Password Managers: A Survey

Automatic Password Change Using Headless Browsers

[1]Yasasw. Kumarakalva , [2]Vidhya Shankar. G. S , [3]Shreekara. K. K, [4]Sridhar. P. H, [5]Asha. G. R
[1,2,3,4] IV year B.E. (2016-2017), [5] (Under the guidance of) Assistant Professor
[1,2,3,4,5]Department of Computer Science and Engineering (CSE)
[1,2,3,4,5]BMS College of Engineering, Bangalore, India.

***Abstract*— Password Managers are widely available software, but only a small fraction of internet users make use of them. This is mainly due to the fact that the majority of them do not recognize the need for such a software as they feel there is no threat to their online services. The others who recognize this threat do end up using one of the many password managers available. After the initial setup, the user is left with a secure way of logging in to his / her online services; albeit a bit cumbersome. But then news hits that the user data of an online service has been compromised and the users are requested to change passwords to avoid any further damage. This might not be a scenario that everyone finds themselves in very often; but when they do, they are left with a very cumbersome method of manually changing passwords.**
Even when there is no threat to your account, it is usually a good practice to change your password ever so frequently. To make this process easier, a few password managers such as Dashlane and Last Pass have introduced a feature called automatic password change. This helps in the periodical change of password without the user's involvement. But these software's use methods that, while adequately secure for a normal user, is still very vulnerable in many areas. In this paper we specify some of these loopholes and try to provide a comprehensive solution. We also propose some new methods of security that improve upon the methods used in these software's.

***Index Terms*—Password Manager, Automatic Password Change, Headless Browser, Security, Two - Factor Authentication, OTP, Biometrics, TPM, USB Key.**

I. INTRODUCTION

Most of the time, we use the same passwords for different websites, usually because it is easier to remember. But when one of the password is compromised, all of your online services and data are at the risk of being hacked. The best solution is to use different passwords for every website, but not everyone is blessed with good memory to remember all these passwords. The next best solution is to use a password manager which stores all the different passwords for you and later lets you retrieve them. This liberates the human brain by enabling a password manager to generate a longer and stronger password without worrying about memorise it.

A strong password is one that cannot be guessed by either a human or a machine. Password strength can be achieved by embedding the following characteristics:

- Uppercase and lowercase characters
- Letters and numbers
- Special characters: e.g., & @ # ! ?]

(**Note:** < or > should not be used in your password, as both these characters can cause problems in Web browsers).

A strong password chooses these characteristics at random so as to make it uncrackable (except by brute force, which would take many years to crack even by using sophisticated machines, thus making the method impractical to use).

The minimum length of a password in almost all cases is 8 characters while the upper bound remains variable to each website. It must be kept in mind that many websites have different rules and restrictions governing password creation. The password creation process on different websites can be a bit like visiting foreign countries with unfamiliar social customs. While one requires eight characters; the other one lets you have up to 64. While one allows letters and numbers only; the other one allows hyphens and some others might allow special characters; and so on. All these restrictions must be kept in mind during the process of password generation.

The password manager stores the passwords in a centralized database, located either locally or on the cloud, which is encrypted by a master key. This master key is all the user has to remember to access the database. Latest implementations use biometrics instead of a password to identify the user, thus eliminating the prospect of the user forgetting the master key.

The above features entail the basic functions of any password manager. Some password managers throw in some extra features like app encryption (app lock), file encryption, auto form filling (remembering data other than passwords), and some also throw in extra security features such as two-factor authentication. Most of these features are an incremental improvement over the basic premise of a password manager and extend the definition to include apps and files. We move on to the main feature that the title indicates- Automatic Password Change.

Automatic Password Change is the process of changing the password for a website by the click of a button, without the need to visit the 'change password' page for that website. This helps when you want to change the passwords of many websites at once, saving you the time and hassle of manually going to each website and changing the password.

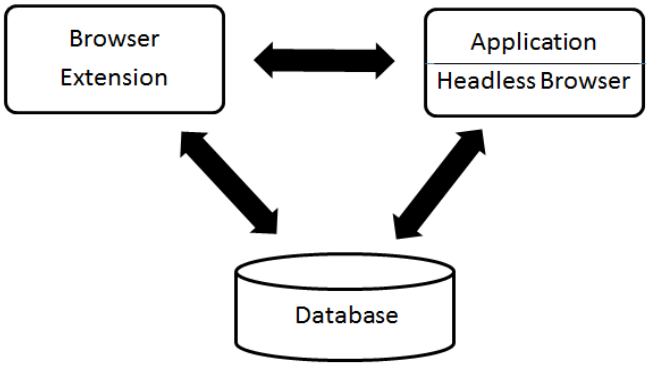
II. DESIGN

The implementation of automatic password change is done using a Headless Browser.According to arhg.net, " A headless browser is a web browser without a graphical user interface. Headless browsers control web pages automatically in an environment similar to popular web browsers. But the difference is that they are executed using a command-line interface or communicate through a network "[25]. Thus to achieve the feature of automatic password change, a script must be written for the browser to log into the website on your behalf and change the password. This cannot be achieved by the headless browser alone.

