Securing Password Using Dynamic Password Policy Generator

Areddy Rekha, K.Rubiya Alam

1PG Scholar, Department of CSE, PVKK Institute of Technology-Anantapuramu
2Assistant Professor, Department of CSE, PVKK Institute of Technology-Anantapuramu

ABSTRACT: It is very difficult to keep password users from creating simple and common passwords; generally the application and websites provide a measurement which is also named as password checker or password strength measure. For every critical password checking requirements need password checker to be stringent have prevailed in the study of password security, we can do that regardless of the stringency, such as there might be always a risk for password checkers like static checkers can leak information and also help the adversary to enhance the performances of their attacks. To increase the strength of the static password checker, introduce the Dynamic Password Policy Generator, namely DPPG, to be an effective and usable alternative to the existing password strength checker. The proposed method DPPG aims about dynamic rules to generate dynamic password policies and gives opportunity users to create passwords that are diverse and that contribute to the overall security of the password database. Since DPPG follows metrics for policy generation and can function like modular. Further keep an extension to introduce a diversity-based password security metric that evaluates the security of a password database in terms of password space and distribution. The metric can be very useful as a counter measure to a well written code for offline cracking algorithms and theoretically illustrates why DPPG works well.

INTRODUCTION

Generally, widely used password is Text based for both online and offline applications for decades. Since passwords are personal and portable, they are not likely to bare placed in the foreseeable future. However, the methodology for that people choose simple passwords and reuse common passwords so that they should remember it easily, have raised great security concerns as such passwords are vulnerable to offline cracking attacks. To make things worse, a number of password leak incidents have happened recently and frequently [6]. Large datasets of leaked passwords can greatly enhance attackers’ capability in conducting training-based password attacks, thus posing significant threats on password security. The most direct and pervasive protective mechanism used by major websites and applications designed as the password strength checker, which evaluates the strength of passwords proactively during user registration. While the goal is to guide users to create strong passwords, in previous work, the lack of accuracy and consistency in the strength feedback has been widely observed and examined. That is, existing checkers do not demonstrate effective or uniform
characterization of strong passwords. Furthermore, the space for the rules and policies of the checkers to be stringent is very limited as researchers have shown that the complexity of a password is a trade-off with the usability [7]. Password strength checker itself can be leaked which has not researched in the studied done. By defining a set of password creation policies and showing users password strength scores, password checkers can exert a strong bias on password characteristics [8].

When the policies and scoring mechanisms remain static. Although the common things in password checker, they always rely on similar rules that focus on specific password properties (e.g., length, number of digits and special characters). When rules are relaxed, password users can create simple passwords by following the common distribution policy.

When rules are relatively demanding, the password distributions closely correlated to the scoring metrics and can be inferred. Since the password checkers are publicly available, attacker scan easily make use of the password checkers to learn the password characteristics distribution that is shaped by the password checkers.

**ISM Security Policy:** It is required for ISM security policies cover all areas of security, be appropriate, meet the needs of business and should include the policies shown in the following diagram:

![ISM Security Policy](image)

**LITERATURE SURVEY:**

**Passwords: If We’re So Smart, Why Are We Still Using Them**

There is a changed in the last 10 years in Internet security, a lot has gained the same use of alphanumeric passwords. Passwords dominant means to authenticate the internet, even they face the significant problems related to password forgetting and theft. After a research we found large numbers of proposed alternatives, we need to remember our passwords that ever before [9]. As we have a large number of proposed alternatives, now we need to remember more passwords than ever before. Do we need alphanumeric passwords still to be unique in 2019, or going to adopt the alternative proposal which should be commonplace? This
raises so many questions, following a panel discussion at Financial Cryptography and Data Security 2009. Since many years Passwords have server as well but then also, they suffer from a number of problems that suggest their reign should be coming to an end. Users’ mindset is always to choose weak passwords, making guessing and brute-force dictionary and exhaustive attacks feasible [10]. Users also gets so many unnecessary customer support calls and even automated backup authentication schemes which involves some challenging queries, which weaker forms of authentications. Because of the above challenges users always tries to store copies of passwords in different places which can be leaked and use the same password for multiple systems. Users also went through passwords stolen by phishing, social engineering, man-in-the-middle techniques. The password with static natures allows the repeated unauthorized access by attacker. The popular stories about identify and fraud, and there appears to be increment of awareness, even the unsophisticated users. As economic gain has emerged as a primary motivation for computer security exploits, there should be increased motivation to move beyond simple passwords. On the other hand, despite these signs of real need and a desire for change, adoption of authentication alternatives has been very slow.

