Passphrase and keystroke dynamics authentication: Usable security

This is a research that works along the same lines of concept as the last paper discussed. It deals with profiling a user based on metrics such as keystroke dynamics, in order to act as a layer of authentication in a Multi Factor Authentication System. The motivation for this paper comes from the fact that conventional password based systems are not sufficient anymore considering the level of proficiency cyber criminals have risen to, along with computing resources. Security policies were used to increase security, however often due to the way humans use it, it ends up acting in a counter productive way decreasing the overall level of security. The advantages of the two factor authentication system over the basic one layered password system is that, they would be resistant against attack techniques such as phishing & social engineering, dictionary attacks, brute force, keyloggers, shoulder surfing, database attacks, etc.

As the first layer of security, the authors suggest using passphrases instead of passwords. A passphrase can be defined as a sequence of words, in the context of the paper normally in lower case alphabets with more than one word and a minimum of 16 characters. However the definition for this in the real world normally include abbreviations and symbols.

As the second layer of security, the authors use keystroke dynamics. As part of the keystroke dynamics, the authors measure Dwell time, flight time and pressure of the keystrokes. Based off these attributes which form a user's typing pattern the user is authenticated which forms the second layer of authentication. The keylogging software that runs can be of 3 different types -

1. Static - Runs on a specific page
2. Non-static - Continuous logging of all system interactions
3. Semi-static - A mix of static and non-static with logging done in specific time intervals

The difference should be noted between authentication and identification. For identifying, the user would be located using his keystrokes from a list of users and their user profiles. For authentication, finding a user from a collection of users is not necessary. Instead, the profile is just compared against the stored profile of the user they claim to be. In order to see if such an authentication system would be able to authenticate the user, the tools and techniques used were the Shannon Entropy Theory, the Chunking Theory and the Keystroke Level Model.

The Chunking theory deals with how much chunks of information can a person hold in his short term memory. Chunking itself deals with these chunks and their organization along with the relation between them. This is the reason why passphrases are better than passwords in terms of usability. The users would be able to have personal associations with the passphrases and hence would take less chunks to remember the phrase than a password. On the other hand a password that is under strict security policies would have less meaning and hence would require more chunks for the person to remember, reducing usability. The Keystroke Level Model is a tool that is used to predict how long it takes a user to carry out a task in terms of the number of keys pressed, the mental preparation required, etc. Shannon's entropy quantifies the amount of information in a variable, which acts as a metric to test the strength of the authentication metric such as passwords. passphrases, etc. The more the entropy, the stronger it is, and the harder it would be for attackers to bruteforce the key.

The paper was also reviewed by experts in the field who gave their feedback about the solution proposed by the paper. Eight out of ten experts are of the opinion that the solution would enhance both security and usability. Also, it was suggested that the keystroke dynamics would not be sufficient on it's own and hence would need to be supplemented with other factors. Also, the people who are adopting this solution should decide the leniency of the approach based on the context.

Discussion

- Human beings, by nature, try to reduce the effort possible. Hence they consistently pick easier passwords which was why password policies needed to be enforced. However, in the constraints enforced by the password policies, there are patterns when humans try to reduce the effort. For example, if there is a password policy that there should be a capital letter in the password, then in majority of passwords the capital letter would be the first letter. This is an added piece of information that the attacker can leverage, that was not there before the policy was enforced. Similarly when there is a requirement for a number to be in the password, there is a good chance that the number would be 123 at the end of a regular password.
- The reason why passphrases are suggested by the authors over passwords, because typically passphrases would be of more length than passwords. So the argument is to use passphrases instead of enforcing password policies on passwords that would lead to more predictable patterns.
- Furthermore, the paper's strategy is to cover for the low entropy in passphrase/password section, with a high entropy layer formed by the keystrokes.
- When keystroke dynamics are included as part of the pipeline, with it being invoked in certain circumstances, like say, based on location, it introduces further issues. The attackers would always try to utilize these circumstances to bypass the more difficult parts of the pipeline. For example, if a biometric authentication system has a backup authentication that can be invoked by the admin, the attackers would be looking for ways to go through the backup authentication instead of focusing on trying to bypass the biometric system.
- One major assumption in the paper is that they focus only on bruteforce attacks and not on other attacks such as dictionary attacks. The entropy that they define doesn't take into account the characteristics of a language. For example, some letters in the English alphabets would be more frequent compared to others, and hence the entropy would depend on the usage of such letters among other such factors. However, the authors are focusing on bruteforce attacks where all possible combinations and permutations are tried out. Moreover, the definition of a passphrase by the authors is also very arbitrary with it including only lower case alphabets when in reality it could have a mix of numbers, symbols and capital letters.

- While on the topic, it also helps to revise how passwords are stored. The accepted way for authentication is to never store the password as it is in the server, but to store its hash which is a one-way mathematical function. In this form, even if the server is compromised or if the server has a malicious admin, the passwords of the users are not compromised. This makes attacks like bruteforce and dictionary attacks the only forms of attacks for the attacker, even if he manages to get his hands on the password hash. To make this process easier, the attackers used rainbow tables which are pre-hashed values so that it can be compared against the hash they managed to acquire. To combat this, the defenders started using salted hash which is basically a random string appended to the password before hashing it. It is interesting to note, that this salt is not kept a secret. However, it is still enough to ensure that the attacker would have to hash each entries while bruteforcing instead of using a prehashed rainbow table.
- However, the more common method used by attackers is a dictionary attack and not pure bruteforce, which brings down the possibilities further. The attackers use information acquired about the target from mediums like social media to construct specific wordlists focusing on more probable words and their combinations.
- Another form of attack discussed was typosquatting where very similar words to legitimate services are used to deceive users who make a typo. For example, facebok.com instead of facebook.com. There are new dimensions to the typosquatting attack with technologies like voice detection virtual assistants, like Alexa. For example, the attackers can create fake services that trigger on words that are very similar to the triggers for legitimate services.
- Yet another form of attack discussed was side channel attacks, where things like electromagnetic fields, heat, etc are utilized to identify the activity in a system, maybe even to the extent of the keys pressed. However, most of these kinds of attacks are not practical unless you have high end equipment like maybe military-level ones.