iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.39970

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

A Study for an Ideal Password Management System

Shivam K. Shinde[1], Mohit V. Deshpande[2]
[1, 2]*Department of Computer Engineering, Shatabdi Institute of Engineering & Research, Nashik, India*

Abstract: ***The growing number of online services needs users to have control over their password management system (generation, storage, recall). But the demand for total randomness and exclusivity of passwords is impractical in day-to-day life. Each component of a password management system comes with its cognitive burden on a user. There are many password management solutions available for users but every one of them has some drawbacks. Password managers have the ability to help users manage their passwords more successfully while also addressing many of the problems about password-based authentication. In this study, We're analyzing various previous studies regarding the effectiveness, usability, and security of password managers of all categories. Also, we're trying to come up with an ideal set of parameters to build the best possible password management system in 2022. This study will help to understand the key parameters and algorithms that we can use while building the ideal password generation, storage, and recall system for the user.***

I. INTRODUCTION

A. *Why Password Managers?*

Entropy-based exclusive passwords from the point of security are a valid demand from many online services. But passwords that are difficult for an attacker to guess are also hard for users to remember, users often create weaker passwords to avoid the cognitive burden of recalling them. Unfortunately, the number of passwords a user must remember is growing, with the average Internet user having 25 different online accounts [6]. The fact that various sites have varying complexity rules frequently mitigates exact password repetition. Users, on the other hand, frequently employ easy tactics to get around these regulations, such as making minor changes to a popular password (e.g., adding a 1 to the end of a password used on another site). Users often make these adjustments using a limited set of simple guidelines, which can greatly increase an attacker's ability to guess passwords on other sites [6].

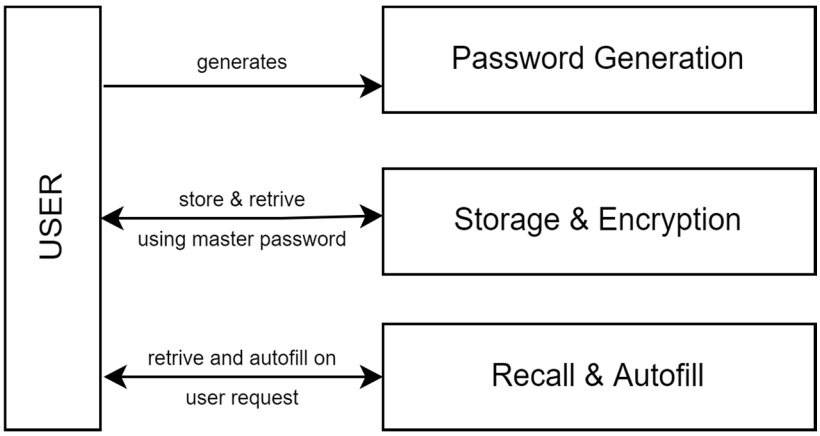
Password managers seek to tackle this problem by generating & storing passwords on a computer device rather than the user having to do that work and then delivering(recalling) them to the user as needed [9]. Many password managers are available; some are built into browsers, others are offered by third parties, and many are network-based, such as 1password, where credentials are backed up to the cloud and synchronized across the user's devices [10].

When it comes to password formation, human memory focuses on familiarity and repetition, putting us open to assault. People prefer to use the same password with personal information tacked on, according to research on building secure passwords done by Lo (2016). When requested to change a password, for example, adding a birthdate after the original password or adding the current month. Our minds can only carry around seven characters, so we can't store the long sequences of random characters required to be deemed safe in our internet world (Lo, 2016). Yan, Blackwell, Anderson, and Grant (2004) found that personal features such as a birthdate or pet's name are worthless against specific hacks in a survey of 288 college students[2]. For example, if a hacker uses a dictionary attack to generate potential passwords using the same phrase, then variants of a password are just as simple to crack (Yan et al., 2004)[2].

B. *What is a Password Manager?*

A password manager, in the most basic sense, is a tool that saves a user's credentials (i.e., username and password) to reduce the cognitive load involved with remembering many unique login credentials. A password vault is a name given to this collection of passwords. The vault should ideally be stored encrypted, with the encryption key most typically generated from the master password, which is a user-chosen password. The password vault can be kept online if desired, allowing for synchronization across several devices[1]. Most current password managers can assist users to create passwords in addition to storing user-selected passwords. The length of the intended password, the desired character set, and any particular attributes the password should have (e.g., at least one number and one symbol, no difficult to identify characters) are all inputs into the password generator. The password generator generates a password at random that fits the supplied criteria.

