International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 12, December-2014

Alpha-Numerical Random Password Generator for Safeguarding the Data Assets

A. Sandanasamy
Department of Computer Applications,
Bishop Heber College,
Trichy – 620017,
Tamilnadu.

D. Muthulakshmi
Department of Computer Applications,
Fatima College,
Madurai – 625018,
Tamilnadu.

***Abstract* - The users of computer technology and internet are increasing day by day. As the users are growing , the need for security is also felt very much. The data assets and other valuable facts are stored in the computer systems. One of the way to safe guard the data assets is to have a proper authorization method to access the data. This is achieved by user identification and password mechanism. The selection of password is important, since the entire authorization is dependent on the password. The password need to be strong enough to avoid brute force attack and other attacks. Here we discuss a method of generating random passwords, which are strong enough to combat the attacks.**

Keywords – Password; Random password;Encryption; Decryption; Security;Symmetric Key.

I. INTRODUCTION

Password is indispensable and inevitable one in today's communication process and provides security to user's data. Password is a sequence of character string used to authenticate personal identity of user and to provide or refuse the access to system resources. The password is not only denying any access to the system from unauthorized person, but also prevent users who are previously logged in from doing unauthorized process in system.

Security risk from unauthorized entry involves more than the risk to a single user via their system account[1]. User ID and password combination is the one of the simplest forms of user authentication. Password is a secret word, which is used to authorize the user to particular system or particular application. The identity of the user is tested using the password. If the passwords are not strong or easily guessable, the intruders can attack the system and attack the data assets. Guessing attacks have had major business implications, such as a 2009 incident in which a vandal guessed a Twitter executive's password and was able to leak all of the company's internal documents [2]. Because of the scenario of widespread re-use of passwords across sites [3], an emerging attack model is to compromise accounts by a guessing attack against a low-security website and attempt to re-use the credentials at critical websites [4].

The length and diversity contribute to the size of the domain set containing all possible, that increases the difficulty of brute force detection [1]. Most organizations specify a password rules that form the requirements for the composition and usage of passwords.They are minimum length, required categories such as upper and lower case, numbers, and special characters, prohibited elements such as own name, D.O.B., address, telephone number. Some governments have national authentication framework that define requirements for user authentication to government services, including requirements for passwords.

The aim of this paper is to provide an Automated Password Generator Model by specifying an algorithm to generate passwords for the protection of computer resources, which provides basic security criteria for the design, implementation, and use of passwords. The algorithm uses random numbers to select the characters that form the random pronounceable passwords. The genereted password is protected through encryption and decryption mechanism.

II. RELATED WORK

Art Conklin, Glenn Dietrich, Diane Walz [1] have discussed about conceptual model of password-based security across multiple systems connected by user activity. They have showed a light on user authentication with examples of user ID and password combination, one of the simplest forms of user authentication, smart card system where a user typically has an ID, a password, and also a time-generated passkey from thesmart card which changes every 60 seconds. They have also discussed about human cognitive ability in remembering passwords, potential risks in weak passwords. According to them, adherence to password rules does produce passwords that are more difficult to break. They also recognized the problem of remembering it. Adherence to system rules produces passwords that are more difficult to discover.

Manoj Kumar Singh [5], proposed a method to exploit the artificial neural network to develop the more secure means of authentication, which is more efficient in providing the authentication. He discussed about architecture of neural Network, learning rule, target definition and process, which will apply for authentication. He has highlighted that neural network with intrusion detection capability can handle the challenge associated with password and authentication.

Michel D. Leonhard and V.N. Venkatakrishnan [6] analyzed three random password generation algorithms namely ALPHANUM, DICEWARE, and PRONOUNCE3. They used the metrics such as security, memorability and affinity to find out, which of the three methods produces best passwords. They used six character length random password,

which contain upper-case letters, lower-case letters, and numbers. They have highlighted that random passwords have several benefits over user-chosen passwords mainly security and confidentiality. DICEWARE generator produces random lists of words. This is based on the concept of memorization and the mental connection required to remember the password. The PRONOUNCE3 produces pronounceable words in English. This aims at the speech facilities of the user's mind to assist in remembering the password.

Ayushi [7] proposed a Symmetric Key Cryptographic Algorithm for increasing the security of data. Secret key cryptography methodologies are classified as either stream ciphers or block ciphers. Stream ciphers operate on a single bit at a time, and feedback mechanism is used. Block cipher uses one block of data at a time using the same key on each block. The same plaintext block is always encrypted to the same cipher text when using the same key in a block cipher whereas the same plaintext will be encrypted to different cipher text in upstream cipher. In this paper, bit manipulation is taken for encryption and decryption. The data is converted to binary digits and reversed. The reversed digits are divided by the key and the result is converted as the encrypted data.

Kamini H. Solanki and Chandni R. Patel [8] introduced new type of symmetric key cryptographic algorithm to improve the security of data. They have also used bit manipulation for encryption and decryption. The character is converted to ASCII and then converted to binary digits. The binary digits are complemented. The key constant 10 is multiplied with the complemented binary digit and the result is converted to hexadecimal value. They have proposed this algorithm to design and implement a new algorithm to address this issue of cost effectiveness to encrypt a small amount of data.

This paper is organized as follows: section III describes the proposed work for random password creation and ciphering methods, section IV consists of experimental study and section V describes conclusion.

III. PROPOSED WORK

The proposed system defines a new method for generating random password and invents the novel cryptographic techniques to protect the password. The random password is generated and encrypted with innovative symmetric key algorithm. The proposed model provides the principles and guidelines for random password creation, password encryption and decryption. The process of encryption and decryption of password can also be adopted for various applications such as online registration, online exams etc.

A. Random Password Generator

Random password generator is to produce random password with high security. Generally, random passwords have various benefits over user-chosen password where it enhances security and confidentiality. The new methodology has been created to generate random password which consists of both upper & lower case letter and digits from 0 to 9, with fixed length. The alphanumeric is a simple algorithm that generates random password with predetermined length. The password generator algorithm selects a random character form random character list and forms the password, which is combination of numbers, lower & upper-case letters.

The total length of the password is 12. This algorithm produces the password with the combination of randomly selected lower-case character, upper-case character and numbers. The entire alphabet size is 62 [10+26+26=62], which indicates 10 digits (0 to 9), 26 upper-case letters and 26 lower-case letters. There are 62 possibilities of occurrence of each character in password. So the number of possible passwords is:

$62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 = 62^{12}$

Procedure:

Step 1: Start the process
Step 2: Create random character list with numbers, upper & lower-case letters.
Step 3: Password must be in fixed length of 12.
Step 4: Create Random Password Generator method to generate the password.
Step 5: Random Password Generator chooses any of the three character set.
Step 6: The index position of any one of the characters from the random character set is returned.
Step 7: Append the characters selected through the index, one by one.
Step 8: Print the password.
Step 9: End.

B. New Password Encryption Method

Cryptography is a process of converting ordinary text or plain text to cipher text (encryption), then converting back to the original text (decryption). There are several ways to classify the various algorithms. The most common types are i) Secret Key Cryptography which is also known as Symmetric Key Cryptography and ii) Public Key Cryptography which is also known as Asymmetric Key cryptography. The algorithm that we are adopting is the symmetric key cryptographic technique. The key that is used to encrypt the password is 4 digits binary number (>=1000). This algorithm maintains three separate variable lists that contain upper and lower case letters and numbers. To encrypt the password, the algorithm checks the alphabets in an inputted password, and finds out whether the alphabet is an upper-case/lower-case/number and then corresponding constant is used with the character.

The algorithm consists of three parts. The first part is to convert the password to decimal value. The second part is to perform binary manipulations. The third part is to converting binary value to alpha-numerical value. In the first step there are three cases to change alphanumeric into decimal value: (1) If the alphabet is upper-case letter, and then the ASCII value of alphabet and constant 50 are summed. (2) If the alphabet is lower-case letter, then the constant 20 is subtracted from the ASCII value of alphabet (3) If the alphabet is number, and then the constant 10 is subtracted from the discovered ASCII value.

The second step converts the decimal value received from the previous procedure into binary and reverses the binary value. The reversed binary value is divided by the key. The key is selected by the user. The remainder and the quotient are formed as resultant binary value. The remainder in first 3 digits and quotient in next 5 digits are formed. The final step is to convert the resultant binary value to alpha-numerical value.

Procedure:
Step 1: Start the process.
Step 2: Store upper-case letters, lower-case letters and numbers as separate variable lists.
Step 3: Random password is used as input.
Step 4: Conversion from the alphabets to decimal notations.
4.1. Check the alphabet is upper-case.
Find out its ASCII value and add with constant 50.

4.2 Check the alphabet is lower-case.
Find out its ASCII value and subtract the constant 20 from the ASCII value.
4.3 Check the alphabet is Number.
Find out its ACSII value and subtract the constant 10 from the ASCII value.
Step 5: Convert the resultant decimal value to binary digits.
Step 6: Reverse the binary digits.
Step 7: Get the key from the user.
Step 8: Divide the reversed binary digit by the key.
Step 9: Store the remainder in first 3 digits & quotient in next 5 digits (remainder and quotient wouldn't be more than 3 digits and 5 digits long respectively. If any of these are less than 3 and 5 digits respectively we need to add required number of 0s (zeros) in the left hand side.
Step 10: Convert the binary to decimal value
Step 11: Consider the decimal value as the ASCII value and find the related character as encrypted value.

Example:
Let the character of the plaintext of the Random Generated Password be 'B' (Upper-case letter).
1. ASCII equivalent of 'B' is 66 in decimal.
2. ASCII value and constant value are added.
= 66+50 (ASCII value + constant 50)
= 116
3. The binary value of 116 is 01110100.
4. Reverse of this binary number would be 00101110.
5. Let 1000 be the divisor i.e. Key.
6. Divide 00101110 (dividend) by 1000(divisor).
7. The remainder would be 11 and the quotient would be 101. The binary digits after division are 11000101.
8. 11000101 is converted to decimal 197.
9. The character for the ASCII value 197 is 'Å' and stored as encrypted value.

C. New Decryption Method

Encryption is the process of converting plaintext into cipher text. Decryption is the reverse of encryption where the cipher text is converted into plaintext. New type of decryption algorithm has to be generated to decrypt the encrypted password. The symmetric key cryptography is using same key for both encryption and decryption.

The decryption process will be taken according to the following step: This algorithm maintains the separate arrays such as upper case array list, lower case array list and numbers array list that hold certain values for both upper and lower case alphabets and digits from 0 to 9. The key used for decryption that must be same as key used for encrypting the password. The encrypted password is in hexadecimal format.

First step of decryption is to convert the each character to ASCII value. Each ASCII decimal value is converted to binary digits.The last 5 digits are multiplied by the key. The first 3 digits of the cipher text are added with the result produced through the multiplication. If the result produced is not an 8-bit number, then zeros are added at the left to make it 8- bit number. Now the 8-bit digits are reversed.

In the second step of decryption reversed binary digits are converted to a decimal value. The decimal value is checked against the array list using the different range of values for each array. For Upper-case array, the values start from 115 to 140, for Lower-case array, the values start from 77 to 102, and Number array start from 38 to 47. The range of the decimal value is checked to find the array list. If the decimal is in upper-case array list, subtract the constant 50 from the decimal value. This is the ASCII value of alphabet in random password. If the decimal is in lower-case array list, then add constant 20 with the decimal value. This process produces the ASCII value and then the alphabet for ASCII value is discovered. If the decimal is in number array list, then the constant 10 added with the decimal value. The resultant value is the ASCII value of the character. In the final step, the ASCII value is converted to character, which produces the decrypted form.

Procedure:
Step 1: Start the process
Step 2: Convert the each encrypted character to ASCII decimal value.
Step 3: Convert the decimal value to binary digits.
Step 4: Multiply last 5 digits of the cipher text by the Key.
Step 5: Add first 3 digits of the cipher text with the result produced in the previous step.
Step 6: If the result produced in the previous step i.e. step 5 is not an 8-bit number we need to make it an 8- bit number.
Step 7: Reverse the binary digits
Step 8: Find the decimal value for the reversed digits.
Step 9: Maintain Upper-case array (values start from 115 to 140), Lower-case array (values start from 77 to 102) and Number array (values start from 38 to 47). Check the resultant decimal is available in either upper case array or lower case array or number array.
9.1 If the decimal value is in upper case array, subtract constant 50 from the decimal value. This process produced ASCII value of any one upper-case letters in password.
9.2 If decimal value is in lower case array, add the constant 20 with the decimal value. This answer is the ASCII value of one of the lower case alphabet in password.

9.3 If the decimal value is in number array, add the constant 10 with the decimal value.The result is ASCII value of numbers in password.

Step 10: Find out alphabet for resultant ASCII value and append one by one to form decrypted password.
Step 11: End.

