Advances and Applications in Mathematical Sciences
Volume 21, Issue 5, March 2022, Pages 2765-2780
© 2022 Mili Publications, India

AN APPROACH FOR DETECTING PASSWORD PATTERN IN DICTIONARY ATTACK

TANVI GAUTAM[1] and UTKARSH SINGH[2]

[1]Assistant Professor
Department of Information Technology
IPEC, Ghaziabad, AKTU, U. P. – 201010, India
[2]Student
Department of Information Technology
IPEC, Ghaziabad, AKTU, U.P. – 201010, India
E-mail: singhutkarsh2799@gmail.com

Abstract

Cyber Crime encompasses such criminal activities dealing with computers and networks (also known as hacking). Additionally, cyber-crime also includes crimes conducted through the Internet such as online fraud, Identity theft and credit card account thefts. In both online and offline activities password cracking is one of the loopholes of cyber-crime. In password protected remote login services password cracking has increased using several password-guessing techniques. The authorized user login into the online services conveniently while blocking such password – guessing techniques is difficult. Using Automated Turing Tests (ATT) in preventing password attacks is effortless and constructive method to execute, but cause average amount of hindrances to the user. In this paper, we discuss the existing dictionary attack methodologies: Automated Turing Test (ATT), MD5 Hash Generator, Cain and Abel to study large-scale password cracking methodologies. Also, proposed the new dictionary attack mechanism using linear methodology of analyzing the pattern of password guessing. The password pattern analysis is study using the existing password datasets available online.

1. Introduction

Password cracking is a process of attempting to guess or crack password to gain access to a system. It can also be signifying as a process for recovery of

2020 Mathematics Subject Classification: 68T01.
Keywords: Password Attack; Dictionary attack; Brute force attack; linear methodology; Automated Turing Test; Cain and Abel.

[1]Corresponding author; E-mail: tanvi.gautam@ipec.org.in

Received September 29, 2021; Accepted November 30, 2021

password from data that are stored into the system. Therefore, password cracking is an approach, which uses repetition in order to try to guess the password [1]. Moreover, the purpose of password cracking might be help as a user to recover a forgotten password, gain unauthorized access to a system, or perform preventive measure by system administrator to check for password strength [2]. Identity authentication through password is still a widely method of ensuring system security, despite the increased use of alternative techniques such as graphical password, smart card and biometrics. In order to force users to create strong password, system administrator policies often enforce several complex rules intended to force users into creating strong password. Under such rules [3], users may be required to use numeric or special character, have to enter password of a minimum length and avoid word found in a dictionary. In this way, methods of password cracking can be paraphrase as a test for password guessing, because we do not know if the proper method/test is going to be efficient. Hence, we are going to see the different methods of performing password guessing. There are three types of password cracking methods [4] that can be automate with the tools:-

- Dictionary attack
- Brute force attack
- Rainbow table

Dictionary attack [4]**.** A file of words runs against user accounts and if the password is a simple dictionary word, it can find easily and quickly. The attackers use a dictionary comprised of words act as a suspect that the target has used in their password. The attacker then applies colander rules [5] to these input words, such as capitalization the first letter, adding three digits to the end, changing the letter 'a' to '@' etc. to further match the target password. The success of dictionary attack depends not only on the input dictionary selected, but also on the word colander rules applied. Attacker first tries with little input dictionaries, if this fails, then the attacker cracks the password using much larger input dictionaries.

Brute force [4]**.** The process of password – guessing is the most time – consuming because it repeatedly tries until the password is found. Brute force attack is most common mechanism in password cracking [6]. A method does not use any input dictionary of human generated words. These attacks are popular because the most input dictionary only covers a fraction of the total words that are mostly use by users while creating their passwords.

There are four types of attacks, which is performing during the brute force attack methods [7]: -

• Pure brute force is brute force attack that does not use outside probability information also not found inherently in the key space searched.

• Letter frequency analysis attack is an attempt to use the frequency of characters appearing in a training set to increase the effectiveness of brute force attacks.

• A Markov model is a way to represent the join probability of different characters appearing together.

• Targeted brute force attacks can comprise letter frequency analysis and Markov models, but applies outside logic to these attacks. For example, performing the letter frequency analysis attack mechanism by using a different character set for each character position.

Rainbow table [4]**.** The idea of this technique was first pioneered by Philippe Oechslin as a fast form of time – memory trade – off, which he implemented in the Windows password cracker Ophcrack [8]. It uses a refined algorithm by using a number of different reduction functions to create multiple parallel chains within a single “rainbow” table. It also reduces the probability of false positives from accidental chain collisions, thus increasing the probability of correct crack for a given table size. The use of multiple reduction functions also greatly increases the speed of look – ups.

Rainbow tables are specific to the hash functions that created for md5 tables only. Later, the more powerful rainbow crack program developed that generate and use rainbow tables for a variety of character sets and hashing algorithms [9] (hashes include lm hash, md5, sha1, ntlm). Rainbow tables based on the idea of hash chains where the important concept is the index value in a standard offline password cracking attack, the attacker possess a password hash, and its attempting to guess the password that created it. That is why rainbow tables can be thought of as a very efficient, compression algorithm for hash look – up tables the index value ranges from 0 to (key max-1).

2. Related Work

A dictionary attack is a technique or method use to breach the computer security of a password-protected machine or server. A dictionary attack attempts to defeat an authentication mechanism by systematically entering each word in a dictionary as a password or trying to determine the decryption key of an encrypted image or a document to try to crack the password [5]. Dictionary attacks are often successful because many users and businesses use ordinary words as passwords and it becomes easy for an attacker to crack the password [10]. These ordinary words are easily accessible in a dictionary, such as an English dictionary. For making the security essential password usage is must but sometimes it is easy to guess password using several existing automated programs and tools developed by the hackers in dictionary attack [11]. The existing system of preventing the dictionary attacks are: Automated Turing Test [12] (Captcha image) [13], MD5 Hashing using MD5 Hash generator [14], Cain and Abel [15].

2.1. Automated Turing Test

The Automated Turing Test (ATT) is a high graded level security technique for directing the risk of unwanted or malignant bot programs [16]. Currently, CAPTCHA (Completely Automated Turing Test to Tell computers and Humans Apart) a high graded level mechanism of ensuring the standard security in addressing the unwanted or malignant programs. The existing bot programs are follows such as: -

- **Voting bots.** Enables the casting of thousands of votes as masquerading humans in online polls.
- **Email account registration bots.** Wherein sign up for thousands of accounts could be done in every minute with free email service provider.
- **Email spam bots.** That send thousands of spam messages in every minute.
- **Weblog bots.** Posting of false comments in weblogs directing both reader and search engines to irrelevant sites.
- **Search engine bots.** Raised the ranking of web pages in a search engine.

The principle of CAPTCHA is that – it performs as a simple two – round authentication protocol as follows.

S(ystem) → U(ser): a CAPTCHA test

S → U: reply

It is a test conducted pass by most of the humans but recent computer programs cannot pass. ATT test is repeatedly depending on a hard, open problem of AI such as automatic recognition of distorted text, or of human speech against a noisy background. Humans are the only one who returns sensible response for determining whether there is a human behind the computer, a one accessing the system. CAPTCHA acts as a mechanism for measuring the progress of security level like as in AI. For preventing a bot program such as forwarding spam emails, hence for stopping it CAPTCHA used that is an email not forwarded until CAATCHA apply to it before authentically login to the account. In the existing system, an automated test is a program that has high probability of decreasing the security risks of guessing the passwords. In CAPTCHA, attackers restrict the number of password guesses from particular system before being lock. Using the high computation skills CAPTCHA can be broken easily. Such process involves an entire system test that ensures a successful login.

2.2. MD5 Hashing [14]

MD5 Hashing can be defined as "Message Digest Algorithm" which is a widely used cryptography hash function that was invented by Ronald Rivest in 1991. The idea behind the MD5 hash algorithm is to take up a random number or data such as text or binary as an input and generating a fixed size "hash value" as the output. The input data can be of any size or length, but the output "hash value size" is always fixed. Like as in the following Figure 1 MD5 Hash Generator Process, given below explains the MD5 Hash generator mechanism.

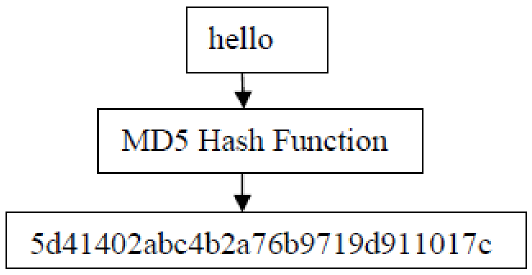

Figure 1. MD5 Hash Generator Process.

As you can see from the above example, the input you provide to the file such as in this case hello the mD5 hash generator output a fixed size (32-digit hex). Using the MD5 hash value [9] of the input enter helps in generating the unique and non-repeatable hash which is difficult to break and helps in preventing the password guessing mechanism.

Advantages of MD5 Hashing: -

• It has an advantage over storing password with unique hash value.

• Each and every string, characters of alphanumeric characters have unique hash value.

Disadvantages of MD5 Hashing: -

• Collision vulnerabilities can be defining as when two hash values can be constructed from the same hash file, which can act as a template that contains the same hash value generated file due to which the unique property of the MD5 hash which is unique hash value has been affected.

• Security of the MD5 Hash value has been compromise using the digital certificates and documented file that is containing the hash files.

2.3. Cain and Abel [15]

Cain and Abel is a password recovery tool for Microsoft Operating System. It allows easy recovery of various kinds of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute force, Cryptanalysis attack, recording VOIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analysing routing protocols. The program does not exploit any software vulnerabilities or bugs that could not be fixing with little efforts. Cain and Abel wordlist used instead of the dictionary that comprise of the meaningful words. Cain and Abel improves the dictionary attack success: the first method is to use a larger dictionary or more dictionaries (technical dictionaries and foreign language dictionaries will increase the overall chance of discovering the correct password), the second method is to perform the string manipulation. For example, the dictionary may have the word "password" in it. Common string manipulation techniques will try the word backwards (drowssap), appending the numbers to the end of the string like password123 or using the password string with different capitalization like Password, pAssWord etc.

Cain's Dictionary Password Cracker can be configured to use a list of dictionary files and it also offers the possibility to apply a number of variants for each word:

• As Is: -> the entered password checked as it is written in the dictionary file, such as (hello->hello).

• Reverse -> the password reverse form is tried such as (hello->olleh).

• Double -> twice times the password tried, such as (hello->hellohello).

• Lowercase -> the password lowercase form tried, such as (Hello -> hello).

• Uppercase -> the password uppercase form tried, such as (hello -> HELLO).

• Two numbers Hybrid-Brute: a maximum of two digits appended after each word, such as (hello-> hello10, hello11………. hello20).

In Cain and Abel, the password cracked using NTLM hashes only. Figure 2. Dictionary Attack demonstration using Cain and Abel shows the password cracking using NTLM hashes only.

Advantages of Cain and Abel:

• Supports all types of hashing to generate the unique hash value of the entered text, number, alphanumeric character or any string such as MD5, SHA-1, SHA-3 etc.

• Decoding of the scramble password is possible.

• Ability to crack- LM and NTLM hash, MD5 hash, SHA-1, SHA-2 hash etc.

Disadvantages of Cain and Abel:

• In the system having Avast set up detects Cain and Abel as malware.

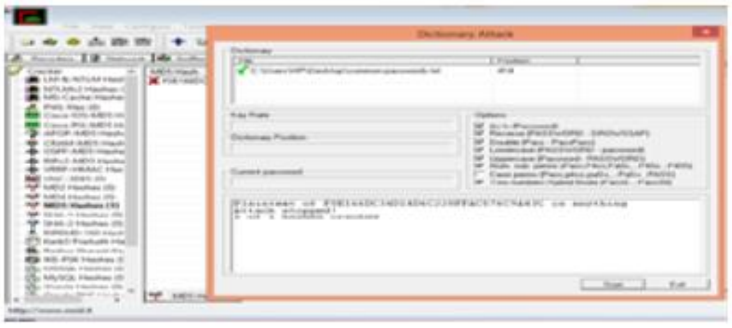

Figure 2. Dictionary Attack demonstration using Cain and Abel.

3. Proposed Approach for Dictionary Attack

Dictionary attack defined as "a technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or paraphrase by trying hundreds or sometimes millions of likely possibilities, such as words in dictionary." A method used to break security systems, specifically password – based security systems, in which the attacker systematically tests all possible strings beginning with words that have a higher possibility of being used such as names and places. The word "dictionary" refers to the attacker exhausting all of the words in a dictionary in an attempt to discover the string. Dictionary attack is faster than the brute force attack. Unlike checking all possibilities using brute force attack, the dictionary attack tries to match the entered string with most occurring words or words of daily usage. Many users generally chose passwords by using the words related to the name, birth of date, address, and famous actor/actress name. The practice of choosing the good passwords by the users depends on the password composition policies makes password harder to guess and more secure from the attackers. Although the dictionary attack is much faster than the brute force attack, it has some limitations too that is it remains a probability that password to be cracked may not be present in the dictionary itself. We know that as long as password remain human – memorable, they are vulnerable to "smart – force attack" even when the space of potential passwords is very large. The case of dictionary attack when the attacker uses a dictionary comprises of the commonly meaningful words in daily usage like common punctuations, meaningful words etc., may be a target by the suspect. The attacker then applies the colander rules which states that "Capitalizing the first digit of the string then adding the three digits at the end, changing the letter a to @ etc" to further match the target password. The important thing behind this concept is that the success of dictionary attack not depends only on the type of the input dictionary selected but also on the word colander rules applied. Attacker first tries with the little input dictionary words if the password not cracked then use the large input dictionary word to crack that password. Therefore, it is advisable to use the password composition policies in making the password strong and resistible from dictionary attack. In order to create the strong password, password composition policies often enforce several complex rules which is intended to force users into creating strong passwords. In these rules, users required to include the special characters, numbers, lower alpha number strings and upper-case alpha strings, have to enter a password of minimum length, and avoid words found in a dictionary with the colander rules. For example, consider the password 'emmer123!', for satisfying the requirements of the colander rules have to append the numeric characters '123' with the characters string "emmer" which is chosen by the user as his password and a special character "!". In, this way the password security enhanced for reducing the password cracking possibility. Hence, the habit of choosing the passwords for protecting the system and user accounts depends on the various online practices studied by many researchers [17]. Even though adding the three digits at the end of the character string increases the probability of guessing the password, if the dictionary word lies in the dictionary comprises of 800,000 words would result in 80,000,000 guesses. As Richard Smith paradoxically notes [18], choosing strong password practices imply that there is very less possibility of recollecting the forgotten password" which in turn causes the construction of weak password even after have idea of password composition policies and colander rules. In a 2007 Florencio and Herley [19] study, conducted the analyses of the password datasets about 500,000 users. Resulting in the interested insights of user habits of password selection, quantifies password strength with a simple "bit strength" measure based on their length and on the use of the uppercase, numeric, and non – alphanumeric characters. It is impossible to conclude the strength of advanced password cracking techniques against the research conducted by many researchers [17] [18] [19]. The password pattern analysis done using the linear searching methodology wherein password guessing done in linear manner. In addition, the study of run-time, complexity and password guessing time analysis also done. The string entered from the existing password datasets [20] [21] [22], if the string is in the given file, then the buffer_count is incremented. Then again, password checking has done in a linear manner from the starting alphabet. The methodology mapped into flow chart explained in the Figure 3 Flow Chart of Proposed Methodology given below.

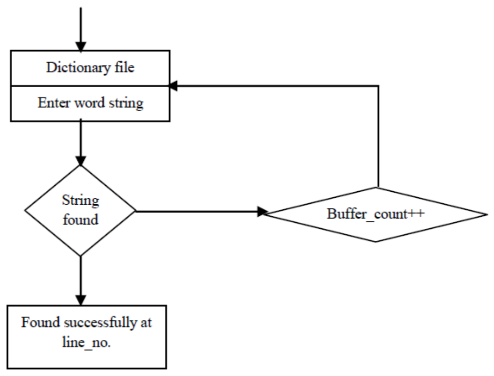

Figure 3. Flow Chart of Proposed Methodology.

The above-proposed approach of dictionary attack implemented using the Java Net Beans IDE. It supports the inbuilt functions and library with the help of which the functions create helps in finding the possible minimum amount of password guessing time and determining its run time complexity using linear searching methodology.

Specification of code: The specifications of code are given below-

• The code executes in the platform of NETBEANS IDE 7.1. Reason of choosing the Net Beans IDE 7.1 is that it gives the run time of the execution of the code whereas the other like Eclipse does not give the run time of the execution of the code.

• Most importantly, Net Beans IDE does not support the execution of the program without the presence of the main due to which the chances of ambiguity are less and successful result is given.

Reasons of choosing the JAVA programming language over VB, C# etc. are follows: -

• Forward compatibility unlike something like VB undergoes several changes in syntax and library from version to versions, JAVA syntax is same 99% and libraries getting add up more and more.

• Tool availability as Eclipse and Net beans are easily free available tools online like web server tools Tomcat, Glassfish etc. while for working in the VB net the web server tools are not easily compatible.

The existing dictionary dataset [20] [21] [22] is of varying size and comprises of the meaningful words. The summarized information of the existing password dataset explained in the given Table 1 Summarized information of Existing datasets.

Table 1. Summarized information of existing datasets.

Dataset	Size(kb)	Dictionary-name	File-type
tt	82	Online internet dictionary	txt
web	2,658	Dictionary wordlist comprises of meaningful words	doc
common-dictionary	6	Dictionary wordlist comprises of the common passwords	txt

3.1. Test Case Generation

The password complexity study based on considering two factors that is the position of the word occurrence and time for cracking the password. On the basis for studying the above two characteristics for studying the dictionary attack the test cases are generated using the knowledge of software testing.

Preparing the test data is the core part of the "project test environment setup". Tester cannot pass the bug responsibility saying that complete data not available for testing. For this, tester needs to create his/her own test data additional to the existing standard production data. Your test data should be ideal in terms of cost and time. The test data categories in which the test case is generated are: Functional testing, Structural testing, Performance testing and many more. Whereas, here only considering the Functional testing test case scenario. In addition, the test data designed by considering the following categories: -

• **No data:** Running the test cases on blank or default data and checking that proper error will be generate or not.

• **Valid data set:** Creating it and checking that if the valid test cases give the result or not in the output screen.

• **Invalid data set:** Creating it the invalid test cases and validating whether "the invalid message appears on the screen or not" by generating the test data whose values are out of range.

3.2. Functional Testing

Functional testing comes under the category of the validation in which the execution of the program is an essential thing in which both valid and invalid inputs for observing the behaviour of the program. The functional testing can be used at all levels of software testing like unit, integrations, system and acceptance testing which helps tester to design efficient and effective test cases for determining the faults in the software. The categories, which come under the functional testing are: -

3.2.1. Boundary Value Analysis

It is a simple technique in which we concentrate on the input values and design test cases with input values as.

a. Minimum value (a)

b. Just above minimum value (b)

c. Maximum value (z)

d. Just before maximum value (y)

e. Nominal value (m)

The test cases generated by the combinations of the boundary value. The test cases generated for studying the time complexity, password data variations, occurrence of the word and position of the dictionary word in the database. The boundary values generated are a, b, p, y and z and test cases generated as the combinations of the boundary value comprises of the dictionary word (meaningful word) are given in the following table. The meaningful words considered based on the online survey [15] [23] [24] of the most commonly used words by the users in protecting their accounts from the hackers. The dictionary file which is used for cracking the password string which is chosen from the online survey [15] [23] [24] of the most popular and hack able passwords are txt file and it is tt.txt [20], web.doc [21] and common-passwords.txt [22]. In given Table 2. Run time of Existing Password Dataset using Proposed Approach for Dictionary Attack is below.

Table 2. Run time of Existing Password Dataset using Proposed Approach for Dictionary Attack.

S. No.	Password String	tt.txt	web2.txt	common-password.txt
1.	admin	2ms	141ms	2ms
2.	alphabet	3ms	373ms	2ms
3.	anything	3.2ms	696ms	3ms
4.	batch	3.5ms	1000ms	3.1ms
5.	beloved	3.7ms	1118ms	4ms
6.	bridge	4ms	1279ms	6ms
7.	password	15ms	7000ms	14ms
8.	police	90ms	7689ms	53ms
9.	python	150ms	8316ms	55ms
10.	yahoo	412ms	12000ms	65ms
11.	yellow	442ms	12495ms	69ms
12.	ummy	437ms	12555ms	72ms
13.	zang	520ms	12002ms	73ms

14.	zebra	534ms	12437ms	74ms
15.	zoom	537ms	12471ms	75ms

4. Implementation Results

The stimulation of the proposed dictionary attack methodology done using JAVA Net Beans IDE. The pattern analysis of the entered string is study by graph plotting. In addition, the password guessing pattern analysis done on the occurrence of the words in alphabetical manner. Such as if the entered word initial starting from a then proceeding in a chronological way till *z*. The linear searching methodology used in the searching of the entered string within the existing dataset. Also, in addition for studying the password guessing pattern, the linear searching methodology look for the entered string using pattern matching methodology (that is brute force attack). For analysis, the string chosen from the existing datasets for studying the password pattern analysis. The string chosen in a probabilistic way for studying and analysing the string - guessing pattern of words.

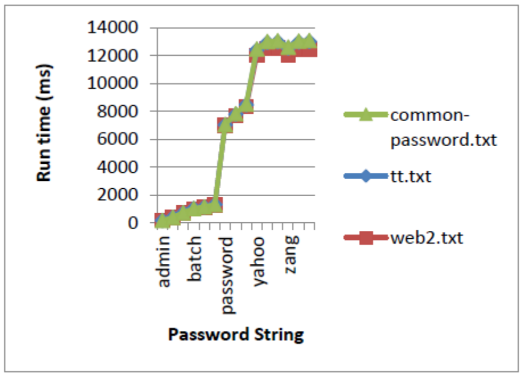

Figure 4. Run time complexity vs. Password String.

5. Observations and Future Work

The password cracking time also depends on the occurrence of the password at the position in the wordlist: -

• If the password is at the first position in the wordlist and the password is starting with the letters like a then time complexity is best.

• If the password is at the middle position in the wordlist and starting with the letters like *l*, *m* and *n* then time complexity is average.

• If the password is at the last position with the starting letter *x*, *z* then the time complexity is worst.

In future work, instead of using the linear methodology in dictionary attack we can apply binary methodology for improving the run time complexity and searching methodology. The binary methodology can be use in conjunction with the MD5 Hashing mechanism to make the password more secure.

References

[1] Predarg tasevski, Password attacks and generations strategies, Taru University.

[2] J. A. Cazier and D. B. Medlin, Password security: An empirical investigation into e-commerce passwords and their crack times, Information Security Journal: A Global Perspective 15(6) (2006), 45-55.

[3] L. St. Clair, L. Johansen, W. Enck, M. Pirretti, P. Traynor, P. McDaniel and T. Jaeger, Password exhaustion: Predicting the end of password usefulness, in Proc. ICISS, 2006.

[4] 15-A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication, Mudassar Raza, Muhammad Iqbal, Muhammad Sharif and WaqasHaider.

[5] R. V. Yampolskiy, Analyzing user password selection behaviour for reduction of password space, Proc. of the IEEE International Carnahan Conferences on Security Technology (2006), 109-115.

[6] Russel Dan Vines, Ehtical hacking tools and techniques: Password cracking.searchsecuritychannel.techtarget.com, 2007.

[7] Brute force algorithms. Available FTP:

http://faculty.simpson.edu/lydia.sinapova/www/cmsc250/LN250_Levitin/L05.-BruteForce.htm.

[8] OphCrack, Available FTP: http://ophcrack.sourceforge.net/.

[9] Hashing Algorithms, Available HTTP https://crackstation.net/.

[10] D. Florencio and C. Herley, A large – scale study of web password habits, in WWW '0: Proceedings of the 16th international conference on World Wide Web. ACM (2007), 657-666.

[11] J. Yan, A. Blackwell, R. Anderson and A. Grant, Password memorability and security: Empirical result, IEEE Security and Privacy Magazine 2(5) (2004), 25-31.

[12] Subbarao Gogulamudi, Mahanthi Pasumarthi, PGRP: Online Brute Force Attacking, in

International Journal of Research in IT, Management and Engineering, ISSN 2249-1619.

[13] Captcha image definition, Available FTP:

http://searchsecurity.techtarget.com/definition/CAPTCHA.

[14] Online MD5 Hash generator tool, Available HTTP http://onlinemd5.com/.

[15] A list of most popular and hackable passwords on internet, 2015, [Online Document] [cited April 2015] Available HTTP,

http://www.telegraph.co.uk/technology/internetsecurity/10303159/Most-common-and-hackable-passwords-on-the-internet.html.

[16] J. Yan, A. Blackwell, R. Anderson and A. Grant, Password memorability and security: Empirical result, IEEE Security and Privacy Magazine 2(5) (2004), 25-31.

[17] R. A. Grimes, MySpace password exploit: Crunching the numbers (and letters)," InfoWorld online article, November 2006, Available FTP.

http://www.infoworld.com/article/06/11/17/47OPsecadvise_1.html.

[18] R. E. Smith, The Strong Password Dilemma, Addison – Wesley, 2002, ch.6.

[19] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, Password cracking using probabilistic context free grammars, in IEEE Symposium on Security and Privacy, IEEE, May (2009) 391-405.

[20] Common download dictionary word web, 2011, [Online Document] [cited 2004 March 07] Available HTTPhttp://wordweb.info/free/.

[21] A list of popular common password wordlists, 2014, [Online Document] [cited 2004 March 07] Available HTTPhttp://www.outpost9.com/files/WordLists.html.

[22] A list of popular family-names wordlists, 2014, [Online Document] [cited 2004 March 07] Available HTTPhttp://www.outpost9.com/files/WordLists.html.

[23] A list of Top 25 common hackable passwords, 2012, [Online Document] [cited October 2012] Available HTTP http://www.zdnet.com/article/top-25-common-hackable-passwordsqwerty-ninja-jesus/.

[24] A list of Top 20 most popular passwords stolen from ADOBE, 2013, [Online Document] [cited November 2013] Available HTTP http://mashable.com/2013/11/05/20-most-popular-passwords-adobe/