Development of Myanmar Dictionary Based Proactive Password Checker

La Wynn Sandi, Nyein Aye
University of Computer Studies, Mandalay
lawynnsandi@gmail.com,nyeinaye@gmail.com

Abstract

With the rapid increase in multi-user systems, the security issues relating to the password authentication become more and more important. Numerous cryptographic protocols also rely on passwords selected by users (people) for strong authentication. Passwords are an important aspect of computer security and are also the front line of protection for system and user accounts. Most of the people enjoy using easy and memorable passwords that are weak against dictionary attacks. The proposed system will selected user passwords by using proactive password checker, especially this system can resist against Myanmar language based dictionary attacks.

1. Introduction

Nowadays, passwords become extremely important and sensitive information to gain access to many online services including e-mail, networks and e-commerce applications. People are inclined to use easy passwords that are weak against dictionary driven attacks. 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 [Neuman and Tso 1994]:, an attacker can easily mount a password-guessing assault.

Password security is an old problem. Due to the limitation of human memory, people are inclined to choose easily guessable passwords (e.g. phone numbers, birthdays, names of family or friends, or words in human languages) that might lead to severe security problems. Proactive password checking has been a common means to enforce password policies and to prevent users from choosing easily guessable passwords in the first place. When a user chooses a password, a proactive checker will determine whether his password choice is acceptable or not.

All such tests are important, but the heart of a proactive password checker has to deal with detecting membership in large dictionaries. There are two problems in this simple approach: space required to store the dictionaries, and time required to detect membership. Time is important because the user has to wait for the command prompt while the password is being checked.

Therefore, proactive password checking becomes important and it should be performed by websites before choosing a password. Nowadays, more and more Myanmar language-specific contents are being used on the Internet. This paper proposes a proactive password checking framework/design for passwords written in Myanmar language.

2. Related Work

Bloom introduced Bloom filters in conjunction with an application to hyphenation programs [8]. Most words can be hyphenated appropriately by applying a few simple rules. Some words, said around ten percent, required a table lookup. To avoid storing all the words that could be handled via the simple rules, Bloom suggested using a Bloom filter to keep a dictionary of words that required a lookup. False positives here caused words that could be handled via the simple rules to require a lookup.

In literature there were several approaches for proactive password checking: Spafford suggested Bloom filters in the OPUS system [1]. Proactive password checking had been a common means to

enforce password policies and prevented users from choossing easily guessable passwords in the first place. Proactive password checking scheme, based on second order Markov model.

Word tokenizing [4] played a vital role in most Natural Language Processing applications. It was therefore useful to syllabify texts first. Syllabification was also a non-trivial task in Myanmar. C.Herley, in [5], proposed that an overly restrictive password policy could be the cause for a bigger harm (particularly economic) than the harm the policy has meant to prevent. The research was depended on the system and on user expectations; password policies cauls have a severely negative impact on the security of the system instead of improving it. This led to the conclusion that usability of passwords and password creation policies might be even more important than security measured in bit strength or time needed to crack a password for an account.

M Walsvogel [7] described a class of best matching algorithms based on slicing perpendicular to the patterns and performing a modified binary search over these slices. And also analyze their complexity and performance. Then introduce schemes that allowed the algorithm to "learn" the structure of the database and adapted itself to it. Furthermore, showed how to efficiently implement our algorithm both using general-purpose hardware and using software running on popular personal computers and workstations.

3. Password Management

Password management is the process of defining, implementing, and maintaining password policies throughout an enterprise. Effective password management reduces the risk of compromise of password-based authentication systems to the extent possible.

To protect the confidentiality, integrity, and availability of passwords so that all authorized users and no unauthorized users can use passwords successfully as needed. Integrity and availability should be ensured by typical data security controls, such as using access control lists to prevent attackers from overwriting passwords and having secured backups of password files.

Ensuring the confidentiality of passwords is considerably more challenging and involves a number of security controls along with decisions involving the characteristics of the passwords themselves. For example, requiring that passwords be long and complex makes it less likely that attackers will guess or crack them, but it also makes the passwords harder for users to remember, and thus more likely to be stored insecurely. This increases the likelihood that users will store their passwords insecurely and expose them to attackers. Password management may also be concerned about protecting the confidentiality of user identifiers, such as usernames.

Passwords are used in many ways to protect data, systems, and networks. For example, passwords are used to authenticate users of operating systems and applications such as email, labor recording, and remote access. Passwords are also used to protect files and other stored information, such as password-protecting a single compressed file, a cryptographic key, or an encrypted hard drive. In addition, passwords are often used in less visible ways; for example, a biometric device may generate a password based on a fingerprint scan, and that password is then used for authentication.

This publication provides recommendations for password management, which is the process of defining, implementing, and maintaining password policies throughout an enterprise. Effective password management reduces the risk of compromise of password-based authentication systems. .

4. Password Selecting Strategy

There are many approaches to improving password security by selecting good passwords, such as user education, program-controlled password generation and reactive password checking (i.e., system administrators periodically run password cracking programs to search weak passwords), proactive password checking has been widely regarded as the best method.

4.1. Proactive Password Checking

Proactive password checking has been a common means to enforce password policies and prevent users from choosing easily guessable passwords in the first place. When a user chooses a password, a proactive checker will determine whether his password choice is acceptable or not, and this proactive checking is done online and the user will be immediately responded the result. Among common approaches to improving password security by selecting good passwords, such as user education, program-

controlled password generation and reactive password checking (i.e., system administrators periodically run password cracking programs to search weak passwords),proactive password checking has been widely regarded as the best [6].

5. Dictionary Attacks

A dictionary attack is secured than other methods. A common problem with systems that use passwords for authentication results when users choose weak passwords. Weak passwords are passwords that are easy to guess, or likely to be found in a dictionary attack. Thus, the choice of weak passwords may lead to system compromisation. Methods exist to prevent users from selecting and using weak passwords. One common method is to compare user choices against a list of unacceptable words.

When a hacker cracks passwords, he can use the following two methods: 1) to do a dictionary attack, which tries each of a list of word and other possible weak passwords, and simple transformations such as capitalizing, prefixing, suffixing or reversing a word as a candidate until the hashed value of the candidate matches a password hash; and 2) to launch a brute force attack to search the whole key space, which is commonly huge. Hackers, however, always prefer to use dictionary attack, because it has proved to be very effective in history [6].

Current proactive password checkers are based on the dictionary attack. They check each user-chosen password candidate against a dictionary of weak passwords. If a candidate matches a dictionary item, or anyone of its variants that are generated by common transformations, then the candidate is an unacceptable password and is rejected.

In computer security, a dictionary attack is a technique for defeating a cipher or authentication mechanism to catch its decryption key or pass phrase by trying likely possibilities, such as words in dictionary(s).A dictionary attack uses a targeted technique of successively trying all the words in an exhaustive list called a dictionary.

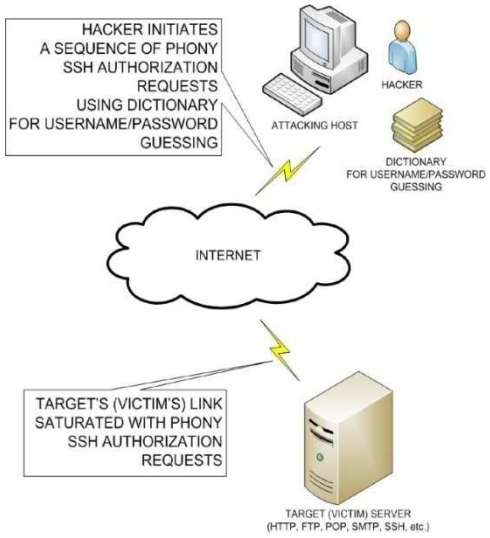

Figure 1. Dictionary Attacks

6. Syllable Segmentation

Syllable segmentation is the process of identifying syllable boundaries in a text.

6.1. Myanmar Syllables

A basic syllable consists of an initial consonant with optional medial, dependent vowels and dependent various signs.

Examples of Myanmar Syllables are as follow:

က	= က
ကာ	= က + ာ
ကံ	= က + ံ
ကျ	= က + ျ
ကား	= က + ာ + း
ကျာ	= က + ျ + ာ
ကျား	= က + ျ + ာ + း
ကျု့	= က + ျ + ာ + ့
ကော်	= က + ေ + ာ + ်
ကျော်	= က + ျ + ေ + ာ + ်
ကျောင်	= က + ျ + ေ + ာ + င + ်
ကျောင်း	= က + ျ + ေ + ာ + င + ် + း

6.2. Grammar Rules

A "grammar rule" is the right side of a production. It consists of a sequence of rule elements. The following are two of the sequence of rule elements.