For successful implementation of the above features, the password manager must have the following components::

- Browser Extension
- Application
- Headless Browser

These three components work together behind the scenes to provide a seamless experience to the user. The application and the headless browser components can be merged, but it is essential that the components remain, because the application is required to store the script required to run the headless browser.



The database that stores the passwords is usually a separate entity. The encryption and decryption process is taken care by the application. The storage location of the database is usually on the local storage, so as to implement the best possible security, but it can be stored on the cloud as well. Another option is to use a private server, but it requires a lot of investment and maintenance to run smooth

The diagram shows a model representation of the architecture of a password manager. The browser extension communicates seamlessly with application and the database without any interruption to the user's work. The application does the main work, doing jobs like communicating with the headless browser and dealing with command that are given to it. The database is encrypted and can be accessed only after entering the master key whenever you use either the application or the extension.

III. LITERATURE SURVEY

The first concern we will address is the use of the computer memory as discussed in [1]. This paper discusses the benefits of using TPM over memory to store passwords. According to Howtogeek.com, "TPM stands for Trusted Platform Module and is a very helpful tool for encryption. It is basically a chip on the motherboard that helps in encryption without the requirement for extremely long passwords and makes it tamper-resistant. The TPM generates encryption keys, keeping a part of the key to itself". So, a section of the key is stored in the TPM itself instead of just on the disk, if you're using encryption with a TPM. This means an attacker cannot remove the drive and attempt to access its files on any other computer. An attacker can't remove the chip and place it on another motherboard, or tamper with it to attempt to bypass the encryption because the chip provides hardware-based authentication and tamper detection [26].

There are many other methods if you do not wish to store passwords on the local storage of a machine, with the fear of it being stolen. Secret sharing and personal servers can be used to satisfy security and usability criteria for a password manager [2]. This needs investment; which, however small, is still something many people are not willing to shell out. Another implementation is to use a cloud based storage instead of a personal server, which for the intents of storing passwords is completely free. The implementation in [3] shows a cloud based password manager, which while being a good model, still lacks the security features a local encryption model could have. The account can be hacked, the server of the cloud service provider could be compromised and the worst case scenario is that the local government retrieves all the data from the cloud as government rules allow this. Thus we stick to the local storage model.

Another good method of storage would be to decentralize the data; ie, not store it in one single space. One idea would be to use the method proposed in [11] which says that data be shared between local and cloud storage, so that even if the data on one end is compromised, there is no problem, as both local and cloud data are required to access the database. Another implementation proposed in [5] uses the same decentralized storage formula, but all files are stored locally. It creates faux passwords using Transformed-Based Algorithm and a decentralized file format architecture is used to distribute credential

information into different files for storage. This method increases security of the password manager while restricting the data to the local storage, which is a win-win situation as lesser amount of resources are needed. Both the above proposed decentralized methods of storage are much better implementations than the rest as the risk of your data being compromised reduces drastically.

The next problem we face is the problem of secure access on more than one computer. This is required as we sometimes need to access websites on other computers due to a myriad of reasons. This is difficult to implement as it is very easy for the master key to be compromised in a computer whose security levels you do not know. Secure biometrics can be used as shown in [3], but these is no guarantee that there are any biometric sensors present on the computer. Biometrics also pose a lot of threats and thus must not be used everywhere. The alternative is to make another requirement necessary to access the database other than the master key. This could include an OTP Solution or a USB Key solution. Remember, this is possible only if the architecture allows remote access of data or a cloud based solution.

Paper [4] gives one of the comprehensive solutions. It gives a method similar to OTP to access the data. You need your mobile phone along with your master key to prove you are the owner. The master key is used to generate a unique code each time and the code is sent to the phone. The phone has data that decrypts the data and provides the key to log in. another solution is provided in [6] where a USB key is used along with the master key for unlocking the password manager. A USB key is usually A usb thumb drive, which in this case is used as a key. This usb drive usually contains a unique code hidden in it that can unlock the password manager, along with the master key. The thumb drive can be imagined as a physical key with sophisticated ridges that is very difficult to forge. These solutions can be used to implement secure access on remote pc.

Another new method of authentication is given in [7], where the user's voice is recognised by the voice-recognition software when he speaks out a certain passphrase. This is a highly experimental as voice recognition is difficult to implement on a computer with average hardware components.

The virtue of a good password manager is also to be available on the most popular operating systems used. The most popular OSes in 2016 are Ios, Android, windows, MacOS and Linux. A Password manager caters to a vast majority of users if it is available to these operating systems. The main piece of software required is the browser extension, which provides the seamless interface for the password manager. This cannot be implemented in a mobile OS (Ios and Android) as many browsers still do not support extensions in these platforms. Paper [8] provides a solution where a keyboard is used instead of a browser extension. This is a better solution on a mobile platform as other passwords like app passwords can also be easily incorporated.

We now come to the problem of encrypting the database of passwords. We need to stay ahead of the curve in encryption methods because hackers always find a way to break the toughest of encryptions, given time. In[14], an outdated encryption method called Blake 264 Hash function is used for Password encryption and Parallel CRC. A method for Secure Login is proposed in [15] by Using One-time Password Authentication Based on MD5 Hash Encrypted SMS. This is similar to the method discussed before. Though we use sophisticates encryption algorithms, our master password is still weak, because we have to remember it. To avoid dictionary attacks, Encrypted Key Exchange is used [16.].

A new approach for improvising password encryption is using the process of Jumbling-Salting [17]. In order to augment the security aspect regarding passwords, we are devising Jumbling Salting algorithm which prevents dictionary and brute force attacks by increasing the length of cipher text. In this algorithm, modulus function is used in the jumbling process which selects characters from pre-defined character set and adding them into the unencrypted password; salting consists of adding a random string into jumbled password. Ultimately AES block is implemented which obtains a fixed length password which is stored in the database located on the server. A random version of JS algorithm ensures that the time required to crack the password is increased, by forming a highly secured version of encrypted password. Keywords. Whenever a server or a third party is involved, Three-party Encrypted Key Exchange Protocol with Protected Password Authentication nam be used [13]. We can use encryption techniques widely used in other areas such as online payment and internet banking where more secure implementations are made [18 - 23]. These, however are too complex and deprecative to the above methods, so we shall avoid using them.

Our last step is to compare the most widely used password managers out there. Papers [9, 10,12] provide a comprehensive comparison between the most widely used password managers in 2016. We shall concern ourselves with only Dashlane and LastPass for this discussion, as they are the only ones that provide the feature of Automatic Password Change. As discussed above, Automatic password change is a feature implemented by using a headless browser. Both the above softwares provide this feature by using a headless browser as well, but they do the process of changing the password on their respective servers, which means they require access to our passwords[24]. Instead we should prefer a method in which the change is done on the local machine as much as possible.

Another huge problem with these password managers is Data Legacy. Data Legacy refers to the data you leave behind after you have left a particular service. An application which does not leave any data legacy behind clears all data when the service is opted out of. Most of the password managers have data legacy, which means there is a chance that they still retain passwords somewhere even after the user has unsubscribed to their software. This leaves the users vulnerable as the passwords still might be in company's possession.

The main reason that users think twice before using these softwares is because of their cost. Both softwares cost about $5 a month for the lowest subscription. In a developed country like USA, this might be nothing, but most users elsewhere usually cannot justify using so much money just for the sake of managing passwords. They would rather prefer to remember the passwords instead of wasting that money. This is why we propose to build a free software that is secure and robust so that all users can use it for FREE!

IV. CONCLUSION

Even though digital security is a huge part of our daily lives, perpetrators find new ways of circumventing these restrictions and break into the most secure of locations. This leads to invasion of privacy and sometimes results in huge losses for the victims of these attacks. Password managers are needed now more than ever because the threat to individual privacy is increasing drastically and using a strong password stops these perpetrators in their path. This might not completely stop them, but it will drastically slow them down. Automatic password change must become a staple feature of every password manager, because frequent change of passwords ensures the privacy of the user to a large extent, and once the feature is ubiquitous, cyber-crimes will diminish to a very large extent, thus ensuring our online safety….

V. ACKNOWLEDGEMENTS

VI.

The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

REFERENCES

[1] Wang, Hua; Guo, Yao; Zhao, Xia; Chen, Xiangqun -- [IEEE 22nd International Conference on Advanced Information Networking and Applications (aina 2008) - Gino-wan, Okinawa. " Keep Passwords Away from Memory: Password Caching and Verification Using TPM "

[2] Fukumitsu, Masayuki; Hasegawa, Shingo; Iwazaki, Jun-Ya; Sakai, M -- [IEEE 2016 IEEE 30th International Conference on Advanced Information Networking and Applications " A proposal of a password manager satisfying security and usability by using the secret sharing and a personal server "

[3] Yang, Bian; Chu, Huiguang; Li, Guoqiang; Petrovic, Slobodan; Bus -- [IEEE 2014 IEEE International Conference on Cloud Engineering (IC2E) - Boston, MA, USA " Cloud Password Manager Using Privacy-Preserved Biometrics "

