experimental architecture of Honeypot	25
Figure 3.4	Logical Design	25
Figure 3.5	Physical Design	26
Figure 3.6	Find the honeypot.py directory	28
Figure 3.7	Run the honeypot.py script using python	28
Figure 3.8	Start the service	28
Figure 3.9	Insert the banner	29
Figure 3.10	View logs file on command prompt	29
Figure 3.11	Honeypot User Interface	30
Figure 3.12	Honeypot activity flow	31
Figure 3.13	Intruders data captured example	32
Figure 3.14	Banner on the decoy system	32
Figure 3.15	Flow of attacker	33
Figure 4.1	System Architecture for user	39
Figure 4.2	Sytem architecture for Administrator	40
Figure 4.3	User interface of Honeypot	42
Figure 4.4	The web application act as decoy system	42
Figure 4.5	Insert Banner Information	44

Figure 4.6	Insert IP Address of the Honeypot	45
Figure 4.7	Insert Port No.	45
Figure 4.8	Start button	46
Figure 4.9	Result display	46
Figure 4.10	Exit	47
Figure 5.1	Honeypot activity	49
Figure 5.2	Honeypot flowchart	50
Figure 5.3	Scanning malicious activity flowchart	51
Figure 5.4	Malicious attack flowchart	52
Figure 5.5	First data captured	55
Figure 5.6	Second data captured	56
Figure 5.7	Honeypot GUI	56
Figure 5.8	Data captured using GUI	57
Figure 5.9	Interface of honeypot web browser	57
Figure 5.10	Pyloris software	58
Figure 5.11	An attack from Pyloris	58
Figure 5.12	Attacking data captured and stored in log file	59
Figure 6.1	Physical design	63
Figure 6.2	Logical design	64
Figure 6.3	First data captured	66
Figure 6.4	IP Address of the attacker	67
Figure 6.5	User-Agent	67
Figure 6.6	Date and time of the attack	67
Figure 6.7	Data captured during the attack by Pyloris	68
Figure 6.8	Honeypot web interface	68
Figure 6.9	Pyloris software	69
Figure 7.1	Data captured by honeypot	73

LIST OF ABBREVIATION/SYMBOLS

ACRONYM	WORD
Honeypot	Decoy system
IP	Internet Protocol
Port Number	Addressing information
Web server	Computer program that dispenses web pages
Python	Programming Language
Malicious activity	An activity that aims to destroy a system
Intruders	A person that intend to attack a system
IP Address	Unique address that is use to communicate with other devices
Banner information	A sentence that will display on certain interfaces
Interface	Device or system that unrelated entities use to interact
DoS	Denial of services
GUI	Graphical User Interface

CHAPTER I

INTRODUCTION

1.1 Project Background

A honeypot is a computer system on the Internet that is specifically set up to attract and "trap" people who attempt to penetrate private and unknown users computer systems. Unlike firewalls or IDS sensors, honeypot is something that the intruders to want to interact with. Honeypot acts as a decoy system to detour any suspicious activity that might harm our computer system.

Conceptually, honeypots are very simple. They are a resource that has no production value, it has no authorized activity. Whenever there is any interaction with a honeypot, this is most likely malicious activity. For example, if someone is in the internal network scanning for vulnerable desktops, and the intruder scans the internal honeypot, your honeypot will easily detect and log this unauthorized activity as no one should be connecting to your network.

Honeypots are unique; they do not solve a specific problem. Instead, they are highly flexible tool with many different applications to security. It depends on what it wanted to achieve. Some honeypots can be used to help prevent and detect malicious attacks. Likewise, other honeypots can be used for data collection in any on-going research project.

The Advantages:

- a.) Small data sets: Honeypots collect small amount of data, but almost all of this data is real attacks or unauthorized activity. Instead of dealing with 5,000 alerts, honeypots collect only malicious activity, it is easier to analyse and react to the information they collected.
- b.) Reduced false positives: Honeypots detect or capture is an attack or unauthorized activity, vastly reducing false positives.
- c.) False negatives: Unlike most technologies, it is very easy for honeypots to detect and records attacks or behaviour never seen before in malware attacks.
- d.) Cost effective: Most honeypots can easily run on any Pentium processor computer with 128 MB of Ram
- e.) Simplicity: Honeypots are very simple; there are no advance algorithms to develop, nor any rule bases to maintain.

In general, there are two different types of honeypots, **low-interaction** and **high-interaction**. Level of interaction measures how much activity, or interaction, an attacker can have with a honeypot. Low interaction honeypots limit the level of interaction by emulating services. The interaction an attacker has with the honeypot is limited by how advance the emulation of the service. An example of a low interaction honeypot is **Honeyd**. In contrast, high interaction honeypots do not emulate services; instead they provide real applications for attackers to interact with. An example of a high interaction honeypot is **Honeynets**. Neither is better than the other. Low interaction is simpler and has less risk (as the attacker can do less while interacting). High interaction allows us to learn more about the intruders including chatting with the intruders however this is not advisable as it poses great risk being hacked or attacked while communicating.

1.2 Problem Statements

Stealing someone's identity is one of the best known techniques for hackers to access confidential information in a corporate environment.

It is known to the public users that the internet is not the safest media. Intrusions into foreign networks have become easier and too convenient for hackers to hack regardless of the firewall or security protection imposed onto the selected network. It is just a matter of time for these bots scan networks to insert and infect fully automated malicious code into foreign remote machines.

Network security issues are of major concern for all businesses to keep the affirmation of the nature of their business under strict confidentiality. These issues are not recent as it started since information was transferred from sender to respective recipient. Data transition has been prone to attack and intrusion as useful pieces of information is crucial to survive tough competition in corporate world.

Table 1.1 shows the research problems in this project.

RP	Research Problem
RP 1	There are many type of techniques used to track intruders.
RP2	Lack of security in Honeypots.

Table 1.2 shows the research problems and research questions in this project.

RP	RQ	Research Question
1	1	How to choose from the techniques to prevent the attacks?
2	2	How to increase Honeypots security?

1.3 Objectives

The objectives of this project are:

- To develop simple alert system whenever unknown source probes the network.
- To detect malicious attack launch by the intruders.
- To develop GUI for the honeypots so that it will be easy to track and analyse intruder's activities.

From the research problem statements and research questions, the research objectives have been create to overcome the problems and questions as shown in Table 1.3

Table 1.3 shows the research problems, research questions, and research objectives.

RP	RQ	RO	Research Objective
1	1	1	To develop simple alert system whenever unknown source probes the network.
2	2	2	To detect malicious attack launch by the intruders.
2	2	3	To develop GUI for the honeypots so that it will be easy to track and analyse intruder's activities.

1.4 Scope

Scope of this project will involve data capturing at Experiment wireless Lab, Faculty of Information and Communication, Universiti Teknikal Malaysia Melaka. The operating system for main server is windows server 2008. The software used is python 2.5 to write honeypot script. Data capturing duration will be held from 2nd April 2012 until May 2012.

1.5 Project Significance

This project is important to introduce and explain to user how honeypot works does and will do to protect their server or network.

1.6 Expected Output

The expected output from the honeypot includes the ability to detect an early hacking from the intruders and detecting malicious activity launched by the intruders to the honeypot. A simple and easy graphical user interface is expected to be done at the end of the project.

1.7 Conclusion

Honeypot is important in tracking and avoiding the intruders breaching into the main important system. It is useful and can con it ways to attract the intruders into believing they are communicating with the main security system. Since honeypot is an attractive system to attract and track the unknown intruder, honeypot will be useful in protecting the real important system.

CHAPTER II

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter will discuss the literature review and analysing the tools that were used for this project. These include honeypot, firewalls, and the tools that will be used in conducting this project. This is followed by reviewing on several articles which is relevant to this project. The articles that have been reviewed are based on the real projects that were conducted and all the tools, techniques and results of the project are shared with others. These previous projects gained successful outcomes in their project experiments.

2.1.1 History of Honeypot

The idea of honeypot began in 1991 with “The Cuckoos Egg” and “An Evening with Bredford” by Clifford Stoll and Bill Chewick . The first publication, “The Cuckoos Egg” was about the experience catching a computer hacker that was in Clifford Stoll Corporation searching for secrets about the corporation. The other publication, “An Evening with Berferd” was about a computer hacker’s moves through traps where Bill Chewick and his friend used to catch the intruder. Both of this publication is the beginning of honeypot. The first type of honeypot called the Deceptive Toolkit was released in 1997. This tools main function is to attack at the intruder. First commercial honeypot came out in 1998, it was called Cybercop Sting. In 2002 the honeypot started to be shared and

being used all over the world. In year 2005, The Philippine Honeypot Project was launched. Today, a very popular honeypots project is taking place which is called Honeynets Project.

2.2 Fact and Findings

In this literature review some details regarding the project which is "Tracking Intruders using honeypots" will be discussed. Discussion will be based on the objectives of the project.

First objective is to determine how intruders probe the honeypots and what they are looking for. This objective shows the purpose of honeypots. Probing the system without authorization is illegal and the intruders sometime will succeed in their mission. However, many computer users do not know the intruders motive and as why they probe the system. Honeypot is one of the ways to track the intruder's activities in our network system.

Second objective is to build a simple alert system whenever someone probes the network. The new system is secure with specific security features, thus, if a third party or unknown user can access through it, it is most probably the intruder's activities. In this case, honeypots will acts as an alert buoy to the whole system.

The third objective is to build Graphic User Interface (GUI) for the honeypots. This will enable the administrator to track the intruder who is probing the network and some of the threats that they use to attack the system. GUI will be able to identify malware that is in use to attack the system.

2.2.1 Domain

The domain for the project is the ICT in Advance manufacturing Technology based on Network Security. It involves the activities in organizations, enterprises, and institutions undertake to protect the value and current usability of their asset. Hence, to uphold the integrity and continuity of operations. An effective network security strategy requires identifying threats and then choosing the most effective set of tools to combat them. Specifically, network security protects the usability, reliability, integrity, and safety of your network information and data. Effective network security targets a variety of threats and prevents them from entering or spreading on your network.

2.2.2 Keyword

1.) Firewalls

A firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls will be put to work before Honeypots. Firewall will act as first layer of security in the system including tracking intruders this is due to several security systems in a network. Moreover, firewall is a tool that has log on all traffic that through it. Using firewalls log, we can track every moves of the intruders and find out what they do and want on our system.

Second, firewalls have alerting ability. Using this capability, firewalls will alert us if the intruder has penetrated through the firewall entering honeypots section. Firewalls alert is simple and easy to build. We can detect the intruders by viewing the traffic in and out of our system, since none can connect to the honeypot; it is most-likely the intruders will be kept busy by honeypots via fake authentications.