Many password managers additionally assist users in logging into websites by automatically picking and filling in the necessary username and password (i.e., autofill). If a user has numerous accounts on the site, the password manager will let them choose which account to use for autofill [1].



General working of a password manager

The inclination to utilize password managers has a mixed connection with trust. When other characteristics (such as perceived severity, vulnerability, and so on) are taken into consideration, trust in general technology has little effect, and faith in password managers has no impact. Individuals' threat assessments of password loss are a more major motivator of password manager adoption than faith in technology, according to the findings. When comparing technology vs. non-technology solutions, trust is likely to play a larger role. Password managers are used because of the perceived vulnerability and severity of password loss[3].

C. Pros & Cons of Password Manager

A password manager may provide a number of concrete benefits to the user if it is correctly configured and used.

1) It eases the load of memorizing usernames and passwords on the brain.
2) It is simple to set a unique password to each website, preventing password repetition.
3) Creating passwords that are resistant to both online and offline guessing assaults is simple.[1]

Password managers may become a single point of failure if done poorly, putting all of a user's credentials at risk. To keep passwords safe, password managers should only fill credentials when the user has explicitly allowed the operation, the credential is mapped to the web domain or app to be filled, and the filled credential is only visible to the mapped app or web domain [5].

D. Objectives of this Study

The primary objective of this study is to figure out what the most important components of a password management system are, and how we can integrate the findings of other researchers' studies and methodologies to create an ideal password management system.

1) Acknowledge the importance of finding a balance between usability and security.
2) Analyze how the components of generation, storage, and recall might be strengthened.
3) Describe the best set of approaches to utilize in the generation, storage, and recall of information.

II. PASSWORD GENERATION

The usage of truly random password strings in existing password managers provides strong authentication. Most existing systems provide users the choice of keeping pre-existing, non-random credentials or creating fresh random passwords at registration. If previous passwords are saved, the method offers no additional security, just the ease of quick password retrieval[9]. While all password managers support the same set of characters and numbers, their symbol sets were all distinct.

KeePassXC featured the most extensive symbol set, including all ordinary ASCII symbols (excluding space) as well as the extended ASCII symbol set. Other than space, KeePassX and Dashlane support the regular ASCII symbol set, but not the expanded ASCII symbol set. Only 19 ASCII symbols are supported by 1Password, whereas the other systems only accept 8 or fewer symbols. The strength of created passwords is significantly reduced when the symbol set is limited. Almost all passwords with a length of 12 or more were determined to be sufficiently secure to survive both online and offline guessing attempts[1].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

Password Manager	Supported Lengths	Default Length	Symbol Set
KeePassX	3-64	16	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
KeePassXC	1-128	16	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
1Password	8-50	20	!#%)*+,-.:=>?@]^_}~
Bitwarden	5-128	14	!#$%&*@^
Dashlane	4-28	12	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
LastPass	4-100	12	!#$%&*@^

The significance of these password generator functions in terms of maximizing the security of their passwords and so assisting them in remaining secure on the internet by generating 'lengthy and complicated' passwords[4]. We looked at unpredictability and guessability as indicators of its quality. There is no method to verify that a pseudorandom generator is indistinguishable from random, as far as we know.

III. PASSWORD STORAGE

The second step of the password manager's lifecycle is password storage.AES-256 is used to encrypt the databases of both app-based and extension-based password managers. App-based and extension-based password managers have greatly improved their ability to secure metadata. All metadata is encrypted in both KeePassX and KeePassXC. Most metadata is encrypted by extension-based password managers, but at least one component is not. An attacker might view or modify extension settings in 1Password X since they are stored in plaintext. Security-related options include whether auto-lock is enabled, the default password generating settings, and whether or not to display notifications. While Dashlane encrypts website URLs, it does not encrypt the website icons it connects with those URLs, making it possible for an attacker to deduce which websites a user has accounts. The email address used to log in to the password manager is leaked by all extension-based password managers[1].

Password Manager	Location of Storage	Encryption
1Password	Local Device Storage	AES 256-bit
Lastpass	Own Cloud Storage	AES 256-bit
Dashlane	Local & Cloud Storage	AES 256-bit

The encryption key is referred to as the user's master key if the password manager enables credential encryption. LastPass, for example, use JavaScript to decrypt and encrypt the user's credential database with a key obtained from the user's master username and password[7]. By combining a master secret with domain names to dynamically create per-domain passwords, several systems exist for strengthening user passwords (and eliminating direct password reuse)[8].

IV. PASSWORD RECALL & AUTOFILL

Password managers utilize a variety of approaches to recall and autofill passwords, such as app-based password managers that generate a local duplicate of cloud data and encrypt it with a master password. Extension-based databases also create a local copy that is password-protected. When the device or autofill service manager generates an autofill request after the local database has been populated. The password manager decrypts the vault locally before decrypting the relevant autofill password. On both iOS and Android, autofill frameworks are available. The app extensions framework and the Password AutoFill framework are both available on iOS. The autofill service is available on Android. Autofill frameworks for browsers and other apps loaded on mobile devices are provided by smartphone operating systems. These frameworks are meant to interface with password managers to enable safe and useful autofill for browsers and other apps installed on mobile devices[5].

If a password manager autofill passwords without prompting the user, the user's password can be stolen without them even knowing it by visiting a hacked website. As a result, user engagement should be necessary before autofill takes place. User participation is required by default in KeePassXC, Bitwarden, and RoboForm, although it may be deactivated. Dashlane, Lastpass, and Firefox all default to auto-filling passwords without user input, however, there is an option to force user input. User credentials are always auto-filled in Chrome, Edge, Internet Explorer, and Opera. While having the option to demand user interaction[1]. Several autofill rules can have devastating implications, allowing a remote network attacker to retrieve several passwords from a user's password manager without requiring the user's participation[10].

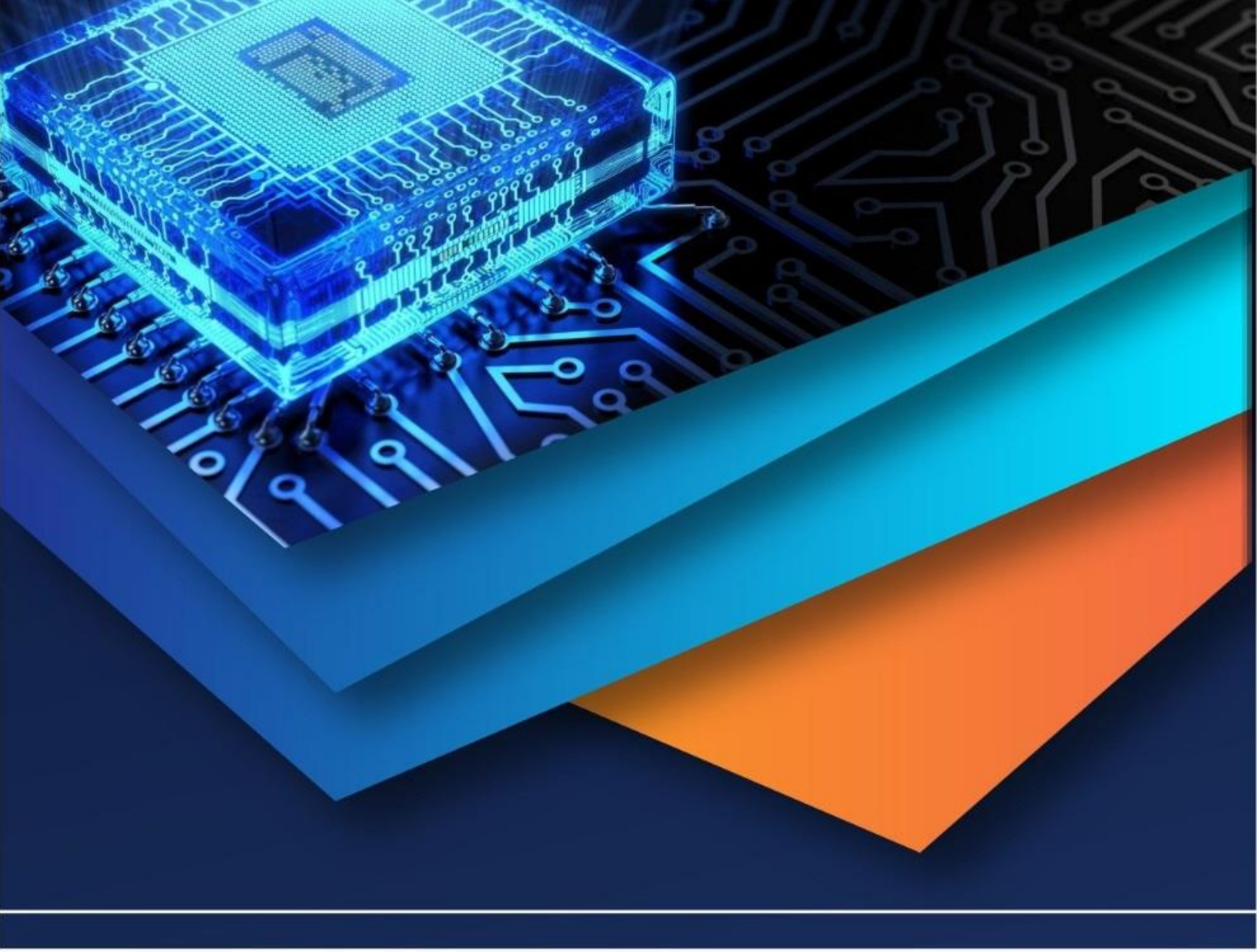
Regardless of user participation, auto-filling passwords within iframes are extremely risky. Clickjacking, for instance, can be used to deceive users into giving the required user input for auto-filling their credentials, allowing an attacker to obtain passwords for susceptible websites loaded in an iframe (same-origin or cross-origin). Worse than that, if autofill is enabled for cross-domain iframes and no user involvement is necessary, the attacker may automatically collect the user's credentials for any website where a network injection or XSS attack can be performed (by loading hacked webpages into iframes)[1].

