

Password Management kit for Secure Authentication

Natanmai Deepak Sundararajan¹, Manikandan M.K¹, Karthickkumar J. ¹, Dr.Askarunisa A ²,
Saraswathi Meena R²

^{1,2} Department of Computer Science and Engineering, K.L.N College of Information Technology

Abstract—Password getting harvested by hackers have become very common today. The highly secure password in the world is useless if intruder steals it, but it becomes much useless if it is not the same password you use for every log-in. The time has come to throw away the passwords and get authenticated into the application via passwordless api, where you don't need to store multiple login credentials for each account instead an unique identification tokens will be generated during the time of authentication - while also controlling costs and maintaining the user experience.

Keywords—Authentication, Password Mobile Device, Human Computer Interaction, Security, Usability, Deployability.

1. Introduction

Millions of people use Internet on everyday basis for various purposes which includes email, news, music downloads, browsing information about anything. Peoples frequently access internet in their daily lives. Nowadays, it is destined for users to have a multiple accounts for Email accounts, websites, social networks, and many other services, all of which employs authentication method as passwords and thus having different passwords and security policies for each account. Remembering all the passwords is difficult and troublesome, so people end up in using simple passwords and hence compromising security. when we perform online transactions these practices are bound to help hackers, especially using computing devices. Hence, what we really require is a new and an innovative way to access internet services that does not involve remembering passwords with dozens of alphanumeric combinations, as well as does not add complexity for users. The security in password-based authentication is determined by the task of successfully guessing password. Unfortunately, passwords are easier to guess. To enhance the security of password-based authentication, a favorable solution is to make use of technology called multi-factor authentication, wherein a user is required to provide more than one authentication factor. The other piece of authentication information is either generated by a physical token, for example, RSA Secure ID or with Google Authenticator application. Although the two-factor authentication is able to enhance the security, different service providers may require setting up their own two-factor authentication services. In addition, users have to undergo painful registration and login procedures.

2. Authentication

Generally User authentication occurs in most human computer interactions. In most cases, a user has to enter an id and provide the corresponding password to start the use of a

system. Authentication authorizes human-to-machine interactions among applications and also allows both wired and wireless networks to enable access to network. In private and public networks, authentication is frequently done through the use of login id and passwords. Knowledge about login credentials is supposed to guarantee that the user is genuine. Each user registers to the system, with the help of assigned or self-declared password. Upon each use, the user must know and use the previously declared password. Nevertheless, password based authentication is not considered to give more security for any system that contains sensitive data. The domination of password based authentication is been there from the early days of authentication and still the only method being used widely. Certain characteristics like ease of using, should be faster and at the same time secure as well should have in an authentication method. Different provider use certain rules in defining passwords like password should have certain number of upper case, lowercase, number, special character. Example -google mail, which makes authentication process troublesome and more difficult for users to remember. The proposal have been made to replace text based method, some of the scope of proposal include management software, federated login protocol, graphical password scheme, one time password, hardware tokens, phone aided schemes and biometric methods. When certain method provide significant security then the problem is that it will be more costly to implement as well more difficult to use usability, deploy ability, security hence serves major factors in any method.

User benefits must be considered the method that must be memory wise easy to remember, simple for user so that can implement in large scale without any complexity to user, and which must avoid carrying object for the purpose but at the same time nothing to carry like mobile devices that everyone carries always can be used, physically effortless and easy to use, learn and also easy to recover from loss of token and credential like use backup methods. Deployment benefits must be in consideration that are accessible in the sense who uses password based method must be allowed to use the method with same ease, minimum cost per user including both provider side and client side cost, server compatible so that no need to change existing setup to support current case ,browser compatible which ensure no need to change the client side settings and can work on web browser and no extra additional software is required, also mature enough so that any user can implement or use the scheme for any purpose. Security benefits that should be considered are as follows: The attacker cannot be able to impersonate a user after observing them multiple times to their account, Resilient to Targeted Impersonation: It's not only possible for skilled investigator to impersonate a user by exploiting

