Proactive Password Strength Analyzer Using Filters and Machine Learning Techniques

Suganya G
M.Phil Research Scholar
P.S.G.R Krishnammal College for Women
Coimbatore, India

Karpgavalli S
Senior Lecturer
GR Govindarajulu School of Applied Computer Technology
Coimbatore, India

Christina V
M.Phil Research Scholar
P.S.G.R Krishnammal College for Women
Coimbatore, India

ABSTRACT

Passwords are ubiquitous authentication methods and they represent the identity of an individual for a system. Users are consistently told that a strong password is essential these days to protect private data. Despite the existence of more secure methods of authenticating users, including smart cards and biometrics, password authentication continues to be the most common means in use. Thus it is important for organizations to recognize the vulnerabilities to which passwords are subjected, and develop strong policies governing the creation and use of passwords to ensure that those vulnerabilities are not exploited. This work proposes a framework to analyze the strength of the password proactively. To analyze the chosen password, filters and support vector machine are employed. This framework can be implemented as a submodule of the access control scheme.

General Terms

Password strength, machine learning techniques

Keywords

authentication, proactive, password strength, filters, support vector machine

1. INTRODUCTION

Life these days have become largely dependent on passwords. A typical computer user may require passwords for many purposes: logging in to computer accounts, retrieving e-mail from servers, transferring funds, shopping online, accessing programs, databases, networks, web sites, and even reading the morning newspaper online. The problem of selecting and using good passwords is becoming more important every day. The number and the importance of services that are provided through computers and networks increase dramatically and in many cases such services require passwords or other forms of user identification. For different reasons, including obvious security concerns, users have to use different passwords for different systems or services, making it more difficult to remember and protect one’s password. Passwords are not only critical for login identification, but also in more sophisticated service-granting systems, such as Kerberos. Finally, passwords are needed for protecting secret information that cannot be remembered by the user (e.g. private keys) in authentication and encryption software that is becoming essential to many applications.

The average user chooses a simple, guessable, memorable password and cares less about choosing a strong password. Nowadays there is a real and growing threat of data thieves, hackers and other criminals taking advantage of people who aren't security conscious. Thus it is important for organizations to recognize the vulnerabilities to which passwords are subjected, and develop strong policies governing the creation and use of passwords to ensure that those vulnerabilities are not exploited.

2. PASSWORD STRENGTH

A password is a secret word or string of characters that is used for authentication, to prove identity or gain access to a resource [1]. Password strength is a measurement of the effectiveness of a password in resisting guessing and brute-force attacks. The key to a strong password is length and complexity. Length is simply the number of individual characters used in the creation of the password, while complexity refers to the number of characters that could potentially be used in the creation of the password. Using

Most of the password strength checking tools rate the password as very weak, weak, moderate or medium or good, strong and very strong. A password standard describes that a password should satisfy the following criteria: they should have at least eight characters, including a mixture of upper- and lower-case and some numbers and special characters; the password should also not use personal information, the account name, or dictionary words in any language.

For a password of a given length, the number of permitted symbols determines its maximum possible strength. Users rarely make full use of larger characters sets in forming passwords. Most of the survey results reveal that 10% to 15% of the user’s passwords used mixed case, numbers, and symbols. Though there are many alternatives to passwords for access control, password is the more compellingly authenticating the identity in many applications. Hence the requirement of an effective password policy and proactive password checking system of an organization increases, that helps in selecting strong passwords and managing them, to protect the identity and the resources.

3. RELATED WORK

Most of the password meters use lexical rules. Decision trees were also used to check the strength of the password [2]. Enfilter is a Windows based tool that employs decision tree classifier, lexical analysis to proactively assess the strength of the password [3]. In our previous work we employed supervised machine learning algorithms to predict the strength of the password [4]. The problem of choosing weak passwords that are likely to undergo brute force attacks and enforcing the choice of strong passwords is something vital and has to be addressed in a proper manner. Hence the problem has been dealt using various filters and the support vector machine that outperformed other supervised machine learning algorithms namely oneR, C 4.5 Decision tree classifier, Multilayer perceptron and Naïve bayes classifier [5].

4. METHODOLOGY

Support Vector Machine is used for learning the classification model and to classify the given password.

4.1 *SUPPORT VECTOR MACHINE*

Support Vector Machine a new approach to supervised pattern classification that has been successfully applied to a wide range of pattern recognition problems. Support vector machine is a training algorithm for learning classification and regression rules from data. SVM is most suitable for working accurately and efficiently with high dimensionality feature spaces. SVM is based on strong mathematical foundations and results in simple yet very powerful algorithms. [6-8]

The standard SVM algorithm builds a binary classifier. A simple way to build a binary classifier is to construct a hyperplane separating class members from non-members in the input space. SVM also finds a nonlinear decision function in the input space by mapping the data into a higher dimensional feature space and separating it there by means of a maximum margin hyperplane. The system automatically identifies a subset of informative points called support vectors and uses them to represent the separating hyperplane which is sparsely a linear combination of these points. Finally SVM solves a simple convex optimization problem.

The machine is presented with a set of training examples, (x_i, y_i) where the x_i are the real world data instances and the y_i are the labels indicating. Which class the instance belongs to. For the two class pattern recognition problem, $y_i = +1$ or $y_i = -1$. A training example (x_i, y_i) is called positive if $y_i = +1$ and negative otherwise. SVMs construct a hyperplane that separates two classes and tries to achieve maximum separation between the classes. Separating the classes with a large margin minimizes a bound on the expected generalization error.

The simplest model of SVM called Maximal Margin classifier, constructs a linear separator (an optimal hyperplane) given by $w^T x - \gamma = 0$ between two classes of examples. The free parameters are a vector of weights **w** which is orthogonal to the hyperplane and a threshold value γ. These parameters are obtained by solving the following optimization problem using Lagrangian duality

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \quad D_{ii}(\mathbf{w}^T \mathbf{x}_i - \gamma) \geq 1, i = 1, \ldots, l.$$

Where Dii corresponds to class labels, which assumes value +1 and –1. The instances with non null weights are called support

In the presence of outliers and wrongly classified training examples it may be useful to allow some training errors in order to avoid overfitting. A vector of slack variables ξ_i that measure the amount of violation of the constraints is introduced and the optimization problem referred to as soft margin is given below

$$\underset{\mathbf{w},\gamma}{\text{Minimize}} \quad c\sum_{i=1}^{l} \xi_i + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \quad D_{ii}(\mathbf{w}^T \mathbf{x}_i - \gamma) + y_i \geq 1, i = 1, \ldots, l.$$

The minimization of the objective function causes maximum separation between two classes with minimum number of points crossing their respective bounding planes. The parameter C is a regularisation parameter that controls the trade-off between the two terms in the objective function. The proper choice of C is crucial for good generalization power of the classifier. The following decision rule is used to correctly predict the class of new instance with a minimum error.

$$f(\mathbf{x}) = \text{sgn}[w^T x - \gamma]$$

The advantage of the dual formulation is that it permits an efficient learning of non–linear SVM separators, by introducing kernel functions. Technically, a kernel function calculates a dot product between two vectors that have been (nonlinearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still feasible although the dimension of the real feature space can be very high or even infinite. The parameters are obtained by solving the following non linear SVM dual formulation (in Matrix form).

Minimize

$$\mathrm{L_D}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\boldsymbol{Q}\mathbf{u} - \mathbf{e}^{\mathrm{T}}\mathbf{u}$$

$$\mathbf{d}^{\mathrm{T}}\mathbf{u}_{=0.} \ \tilde{} \ \mathbf{0} \leq \boldsymbol{u} \leq C\boldsymbol{e}$$

where Q=DKD and K is kernel matrix. The kernel function $K(AA^T)$ may be polynomial or RBF (Radial Basis Function) is used to construct hyperplane in the feature space, which separates two classes linearly, by performing computations in the input space. The decision function in this nonlinear case is given by

$$f(\mathbf{x}) = \operatorname{sgn}\left| K(x, x_i^T) * u - \gamma \right|$$

When the number of classes is more than two, then the problem is called multiclass SVM. There are two types of approaches for multiclass SVM. In the first method called indirect method, several binary SVM's are constructed and the classifier's output are combined for finding the final class. In the second method called direct method, a single optimization formulation is considered. The formulation of one of the direct methods called Crammer and Singer Method [9] is

Minize

$$\frac{1}{2}\sum_{k=1}^{N} \boldsymbol{w}_k^T \boldsymbol{w}_k + C\sum_{i=1}^{n} \xi_i$$

subject to the constraints

$$\boldsymbol{w}_{k_i}^T \phi(\boldsymbol{x}_i) - \boldsymbol{w}_k^T \phi(\boldsymbol{x}_i) \ge e_k^i - \xi^i \quad , \forall \mathrm{k} \ne \mathrm{k}_\mathrm{i}$$

where k_i is the class to which the training data x_i belong,

$$e_k^i = 1 - c_k^i$$

$$c_k^i = \begin{cases} 1 \text{ if } \mathrm{k_i} = k \\ 0 \text{ if } \mathrm{k_i} \ne k \end{cases}$$

The decision function for a new input data x_i is given by

$$\hat{d}_j = \arg \max_{\mathrm{k}} \; f_k(\boldsymbol{x}_j)$$

where

$$f_k(\boldsymbol{x}_j) = \boldsymbol{w}_k^T \phi(\boldsymbol{x}_j) - \gamma_k$$

5. THE PROPOSED FRAMEWORK

Proactive password strength analyzer is designed with the following filters to reject the passwords that are commonly chosen by the user, by considering the human tendency to choose passwords that are easy to remember, simple and short.

Filter 1: It verifies the given password is same as the username or empty. In that case an error message is displayed and the password is classified as very weak password.

Filter 2: After passing through Filter1, Filter2 verifies the password against the list of most commonly used passwords. A list of 200 words is maintained and is used for verification. The words are collected from websites. If the given password matches with the list, it is classified as very weak password and appropriate alert will be displayed. Figure 1. Shows the framework of the proactive password strength analyzer.

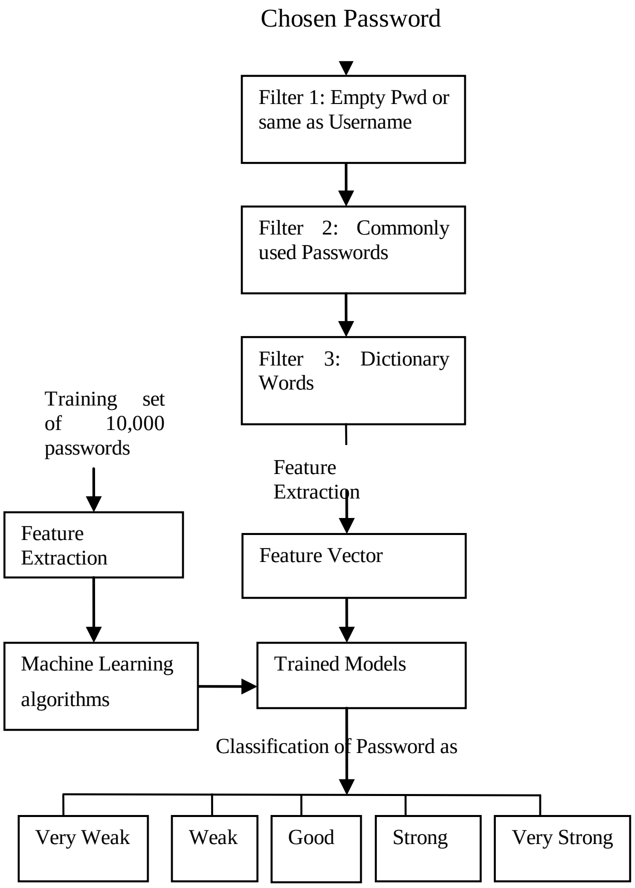

Chosen Password
Filter 1: Empty Pwd or same as Username
Filter 2: Commonly used Passwords
Filter 3: Dictionary Words
Training set of 10,000 passwords
Feature Extraction
Feature Extraction
Feature Vector
Machine Learning algorithms
Trained Models
Classification of Password as
Very Weak
Weak
Good
Strong
Very Strong

Fig. 1 Framework of the proactive password strength analyser

Filter 3: After passing through filter1 and filter2, to check the vulnerability of a weak password which is suitable for a dictionary attack, a list of 40,000 dictionary words are collected. The word size ranging from 5 to 8 characters are grouped in alphabetical order. The words are maintained in different files. The password chosen by the user are checked for its strength against a set of dictionary words. If the password is a dictionary word, it is considered as a weak password and appropriate warning will be displayed to the user.

After passing through all the filters, the given password is analysed and its features are extracted and tested against SVM classifier which is built prior by a training set of 10,000, 8 character length passwords. Those models categorize the password chosen by the user as very weak, weak, good, strong, and very strong according to its strength.

6. FEATURE EXTRACTION

Feature selection plays an important role in improving classification effectiveness, computational efficiency or both. Distinctive features describing the characteristics of the password are extracted from a set of 10,000 passwords. These passwords are generated using PC Tools Password Generator. A weight is assigned to each relevant feature. The twenty-seven descriptive features are created as a fixed length vector for password analysis. They are Length of the password, Weight of the password, Number of lowercase characters, Lowercase character weightage, Number of uppercase characters, Uppercase character weightage, Number of digits, Digits weightage, Number of symbols, Weightage of symbols, Number of middle number and symbols, Middle number/symbol weightage, Password contains characters only [case insensitive], Characters only weightage, Password contains digits only, Digits only weightage, Number of repeat characters, Repeat characters weightage, Number of consecutive uppercase characters, Consecutive uppercase characters weightage, Number of consecutive lowercase characters, Consecutive lowercase characters weightage, Number of sequential characters, Sequential characters weightage, Number of sequential digits, Sequential digits weightage and Password requirement weightage, Keyboard Patterns, Keyboard Patterns weightage, Mirrored Sequences, Mirrored Sequences weightage and Sequence was repeated, Sequence was repeated weightage.

TABLE I SCHEME OF WEIGHTS ASSIGNED

Feature	**Weight Assigned**
Number of characters in the password	Number of characters*4
Number of LC characters	(length – number of lowercase characters) * 2
Number of UC characters	(length – number of uppercase characters) * 2
Number of digits	(number of digits * 4)
Number of symbols	(symbolcount * 6)
Number of Middle number /symbols	(numbersymbolcount * 2)
Characters only	- 1 * number of characters
Digits only	- 1 * number of digits
Number of repeat characters	-(n (n –1))
Number of consecutive uppercase characters (n)	- (n * 2)
Number of consecutive lowercase characters (n)	- (n * 2)
Number of sequential characters (n)	- (n * 3)
Number of sequential digits (n)	- (n * 3)
Requirements (n)	(n * 2)
Keyboard Patterns	- (n * 2)
Mirrored Sequences	- (n * 2)
Sequence was repeated	- (n * 3)

A weighting method is adopted for computing the strength of the password. The strength is decided based on the overall score. Overall score is determined using positive and negative weightages based on a predetermined scheme given in Table I. Final score is capped with a minimum of zero and a maximum of 100. The features that make the password strong are given more weightage and the features that weaken the password are given negative weightage. Final score is the cumulative result of all bonuses and deductions.

TABLE III PASSWORD CLASSIFICATION

Class	**Score**
Very Weak	Less than 20
Weak	20 - 39
Good	40 - 59
Strong	60 - 79
Very Strong	Greater than 80

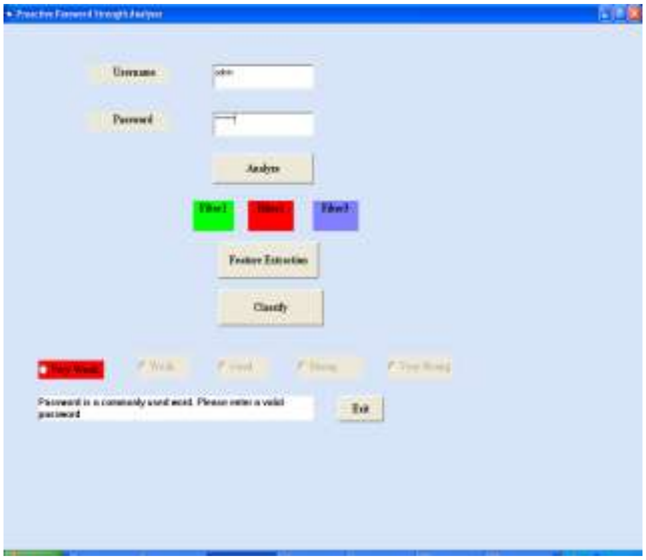
The password classification scheme is designed based on the cumulative score which is given in Table II.

7. EXPREIMENT AND RESULTS

Proactive password strength analyzer is implemented in Visual Basic. Visual Basic provides rich set of built-in functions and integrated development environment. To train and classify the password, SVM Light for Windows is used. It is an open source tool [6].

Passwords with length 8 are considered for classification. In the experiments, filter 1 designed to eliminate the empty password and passwords same as user name. After passing through filter1, the password enters filter2. Filter 2 is specially designed to cross check the chosen password is one of the most commonly used password against a list of 200 most commonly used passwords. After filter2, Filter 3 overcomes the problem of choosing weak passwords that are vulnerable for dictionary attack. A list of 20,000 dictionary words is used. After filter3, the password is further analysed using the trained models of RBF kernels. The passwords for the training data set are generated using PC tools password generator. The password dataset consists of 10,000 passwords of varying strengths, Very weak, Weak, Good, Strong and Very strong. The features are extracted from the password dataset and the respective training set is created for constructing the appropriate model.

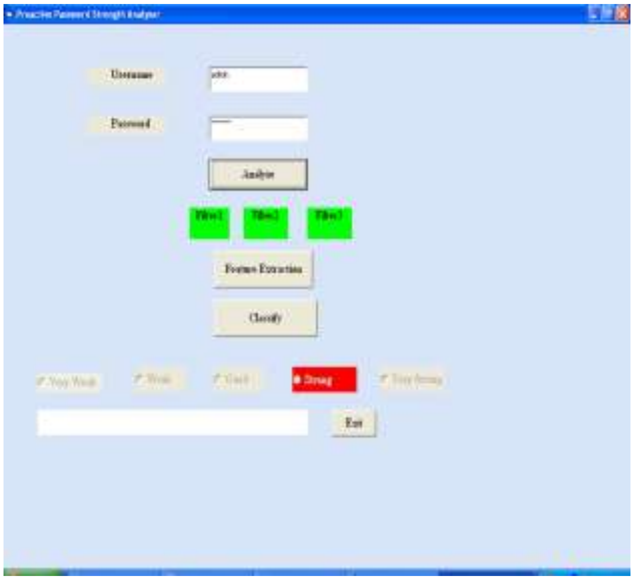
The training set consists of equal number of instances of password of different categories. The class labels are designated as 1, 2, 3, 4 and 5 to represent password strength as Very weak, Weak, Good, Strong and Very strong respectively.



admin

Password is a commonly used word. Please enter a valid password

Fig. 1 Snapshot of the proactive password strength analyser



admin

Fig. 2 Snapshot of the proactive password strength analyser

Before employing RBF Kernel to classify the chosen password in our work, the models are built using various machine learning algorithms. The performances of the classifiers are summarized in Table III.

TABLE IIIII COMPARATIVE RESULT OF THE CLASSIFIER

Evaluation Criteria	**One R**	**NB**	**J48**	**MLP**	**Linear**	**Polynomial**	**RBF**
Training time (secs)	0.09	0.03	0.16	101.83	0.094	538.38	10.24
Prediction Accuracy (%)	73.6	73.6	87.2	89.8	80.1	98.2	98.3

8. CONCLUSION

Despite the existence of more secure methods of authenticating users, including smart cards and biometrics, password authentication continues to be the most common means in use. The problem of choosing weak passwords that are highly vulnerable and likely to undergo brute force attacks and enforcing the choice of strong passwords is something vital and has to be addressed in a proper manner. As a solution, Proactive password strength analyzer is designed with various filters and employed RBF kernel to classify the chosen password. The performance of the various machines learning algorithms are studied and the best model is employed in the proactive password strength analyzer. The proposed proactive password strength analyzer can be implemented as a sub module of access control mechanism that will enable the users in selecting strong passwords, to protect the identity and the resources.

9. REFERENCES

[1] http://en.wikipedia.org/wiki/Password

[2] F.Bergadano, B.Crispo, G.Ruffo, “Proactive password checking with decision trees”,Proc. of the 4th ACM conference on computer and communications security, Zurich, Switzerland, 1997, pp 67-77.

[3] Giancarlo Ruffo, Francesco Bergadano, “EnFilter : A Password Enforcement and Filter Tool Based on Pattern Recognition Techniques”, Springer Berlin / Heidelberg, 1611-3349 (Online), Volume 3617/2005.

[4] Vijaya MS, Jamuna KS, Karpagavalli S,”Password Strength Prediction using Supervised Machine Learning Techniques”, IEEE, 978-1-4244-5321-4,pp 401-405, 2009.

[5] lan H. Witten, Eibe Frank, “Data Mining – Practical Mahine Learning Tools and Techniques,” 2nd Edition, Elsevier, 2005.

[6] John Shawe-Taylor, Nello Cristianini, “Support Vector Machines and other kernel-based learning methods”, 2000, Cambridge University Press, UK.

[7] Vapnik V.N,”Statistical Learning Theory”, J.Wiley & Sons, Inc., 1998, New York.

[8] Soman K.P, Loganathan R, Ajay V, “ Machine Learning with SVM and other Kernel Methods”, 2009, PHI, India.