ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 8, Issue 3, pp: (264-267), Month: July - September 2020, Available at: www.researchpublish.com

Security Analysis on Password Managers Applications

[1]Abdulaziz Alrushaid, [2]Reem Algarawi

Saudi Aramco, Dhahran, Saudi Arabia

***Abstract*: Passwords are essential and fundamental for information security. They are used as a first security layer in defending electronic access. People are surrounded by tens of technologies, web services and mobile applications that require a password to grant access. Among all of these applications, people should worry about maintaining all different passwords securely. In this research, we will highlight different applications of Password Managers form security perspective. We will demonstrate the most common and used Password Managers showing their mechanism in storing confidential data, Master Key derivation and user authentication security. Finally, will conclude with security assessment result and recommendation for using Password Managers.**

***Keywords:* Password Managers, Native Application-based Password Managers, Web Browser Password Managers, security Analysis.**

I. INTRODUCTION

Approximately 80% of all cyber security attacks involve a weak or stolen password. This number shows the huge impact and importance of password confidentiality. In addition, major consequences can happen as a result of leaking passwords. And 64% of end users report that they have written down their password at least once. Secure password management requires that the password must be long enough (in terms of number of characters), and comprised of numerals, mixed-case letters, and special characters. as the password must also be kept secret and changed frequently.

Secure password practices result in numerous cryptic passwords, which are very difficult to keep track of. In addition, the fact that people are relying solely on memory is a clear indication that they are not following secure password practices, because if they can remember all of their passwords then they must create simple passwords or re-use passwords for multiple accounts.

Password managers are designed to relieve password fatigue, reduce mental load on users, and reduce log-in time. They can also indirectly facilitate better password quality and a reduction in password reuse. A naive password manager simply stores the passwords, while advanced password managers secure the stored passwords under a master password. Password managers can be divided into subcategories based on the credential management approach taken. In this research we have grouped the most common password managers to web browsers and native application-based password managers.

II. PRODUCTS SECURITY ANALYSIS

A. Web Browser Password Managers

1. Google Chrome

Google Chrome stores usernames and passwords in an SQLite database file in the user profile directory. In this database on Windows, the passwords are encrypted using Windows provided API (Windows Data Protection DPAPI) function, which makes the encrypted data only decipherable, by the Windows user account used to encrypt the password. DPAPI initially generates a strong key called a MasterKey, which is protected by the user's password. DPAPI uses a standard cryptographic process called Password-Based Key Derivation to generate a key from the password. This password-derived key is then used with Triple-DES to encrypt the MasterKey, which is finally stored in the user's profile directory. So essentially, your master password is your Windows account password. As a result, once you are logged into Windows

using your account this data is decipherable by Chrome. In addition, Google Chrome can optionally store all browser preferences (including passwords) on Google's servers to allow synchronization between different devices.

2. Mozilla Firefox

Mozilla Firefox stores usernames and passwords in an SQLite database too. Users can specify an optional master password that is used to encrypt the database content. URLs are always stored unencrypted regardless of whether a master password is used or not. Firefox makes use of an API developed based on their own open source library called the "Secret Decoder Ring" (SDR) to facilitate the encryption and decryption of account credentials. When a Firefox profile is first created, a random key called an SDR key and a salt are created and stored in a file called "key3.db". This key and salt are used in the 3DES (DES-EDE-CBC) algorithm to encrypt all usernames and passwords. These encrypted values are then base64-encoded and stored in a SQLite database called signons.sqlite. Since the database is part of Firefox' user profile, it can be automatically synchronized across multiple devices, either through Firefox Sync, manually, or stored on a USB stick and used on different computers.

3. Microsoft Internet Explorer

Internet Explorer stores usernames and passwords in the registry. Each record is stored as a separate registry entry and encrypted using the system login credentials. When a user fills-in a password form at address URL, Internet Explorer computes h = SHA-1(URL) and encrypts username and password as c = Ek(metadata || username || 0x00 || password || 0x00), where metadata contains additional information such as the size of encrypted elements. Similar to Google Chrome, the encryption is performed using the CryptProtectData system call, which uses Triple-DES in CBC mode and a hash-based MAC. The key is derived from three parameters: (1) a random salt (2) URL and (3) the Windows login credential for the current user. Finally, Internet Explorer creates a new registry entry with key h and value c. As a result, the security of Internet Explorer's password manager depends on the strength of the user account password.

B. Native Application-based Password Managers

1. LastPass

LastPass is one of the most popular online password managers that is available on phones, tablets, and desktops for most of the operating systems and major web browsers. One of the advantages of LastPass password manager is that the master password that is used to derive the encryption key for the password database is never stored or sent over the network in a plain text form.

LastPass stores all the credentials locally in the application, and in LastPass's production servers. These credentials are encrypted and decrypted locally in the client device with 256-bit AES encryption by the MasterKey that is generated from the master password. Therefore, when the user requests access to the password vault, LastPass will return all encrypted credentials over SSL, and the MasterKey that is saved locally in the device will do the decryption.

The entire MasterKey derivation process is done locally on the client device. It uses Password Based Key Derivation (PBKDF2) with SHA-256 hash function. PBKDF2 is a standard function and part of the Public-Key Cryptography Standards (PKCS). PBKDF2 requires a hashing algorithm, a seed, salt, number of iterations, and the plain text, which is the master password. LastPass uses the SHA-256 as the hashing algorithm, the user id as the salt, a random value as the seed value, and the number of iterations is configured to be 5000 iterations. The final derived key is a symmetrical key that is stored locally to encrypt and decrypt the password vault.

Same master password is used to derive another key used to authenticate the user before the login to the password vault. To derive the authentication key, the MasterKey undergoes an additional round of PBKDF2 where the encryption MasterKey will be used as a password and the master password will be used as the salt value. Finally, the generated authentication key is encrypted again by PBKDF2 function again with a large number of rounds to store it on the server side. The number of rounds used to generate the server key encryption is not published for security reasons.

2. KeePass

KeePass is free and open source password management application. Unlike LastPass, the KeePass application only exists locally in the client device and does not have a cloud component or a web server to log into. Therefore, the synchronization of passwords over the Internet is not supported.

KeePass offers multiple options on how to protect the password database. The user has the option to use a master key, a key file, or even both. The key file is an additional layer of security that can be stored locally on the device or externally on a storage device. KeePass doesn't only protect the user's password; it encrypts the entire database including passwords, usernames, URLs, and other notes using 256-bit AES encryption.

Cipher Block Chaining (CBC) block cipher mode is used with a randomly generated initialization vector (IV) is used in encrypting the database, which helps in concealing the plaintext patterns. Each time the database is saved, a new IV is generated randomly. Also, KeePass is following the design of encrypt-then-MAC (EtM) scheme to ensure the authenticity and integrity of the data by using the HMAC-SHA-256 hash of the ciphertext.

To Generate the MasterKey for the encryption algorithm, KeePass uses a key derivation function that requires a hashing algorithm, salt, random seed, and number of iterations. KeePass uses SHA-256 to compress all the password components that may include master password, a key file, or both to a 256-bit MasterKey. If the master key is used in conjunction with the key file, the MasterKey derivation formula will be SHA-256 (SHA-256 (password), key file). The default value of iteration is 6000; however, this number of configurable. Salts and random seed are derived from different values, such as data and time, performance counter, mouse cursor position, and random bytes provided by the system's random generator. As a result, it will prevent precomputation of keys and makes dictionary and guessing attacks harder. Unlike other web-based password manager tools, separate authentication and server keys are not required since all data is stored locally on the user's device. and it doesn't require any communication with a web server.

3. 1Password:

1Password is also a password and identity manager application that is available on OSX, Windows, iOS, and Android. Other than storing the user's credentials, 1Password provides a place for the user to keep track of secure notes, passport numbers, medical information, and even the credit card details and store them in the virtual and encrypted vault. Unlike LastPass, 1Password gives users a greater level of control over their confidential data by giving them the option to disable or enable the Internet cloud synchronization.

Items in the vault are encrypted and decrypted with Advanced Encryption Standard AES using the 256-bit MasterKey. 1Password provides authenticated encryption to protect the encrypted data from tampering by using the Aslois Counter Mode (GSM). As a result, this will offer a great defense against the Chosen Ciphertext Attacks.

The MasterKey used in the data encryption is generated by the combination of two different secrets in the Two-Secret Key derivation process (2SKD). The first secret is the Master Password provided by the user, which is never stored in the application. And since that password chosen by human tend to be guessable by automated password guessing systems, 1Password has the feature to suggest a Master Password for the user during the account creation. The suggested Master Password is generated locally on the device by randomly choosing four words out of a list of more than 18000 English Words. Therefore, a Master Password with four words will be 18000^4 possibilities. Second secret used in the two-secret key derivation is the Secret Key that is generated on the user's device during the account creation by combining different values such as non-secret version setting, the non-secret account ID, and a sequence of 26 randomly chosen characters using a Hash-based Key Derivation Function (HKDF) to create a new 32-byte salt. Since the Secret Key is unguessable and very difficult to be memorized by the user, it is stored in 1Password application. Master Password and Salt are passed to PBKDF2-HMACSHA256 with 100,000 iterations and the result will be XORed with the result of processing the Master Password.

There is also an authentication component to ensure that users only received the encrypted data they should have access to. To avoid the traditional authentication issues, 1Password uses password-authenticated key exchange (PAKE) process along with Secure Remote Password (SRP) mechanism to satisfy the authentication requirements such that client and server authenticate each other without either of them revealing any secrets in the process. Moreover, with the use of two-secret key derivation (2SKD), 1Password ensures that the data held by the server doesn't help in launching any password cracking attempts on the user's Master Password.

III. CONCLUSION

Password Managers are very critical piece of software designed to securely store and manage users' password and sensitive information. The security level of these Password Managers applications depends on the techniques that are used to protect the private data such as the application's database and the Master Password. It also depends on how efficient

and secure the encryption and authentication algorithms are. In this research we have presented the most popular Password Managers applications and found that most browser-based Password Mangers are not secure enough and they are easy to be broken against many attacks. Native application-based Password Managers that were presented in this research are considered to be safe, practical, and easy to use by users to follow the recommended secure password practices. Choosing from these password managers depends on the individual's preferences. Although the overall security in this category of applications is accepted, the security level will vary from one application to another based on the provided features that may degrade the application security.

REFERENCES

[1] D. Silver, S. Jana, E. Chen, C. Jackson, and D. Boneh "Password Managers: Attacks and Defenses" https://crypto.stanford.edu/~dabo/papers/pwdmgrBrowser.pdf

[2] Z. Li, W. He, D. Akhawe, D. Song "The Emperor's New Password Manager: Security Analysis of Web-based Password Managers" https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-li-zhiwei.pdf

[3] S. Standridge "Password Management Applications and Practices" https://www.sans.org/reading-room/whitepapers/bestprac/password-management-applications-practices-36755

[4] P. Gasti, K. Rasmussen On "The Security of Password Manager Database Formats" https://www.cs.ox.ac.uk/files/6487/pwvault.pdf

[5] "1Password Security Design" https://1password.com/files/1Password%20for%20Teams%20White%20Paper.pdf