6.2.1. Noun Phrase

- နာမ်+နာမ် (e.gဈေးနှုန်း၊ စစ်မြေမီးတိုင်)
- နာမ်+နာမ်+နာမ် (e.gကုန်ဈေးနှုန်း ၊စစ်မြေပြင်၊ ဆီမီးတိုင်)
- ကြိယာ+ကြိယာ (e.g ဖြတ်ပိုင်း ၊ပြုတ်ကျော် ၊အုပ်ဆောင်း)
- နာမ်+ကြိယာ (e.g ရုံပိုင်၊ ရေမန်း၊ ပဲ့ကိုင်)
- ကြိယာ+နာမ် (e.gတိုင်စာ၊ ထွက်ငွေ၊ ကဇာတ်)
- နာမ်+နာဝိ (e.gလူငယ်၊ရေမွှေး၊ပန်းနီ)
- နာမ်+ကြိယာ+နာဝိ (e.g ဆံရစ်ပိုင်း၊ဂျေးဖြစ်ဟောင်း)
- နာမ်+နဝိ+ကြိယာ (e.g မုန်စိမ်းပေါင်း၊မုန်ညှိသွင်း)
- ကြိယာ+ကြိယာ+နာမ (e.g ကင်းလွတ်ကုန်)
- နာမ်+နာမ်+ကြိယာ (e.gဆားမီးဖုတ်၊ဘီးဆံပတ်)
- နာမ်+နာမ်+နိဝိ (e.g ဆွမ်းဆန်စိမ်း၊ ရွှေပုစွန်ခြောက်)
- နာမ်+ကြိယာ+နာမ် (e.g ကမ်းတက်သဘော်၊အားတိုးဆေး၊)
- နာမ်+နာဝိ+နာမ် (e.g ရေပူစမ်း၊ရွက်လှပန်း၊လေအေးစက်)
- နာမ်+ကြိယာ+နာမ်+ကြိယာ (e.g စာစီစာကုံး)
- နာမ်+ကြိယာ+ကြိယာ+နာမ် (e.g ခါးပိုက်ဆောင်တပ်)
- နာမ်+နိဝိ+ကြိယာ+နာမ် (e.g လူနာတင်ကား)
- နာမ်+ကြိယာ+နာမ်+နာမ် (e.g ရုံသုံးဘာသာစကား)
- နာမ်+ကြိယာ+နာမ်+ကြိယာ+နာမ် (e.g ရုပ်မြင်သံကြားစက်)
- နာမ်+ကြိယာ+ကြိယာ+ကြိယာ+နာမ် (e.g မျက်နှာစုံညီစည်းဝေးပွဲ)

6.2.2. Verb Phrase

- ကြိယာ+ကြိယာ (e.gကာကွယ်၊ ရောင်းဝယ်)
- ကြိယာ+ကြိယာ+ကြိယာ (e.g ပြီးပြည့်စုံ၊ပြန်ပေါင်းထုပ်)
- ကြိယာ+ကြိယာ+ကြိယာ+ကြိယာ (e.g ဝယ်ယူတင်သွင်း)
- နာမ်+နိဝိ+ကြိယာ (e.gရေတိမ်နစ်၊စိတ်ကြီးဝင်)
- နိဝိ+နာမ်+ကြိယာ (e.gမိုက်ကြေးခွဲ၊ပြောင်းသလင်းခါ)
- နာမ်+ကြိယာ+ကြိယာ (e.gခြေလှမ်းပျက်၊စကားနိုင်လု)
- ကြိယာ+နာမ်+ကြိယာ (e.gထိုးစစ်ဆင်၊မှတ်ကျောက်တင်)
- နာမ်+နာမ်+ကြိယာ+ကြိယာ (e.gစကားလက်ဆုံကျ)
- နာမ်+ကြိယာ+နာမ်+ကြိယာ (e.g မြေစမ်းခရမ်းပျိုး)
- ပစ္စည်း(အ၊တ၊မ(နာမ်ပုဒ်ပြောင်းပစ္စည်း))+ကြိယာ (e.gအစား၊တလွဲ၊မပြတ်)
- ကြိယာ+ပစ္စည်း(နာမ်(ခြင်း၊မှု၊ရေး၊ချက်၊ဖွယ်၊စရာ၊အ-အ(ပုဒ်ပြောင်းပစ္စည်း))
(e.gကျန်းမာခြင်း၊ကျန်းမာမှု၊အကျွေးအမွေး)

6.3. Checking Spelling and Grammar Rules

Firstly, the system accept all fronts of Myanmar language and then the system changes the input password to Myanmar 3 font. And then check the password that include directly in a Dictionary or not. If the password is not include in dictionary the system check the spelling. If the password is not including in dictionary but spelling is true the system check the password grammar using grammar rule that builds in this system. Then the system gives the output to check with the bloom filter.