Example:

After encrypting 'B', we get the cipher text as 'Å' which is the ASCII character for 197, now we will decrypt the cipher text to get the plaintext.

1. The cipher text 'Å' is converted to decimal by finding ASCII value.
2. The binary value of 197 is 11000101.
3. After multiplying 00101 (last 5 digits of the cipher text) by 1000 (Key) the result would be 101000.
4. After adding 110 (first 3 digits of the cipher text) with 101000 the result would be 101110.
5. Since 101010 is not an 8-bit number we need to make it 00101110.
6. After reversing the number it would be 01110100.
7. The decimal value of 01110100 is 116.

The decimal value '116' will available in upper-case array list hence 50 should be subtracted
8: Detection of ASCII value
= 116 – 50 (decimal value–constant 50) = 66 (ASCII value)
9: Convert the given ASCII value into alphabet which is "B".

Thus the character 'B' from random password is encrypted as 'Å' using the proposed encryption method and the encrypted character is decrypted back to 'B' using the proposed decryption method.

IV. EXPERIMENTAL STUDY

The concept is well experimented and demonstrated through a software program. Alpha-Numeric Random Password Generator Algorithm is tested by generating password for the users of a browsing centre. When the new users are registered in the browsing centre, all the details are stored in the server and the password is generated and sent to the user through a message. At each login, the users can use the randomly generated passwords. This provides security to the regular customer of a browsing centre, to safely maintain his/her work in the server based environment. Whenever the user is logging on to the system, he/she has to submit the respective user name and password. Password that is entered by the user is always being in the form of a plaintext but the information which is stored in the database is an encrypted form of the plain text. Hence, the plain text is converted between encrypted and decrypted form accordingly while comparing the user entered password and already stored password. The security measure is increased by storing the password using the encrypting and decryption method proposed in this paper.

The proposed method could also be used for providing username and password for the lab users of colleges or any other institutions. The students may register his/her detail and the password for the login could be generated using the proposed method. This random password generation mechanism can be used for any application where authentication is important.

V. CONCLUSION

The password generated using alpha-numerical random password mechanism that was illustrated above is practical and can be used with great results. When the password is selected manually, most of the time, the users select the password that are related to himself or herself and related to any of the event. This gives the space for the intruders to deploy various attacks in breaking the passwords. The random generated passwords avoid this particular situation. One of the drawbacks could be the difficulty in memorizing the randomly generated password. But when comparing the security achieved through the randomly generated password, it is much preferable than the manually chosen password. The encryption and decryption standard provided here also strengthens the security measures. Since, the encryption and decryption standards are simple, it is cost-effective. The above done work also creates awareness and interest to start exploring this field more.

VI. FUTURE ENHANCEMENTS

Since all the applications are protected with passwords, more research can be accomplished for secured automatic password generations. The proposed method uses only the alphabets and numerical values for random character list. Still special symbols could be considered for strengthening the password. The password length also can be extended to make the password strong. New encryption and decryption standard could be implemented with the randomly selected passwords. The experimental study can be done with large number of samples in future.

REFERENCES

[1] Art Conklin, Glenn Dietrich, Diane Walz, "Password-Based Authentication: A System Perspective", Proceedings of the 37th Hawaii International Conference on System Sciences – 2004.
[2] NikCubrilovic, The Anatomy of the Twitter Attack, TechCrunch, July 2009.
[3] Thorsten Brantz and Alex Franz,The Google Web 1T 5-gram corpus, Technical Report LDC2006T13, Linguistic Data Consortium, 2006.
[4] Mike Bond, Comments on authentication, www.cl.cam.ac.uk/~mkb23/research/GridsureComments.pdf, 2008.
[5] Manoj Kumar Singh," Password Based a Generalize Robust Security System Design using Neural Network", IJCSI-International Journal of Computer Science Issues, Vol. 4, No. 2, 2009.
[6] Michael D. Leonhard, V. N. Venkatakrishnan, "A Comparative Study of Three Random Password Generators", IEEE EIT 2007 Proceedings.
[7] Ayushi, "A Symmetric Key Cryptographic Algorithm", International Journal of Computer Applications, Volume 1 – No. 15, 2010.
[8] Kamini H. Solanki, Chandni R. Patel, "New Symmetric Key Cryptographic algorithm for Enhancing Security of Data", International Journal of Research in Computer Engineering and Electronics, volume 1, issue 3,Dec 2012.