knowledge of personal details like date of birth, relatives name etc, Resilient-to-Throttled Guessing: An attacker whose rate of guessing is constrained by the server and attacker cannot successfully guess the secrets, Resilient to Internal Observation: The hacker can't impersonate a user by intercepting the input from the user's device, Resilient to Leaks from Other Verifiers: A verifier could not possibly leak anything which can help an hacker to impersonate the user to the verifier, Resilient to Phishing: An attacker who simulates an genuine verifier cannot collect details that can be used later to impersonate the user to the valid server, Resilient to Theft: If the scheme uses a object for authentication, then the object cannot be utilized for authentication by another person who gains possession of that object, No Trusted Third Party: The scheme does not rely on a trusted third party who offers authentication mechanism, Requiring Explicit Consent: The authentication process cannot start without the explicit consent of the user. If users can be educated then common concept is that to select "correct" password which is complex task, offline brute-force attack to recover information will surpass the computational ability of machines. In real time, the entropy is a perfectly random 6 character password. However, the most common password length, is less than that of a DES key. Since DES was effectively broken by brute-force attacks due to error available in algorithm, this assumption is questionable. Nowadays, a variety of password policies request 18 character passwords. In such case, the entropy is comparable to AES. Also, a prevalence of password policies is given for guiding the users to choose passwords that are efficient. The main interpretation is that the community is demarcating the future viability of password increases in length and policies to ensure effective use of the password length, but users are capable of remembering approximately 7 random things. Also an increase in password length does not mean a commensurate increase in entropy. The basic limit amount of protection of current passwords can provide is no longer sufficient to protect password-based authentication systems exploitable to offline brute force attacks by the rapidly growing computing resources available. As all passwords are recoverable, the security of any system based on passwords will depend on the availability of hacking items, not how random passwords are generated. As such, protocols must be designed to not allow any type of offline attack, and the material that can be used to mount such an attack must be secured with the understanding that its confidential and is equivalent to the security of the authentication.

3. Existing system

The most ubiquitous method is the password based and has number of issues, which includes susceptibility to unintentional exposure through phishing and cross-site password reuse. There are many existing systems other than password based authentication being in use like OAuth 2.0 is the up gradation of OAuth protocol. It focuses on client developer simplicity along with providing specific authorization flows for applications. Double factor authentication schemes have the potential to increase security mechanism but faces usability and other challenges. Mobile Authentication is a system intended to provide

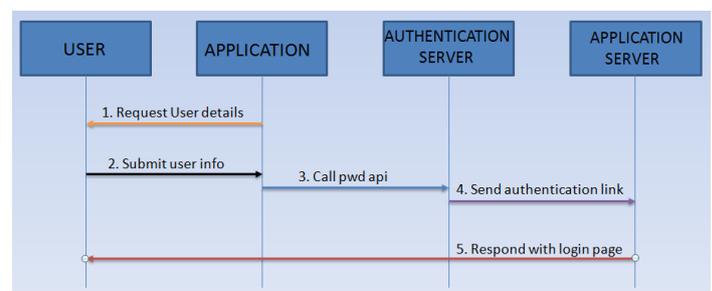
security assurances in comparison to or greater than that of conventional double factor authentication systems, in addition to offering the same authentication experience as traditional passwords. First, a user's personal device (phone) can communicate directly with the user's computer with no interaction with the user. Second, it is possible to provide a layered approach to security, by which a web server can impose different policies depending on whether or not the user's personal device is present. Kerberos is a distributed authentication service that enables a process (a client) running on behalf of a principal (a user) to prove its identity to a verifier (an application server, or just server) without sending data across the network.

4. Proposed system

During authentication, when the user logs in using their credentials, a Unique Token will be created and returned back and must be saved locally, Creating a session in the server and returning a cookie. There are security considerations that must be taken into account with regards to the way tokens are stored. Where to Store Tokens are enumerated. Whenever the user wants to access a protected route or resource, the user agent should send the Token, in the Authorization header using the Bearer schema. The content should look like the following:

Authorization: Bearer <token>

It is a stateless authentication mechanism as the user state is never saved in memory. The server's protected routes will check for a valid Token in the Authorization header, and if it's present, the user will be allowed to access resources. As Token will be self-contained, reducing the need to query the database multiple times. This allows you to fully rely on data APIs that are stateless and even make requests to downstream services. It doesn't matter which domains are serving your APIs, so Cross-Origin Resource Sharing (CORS) won't be an issue as it doesn't use cookies. The following diagram shows this process:



The following example Header declares that the encoded object is a Plaintext Token:

```
{ "alg": "hmac" }
```

Base64url encoding the octets of the UTF-8 representation:

```
eykxhbGciRiJoB25lIm0
```

