IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 20-23
www.iosrjournals.org

Strong password generator scheme Using image fusion

[1,2]Miss. Snehal A. Mahajan, M.Tech.- C.S.Engineering
G. H. Raisoni Institute of Engineering and Technology, Nagpur.

ABSTRACT: *Password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. The strength of a password is a function of length, complexity, and unpredictability. Using strong Passwords lowers overall risk of a security breach, but strong passwords do not replace the Need for other effective security controls. So, two or more images are used for this password. We perform Image Fusion Algorithm that combines all selected images into a single Image. Significance of image fusion algorithm is only to avoid the dependency of generated password on single image. Image fusion modifies input image pixels & at the result end, we find two images are mixed up. Once the images are fused, we will apply two shares Visual Cryptography that Encrypt the image & converts it into unreadable format. In proposed work; we will try to Mix up all the generated sections with permutations so that every time & in every round an Unique Password will generate. Once the password is generated we pass this password from strong password definition filter if it pass through it than we will use it for the Authentications else we forward it to proposed password generator algorithm again. This paper gives an idea about password generation by using image fusion technique.*

I. INTRODUCTION

Password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. In its usual form, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. The strength of a password is a function of length, complexity, and unpredictability. Using strong passwords lowers overall risk of a security breach, but strong passwords do not replace the need for other effective security controls. The effectiveness of a password of a given strength is strongly determined by the design and implementation of the authentication system software, particularly how frequently password guesses can be tested by an attacker and how securely information on user passwords is stored and transmitted. Risks are also posed by several means of breaching computer security which are unrelated to password strength.

A strong password:

- has at least 15 characters;
- has uppercase letters;
- has lowercase letters;
- has numbers;
- Has symbols, such as `! " ? $? % ^ & * ( ) _ - + = { [}] : ; @ ' ~ # | \ < , > . ? /
- is not like your previous passwords;
- is not your name;
- is not your login;
- is not your friend's name;
- is not your family member's name;
- is not a dictionary word;
- is not a common name;
- is not a keyboard pattern, such as qwerty, asdfghjkl, or 12345678.

Password Strength:

There are two factors to consider in determining password strength: the average number of guesses the attacker must test to find the correct password and the ease with which an attacker can check the validity of each guessed password. The first factor is determined by how long the password is how large a set of characters or symbols it is drawn from and whether the password is created randomly or by a more predictable process. Users of password-protected resources often have control of this factor. The second factor is determined by how the password is stored and used. This factor is determined by the design of the password system and beyond control of the user. The rate at which an attacker can submit guessed passwords to the system is a key factor in determining system security. Some systems impose a time-out of several seconds after a small number (e.g., three) of failed password entry attempts. In the absence of other vulnerabilities, such systems can be effectively secure with relatively simple passwords. However the system must store information about the user passwords in some form and if that information is stolen, say by breaching system security, the user passwords can be at risk.

Password Creation:

Passwords are created either automatically (using randomizing equipment) or by a human. While the strength of randomly chosen passwords against a brute force attack can be calculated with precision, determining the strength of human-generated passwords is challenging and the latter case is more common. Typically, humans are to choose a password, sometimes guided by suggestions or restricted by a set of rules, when creating a new account for a computer system or Internet Web site. Only rough estimates of strength are possible, since humans tend to follow patterns in such tasks, and those patterns can usually assist an attacker.[2] In addition, lists of commonly chosen passwords are widely available for use by password guessing programs. Such lists include the numerous online dictionaries for various human languages, along with other common passwords. All items in such lists are considered weak, as are passwords that are simple modifications of them. For some decades, investigations of passwords on multi-user computer systems have shown that 40% or more are readily guessed using only computer programs, and more can be found when information about a particular user is taken into account during the attack.

Password Guess Validation:

Systems that use passwords for authentication must have some way to check any password entered to gain access. If the valid passwords are simply stored in a system file or database, an attacker who gains sufficient access to the system will obtain all user passwords, giving the attacker access to all accounts on the attacked system, and possibly other systems where users employ the same or similar passwords. One way to reduce this risk is to store only a cryptographic hash of each password instead of the password itself. Standard cryptographic hashes, such as the Secure Hash Algorithm series, are very hard to reverse, so an attacker who gets hold of the hash value cannot directly recover the password. However, knowledge of the hash value lets the attacker quickly test guesses offline. Password cracking programs are widely available that will test large number of trial passwords against a purloined cryptographic hash.

Improvements in computing technology keep increasing the rate at which guessed passwords can be tested. For example, in 2010, the Georgia Tech Research Institute developed a method of using GPGPU to crack passwords much faster. Another company Elcomsoft invented and started using common graphic cards for quicker password recovery in August 2007 and soon filed a corresponding patent in the US (Use of graphics processors as parallel math co-processors for password recovery, 7,929,707). As of 2011, commercial products are available that claim the ability to test up to 2,800,000,000 passwords per second on a standard desktop computer using a high-end graphics processor.[4] Such a device can crack a 10 letter single-case password in one day. Note that

the work can be distributed over many computers for an additional speedup proportional to the number of available computers with comparable GPUs. Special key stretching hashes are available that take a relatively long time to compute, reducing the rate at which guessing can take place. Although it is considered best practice to use key stretching, many common systems do not.

Another situation where quick guessing is possible is when the password is used to form a cryptographic key. In such cases, an attacker can quickly check to see if a guessed password successfully decodes encrypted data. For example, one commercial product claims to test 103,000 WPA PSK passwords per second. If a password system only stores the hash of the password, an attacker can pre-compute hash values for common passwords variants and for all passwords shorter than a certain length, allowing very rapid recovery of the password once its hash is obtained. Very long lists of pre-computed password hashes can be efficiently stored using rainbow tables. This method of attack can be foiled by storing a random value, called a cryptographic salt, along with the password. The salt is combined with the password when computing the hash, so an attacker pre-computing a rainbow table would have to store for each password its hash with every possible salt value. This becomes infeasible if the salt has a big enough range, say a 32-bit number. Unfortunately, many authentication systems in common use do not employ salt and rainbow tables are available on the Internet for several such systems.

Problem Definition:

Sequence of characters letters, numbers, symbols used as a secret key for accessing a computer system or network. Passwords are used also for authentication, validation, and verification in electronic commerce.

Proposed Strong Password generator scheme will works as follows:

1. Select input Images which may be of any type like RGB, Gray and Binary etc.
2. We perform Image fusion algorithm that combines all selected images into a single Image. Significance of image fusion algorithm is only to avoid the dependency of generated password on single image. Image fusion modifies input image pixels & at the result end, we find two images are mixed up. Let us consider the following example of two image fusion(RGB Format)
3. Once the images are fused, we will apply two shares Visual Cryptography that Encrypt the image & converts it into unreadable format, result of visual cryptography becomes as follow:
4. The Cryptographic image is unreadable in format that's why an intruder will find difficulty in reading Plain image for password decryption. Cryptographic image contain a decimal pixel value either 0 or 255.
5. Crypto image is a input to our Proposed Password Generator Algorithm we choose the pixels from Crypto image based on the values of Keys, we suggest the multiple key selection to create more patterns of selection. Finally we assemble all these selected pixels into a single dimensional array which we will divide into 04 sections that is Digits, Characters, Special Symbols, & Special Character.
6. Strong password definition says that, " Password should contain Digits, Characters, Special Symbols, & Special Characters and it should not be breakable by Non of the Brilliant Intruder easily, in proposed work; we will try to Mix up all the generated sections with permutations so that every time & in every round an Unique Password will generate.
7. Once the password is generated we pass this password from strong password defination filter if it pass through it than we will use it for the Authentications else we forward it to proposed password generator algorithm again.

References:

[1] Arun Ross and Asem Othman, "IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY", VOL. 6, NO. 1, MARCH 2011.
[2] S.Pradeesh Hosea, S. Ranichandra, T.K.P.Rajagopal "International Journal of Scientific & Engineering Research Volume 2", Issue 3, March-2011 1 ISSN 2229-5518.
[3] Th. Rupachandra Singh "International Journal of Computer Applications (0975 – 8887) Volume 39– No.1", February 2012.
[4] Tranos Zuva, Oludayo O. Olugbara, SundayO. Ojo and Seleman M. Ngwira "Canadian Journal on Image Processing and Computer Vision Vol. 2", No. 3, March 2011.
[5] Subrahmanyam Murala, R. P. Maheshwari, and R. Balasubramanin, "IEEE TRANSACTIONS ON IMAGE PROCESSING", VOL. 21, NO. 5, MAY 2012
[6] "H. B. Kekre et al. "(IJCSE) International Journal on Computer Science and Engineering." ISSN : 0975-3397 368.Vol. 02, No. 02, 2010, 368-372.