In this note we consider possible reasons why we are moving so slowly in replacing problematic password systems, how we might accelerate the progress, and where we might be in ten years. Rather than focus on the specifics of particular technologies, we prefer to consider forces that drive or retard progress, including technology, economics, and usability. For example, graphical passwords (e.g., see Chiasson for a recent survey) offer the possibility of improved strength, memorability, and usability.

SYSTEM ANALYSIS:
Existing System: In existing system, the text-based passwords have been used widely in both online and offline applications for decades. Since passwords are personal and portable, they are not likely to be replaced in the foreseeable future. However, the phenomenon that people choose simple passwords and reuse common passwords has raised great security concerns as such passwords are vulnerable to offline cracking attacks. To make things worse, a number of password leak incidents have happened recently and frequently. Large datasets of leaked passwords can greatly enhance attackers’ capability in conducting training-
based password attacks, thus posing significant threats on password security. The most direct and pervasive protective mechanism used by major websites and applications is the password strength checker, which evaluates the strength of passwords proactively during user registration. While the goal is to guide users to create strong passwords, in previous work, the lack of accuracy and consistency in the strength feedback has been widely observed and examined.

**Proposed System:**
In this proposed system we evaluate the impact of misusing current commercial password strength checkers from the attacker’s perspective and explore the possibility and potential to leverage the checkers in offline cracking attacks. Using an attack-based model, we show that the password checkers are effective for attackers to facilitate password cracking. With a certain amount of computational power, the attacker can compromise more passwords with a specific rating with the help of the strength checkers. This implies that the static policies and scoring mechanisms used by password strength checkers exert bias on the password characteristics distribution. Passwords with the same rating follow an obvious pattern which can be exploited by the attacker to refine the training data. We also proposed a countermeasure to protect the information on password distribution and to reduce the efficacy of well-crafted training-based attacks; we devise the Dynamic Password Policy Generator, namely DPPG, which generates dynamic password policies for users. Each new user obtains a different password policy to follow, which is generated in real-time from the server based on but not reflecting the current password distribution.

**SYSTEM ARCHITECTURE:**

![Fig-2: System Architecture](image)

**IMPLEMENTATION:**

**Modules:**

**COMMERCIAL PASSWORD CHECKERS**
Traditional password policies have become less popular as the more user-friendly password strength checkers become widely adopted by major websites and software. The main reason is that good password policies can easily be too stringent to use, while password strength checkers push users to
create “strong” passwords subtly. However, most of the existing research only evaluates the effectiveness and helpfulness of the password strength checkers.

The fact that the checkers are based on unchanged policies which indirectly bias the password characteristics distribution has not been studied. Furthermore, due to the exposure of the policies and scoring mechanisms, careful attackers can utilize the password checkers to mount more powerful attacks on passwords with high strength ratings.

**Dynamic Password Policy Generator:**
One could argue that a potential solution to the password checker limitations is to have better web technologies to hide the policies and detect malignant password strength querying. However, it can result in delay in strength feedback and high false-positive rates in detection. Therefore, we take another approach to the problem and explore the feasibility of providing dynamic password policies to users.

**PASSWORD DIVERSITY**
In this module, we propose to measure the strength of a password dataset in terms of password distribution, by Comp is the number of different character types used in the password. L, U, D, and S represent lower-case characters, upper-case characters, numerical digits, and special characters, respectively. In password policies and checkers, Comp is also a popular measure. In previous analysis, it is shown that requiring more character types reduces usability of the passwords.
RESULT ANALYSIS:

Fig-6: Admin Page

CONCLUSION:

The password space and distribution to understand password dataset security better. Due to the limitation of existing strength measuring mechanisms, we propose a new and usable alternative based on an effective diversity metric to better protect passwords from offline cracking attacks.

Computed time complexity of the algorithm, and it is found that the algorithm works fast. Since the algorithm generates password policies dynamically, it will be challenging for the attacker to guess the characteristics of the password database.

This algorithm also increases password space as most of the special characters like ! % $ # never used in password creation. Including these characters in the password will make the password more complex and more difficult for attacker to crack. The objective of this password policy generator is to secure the existing websites which are vulnerable to various attacks such as brute-force, rainbow table attack, dictionary attack, etc.

Generating password dynamically make the password secure and then hashing it with the enhanced hashing algorithm (like PBKDF-2, Bcrypt, Scrypt, etc.) make the cracking of the password more difficult even
if the attacker has breached the server and compromised the database.

An algorithm which detects the frequency of the characters and then generate the password policy accordingly, it will be very difficult for the attacker to analyze the characteristics of the password distribution in the database. Therefore, it will be very costly for an attacker to crack those dynamically generated passwords. Thus, our dynamic password policy generator can be used to diminish the threat of offline attack.

REFERENCES: