DOI 10.29042/2018-3787-3791

Helix Vol. 8(5): 3787- 3791

Design of Password Guessing Prevention Protocol for Levelled-Security System

[1]R. L. Bagad, [2]A. N. Magar, [3]S. N. Mali, [4]R. R. Rathod

[1, 2, 3] Undergraduate Student, Department of Information Technology, Walchand College of Engineering, Sangli, India

[4]Assistant Professor, Department of Information Technology, Walchand College of Engineering, Sangli, India

Email: maliswapnil982@gmail.com

Received: 09th July 2018, Accepted: 14th August 2018, Published: 31st August 2018

Abstract

Most applications use the password for authentication of the legitimate user. These applications maintain a database of username and corresponding password. In databases, passwords are stored in the form of hash values which are irreversible. A potential hacker may use brute force attack or dictionary attack for guessing the password. In brute force attack; hacker tries all possible combinations of passwords to gain unauthorized access to user's account. In a dictionary attack, the hacker uses dictionary file containing possible passwords and tries every password from that file. It is noted that, even though password with certain patterns is accepted as strong by existing systems, they are vulnerable to dictionary attack. The proposed system allows the user to choose a password which is not present in the dictionary. Also, all possible alterations of passwords are matched against the supplied dictionary. Password containing personal information is not accepted by the system. The present study proposes a security method based on login attempts and respective security levels for the password. The concept of security levels is introduced which aims at increasing the strength of the password on detection of malicious activity such as invalid login attempts, login attempts performed for the same user account from two different locations in little time difference which are far away from each other, etc. On detection of any suspicious activity, the security level of the password gets upgraded to the higher level which increases length and strength of the password by making it more complex and hard to crack.

Keywords: Brute Force Attack, Dictionary Attack, Automated Turing Test, Pinkas and Sander Protocol, Denial of Service, Distributed Denial of Service, Internet Protocol

Introduction

Passwords are the most common way to provide authentication to the user. However, several threats have proven that many people's passwords are vulnerable to being cracked. Thus the security of user's information is threatened all the time [1]. Password is the weakest link in authentication. Due to human memory limitation, it is not possible to have a truly random password [2], [3], [4], [5], [6]. This might be the reason behind the choosing weak passwords by humans which are easily cracked by dictionary or brute force attack.

There are two kinds of attack experienced by password based systems and those are online and offline attacks. During offline attack attacker monitor network traffic and gets insights of the transmitted data over network. Attacker uses those insights to crack passwords [7]. Online attack again have two subtypes first one is Brute force attack and second is dictionary attack [8]. In brute force attack, attacker employs a technique which generates all possible combinations to crack the password. Though it look like impossible to crack password with this technique, attacker can use large network of botnets to speed up this attack. Dictionary attack can be employed by making dictionary of relevant words which are associated user's personal life, user's behaviour or user's habits. An attacker tries every word from this dictionary to gain access of user's account. Dictionary attacks are possible due to the tendency of humans to have simple and memorable passwords [9], [10]. The studies show that human creates passwords by taking help of their personal information [11]. Offline attacks are easy to curb and can be resist by using symmetric key cryptography. However, the proposed solution gives methods to curb brute force and dictionary attacks. These methods are discussed in section 3.

The proposed system also resist Distributed Denial of Service (DDOS) attack by applying the constraint on login from different locations within the appropriate difference between timestamp values.

Existing System

Password-based systems are susceptible to online attacks. Above mentioned techniques are very effective to gain illegal access of user's account and hence create a major problems in setting up of any

password based systems. The measures employed to curb these attacks are as follows:

A. Account Locking

This scheme prevents attacks by setting threshold on number of unsuccessful login attempts. Eventually this technique limits the number of wrong login attempts. One of the major drawback of this scheme is it is vulnerable to Denial of Service (DOS) and DDOS attacks which are carried out by large number of botnets which in turn does not let legitimate user from logging into their systems [12]. Sometimes the legitimate users may mistakenly lock their own accounts. Though this scheme is having above mentioned drawbacks, it is still commonly used in many password based systems [13].

B. Delayed Response

In this scheme, a delayed response is provided by the server to the user request. Thus, it becomes difficult for attacker to check many passwords in reasonable time. This scheme is less effective in a network environment because the attacker can perform DOS or DDOS attack very efficiently [8]. DOS or DDOS attacks are prevented by use of captchas. Captchas are difficult to crack for machine bots. However, the study showed that captchas are often difficult for humans also [14].

C. Pinkas and Sander Protocol

In Pinkas and Sander (PS) protocol, first answering an Automated Turing Test (ATT) challenge is required before entering the {username, password} pair. If user fails to answer the ATT correctly, system prevents the user from proceeding further. User needs to pass an ATT challenge for each password guessing attempt, in order to gain information about the correctness of the guess. While this protocol is effective against online dictionary attacks, legitimate users must also pass an ATT challenge for every login attempt [15]. Therefore, this protocol affects user convenience and requires the login server to generate an ATT challenge for every login attempt [16].

D. Van and Stubblebine Protocol

Van and Stubblebine Protocol (VS) is the modified version of the PS protocol. This protocol traces the number of failed login attempts for a particular username. If the number of failed login attempts exceeds some threshold value, the user is asked to challenge ATT for every next attempt [17]. This protocol offers opportunities for user-friendliness (fewer ATTs to legitimate users), improved security and greater flexibility (e.g., allowing protocol parameter customization for particular situations and users). This protocol is susceptible to minor variations of well-known middle-person attacks [18].

Implementation

The proposed system is divided into two parts. The first part provides measures to prevent dictionary attack and the second part is to defend brute force attack. The proposed system doesn't allow the user to choose a password which includes dictionary words and personal information in it. The system converts the supplied password into the exhaustive list of possible alterations, such as changing a to @, s to 5 or $, etc. which is broadly used by end user to meet the complexity rules. This list is again compared with common password based researches done on latest password database breaches. Additionally, it validates the length and use of multiple charsets in the password. So the minimum possible password combinations provided by the system is

$$(26 + 26 + 10 + 10)^8 = 72^8 = 7.2222041363 \times 10^{14}$$

The present study proposes a security method based on login attempts and respective security levels for the password. For preventing brute force attack, security levels are defined. As the security level increases, the strength of the password is increased. Whenever a suspicious activity is detected, the password is upgraded to the higher security level making it more complex than previous one. Password for specific security level is dependent on the user-specific key provided at the time of registration, username and original password. For identifying the legitimate user, different methods are used such as invalid login attempts, geolocation-based approach, etc.

A. Goals

- To develop a system which will make brute force and dictionary attacks ineffective.
- The proposed system should not have any major impact on user convenience.

B. Methodology

Proposed system is divided into following modules:

1) **New user registration and login -** This is the initial phase of the proposed system. During registration and login, the following steps take place:

i. During registration, the user is asked to fill in details like username, password, and other related information.

ii. Password entered by the user is checked against the following constraints to mitigate the possibility of password cracking during dictionary attack.

a. Ensure that the length of the password is at least 8 characters.

b. Ensure that it contains 4 charsets i.e. uppercase, lowercase, numeric and special characters.

c. The password should not contain personal information in it.
d. The password must not match with the supplied dictionary.
e. The system converts the supplied password into all possible alterations, such as changing a to @, s to 5 or $, etc. which is one of the techniques broadly used by end user to meet the complexity the constraints. This list is again compared with passwords from the dictionary and if it is found in the dictionary, the user is suggested to choose a different password.

a	@
3	E,e
8	B
4	A
1	i,l
o	0
7	T,t
$	s,5
6	b,d

Table 1. Sample Conversion Table

iii. The user then can log in with username and password.

