A PRACTICE TO CREATE USER FRIENDLY SECURED PASSWORD USING CFG

S.VAITHYASUBRAMANIAN, A.CHRISTY

Abstract: Passwords afford the first line of security against illegitimate admittance to computer. Password complexity is a double edged sword. Complex passwords are hard to crack but equally hard to remember. Simple passwords are easy to remember and crack!! However the average user tends to use simple password and more often than not the same password for different logins. This makes them vulnerable to various types of cyber attacks. In this paper we have given a method of creating user friendly password using CFG (context free grammar). Which follows a pattern and user can remember easily.

Keywords: Password – Vulnerabilities - Cracking - CFG

Introduction: Password is as a secret word, a phrase, or combination of miscellaneous characters that authenticates the identity of the user. Usually Passwords are the first and possibly only defense against intrusion. Username ascertains the identity of the user for the computer, while the password, known only to the authorized user, authenticates that he or she the user is claims to be [6]. A typical computer user has passwords for many purposes: logging into accounts, retrieving e-mail, accessing applications, databases, networks, web sites, and even reading the morning newspaper online [2].Passwords are essential in the computing world as they form the first line of defense against all unauthorized intrusions. From using a password to sign in to your operating system to passwords for various transactions on the Internet, passwords can be classified as a “necessary evil”. The existing password protocols are highly inconvenient from a user's point of view when basic security rules are followed, and highly insecure if made convenient, for instance by re-using the same password on all services and devices [3] [8]. A simple mantra to be followed would be that “The stronger the password, the more protected your computer will be from hackers and malicious software”. Users should ensure that they have strong passwords for all web based transactions. However, it is to be borne in mind that passwords are dependent on the human ability to remember. Human Memorability is a major influencing factor in designing passwords. The human mind usually tends to revert to comfort zones and therefore ensures that most passwords are simple and easy to remember. Keeping this in mind / as a base we seek to explore ways to create user friendly, strong passwords using CFG which follows a pattern [10].

Common password vulnerabilities: Normally the user creates password in such a way that they can remember it easily. A difficulty in managing their passwords makes them to create weak password and reuse of common password for multiple accounts. The easier a password is for the user to remember generally means it will be easier for an attacker to crack [4] [7] [8]. While creating passwords they use one among dictionary words, family names, dictionary words in combination with numbers, common misspellings, words spell backwards, their favorites, their personal information, common letter to symbol conversions and obvious password like password, friends. In case if they use numbers in addition mostly it will be birthday, license number, passport number, vehicle number or sequence of numbers [5] [7].

3.Context-free grammar: The formalism of context-free grammars was developed in the mid-1950s by Noam Chomsky [1]. A context-free grammar (CFG) is a set of recursive rewriting rules (or ***productions***) used to generate patterns of strings. Context-free grammars are important in linguistics for describing the structure of sentences and words in natural language, and in computer science for describing the structure of programming languages and other formal languages. In linguistics, some authors use the term phrase structure grammar to refer to context-free grammars, whereby phrase structure grammars are distinct from dependency grammars. In computer science, a popular notation for context-free grammars is Backus–Naur Form, or BNF [9].

3.1 Formal definition: A Context free grammar G is defined by 4-tuple [1] G = (V, T, P, S) where

1. V is a finite set; each element v ε V is called *a* non-terminal character or a variable. Each variable represents a different type of phrase or clause in the sentence. Variables are also sometimes called syntactic categories. Each variable defines a sub-language of the language defined by G.
2. T is a finite set of terminals, disjoint from V, which make up the actual content of the sentence. The set of terminals is the alphabet of the language defined by the grammar G.
3. P is a finite relation from V to (V U T)*, where the asterisk represents the Kleene star operation. The members of P are called the rules or productions of the grammar.

4. S is the start variable (or start symbol), used to represent the whole sentence (or program). It must be an element of V.

3.2 Generation of strings: To generate a string of terminal symbols from a CFG, we:

- Begin with a string consisting of the start symbol;
- Apply one of the productions with the start symbol on the left hand size, replacing the start symbol with the right hand side of the production;
- Repeat the process of selecting nonterminal symbols in the string, and replacing them with the right hand side of some corresponding production, until all nonterminals have been replaced by terminal symbols.

Procedure to create password using context free grammar: The following illustrations generate pattern of string which forms context free language. The user can choose among them and can set as their password. Here are steps to create a CFG password. Create your password of length eight or more since most website sets the requirement criteria of minimum password length. Form them by various combinations. To make it complicate use upper case, lower case, numbers and special symbols. Change them often to make it effective and use distinct password for different login.

