Two Factor Authentication Based Generated One Time Password

Dr. Hilal Hadi
Department of Computer Sciences, University of Technology
Rana Faeez
Department of Computer Sciences, University of Technology
Email: Rana_almashkoor@yahoo.com

Revised on:121/10/2014 & Accepted on: 7/5/2015

Abstract:

This paper explains a method of how the two factor authentication implemented using software token to generate One Time Password (OTP) to secure user accounts. The proposed method guarantees authenticating e-learning features. The proposed system involves generating of OTP by using authentication web service. The generated code is valid for only one login and it is verified using Secured Cryptographic Algorithm. The proposed system has been implemented and tested successfully.

التخويل ثنائي العوامل بالاعتماد على كلمة المرور المتولدة مرة واحدة

الخلاصة:

هذا البحث يوضح طريقة كيفية تمثيل التخويل ثنائي العوامل باستخدام برنامج الرمز لتوليد OTP لتأمين حسابات المستخدمين. الطريقة المقترحة تضمن توثيق ميزات التعلم الالكتروني. النظام المقترح يتضمن توليد كلمة المرور المستخدمة لمرة واحدة باستخدام مصادقة خدمة الويب. OTP ولد يكون صالح لتسجيل دخول واحد فقط ويتم التحقق منه باستخدام خوارزمية التشفير المؤمنة. وقد تم تنفيذ النظام المقترح واختباره بنجاح.

Keywords: Authentication, OTP, HMAC, SHA256, DES

INTRODUCTION

Security is a major concern today in all sectors such as educational institutions, governmental applications, military organization, banks, etc. The rapid growth in the number of online services leads to an increasing number of different digital identities each user needs to manage. But static passwords are perhaps the most common type of credential used today [1]. In spite of static password is short and easy, But it has many disadvantages they are based on subjects close to the user - wedding date, children's name. When different systems have different passwords they can be difficult to remember and may have to be written down raising their vulnerability [2].

To overcome static password drawbacks, a new method was invented that is called "One Time Password (OTP)" OTP is a password that valid for only one login session or

https://doi.org/10.30684/etj.33.3B.1
2412-0758/University of Technology-Iraq, Baghdad, Iraq

transaction. This OTP allow the user to get login into the system by entering their password with OTP [3]. Some solutions based on hardware token to generate OTP, but tokens need cost to purchasing, managing. Tokens are vulnerable to lost or stolen. So we propose a securely generated and verified OTP based web service. The OTP generator installed on each client computer, which users run to generate a new OTP with a predefined expire time.

INCEPTION

E-learning systems represent a new form of learning and are becoming more and more popular every day .Hence security in e-learning has become a fundamental requirement. But the problem of e-learning systems is that only little money invested for security. Also to authenticate an e-learner is a major challenge in an e-learning environment. The role of authentication techniques to prevent unauthorized access by malicious users becomes more significant [4]. Authentication is the act of creating or validating something (or someone) as authentic and claims made about the topic are true. User authentication methods can be classified into three categories:

(1) Methods based on human memory (What you know) such as passwords.
(2) Methods based on physical devices (What you have) such as magnetic, token, etc.
(3) Methods based on biometrics (What you are) such as fingerprint, iris, etc.

Tokens could be a hardware and software, using tokens provide a much safer environment for users, but it can be very costly for organizations. For example, an e-learning foundation with hundred learners will have to purchase, install, and maintain a hundred tokens. Furthermore, the foundation has to provide continuous support for training learners on how to use the tokens [5].

From the user's perspective, having an account with more than one foundation means need to carry and maintain several tokens which constitute a big inconvenience and can lead to tokens being lost ,stolen ,or broken. So we propose a computer-based software token that will save the organizations the cost of purchasing and maintaining the hardware tokens [6].

ONE TIME PASSWORDS

A One Time Password (OTP) is just what the names implies, a password that is only valid for one login. The benefit of OTPs is that it offers much higher security than static passwords, in expense of user friendliness and configuration issues. OTPs are immune against password sniffing attacks. OTP can be generated using different methods [7]:

-Time-based OTPs:

A device with an internal clock generates passwords that are depending on the current time, and the same password is generated at the authentication server. The generated OTP is only valid for a short period of time, before it expires [8].

- Counter-synchronized OTPs:

A counter is synchronized between the authentication server and the device. For each login the counter advance one step in the device and in the server [9].**- Seed-chain**

OTPs:

In this method, a previous entered OTP is used as a seed to generate a new OTP, building a chain of passwords that all depend on the previous password. The passwords will be printed out on a piece of paper, and the user will have to follow the list in the correct order to be able to log in [2].

-Challenge-based OTPs:

These kinds of OTPs are used together with two-factor authentication. A user has to put a challenge into the generating device (often a PIN code) in order to generate the OTP [10].

Exiting HOTP Algorithm

The Hash Message Authentication Code (HMAC) - based One Time Password (HOTP) algorithm [11] is based on an increasing counter value and a static symmetric key known only to the token and the validation service. In order to create the HOTP value, we will use the HMAC-SHA-1 algorithm. As the output of the HMAC-SHA-1 calculation is 160 bits, we must truncate this value to something that can be easily entered by a user. A brief description of HMAC-SHA-1 given in the following:

$$HOTP(K, C) = Truncate(HMAC\text{-}SHA\text{-}1(K, C)) \quad \ldots (1)$$

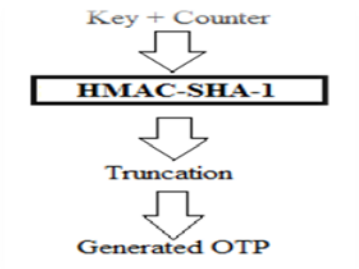
Where:

Truncate represents the function that converts an HMAC-SHA-1 value into an HOTP value

The ***Key (K)***, the ***Counter (C),*** and Data values are hashed high-order byte first. [11]

With a strong foundation of the existing HOTP algorithm, it is necessary to know where the changes should be incorporated to adapt it to our specifications while retaining its strengths. Figure (1) flow diagram provides graphical summary of the working and generation of OTP using this algorithm.

The heart of the algorithm is the HMAC-SHA-1 digest that is produced. HMAC is not a hash function. It is a message authentication code (MAC) that uses the hash function internally. The HMAC-SHA-1 uses a shared "Key" with the user and a "Counter" value as a seed to generate a 160 bits message digest. The counter value changes with every login and thus provides randomness. This 160 bits message digest is then supplied to a truncation function that generates a 6 digit OTP using modulo function [11].



Figure(1)Flow diagram of the existing algorithm [12]

For better security it is recommended the use of 7-digit or 8-digit OTPs. The Chinese group [13] proved that it was possible to break SHA1 hashes. Although HMAC-SHA1 is not compromised due to the breaks in SHA1 hash function, it is a good idea to use another hash algorithm gives more security aspects and finds a schema to increase the randomness of the produced OTP.

System Design and Implementation

In this paper, the proposed computer-based software token was implemented and tested. That is supposed to replace existing hardware token devices. The System involves generation of Secured OTP using Cryptographic algorithm and verify the authenticated users to access to the allowed services, the proposed OTP generator will be installed on each client computer, which users will run to generate a new OTP. The user will type the OTP value when prompted by the web browser and clicks submit. The authentication server call a web service to verify authentication attempts by using the same Cryptographic algorithm and the user information from the SQL server table check the OTP computation to respond with success or failure.

The Proposed OTP Algorithm

To build strong authentication system, the generated OTP must be unpredicted, high randomness, hard to retrieve by hackers; therefore there is a need to develop a secure OTP algorithm. The following factors will be used to generate OTP:

User name: unique name that identify each registered user with the system.
Password: secret phrase for each user, that submits to the system during registration activity. (Something the user knows).
E-mail: electronic mail for the user.
Phone number: phone number of the user.

Algorithm1: Proposed OTP Generation Algorithm
Input :

- The value of current counter, **C**.
- User information in bits.
- The value of the user key, **K**.
-

Output:

- OTP of length 8-digits.

Processing:
***Part A* //** (**HMAC-SHA256**)
Step1: Making concatenate the user information with **C** and encode as a message.
Step2: calculate the (HMAC-SHA256) to the result from step1 with the key, **K.**
Step3: Passing the result of step2 in part A to truncate operation.

***Part B* // (Truncate Operation)**
Step1: Split the received result from HMAC-SHA256 into four sub groups each with (8 bytes) length.
Step2: Apply Data Encryption Standards-Permutation Choice1 (**DES-PC1**) to each of these sub groups to produce four sub groups each with (7 bytes) length.
Step3: Apply (**DES-PC2**) to the result of previous step to produce four groups each with (6 bytes) length.
Step4: Apply (**DES- 8 substitution boxes**) to the result of step3 to produce four sub groups each with (4 bytes) length.
Step5: Making XOR between the results groups as follow
A = group1 XOR group3
B = group2 XOR group4
Step6: making XOR between group A and group B, to produce the final group which is (32 bits).
Step7: Split the final 32 bits into 8 sub-groups each with 4 bits length.
Step8: Converting the groups of step7 to hexadecimal.
Step9: converting the Ascii values to character string.
Step10: Display the final result from step9 to the user as (**OTP).**
End

Fig.2. Show the flow diagram of Algorithm (1).

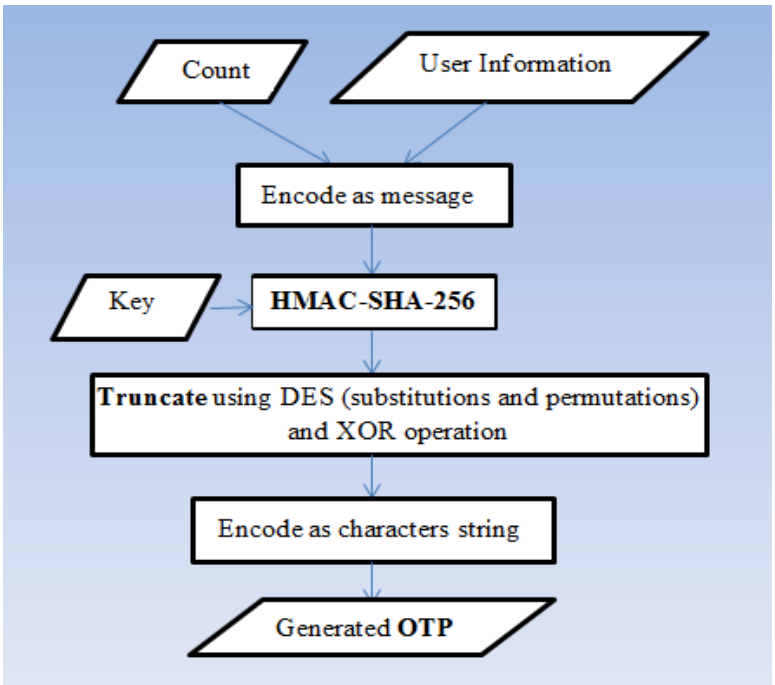
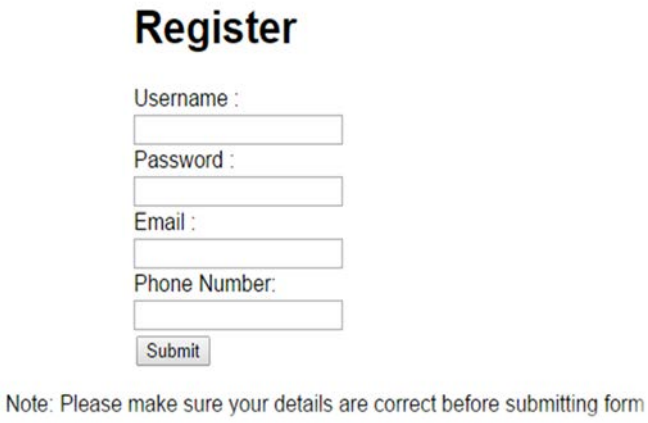


Figure (2) Flow diagram of the proposed algorithm

System Modules

Registration Phase

The service provider displays login screen to the users. Then users create an account by submitting required information and click on submit button in login page. The information entered during the registration is stored in a database .The URL link of the OTP application will be sent to the e-mail address of the user. Now the user downloads this application to generate the OTP. Figure (2) shows the registration page.



Figure(3) Registration page

A. Authentication Phase

Processing of authentication model described in fig(4).

1. The user logs in to the service provider's web site, e.g.: an E.Learning site, requesting access, As a response to this access request secure session established allowing the user enter his authentication privileges (user name, password) The first factor of authentication .

2. The service provider verifies user information by checking if it is exist in the SQL DB.

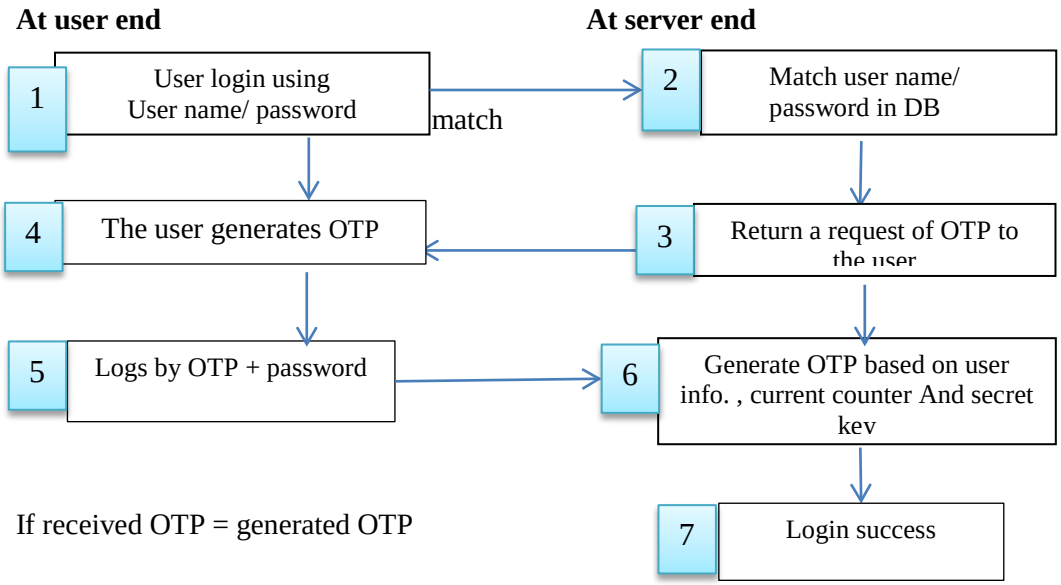
3. If the verification was successful, return to user success message and requires OTP, Else give failure message to the user and suggest visiting the registration page.

4. The user run the OTP application on his/her computer and generates OTP based on the user information, current count, and the secret key.

5. The user logs in web site by submitting the password and the generated OTP.

6. The service provider generates OTP based on the current counter, secret key of this user, and the retrieved user information from the DB.

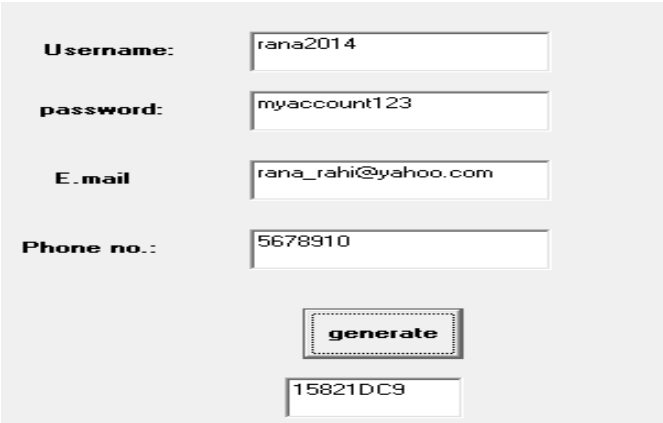
7. Verify the OTP, if the generated OTP by the provider is the same as the received OTP from the user, return "login success" else "login failure".



Figure(4) Processing of the model

OTP Illustration

The user info, random secret key and current counter all are the inputs to the proposed algorithm to produce (32bytes), which reduced to 8 characters OTP code. For example, the user info as in (fig.5), current counter is 64bits"7b4510c4ef07b198" and the random secret key value was 256bits" 77ad4d0d33dd8954b3b3c4f7838870ba6ae1fd31310713167fee0344629e5cac".The result of the HMAC-SHA-256 was "46c96e730ecf3c755e75cdac4a46d50a616 f034011e97054fa9949eefb8b58a8", which passed to truncation operation to be reduced to (32 bits). The result of truncation was encoded to represent the OTP code, which is "15821DC9" based of the above given input parameters. This is code valid for one login only.



Figure(5) Proposed software for OTP

System Tests and Analysis

1. Tests of randomness

The randomness of the generated OTP depends on different factors; the most important factor was the randomness of the secret key. The proposed algorithm use SHA256 hash function, the permutation and substitution of the DES and the XOR operation.

- The permutations and substitutions will increase the randomness of the generated OTP.

- The XOR operation have the property of preserve the randomness meaning that a random bit XORed with a non-random bit will result in a random bit. Multiple sources of potentially random data can be combined using XOR, and the unpredictability of the output is guaranteed to be at least as good as the best individual source [14].

To prove the randomness of the generated OTP, we experiment the randomness by passing it through four popular tests of randomness **[**15]:

A. **Frequency Test**

The purpose of this test is to determine whether the number of 0's and 1's in s are approximately the same, as would be expected for a random sequence. Which approximately follows a α2 distribution with 1 degree of freedom if n ≥ 10

$$X1 = (n0 - n1)^2 / n \quad \ldots (2)$$

B. **Blocks Test**

The Block test divides a pattern into blocks and examines the number of 1s in each block. A random pattern would be expected to have about 50 percent 1s in every block.

C. **Runs Test**

The purpose of the runs test is to determine whether the number of runs (of either zeros or ones) of various lengths in the sequence s is as expected for a random sequence. The expected number of gaps (or blocks) of length i in a random sequence of length n is ei = (n−i+3)/2i+2. Let k be equal to the largest integer i for which ei ≥ 5. Let Bi, Gi be the number of blocks and gaps, respectively, of length i in s for each i, 1 ≤ i ≤ k. The statistic used:

$$X2 = \sum_{i=1}^{K} \frac{(b_i - e_i)^2}{e_i} + \sum_{i=1}^{K} \frac{(g_i - e_i)^2}{e_i} \quad \ldots. (3)$$

Which approximately follows a α^2 (chi-square distribution) with 2k − 2 degrees of freedom.

D. **Serial Test (Two-Bit Test)**

The purpose of this test is to determine whether the number of occurrences of 00, 01, 10, and 11 as subsequences of s are approximately the same, as would be expected for a

random sequence. Let n0, n1 denote the number of 0's and 1's in s, respectively, and let n00, n01, n10, n11 denote the number of occurrences of 00, 01, 10, 11 in s, respectively. Note that n00 + n01 + n10 + n11 = (n − 1) since the subsequence's are allowed to overlap. The statistic used is

$$X_3 = 4/n - 1\,(n^2_{00} + n^2_{01} + n^2_{10} + n^2_{11}) - 2/n\,(n^2_0 + n^2_1) + 1 \qquad \ldots\ldots (4)$$

Which approximately follows a α2 (chi-square distribution) with 2 degrees of freedom if $n \geq 21$.

Example

We test the OTP codes that produce to one user for two times login to the system. The randomness tests given in the following:

User name: Rana2014
Password: myaccount123
E-mail: rana_rahi@yahoo.com
Phone no.: 5678910
Key: "77ad4d0d33dd8954b3b3c4f78 38870ba6ae1fd31310713167fee0344 629e5cac"
Count: 7b4510c4ef07b198
OTP: 15821DC

```
1. Testing input frequencies
pValue for Frequency test = 0.2888
Sequence Passes frequency test for randomness


2. Testing input blocks (block length = 4)
pValue for Block test = 0.7576
Sequence Passes block test for randomness


3. Testing input runs
pValue for Runs test = 0.8367
Sequence passes runs test for randomness


4. Testing input serial
pvalue for serial test = 0.8623
Sequence passes serial test for randomness
```

User name: Rana2014
Password: myaccount123
E-mail: rana_rahi@yahoo.com
Phone no.: 5678910
Key:" 716f6e 863f 744b9ac22c97ec7b76 ea5f5908bc5b2f67c 61510bfc4751384ea7a"
Count: 8b4510c4ef07b198
OTP: 038279B4

```
1. Testing input frequencies
pValue for Frequency test = 0.2888
Sequence Passes frequency test for randomness


2. Testing input blocks (block length = 4)
pValue for Block test = 0.3423
Sequence Passes block test for randomness


3. Testing input runs
pValue for Runs test = 0.3718
Sequence passes runs test for randomness


4. Testing input serial
pvalue for serial test = 0.7232
Sequence passes serial test for randomness
```

2. Time to generate a single OTP

The time to generate the OTP is governed only by the algorithm to generate the digest. This time is almost constant over the length of the OTP generated. The average time for generation of OTP was found to range from (0.3 – 0.5) second for OTP of size 8.

3. OTP Entropy

Entropy define as lack of order or predictability; gradual decline into disorder. Higher OTP entropy the less predictable OTP patterns, so the stronger and more secure OTP [16].

This is a simple equation to demonstrate OTP entropy:
H total binary bits of entropy
L length of your password
N number of possible symbols in password

$$H = L * \log_2 (N) \quad \text{..... (5)}$$

Here below Table (1) Shows the Entropy of the generated OTP codes.

Table (1) OTP Entropy

Characters	Equation	Bits	Passwords
8	$8 * \log_2(16)$	32	4294967296

Conclusions

1. OTP generation algorithms typically make use of pseudo randomness or randomness. This is necessary because otherwise it would be easy to predict future OTPs by observing previous ones.

2. If user lost their pervious password then there is no need of worry for them because OTP give them new password for each session.
3. OTP prevent user id from replay or eavesdropping attack.
4. In this work the proposed method for generating the OTP was implemented and tested, In future more work should be done on how to provide more security in this approach.

References

[1] Josang and G. Sanderud, "Security in Mobile Communications: Challenges and Opportunities," in Proc. Of The Australasian Information Security Workshop Conference on ACSW Frontiers.
[2] E.Kalaikavita , Juliana Gnanaselvi,"Secure Login Using Encrypted One Time Password (OTP) and Mobile Based Login Methodology," Department of Information Technology, Rathinam College of Arts and Science College, 2013
.
[3] B. Rajkumar, C. Yeo, S. Venugopal, S. Malpani, "Cloud computing and Emerging IT platforms: vision, hype, and reality for Delivering Computing"
5th utility, Future Generation Computer Systems, 2009.
[4] Asha, S. , Chellappan, C. ,"Authentication of E-learners Using Multimodal Biometric Technology", Department of Computer Science Engineering, Anna university, Chennai, IEEE, 2008.
[5] Fadi Aloul, Syed Zahidi Two Factor Authentication Using Mobile Phones, College of Information Technology, UAE University.
[6] George Sadowsky James X. Dempsey Alan Greenberg Barbara J. Mack Alan Schwartz, "Information Technology Security Handbook", Washington, 2003.

[7] Markus Johnsson,A.S.M. Faruqe Azam,"Mobile One Time Passwords and RC4 Encryption for Cloud Computing", Master Thesis, School of Information Science, Computer and Electrical Engineering, Halmstad University, 2011.
[8] Young Sil Lee, Hyo Taek Lim , HoonJae Lee , "A Study on Efficient OTP Generation using Stream Cipher with Random Digit", Advanced Communication Technology (ICACT), IEEE, 2010.
[9] Himika Parmar, Nancy Nainan, Sumaiya Thaseen, "Generation of Secure One-Time Password Based on Image Authentication", Computer Science & Information Technology, VIT Unversity, India, 2012.
[10] Vishal Paranjape , Vimmi Pandey ,"An Approach towards Security in Private Cloud Using OTP", International Journal of Emerging Technology and Advanced Engineering (IJETAE),Volume 3, Issue 3, March 2013.
[11] Matthew Allan Ezell , "A Framework for Federated Two-Factor Authentication Enabling Cost-Effective Secure Access to Distributed Cyberinfrastructure", Master thesis, University of Tennessee, Knoxville.
[12] Rohit Tolani, Anjali Yeole, Sachin Gavhane,"An HOTP Based Algorithm to Enhance Wi-Fi Security", Department of Computer Engineering, V.E.S.I.T, 2013.
[13] Pro. Xiaoyun Wang and her associates,"SHA1 Broken",Tsinghua University and Shandong University of Technology, (Online) http://www.schneier.com/blog/archives/2005/02/sha1_broken.html.
[14] Robert B Davies, "Exclusive OR (XOR) and Hardware Random Number Generators", 2002.
[15] J K M Sadique Uz Zaman , Ranjan Ghosh," A Review Study of NIST Statistical Test Suite: Development of an indigenous Computer Package",
Institute of Radio Physics and Electronics, University of Calcutta.
[16] Jacob Nicholson "Password Strength and Security", 2014, (Online)
http://www.inmotionhosting.com/support/website/security/password-strength.