2) **Identifying the user -** This is the second phase of the proposed system. In this phase, the user is identified as valid or invalid. The user is valid if the password entered by the user is correct for the corresponding username. The user is invalid if one of the following conditions is satisfied -

i. Invalid attempts

If user makes invalid attempts crossing the certain threshold, then the password is upgraded to next security level.

ii. Location based number of invalid attempts

In small time difference, if login for certain account is performed from two different locations which are far away from each other and it is not possible to cover that the distance in given time difference, then it is considered as suspicious and login attempts is limited to 1 attempt.

iii. Blacklisted IP

If user crosses invalid login attempts limit, then that Internet Protocol (IP) address is blocked for the day. Thus, that user can't login using that machine unless IP is released from the blacklist.

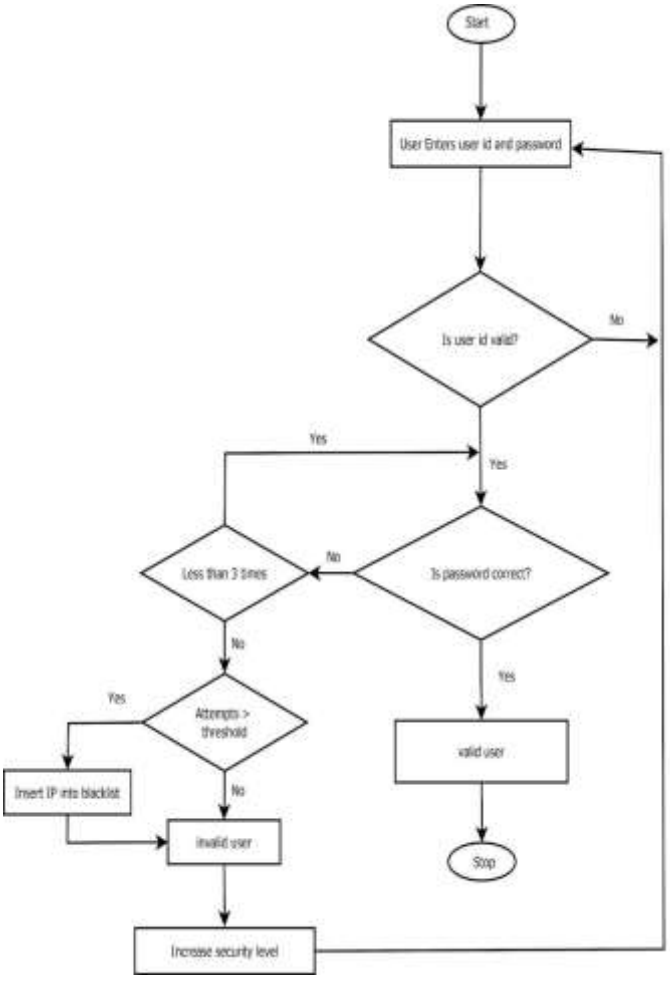

Fig. 1 Proposed Flow Diagram for Identifying the User

3) **Increase security level** - The proposed system is having numerous security levels. Each security level differs from another in terms of strength of the password required for logging into the system. The 4 digit key provided to the user at the time of registration is given as the parameter to the algorithm which is used to calculate next level password. This key provides an extra layer of security. Thus, only knowing the password is not sufficient for breaking the password. For the first time, the user needs to enter the password which is used at the time of registration. If the user makes 3 invalid attempts then the system takes the user to next security level i.e. security level 1 and so on for every 3 invalid attempts. Higher the security level, the more is the complexity of the password. This process of upgrading security levels is

continued till user enters the correct password for the corresponding security level. Security levels are limited to 10. After reaching last security level, One Time Password (OTP) needs to be entered along with the password. At the end of the day, the password is reset to security level 1. Considering above scenario, if the user is at security level 2 then password he needs to enter is calculated on the basis of 4 parameters.

i. Username
ii. Original Password
iii. User Specific key (sent through email)
iv. Security level number

The user can enter all these fields correctly into the level n password calculator provided and can get a valid password required for particular security level.

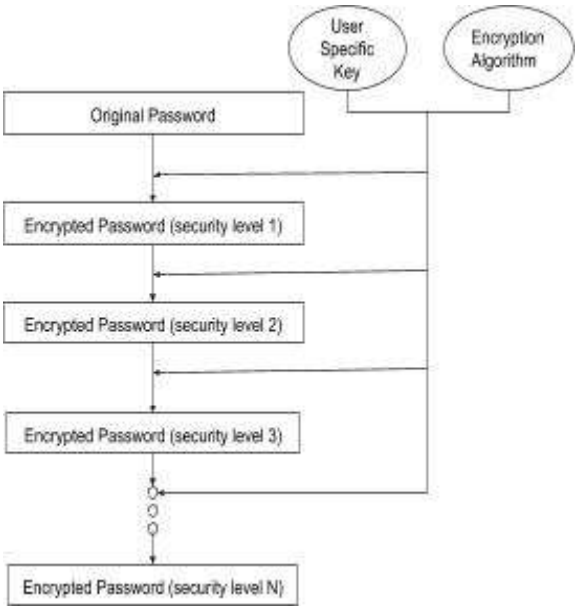

Fig. 2 Security Levels of Proposed System

Experimental Result

MySQL database is used for implementation of the proposed system. Initially at the time of registration record will be created for a user. Details of user like username, password, and other related information are collected during registration.

Table 2 shows the time taken to crack the password for various security levels. Calculation of the time taken to crack the password is done using the project from dashlane i.e. howsecureismypassword.net which lets user know what extent it would take somebody to break their secret password. It likewise checks against the best 10,000 most regular passwords and additionally various different checks, (for example, rehashed strings, phone numbers, and words took after by numbers). [19].

Password Level	Password	Time Taken To Crack The Password
1	D!ng0@83	6 Hours
2	#D!ng06@83	9 Years
3	z#D!nOg06@83	34 Thousand Years
4	#z#D!n6Og06@83	204 Million Years

Table 2.Time Taken to Crack Password for Various Levels

The proposed system considers both location and timestamp to check the validity of user. IP to location REST (Representational State Transfer) API is used to get the location from IP. If login attempts from two different locations are detected for the same user account in little time difference, then such activity is considered as suspicious and thus the login attempts is limited to 1.

VPN (Virtual Private Network) is used for experiment purpose. As given in Table 2, the location is Bangalore, India and login attempt is made at given timestamp for given user account.

IP	User ID	Location	Timestamp (seconds from 1 January 1970)
159.89.163.164	xxxx@xyz.com	Bangalore, India	1520490991

Table 3. Sample Record for User from One Location

After 4 minutes, IP address selected was from Sydney, Australia. When a login attempt performed from this location, security level got upgraded and login attempt limit was set to 1. Since it is impossible to be present at these locations in 4 minutes of the time difference, it is considered as suspicious activity.

IP	User ID	Location	Timestamp (seconds from 1 January 1970)
108.61.185.30	xxxx@xyz.com	Sydney, Australia	1520491218

Table 4. Sample Record for Same User from Different Location

Conclusion

Authentication systems face critical problems due to weak and easily guessable passwords. Since password remains same unless and until the user changes it, so the hacker can easily crack it by brute force method using available tools. The proposed system introduces the concept of security level which corresponds to increasing the strength of the password automatically

when malicious activity is detected. Currently, available schemes use two-factor authentication which requires having secondary devices like a mobile device, but the proposed system does not require to carry any secondary devices with the user. The proposed system doesn't allow users personal information to reside in the password. Hence it is difficult for an attacker to crack the password through a dictionary attack. Also proposed system changes password to higher security level upon detection of brute force attack which does not require any account locking. Therefore attacker cannot employ DOS or DDOS attack. The proposed system does not require to generate ATTs, hence there is no overhead of generating ATTs on the server side.

References

[1] Colombini, Clara Maria, et al. "Cyber Threats Monitoring: Experimental Analysis of Malware Behavior in Cyberspace." *Security Engineering and Intelligence Informatics Lecture Notes in Computer Science*, 2013, pp. 236–252., doi:10.1007/978-3-642-40588-4_17

[2] Bonneau, Joseph. "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords." *2012 IEEE Symposium on Security and Privacy*,2012.

[3] Malone, David, and Kevin Maher. "Investigating the distribution of password choices." *Proceedings of the 21st international conference on World Wide Web - WWW '12*, 2012.

[4] Narayanan, Arvind, and Vitaly Shmatikov. "Fast dictionary attacks on passwords using time-space tradeoff." *Proceedings of the 12th ACM conference on Computer and communications security - CCS '05*,2005.

[5] Veras, Rafael, et al. "Visualizing semantics in passwords." *Proceedings of the Ninth International Symposium on Visualization for Cyber Security - VizSec '12*,2012.

[6] Yan, J., et al. "Password memorability and security: empirical results." *IEEE Security & Privacy Magazine*,vol. 2, no. 5, 2004, pp. 25-31.

[7] Goyal, V., et al. "CompChall: Addressing Password Guessing Attacks." *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, 2005,doi:10.1109/itcc.2005.107.

[8] Kirushnaamoni, R. "Defenses to Curb Online Password Guessing Attacks." *2013 International Conference on Information Communication and Embedded Systems (ICICES)*,2013, doi:10.1109/icices.2013.6508230

[9] Hellman, M. "A cryptanalytic time-Memory trade-Off." *IEEE Transactions on Information Theory*,vol. 26, no. 4, 1980, pp. 401–406., doi:10.1109/tit.1980.1056220.

[10] Morris, Robert, and K. Thompson. *Password Security: a Case History*.1978.

[11] Li, Yue, et al. "Personal Information in Passwords and Its Security Implications." *IEEE Transactions on Information Forensics and Security*,vol. 12, no. 10, 2017, pp. 2320-2333.

[12] S. T. Pirjade, P. K. Deshmukh, "Defend Against Online Password Guessing Attacks" *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.4, pp. 106-111, September 2014.

[13] Adams, Carlisle, et al. "Lightweight Protection against Brute Force Login Attacks on Web Applications." *2010 Eighth International Conference on Privacy, Security and Trust*, 2010, doi:10.1109/pst.2010.5593241.

[14] Bursztein, Elie, et al. "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation." *2010 IEEE Symposium on Security and Privacy*, 2010, doi:10.1109/sp.2010.31.

[15] Mrs. Manjula. G, K. S. Shashikala. "Security For Password Based Systems Using PGR Protocol." *International Journal of Engineering Research & Technology (IJERT),* 3 May. 2013.

[16] Pinkas, Benny, and Tomas Sander. "Securing Passwords against Dictionary Attacks." *Proceedings of the 9th ACM Conference on Computer and Communications Security - CCS '02*,2002, doi:10.1145/586110.586133.

[17] Geethika, I. Naga, and T. Prem Jacob. "A Efficient Approach for Password Attacks." *International Journal of Engineering Trends and Technology*, vol. 9, no. 2, 2014, pp. 78–80., doi:10.14445/22315381/ijett-v9p216

[18] Oorschot, P. C., & Stubblebine, S. (2006). On countering online dictionary attacks with login histories and humans-in-the-loop. *ACM Transactions on Information and System Security*, *9*(3), 235-258. doi:10.1145/1178618.1178619

[19] Collider, Small Hadron. "How Secure Is My Password?" *How Secure Is My Password?* ,howsecureismypassword.net/. Web Source.(last accessed on February 23, 2018).