Here are few Illustrations of creating strings using Context free grammar.

Illustration: 1

A context-free grammar for syntactically correct infix algebraic expressions in the variables x, y and z:

S → x | y | z | S + S | S - S | S * S | S / S | (S) | [S] | {S}

Illustration: 2

A context-free grammar for the language consisting of all strings over {a, b} containing an unequal number of a's and b's:

S → A | B; A → CaA | CaC; B → CbB | CbC; C → aCbC | bCaC | λ

Here, the nonterminal C can generate all strings with the same number of a's as b's, the nonterminal A generates all strings with more a's than b's and the nonterminal B generates all strings with fewer a's than b's.

Illustration: 3

A Context free grammar for the language L = $\{b^n a^m b^{2n}: n \geq 0, m \geq 0\}$ over {a, b}.

S → bSbb | A; A → aA | λ

Illustration: 4

A CFG describing strings of letters with the word "main" somewhere in the string:

S → <Letter*> m a i n <Letter*>; <Letter*> → <Letter> <Letter*> | epsilon; <Letter> → A | B | ... | Z | a | b ... | z

Illustration: 5

A Context free grammar for the language L = $\{ww^R: w \{a, b\}^*\}$ over {a, b}.

S → aSa | bSb | λ

Illustration: 6

A Context free grammar for the language L = $\{wcw^R: w \{a, b\}^*\}$ over {a, b,c}.

S → aSa | bSb | c

Illustration: 7

A Context free grammar for the language L = $\{a^n b^m a^n \mid n > 0, m > 0\}$ over {a, b}.

S → aSa | aAa; A → bA | b

Illustration: 8

A Context free grammar for the language L = $\{a^n b^m c^m d^{2n} \mid n \geq 0, m > 0\}$ over {a, b, c, d}.

S → aSdd | A; A → bAc | bc

Illustration: 9

A Context free grammar for the language L = $\{0^i 1^j 2^k \mid i \neq j \text{ or } j \neq k\}$ over {0, 1, 2}.

S → AC | BC | DE | DF; A → 0 | 0A | 0A1; B → 1 | B1 | 0B1; C → 2 | 2C;
D → 0 | 0D; E → 1 | 1E | 1E2; F → 2 | F2 | 1F2

Illustration: 10

A Context free grammar for the language L = $\{a^n b^m c^k: k = n + m\}$ over {a, b, c}.

S → aSc | B; B → bBc | λ

Conclusion: In this paper we recommend a new way of creating password using context free grammar. Our approach can be effectively and securely used as user friendly evidence mechanism for their web logins. To a great extent further research and user studies are required for CFG password techniques to complete higher levels of maturity and utility. A strong password does not guarantee 100% protection from hackers. However, a strong password / robust password system is an effective deterrent against 90% of the commonly used modes of attack.

References:

1. Hop croft, John E.; Ullman, Jeffrey D. (1979), Introduction to Automata Theory, Languages, and Computation, Addison-Wesley. Chapter 4: Context-Free Grammars, pp. 77–106.
4. Ashlee Vance, "If your password is 123456, just make it HackMe" Technology –The New York times, 2010.
2. http://resources.infosecinstitute.com/dictionary-attack-using-burp-suite.
3. http://www.ghacks.net/2013/10/26/4-simple-password-creation-rules-x-common-sense-tips/
5. http://www.microsoft.com/security/online-privacy/passwords-create.aspx.
6. Sarah Granger, "The Simplest Security: A Guide To

Better Password Practices"http://www.symantec.com/connect/ articles, July 2011.

7. depts.washington.edu/engl/help/pass.html.
8. Bander AlFayyadh, Per Thorsheim, Audun Josang and Henning Klevjer "Improving Usability of Password Management with Standardized Password Policies" The Seventh Conference on Network and Information Systems Security - SAR-SSI 2012 Cabourg, May 2012 ISBN 978-2-9542630-0-7.
9. Akshay Kanwar, Aditi Khazanchi, Lovenish Saluja "Analyzing Ambiguity of Context-Free Grammars" International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 2 Issue 10 October, 2013 Page No. 2921-2926.
10. Jeff Yan, Alan Blackwell, Ross Anderson, Alasdair Grant "Password Memorability and Security: Empirical Results" IEEE security & privacy Volume: 2, Issue: 5, 2004, Page No. 25 – 31.

* * *

Research Scholar, Research Supervisor,
Sathyabama University, Chennai, India.
discretevs@gmail.com,ac.christy@gmail.com