V. CONCLUSION

When it comes to password generation, storage, and recall, there are a variety of approaches. Even randomly weak passwords are likely to be immune to online and offline assaults, therefore choosing passwords of adequate length is still desirable. According to our study, the length of resistance to online assaults is 10 and the length of resilience to offline attacks is 18. When it comes to storing local databases and information, AES-256 is the industry standard. For recalling data security and accessibility are provided by OS-based mobile autofill frameworks, and all frameworks need user engagement prior to autofill. Furthermore, iOS password autofill encrypts the autofill process for native UI components in apps completely. Localstorage with master password encryption is considered the best approach in web extensions.

REFERENCES

[1] Oesch, Sean, and Scott Ruoti. "That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers." USENIX Security Symposium. 2020
[2] Gallagher, Elizabeth A. "Choosing the Right Password Manager." Serials Review 45.1-2 (2019): 84-87
[3] Ayyagari, Ramakrishna, Jaejoo Lim, and Olger Hoxha. "Why Do Not We Use Password Managers? A Study on the Intention to Use Password Managers." Contemporary Management Research 15.4 (2019): 227-245
[4] Alkaldi, Nora, and Karen Renaud. "Why do people adopt, or reject, smartphone password managers?." (2016)
[5] Oesch, Sean, Anuj Gautam, and Scott Ruoti. "The Emperor's New Autofill Framework: A Security Analysis of Autofill on iOS and Android." arXiv preprint arXiv:2104.10017 (2021)
[6] Das, Anupam, et al. "The tangled web of password reuse." NDSS. Vol. 14. No. 2014. 2014
[7] Chiasson, Sonia, Paul C. van Oorschot, and Robert Biddle. "A Usability Study and Critique of Two Password Managers." USENIX Security Symposium. Vol. 15. 200
[8] Chatterjee, Rahul, et al. "Cracking-resistant password vaults using natural language encoders." 2015 IEEE Symposium on Security and Privacy. IEEE, 2015
[9] Karole, Ambarish, Nitesh Saxena, and Nicolas Christin. "A comparative usability evaluation of traditional password managers." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2010
[10] Silver, David, et al. "Password managers: Attacks and defenses." 23rd {USENIX} Security Symposium ({USENIX} Security 14). 2014



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY