International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

Honeywords, an Enhancing Password Security Mechanism: A Comprehensive Survey

[1]Shahad A. Saadullah, [2]Saja J. Mohammed

[1]Student, Dept. of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
[2]Professor, Dept. of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq

***Abstract* - With the increasing computer usage, the increase in threats to digital systems has led to the need to protect these systems by proposed a new technology to protect the system against hackers. So, there was a need to protect users' passwords and provide them with reliability and security. So-called honey words were used, which are fake words that are combined with real passwords side by side in the authentication database. Honeywords are a popular technology in the digital world and are important for enhancing the security of users' real passwords and are considered a strong additional security layer. The advantage of this technology is that it detects attempts to access systems without authorization. This technology provides high protection that confuses the attacker between which words in the database are real and fake. This technology increases the difficulty and complexity of hacking by attackers and enhances system security. Furthermore, it provides an effective strategy to mitigate the risks associated with password breaches, including dictionary and brute force attacks, thus improving the overall system resilience. This paper offers a comprehensive survey of the honeywords and their generating algorithms. The paper can be used as user help to give proper and adequate information about this subject.**

Keywords: Honeywords, Honeywords Generation, Honey Checker, Passwords Security, methods.

I. INTRODUCTION

In recent years, the world has witnessed a significant digital transformation, leading to the digitization of most activities and the exchange of data and information over networks. This has greatly improved speed and efficiency; however, it has also introduced new security risks, particularly the potential loss of user data to unauthorized parties or attackers. To enhance security, unique identifiers or passwords are assigned to each user called passwords, but attackers can still employ various techniques to steal these credentials database [1][2][3].

Password security is a critical issue; therefore, it is essential to enhance and improve password generation over time. Password protection mechanisms help safeguard information from unauthorized users. An attacker who has stolen a file of hashed passwords can employ brute-force techniques to search for a password "p" whose hash value "H(p)" matches the stored hash value of the user's password. This enables the attacker to impersonate the user [3].

Password protection helps safeguard information from unauthorized users. The exposure of password files is a serious security issue that has impacted many users and companies such as Yahoo, LinkedIn, eHarmony, and Adobe, as leaked passwords can lead to numerous potentials cyberattacks. These breaches have revealed that many large organizations use weak hashing techniques, making it easier for hackers to crack user passwords [4]. To effectively mitigate these security challenges, two fundamental considerations must be considered during the development of a secure system. Firstly, it is imperative that passwords are stored within the database employing a strong and suitable hashing algorithm to significantly hinder the ability of malicious actors to deconstruct the hashes. Secondly, it is crucial that breaches of password files are identified in a timely manner to facilitate appropriate remedial actions [4].

One of mechanisms for password secrecy is using a unique password called (Honeywords) and it’s meant a decoy password (fake passwords) added to each account entry in a credential database. The principle behind honeywords is that since the legitimate user does not know the honeywords generated for her account, the only party who is able to enter those honeywords is an attacker who discovered them by breaching the credential database. As such, login attempts using honeywords should be taken as compelling evidence of a database breach, The mechanics of honeywords can be explained as follows [1][2][5]. Where the scenario changes when an attacker tries to breach the database. Suppose the attacker successfully invers the hash protecting the passwords in the database. This provides the attacker with a list of passwords, but they won’t know which ones are valid and which are honeywords. If the attacker submits a password designated as a honeyword, the honey checker will trigger an

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 134-142, January-2025

https://doi.org/10.47001/IRJIET/2025.901017

alarm and alert the administrator. This method is highly effective at detecting database breaches because of the nature of honeywords [1][2][6].

II. Honeyword and Honey Checker

As explained above, Honeyword is a mechanism for password secrecy is using a unique password called (Honeywords) and it's meant a decoy password (fake passwords) added to each account entry in a credential database Honeywords are not simple, easily hackable passwords and are rarely submitted by legitimate users. Therefore, if a honeyword is used as a login password, it strongly indicates that the password hash file has been stolen, due to their infrequent use when properly chosen [1][2][6]. By utilizing honeywords, it provides a realistic and cost-effective method for enhancing security. Additionally, generating honeywords does not require significant storage space. Activating an alert mechanism to notify administrators in the event of a breach further improves cybersecurity measures [7].

The system can include an auxiliary secure server called the "honey checker" to support the use of honeywords. Given that the computer system is susceptible to having its file F of password hashes stolen, it is also vulnerable to having salts and other hashing parameters stolen. Consequently, there is no secure location on the computer system to store additional secret information to thwart adversaries. The honey checker, a separate hardened computer system, addresses this by securely storing such secret information [8][3][9].

2.1 Honey checker working mechanism

A honey checker is a secure assistant server that helps in using honey words. The honey checker should be equipped with extensive hardware to detect anomalies of various types. It is also capable of raising an alarm when any unauthorized access is detected [8][2][9].

Honey checker works with some mechanisms:

a) Communication: The primary computer system communicates with the honey checker during login attempts and when users change their passwords. This communication is secured via dedicated lines and/or encryption and authentication protocols to ensure its integrity and confidentiality.

b) Anomaly Detection and Alarm System: The honey checker is equipped with extensive instrumentation to detect various anomalies. It can raise an alarm upon detecting irregularities, notifying an administrator or another party separate from the primary computer system. This alarm can be immediate or a "silent alarm" that allows the login attempt to proceed while alerting administrators discreetly.

c) Response: When a specific user tries to log in, the honey checker informs the system that there is an unauthorized entry attempt through a specific warning or informs him of this unauthorized entry to stop the system completely. This is called response with reaction. However, if it gives a notification without stopping the system, this is considered a response without reaction, and this depends on the policy chosen for the system [8][3][9].

III. Honeyword Generation Methods

Honeywords generation methods are divided based on whether they affect the user interface [10]:

3.1 Legacy-User Interface procedures

Old UI methodologies are especially considered to be the methodologies for generating fake words or what is called fake words. This application uses the general encryption technique, which is of several types that will be explained above. As explained above, the fake passwords that it generates are stored with the real password in the authentication database. Among these details is the encryption technique by modifying the last letters or what is called the word's suffix or encryption by modifying the first number in the word [6][3]. The alternative may be either a letter by a letter or a number by a number or a symbol by a symbol and so on. These electronic guarantees have been applied to precautionary alert notifications for the security of real knowledge words, which leads to the risk of those who reach the application word and their application in confusion between which words are real, and which are honeycombs. An increase in the creation of Greek words makes it difficult to decode the basic word [6][3]. It is important to realize that although old UI methodologies are secure, they do not require modifications to the current UI designs [6][3][11]. There are three types of legacy UI:

3.1.1 Chaffing by tweaking

This method is one of the methods used to protect systems and improve their security through several techniques applied to original passwords so that they do not affect the basic data but increase their complexity and difficulty for attackers to guess by adding confusion to the word [6][3].

a) Chaffing-by-tail-tweaking

This method is used to generate "honeywords," where it manipulates the tail of the original password to generate a decoy (honeyword). A honeyword is like the original password but differs slightly in its final portion. This method

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

focuses on modifying the letters, numbers, or symbols located at the end of the password. The modification can be simple or complex, ensuring that the generated honeywords are logical and resemble common passwords [3][10][11][12].

b) Chaffing by tweaking digits

This method relies on modifying the numbers in the real password by replacing each digit with another, without altering the rest of the password's letters or symbols. The modification can either involve changing the digits themselves or simply adjusting their positions [3][11][13] [12].

3.1.2 Chaffing with a Password Model

This security concept is based on modification through what is called chaffing and depends in generating honey words on taking a probabilistic model of real passwords [14].

a) Simple model

It operates by mixing real data (digitally signed) with fake or "chaff" data that is either unsigned or contains false signatures, essentially employing a probabilistic model to obscure the real data within. Only the person possessing the correct password or key can distinguish the genuine data from the chaff. The core concept is to conceal the real data within random noise, making it difficult to differentiate [10][15].

b) Modeling syntax

This is a method for generating honeywords, and it is considered a robust approach where letters are substituted with symbols that resemble their shape, making the method more complex and harder to guess, yet easy to remember due to its similarity to the original word [10][18].

The substitutions can be made as follows:

'A' and 'a' are replaced by '@','E' and 'e' are replaced by '3','S' is replaced by '5' or '$','l' is replaced by '1','O' is replaced by '0', and so on [11][15].

3.1.3 Chaffing with "tough nuts"

This method generates honeywords that are more challenging for attackers to detect, as they consist of long, random, and fake strings. These strings are designed to be unbreakable, even with advanced techniques like brute-force attacks. In this approach, the system deliberately includes special honeywords, referred to as tough nuts, which are nearly impossible to reverse-engineer from their hash values. For example, the hash of a honeyword could be set as a fixed-length random bit string, making inverting it computationally infeasible. The number and placement of tough nuts are chosen randomly to prevent the adversary from compromising the entire set of sweet words. As a result, some sweet words remain unknown to the attacker, which could deter them from proceeding with their attack. In such cases, it is believed that the adversary might hesitate or delay attempting to log in with cracked passwords [4][15].

Overall, legacy-UI procedures present a pragmatic strategy for the implementation of honeywords without necessitating substantial alterations to pre-existing systems. They can serve as an effective deterrent against various forms of attacks by amplifying the risk and detection likelihood for illicit access attempts [4].

3.2 Modified-UI Procedures for Password Changes

In an endeavor to enhance the robustness of password management protocols, various improvements have been instituted within the user interface (UI) designated for password modification. A particularly significant enhancement relates to the evolution of honey-word generation, wherein a stochastic value is amalgamated with the user's original password. This approach aims to generate a new password that projects an elevated perception of security [4]. The user interface functions to educate the individual about the incorporation of honeywords or interacts with the user in a manner that may affect the selection of a suitable password [13].

a) Take-a-tail

The take-a-tail methodology bears resemblance to the chaffing-by-tail-tweaking methodology. Nevertheless, this methodology diverges in the selection of the tail; specifically, the tail of the newly generated password is randomly selected by the system and is required for the user to input a new password [13].

b) Random pick

This technique requires the user to provide multiple passwords, after which the system randomly selects one of those passwords to serve as a sugar word, subsequently notifying the user of this selection, while the remaining passwords are classified as honeywords [13].

3.3 Hybrid model

This approach entails the amalgamation of the advantages presented by various methodologies for honeyword generation, such as the chaffing-with-a password framework and the chaffing-by-tweaking-digits technique. For example, should the initial password be designated as apple1903, the honeywords angel2562 and happy9137 may be generated as derivatives for the chaffing-by-tweaking-digits process [3][4].

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

IV. The Artificial Intelligent Algorithms

Artificial intelligence algorithms are considered one of the latest methods used to generate honey words that are characterized by their diversity and difficulty for attackers to distinguish them. These algorithms are simulated, and their method is understood to be able to use them in generation. Among the most prominent algorithms are:

4.1 Bees Algorithm

This algorithm adopts a novel approach to generating honeywords, inspired by the natural foraging behavior of bees [8]. Bees' algorithm operates as follows:

Step 1: Initialization:

- Scout bees are deployed to explore the solution space, searching for potential solutions.
- Elite bees are recruited to search for optimal solutions among those collected, with priority given to the solutions that exhibit the highest frequency of occurrence.
- Stopping criteria for the optimization process are established, where each bee's fitness is evaluated based on the best nectar of the found flowers (the best solution found).
- Additional exploratory bees are recruited to search for new solutions.

Step 2: Exploration and Exploitation: The algorithm balances exploration (searching for new solutions) and exploitation (improving good solutions). Negative feedback mechanisms help discard weak solutions and promote the search for new, improved solutions [8].

Step 3: Honeyword Generation: Honeywords are generated based on the best solutions identified by the algorithm [8].

4.2 Salp Swarm Algorithm (SSA)

This algorithm employs an innovative methodology for the generation of honeywords, drawing inspiration from the natural world. It emulates the innate swarm dynamics of marine organisms classified as sea salps. The algorithm replicates the locomotion of salp chains within the marine environment as they forage for sustenance, where passwords are conceptualized as the nutrient source, and the produced honeywords progress towards them [9]. The swarms are stratified into leaders and followers. For example, in a cohort of 60 salps, 10 individuals assume the role of leaders, while the remaining 50 serve as their followers. The algorithm proposes four potential maneuvers for the(salps) : insertion, deletion, movement, and exchange. The leader evaluates these maneuvers, and the optimal action is executed by the followers. In each iteration, the (salp) exhibiting the least effective performance is supplanted by a newly generated (salp) selected at random [9]. If the input is "Pass123!" the algorithm operates as:

Step 1: Password Encoding:

The characters are represented using ASCII encoding. "P" = 80, "a" = 97, "s" = 115, "1" = 49, "2" = 50, "3" = 51, "!" = 33. The numerical representation is thus: [80, 97, 115, 115, 49, 50, 51, 33].

Step 2: Swarm Initialization:

A collection of initial solutions may be exemplified as follows: [79, 96, 114, 114, 48, 49, 50, 32] [81, 98, 116, 116, 50, 51, 52, 34] [78, 95, 113, 113, 47, 48, 49, 31]

Step 3: Iterations and Solution Improvement:

The numerical values are altered to facilitate the generation of novel solutions. Honeywords that closely resemble the original password are produced, albeit with slight variations to form deceptive passwords.

Step 4: Honeyword Generation:

Upon completion of multiple iterations, the resulting outputs are:

[79, 96, 114, 115, 48, 49, 51, 32] → "Oasr023".

[81, 97, 115, 116, 49, 52, 50, 33] → "Qast124!".

[80, 98, 113, 114, 50, 51, 49, 34] → "Pbqr251".

4.3 Meerkat Clan Algorithm (MCA)

This algorithm derives its inspiration from the social behaviors exhibited by meerkats, which are characterized by their cooperative and adaptive tendencies. The algorithm produces deceptive passwords, referred to as honeywords, to augment the security of authentic passwords. It organizes potential solutions into clusters or "clans" that collaborate to identify optimal solutions, with this categorization predicated upon a defined pattern inherent in the passwords [16].

Functionality of the Algorithm [16]:

Step 1: Segmentation: The password is divided into (letters, numbers and symbols) [16].

Step 2: Classification: The given password is classified based on several criteria such as the length of its letters and the pattern used in it, after that the users are allocated to specific clans [16].

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

Step 3: Character generation: Using the Meerkat clan algorithm, similar alphabetic characters are generated with the help of WordNet. Special symbols are also generated and modified.

Step 4: Merging: Merging the generated numbers, letters and symbols to form honey words.

In the end, diverse words appear that are more complex and less distinct than honey words [16].

V. Security Criteria

The most important criteria for evaluating the strength of pseudo-word algorithms are:

1. Flatness:

This criterion increases the difficulty of distinguishing honey words from the original word. This metric is used to measure the distribution of pseudo-passwords over the total space of real passwords [17].

2. Adaptability to denial of service:

This criterion evaluates the resistance of honey word generation algorithms to denial-of-service attacks [17].

3. Security against multiple system vulnerabilities (MSV):

The password may be repeated in systems, so this criterion measures the strength of honeyword generation algorithms in protecting against security breaches [17].

VI. Attacks

There are many attacks on security systems, and each attack has its own mechanism of action that differs from the other [3]. Among these attacks are denial-of-service attacks, which prevent users from accessing services, and another type is called brute force attacks, which decode the password by following repeated trials until we find another correct word [3][18]. Or attackers may use statistical pattern analysis to identify weaknesses within the security algorithms used, as shown in statistical attacks, among other forms of attack [19]. Therefore, understanding the inherent characteristics of each type of attack, as well as the operational mechanisms they use, becomes essential. This understanding serves as a prerequisite for designing more robust and adaptable systems capable of withstanding these attacks [18][20][21].

6.1 Denial-of-service

This offensive maneuver is designed to render a service inaccessible to its designated users, thereby highlighting a significant concern regarding the efficacy of the honeyword generation methodology. When honeywords are utilized for the purpose of authentication and a security breach is identified, the system typically adheres to its established protocols, which generally include the temporary suspension of the user's login privileges. An attacker can manipulate this situation by deliberately submitting a honeyword during the authentication process, thereby inducing a denial of service for multiple users without necessarily compromising the integrity of the entire database. This scenario becomes plausible when the generated honeywords are relatively straightforward to anticipate. Consequently, the Gen(k) algorithm emerges as a pivotal element in assessing the robustness of a system that implements the honeyword strategy [3][18][22].

6.2 Brute-Force Attack

In the preceding assault, it was observed that the imposition of a rigorous policy regarding honeyword detection could render the system susceptible to Denial of Service (DoS) attacks, consequently impacting the entire system. In contrast, a more lenient policy diminishes the efficacy of honeyword implementation. To elucidate this point, we present a scenario in which a malevolent entity can compromise multiple accounts under a permissive policy framework. Assume that this adversary has acquired a password repository denoted as F and has successfully deciphered a multitude of user passwords. Rather than concentrating on a particular account, the adversary endeavors to gain access to any account listed. It is also posited that the adversary possesses no statistical advantage in deducing the accurate password through the analysis of honeywords, thereby establishing that $Pr(g = pi) = 1/k$. Upon the entry of a honeyword, the system's response adheres to one of the subsequent illustrative policies:

- The login procedure continues as per the standard protocol.
- The user's account is suspended until the user establishes a new password.

The salient characteristic of these policies is that, even in the event of honeyword detection, the system elicits a minimal or localized reaction. As a result, an adversary is afforded the opportunity to conduct a brute-force search until a successful login is realized. For example, even if an account is suspended following the input of a honeyword, the adversary retains the capability to persist in login attempts with additional accounts [3][19][21][22].

6.3 Statistical Attack

Honeywords are susceptible to a specific type of attack known as statistical attacks, which exploit the statistical regularities or inherent properties of the methodology used to generate honeywords to distinguish them from the original

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

password. A key aspect of this type of attack lies in the attacker's understanding of the honeyword generation process, rather than merely focusing on the cryptographic strength of the system [20]. If attacker gains knowledge of the method or algorithm used to generate honeywords, this understanding can significantly reduce the effort and time required to identify the original password, thereby weakening the system's overall effectiveness. By analyzing statistically derived patterns, even without deep expertise, an attacker may identify honeywords that deviate significantly from the original password. Therefore, any algorithm or technique that uses a fixed method to generate honey words gives an opportunity to attackers to gain unauthorized access by taking the methodology or pattern of this technique or algorithm and discovering which words are the real passwords by continuously testing each word or by simply looking at the existing words [20]. In statistical attacks, there is what is called a warning that is given to system administrators when any security breaches occur. It is also of course given if the attacker deliberately uses a honey word if the attacker deliberately provides fake honey words, to drain all the system resources [20][23].

VII. Some of Existing Works

During the previous several years, a lot of research has suggested honeyword generating approaches as mentioned below:

In 2018, the research divided the honey password generation methods into two main groups It also discussed many generation methods for honey password generation such as "Tugh nuts" and "hybrid methods" in addition to emphasizing the importance of the generation sweet words by making the cracking of real password detectable [3].

In 2018, the scholarly investigation focused on the concept of "honey passwords," which was articulated by the eminent researcher Professor Ronald L. Rivest alongside Ari Juels from RSA Laboratories, as a strategic approach to mitigate security vulnerabilities within database systems. This initiative was designed to address the prevalent issue encountered by numerous systems that rely on weak passwords, which subsequently leads to various security breaches; consequently, honey passwords were conceived as a mechanism to enhance the detection of such breaches and to furnish system administrators with timely notifications and alerts concerning potential security incidents within their databases [1].

In 2018, the research presented advanced honeycomb generation methods, especially the "enhanced password" and "user profile" models, to improve, strengthen and enhance the security of the system. The effectiveness of these models against various attack vectors, including brute force attacks, was evaluated. The results of this research showed that the enhanced password model enhances security by updating the login frequency data, making it difficult for attackers to predict forbidden words. In addition, the use of a dataset with many unique passwords and the proposed methods maintained a high degree of flatness, which enhanced the security of passwords while reducing the memory burden on users. This research showed the urgent need to achieve a balance between security and ease of use in honeycomb password generation, considering user behavior and password selection patterns [21].

In 2019, scholars developed honey passwords through the implementation of chaffing techniques by modifying algorithms that altered the characters of the original password to generate deceptive passwords. Moreover, the research incorporated hashing methodologies employing the SHA256 algorithm, along with counterfeit files designed to mislead unauthorized users during hacking attempts. The findings of the study were promising, as honey passwords were successfully produced that were difficult to distinguish from authentic passwords [4].

In 2019, the research proposed a new approach to generate honey passwords by combining user personal information with dictionary attacks, worst password lists, and character mixing. The most important results of this research are improving and enhancing the security and protection of passwords compared to traditional methods and addressing the limitations that have emerged regarding current encrypted password technologies. The new approach has been tested with different types of passwords and has proven effective in enhancing password security and complicating the chances of attackers identifying real passwords and honey or fake passwords [14].

In 2019, studies focused on the Password-Based Key Derivation Function 2 (PBKDF2), and its implementation using HMAC-SHA-1, and to increase the complexity associated with implementing multiple attacks such as brute force or dictionary attacks, the integration of these components was carefully designed. Furthermore, performance and development metrics on the speed and efficiency of key derivation operations were developed using different methodologies, which led to good results [22].

In 2021, research focused on generating fake passwords with additional security measures, such as sending notifications and alerting administrators in the event of suspicious login attempts. Notifications are sent to the administrator regarding unauthorized access attempts, and the attacker is provided with decoy files, misleading them into

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

believing they have successfully breached the system. The aim of the research was to enhance the security of passwords and protect password databases. Its objectives were also to use fake honey passwords and reduce storage costs [6].

In 2022 the proposed honeyword generation method uses the bee's algorithm, a swarm intelligence optimization technique, to create honeywords by considering different password tokens (alphabet, digits, special characters). This method overcomes previous limitations, such as conditional flatness, DoS resistance, storage overhead, and user information security issues. Experimental results demonstrate that the method achieves perfect flatness and strong DoS resistance, improving the overall performance of the honeyword system. Future research may explore additional applications of the bee's algorithm in optimization problems.[8]

In 2022, an innovative honey word generation technique based on the Harmony Search Algorithm (HSA) was introduced, which improves the methodology of generating similar words while addressing previous constraints at the same time. The advantages of this algorithm are fast convergence, and the ability to balance exploration and exploitation within the scope of optimization challenges. HSA uses more advanced generators for alphabetic characters, while the generators of numeric and special characters are random as before. Another advantage is its ability to improve the non-discovery and guessing of the correct password accurately. This algorithm has improved the process of generating honey words and increased the difficulty of guessing real words [11].

In 2023, the research proposed a new approach that uses the meerkat clan algorithm, which is mainly based on the principles of collective intelligence and collaborative problem solving. This algorithm works by simulating the behavior of meerkats to enhance the effectiveness of generating honey words. It is worth noting that the research included WordNet in the process of generating fake passwords, which led to increasing the diversity of honey word generation and its reliability feature. This research also demonstrated clear results in the strength, security, and flexibility of these generated words, in addition to their strength in the face of multiple attacks [16].

In 2023 researchers conducted a comprehensive analysis of the most important algorithms used in generating honey passwords, and what are the strengths and limitations of each algorithm. Two distinct algorithms were highlighted: PassGAN and Honeychecker. PassGAN uses adversarial generation networks (GANs) in the process of generating honey passwords, which are fake passwords that are very similar to the original passwords. The proposed algorithm is trained on datasets containing real passwords and implements methodologies such as IWGAN training optimization, Wasserstein GANs, and ADAM optimizers, all to improve overall performance and make the generated passwords look more realistic. In contrast, one of the interests of this algorithm is to monitor login attempts by generating honey passwords extracted from user information [10].

In 2023, the research proposed a new method for generating similar words, based on the Separate Swarm Algorithm (SSA) . This algorithm mimics natural swarming behavior to enhance the generation of similar words and address the previously found limitations. The research explained the advantages of this algorithm as it improves flatness and resistance to denial-of-service attacks. It is worth noting that it addresses security issues such as storage costs and association problems. Another important feature is that it balances exploration and exploitation. As for the characteristics of similar words, it uses simpler methods for some symbols and thus improves their characteristics. This algorithm is effective, but it lacks diversity, which increases the iterations in the algorithm. Researchers in the future can combine this algorithm with other techniques to enhance and improve generation [9].

In 2023, honey word generation techniques were analyzed and their difficulties in balancing false positives and false negatives were highlighted their struggles with balancing false positives and negatives, especially with human-chosen passwords. For passwords generated by managers, protection is modest. The most effective method seems to be using the same generator as the user, revealing the need for further research in honeyword-generation strategies, various honeyword generation techniques were analyzed and evaluated, but these systems faced challenges in optimizing honeyword characteristics and balancing exploration and exploitation. There was a need to enhance the system by using swarm intelligence algorithms like the Bees Algorithm to generate more efficient and secure honeywords [2].

VIII. Conclusion

In the field of digital security, keywords are essential to enhance password security and protect private user data. Fraudulent passwords and honeycombs improve authentication frameworks and make these systems more difficult to hack. However, these advances face obstacles including brute force attacks, statistical exploitation, and denial of service (DoS) attacks. The design of generation algorithms determines the effectiveness of honeycomb systems. Fairness, diversity, and resistance to statistical auditing are important aspects of generation procedures.

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 134-142, January-2025
https://doi.org/10.47001/IRJIET/2025.901017

Finding the balance between usability and security is an ongoing struggle. To address the shortcomings and improve honeycomb generation techniques, advanced algorithms such as bee algorithm, passive swarm algorithm, meerkat algorithm, and harmony search algorithm are being investigated. Researchers seek to enhance these methods through integration. By leveraging a variety of datasets and incorporating collaborative security frameworks, researchers hope to enhance these methods. Understanding attack tactics and improving existing methods to create reliable, secure, and user-friendly systems is linked to the continued progress in honeycomb generation systems.

REFERENCES

[1] Belding, G. (2018, September 22). What Are Honeywords? Password Protection for Database Breaches. *Security boulevard* (Accessed 2025, jan, 3).

[2] Huang, Z., Bauer, L., & Reiter, M. K. (2023, September 19). The Impact of Exposed Passwords on Honeyword Efficacy. *33rd USENIX security symposium*.

[3] Sawant, S., Saptal, P., Lokhande, K., Gadhave, K., & Kaur, R. (2018, April). Honeywords: Making Password Cracking Detectable. *IJERAT*.

[4] Naik, K., Bhosale, V., & Shinde, V. D. (2016, July 4). Generating Honeywords from Real Passwords with Decoy Mechanism. *IJREEAM*, 2.

[5] Pattabiraman, S., Soms, N., Poovanan, & Ramakrishna, S. (2020). Password Protection Using Honeywords. *ICACCABT (Coimbatore).*

[6] Thite, V., & Nighot, M. (2021, May 5). Honeyword for Security: A Review. *IJASRET*, 6.

[7] Gholap, R. T., & Bhale, N. L. (2018). A Survey of Honeywords Techniques for User Authentication Enhancement. *IJARII*, 4.

[8] Yasser, Y. A., Sadiq, A. T., & AlHamdani, W. (2022). Generating Honeyword Based on A Proposed Bees Algorithm. *IJCCC*, 4.

[9] Yasser, Y. A., Sadiq, A. T., & AlHamdani, W. (2023, April 1). Honeyword Generation Using a Proposed Discrete Salp Swarm Algorithm. *BSJ*, 20.

[10] Ahmed, M. A., & Akif, O. Z. (2023). Honeywords Generation Technique Based on Meerkat Clan Algorithm and WordNet. *WJPS*, 2.

[11] Yasser, Y. A., Sadiq, A. T., & AlHamdani, W. (2022, March 25). A Proposed Harmony Search Algorithm for Honeyword Generation. *Advances in Human Computer Interaction*.

[12] Bhole, M., & Patnaik, G. K. (2018). Managing Passwords Using Honeyword Detection System. *JEPIR*, 12.

[13] Tian, Y.-J., Li, L., Peng, H., Wang, D., & Yang, Y. (2023). Honeywords Generation Mechanism Based on Zero Divisor Graph Sequence. *IEEE Transaction on Services Computing*, 16.

[14] Akif, O. Z., Sabeeh, A. F., Rodgers, G. J., & Al-Raweshidy, H. S. (2019, November 3). Achieving Flatness: Honeywords Generation Method for Passwords Based on User Behaviours. *IJACSA*, 10.

[15] Dionysiou, A., Vassiliades, V., & Athanasopoulos, E. (2021). Generating Honeywords Using Representation Learning. *ASIA CCS.*

[16] Almuhanna, A., Alfaadhel, A., & Ara, A. (2022). Enhanced System for Securing Password Manager Using Honey Encryption. *Conference of women in data science at prince Sultan University*.

[17] Chakraborty, N., & Mondal, S. (2015). A New Storage Optimized Honeyword Generation Approach for Enhancing Security and Usability. *IEEE Conference Computer science cryptography and security*.

[18] Gholap, R. T., & Bhale, N. L. (2018). A Survey of Honeywords Techniques for User Authentication Enhancement. *IJARIIE*, 4.

[19] Gadgil, A. A., Khatawkar, S. D., & Me, C. S. E. (2016). Enhancing Security in User Authentication through Honeyword. *IJSRM*, 4.

[20] Erguler, I. (2016). Achieving Flatness: Selecting the Honeywords from Existing User Passwords. *IEEE Transaction on Dependable and Secure Computing*, 13.

[21] Akshima, Changy, D., Goelz, A., Mishray, S., & Sanadhyax, S. K. (2018). Generation of Secure and Reliable Honeywords Preventing False Detection. *IEEE Transaction on Dependable and Secure Computing*.

[22] Iuorio, A. F. (2019). Understanding Optimizations and Measuring Performances of PBKDF2. *2nd International Conference on Wireless Intelligent and Distributed Environment for Communication.*

[23] Yasser, Y. A., Sadiq, A. T., & AlHamdani, W. (2022). A Scrutiny of Honeywords Generation Methods: Remarks on Strengths and Weakness Points. *Cybernetics and Information Technologies*, 22.

AUTHORS BIOGRAPHY

Shahad A. Saadullah,
Student, Dept. of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 134-142, January-2025

https://doi.org/10.47001/IRJIET/2025.901017

Saja J. Mohammed,
Professor, Dept. of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.

Citation of this Article:

Shahad A. Saadullah, & Saja J. Mohammed. (2025). Honeywords, an Enhancing Password Security Mechanism: A Comprehensive Survey. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(1), 134-142. Article DOI https://doi.org/10.47001/IRJIET/2025.901017

****