[4] Wang, Luren; Li, Yue; Sun, Kun -- [IEEE 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS) - Nara, Japan " Amnesia: A Bilateral Generative Password Manager "

[5] Agholor, S.; Sodiya, A.S.; Akinwale, A. T.; Adeniran, O. J. -- [IEEE 2016 Sixth International Conference on Digital Information Processing and Communications " A Secured Mobile-Based Password Manager "

[6] Wang, Xing; Han, Zhen; Zhang, Dawei -- [IEEE 2012 International Conference on Industrial Control and Electronics Engineering (ICICEE) - Xi'an, China " IDKeeper: A Web Password Manager with Roaming Capability Based on USB Key "

[7] Aliasgari, Mehrdad; Sabol, Nick; Sharma, Ashutosh -- [IEEE 2015 First Conference on Mobile and Secure Services - Gainesville, FL, USA " Sesame: A Secure and Convenient Mobile Solution for Passwords "

[8] Boukayoua, Faysal; De Decker, Bart; Naessens, Vincent -- [IEEE 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS) - Aalborg, Denmark " A keyboard that manages your passwords in Android "

[9] Ziegler, Dominik; Rauter, Mattias; Stromberger, Christof; Teufl, -- [IEEE 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS) - Aalborg "Do You Think Your Passwords Are Secure?"

[10] Zhao, Rui; Yue, Chuan; Sun, Kun -- [IEEE 2013 International Conference on Social Computing (SocialCom) - Alexandria, VA, USA "A Security Analysis of Two Commercial Browser and Cloud Based Password Managers"

[11] Fang, Hao; Aiqun, Hu; Shi, Le; Li, Tao -- [IEEE 2015 International Conference on Wireless Communications & Signal Processing (WCSP) - Nanjing, China " SESS: A Security-Enhanced Secret Storage Scheme for Password Managers"

[12] Arias-Cabarcos, Patricia; Marin, Andres; Palacios, Diego; Almena -- "Comparing Password Management Software-Toward Usable and Secure Enterprise Authentication" [IEEE october-2016]

[13] Hsing-Chung Chen, ; Hsien-Yun Chuang, -- [IET IET International Conference on Frontier Computing. Theory, Technologies and Applications - Taichung, Taiwan, 2010 "A Three-party Encrypted Key Exchange Protocol with Protected Password Authentication"

[14] Fernandes, Floyd; Gupta, Ritu; Sivanantham, S; Sivasankaran, K -- [IEEE 2015 Online International Conference on Green Engineering and Technologies (IC-GET) "Implementation of BLAKE 256 Hash function for Password encryption and Parallel CRC"

[15] Sediyono, Eko; Santoso, Kartika Imam; Suhartono, -- [IEEE 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) - Mysore "Secure Login by Using One-time Password Authentication Based on MD5 Hash Encrypted SMS"

[16] Bellovin, S.M.; Merritt, M. -- [IEEE Comput. Soc. Press 1992 IEEE Computer Society Symposium on Research in Security and Privacy - Oakland, CA, USA "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks"

[17] Churi, Prathamesh P.; Ghate, Vaishali; Ghag, Kranti -- [IEEE 2015 International Conference on Science and Technology (TICST) - Pathum Thani, Thailand ""Jumbling- Salting: An Improvised Approach for Password Encryption

[18] Niansheng Liu, ; Yunfeng Wang, ; Donghui Guo, ; Linmei Jiang, -- [Institution of Engineering and Technology 2013 International Conference on Information and Network Securi.

[19] Mahto, Dindayal; Yadav, Dilip Kumar -- [IEEE 2015 3rd International Conference on Computer, Communication, Control and Information Technology (C3IT) - Hooghly, India

[20] Yeh, Yu-Chang; Ku, Wei-Chi; Chen, Wei-Ping; Chen, Yi-Lun -- [IEEE 2012 1st IEEE International Conference on Communications in China (ICCC) - Beijing, China (2012)

[21] Li, Wenjun; Ji, Jianhua; Zhang, Guirong; Zhang, Wenlin -- [IEEE 2016 Optoelectronics Global Conference (OGC) - Shenzhen, China (2016.9.5-2016.9.7)] 2016 IEEE Optoelect

[22] Erguler, Imran; Anarim, Emin -- [IEEE 2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB) - Avignon, France

[23] Laih, C.S.; Ding, L.; Huang, Y.M. -- Password-only authenticated key establishment protocol without public key cryptography IEEE- 2005

[24] https://csdashlane.zendesk.com/hc/en-us/articles/202699181-What-is-Password-Changer-and-how-does-it-work-

[25] http://blog.arhg.net/2009/10/what-is-headless-browser.html

[26] http://www.howtogeek.com/237232/what-is-a-tpm-and-why-does-windows-need-one-for-disk-encryption/