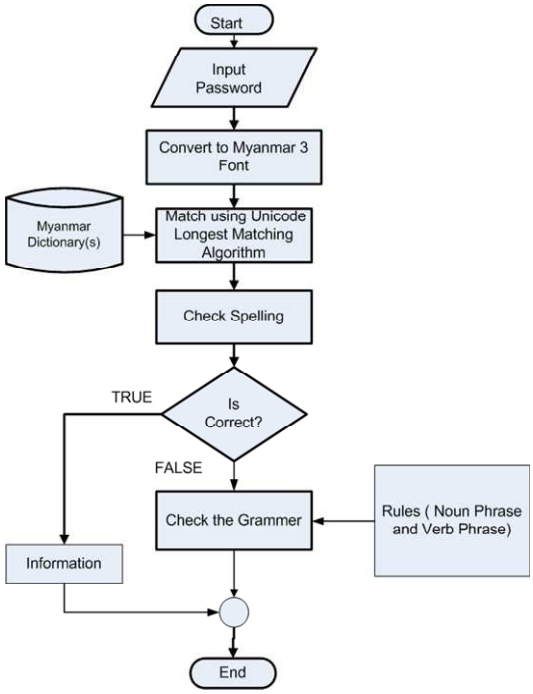

Figure 2. Block diagram for checking Spelling and Grammar (Unicode longest Matching and grammar rules)

7. Bloom Filter

A Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. False positive retrieval results are possible, but false negatives are not; i.e. a query returns either "inside set (may be wrong)" or "definitely not in set". Elements can be added to the set, but not removed (though this can be addressed with a counting filter). The more elements that are added to the set, the larger the probability of false positives.

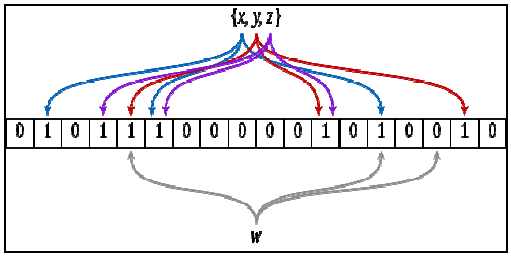

Figure 3. Bloom Filter

An example of figure (3) is a Bloom filter, there must also be *k* different hash functions defined, each of which maps or hashes some set element to one of the *m* array positions with a uniform random distribution. And the figure representing the set {*x*, *y*, *z*} . The colored arrows show the positions in the bit array that each set element is mapped to. The element *w* is not in the set {x, y, z}, because it hashes to one bit-array position containing 0. For this figure, m=18 and k=3. Figure 2 shows the block diagram for checking a word of a string.

7.1. Application of Bloom Filters

Bloom filters have found many applications, there are,

- Dictionary
- Databases and
- Network Application.

A Bloom filter consists of a set of hash functions. A hash function buffer to stored hash results temporarily, a look up array to signify hash values and a decision component is tested the membership of testing string

Firstly, the system accepts the input password that pass from preprocessing task. This password is marching and selecting the nearly same words in Myanmar dictionary by using Unicode longest matching .Check the sentence using grammar rules in True or False Condition. If the password is true it's hashed to use bloom filter the processes of bloom filer password checking are shown in figure.

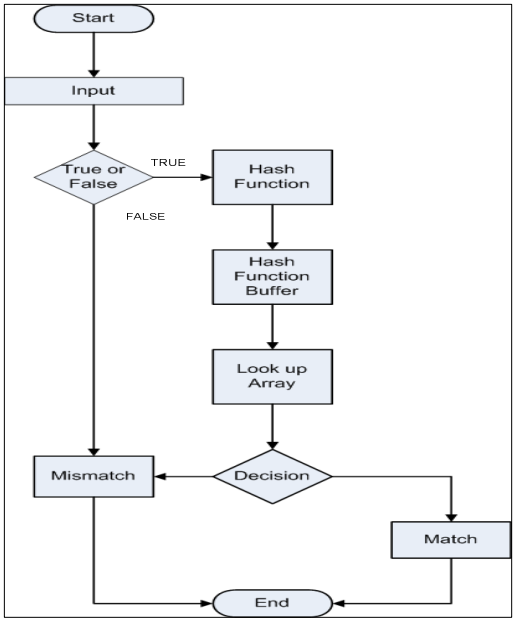

Figure 4. Block diagram for checking a word of a string.
(Bloom Filter Password Checking Tool)

8. Proposed Password Checker

The proactive password checker is constructed with Bloom filter in this paper. The goal is to help the user to properly select strong and acceptable passwords, especially that can resist against Myanmar language based dictionary attacks. In the processing, the proposed system uses KANAUNG Convertor to convert any Myanmar fonts to Myanmar 3 font. And the system allows the passwords length constraints is no more than 20 characters.

The propose system uses longest matching to modify Bloom filter for better result and reduces the storage capacity. Because Bloom Filter do not need to store all of the hash table of every word .Its only need to hash output of the preprocessing tasks.

Firstly, the system accepts the input password that pass from preprocessing task. This password is matching and selecting the nearly same words in Myanmar dictionary by using Unicode longest Matching and then is hashed to use bloom filter. Check the sentence using grammar rules and is hashed to use bloom filter. Furthermore bloom filter checks the password that include in dictionary(s) or not. If the password includes in dictionary, it is rejected.

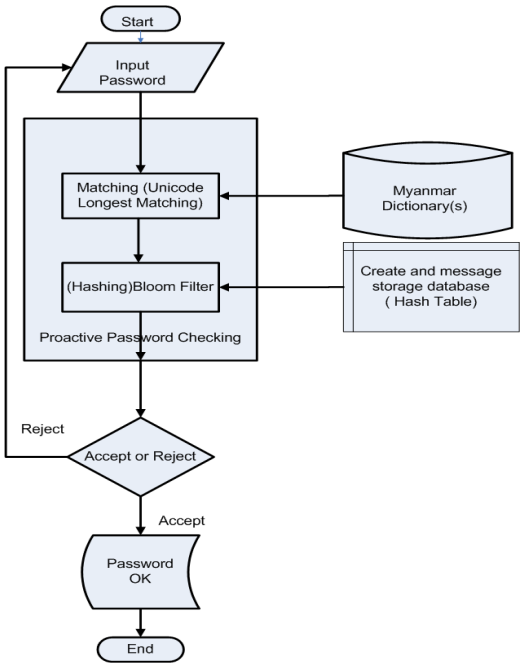

Figure 5. Password Checking Phase of Proposed Password Checker

9. Conclusion

Today, there are many researches and publications in Natural Language Processing (NLP) and researchers are trying to develop Myanmar language based ICT applications. At present, client-server applications in Myanmar language are also still in research area (e.g. Myanmar language search engine).

The dictionary attack has been very effective, and usually it could quickly crack some weak passwords. The brute force attack often has to search the whole key space, which is commonly huge and requires more significant computational power to cover thoroughly. Therefore, the dictionary attack has been considered a more serious threat than the brute force attack. Following this thought, proactive password checkers that have been built so far are (largely) designed to defeat the dictionary attack. They check each user-chosen password candidate against a dictionary of weak passwords.

The system will help the user to properly select strong and acceptable passwords, especially that can resist against Myanmar language based dictionary attacks. This research proposes a proactive password checking framework/design for passwords written in Myanmar language.

References

[1] E H. Spafford, OPUS: Preventing Weak Password Choices Computers and Security,1992.

[2] H Singh Dhillon.,"Second Order Markov Model Based Proactive Password Checker", Department of Electronics and Communication Engineering, IIT Guwahati , India.

[3] H H Htay, "Myanmar Word Segmentation using Syllable level Longest Matching", Kavi Narayana Murthy,Department of Computer and Information Sciences,University of Hyderabad, India

[4] C Antognini, "Bloom Filter", Trivadis AG, Zurich, Switzerland,2008.

[5] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In NSPW '09: Proceedings of the 2009 Workshop on New Security Paradigms Workshop, pages 133–144, New York, NY, USA, 2009. ACM

[6] F Bergadano, B Crispo, and G Ruffo,"*High Dictionary Compression for Proactive Password Checking"* ACM Transactions on Information and System Security, Vol. 1, No. 1, November 1998.

[7] M Waldvogel, "*Fast Longest Prefix Matching: Algorithms, Analysis, and Applications*", Swiss Federal Institute Of Technology, Zurich, 2000.

[8] B. Bloom. "*Space/Time Tradeoffs in Hash Coding with Allowable Errors*." Communications of the ACM 13:7 (1970), 422—426.

[9] G Varghese, Network Algorithms, Lecture 4: "*Longest Matching Prefix Lookups*",2011

[10] Y K Thu and Yoshiyori Urano. 2006. Text entry for myanmar language sms: Proposal of 3 possible input methods, simulation and analysis. In Fourth International Conference on Computer Applications, Yangon, Myanmar, Feb.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: "*A scalable wide-area web cache sharing protocol*", IEEE/ACM Transactions on Networking, 8(3):281{293, June 2000.

[12] K Scarfone ,M Souppaya, "*Guide to Enterprise Password Management (Draft)*", Recommendations of the National Institute of Standards and Technology .