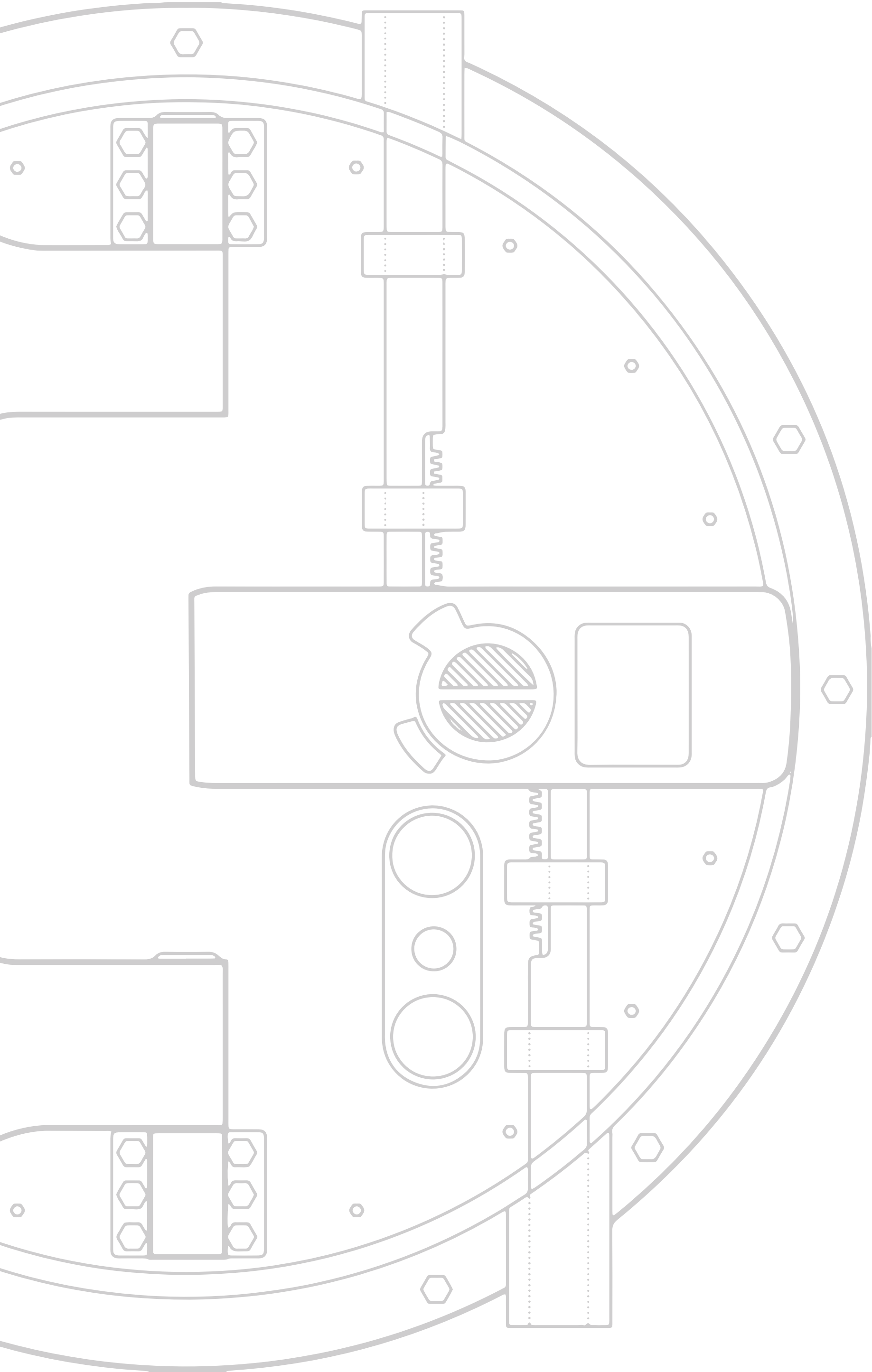
keeper



Why Your Business Needs Enterprise-Strength Password Management

White Paper **2016**

The Emergence of the Password and Hacking

Secret Keys

For over a half a century, the password, a secret sequence of characters known only to the authorized user, provided computer access security. Now, well into the second decade of the 21st century, the venerable password is no longer adequate. At least not in its current form.

In the 1940s, the combination of username, the public identifier for the user account, and the password, the secret key that provided access to the account, was enough security for most people. It primarily authenticated users on stand-alone computer systems, private mainframes, or closed networks.

Back then computing systems were expensive and complex, and physical access to these systems was closely guarded by administrators. Physical access was restricted to those with knowledge of operation of the complex and enigmatic operating systems. This also acted as a de-facto secondary form of authentication.

Emergence of Multi-User Systems

As computing systems became more advanced in the 1950s, a larger population began to use computers. Multiple users shared a single mainframe system simultaneously through dumb terminals, sometimes from remote locations. Physical access was no longer a barrier to use.

New protections and privileges were implemented to prevent unauthorized access to information and damage due to negligence or malicious users. Much of this was done in the operating systems themselves. In UNIX and similar operating systems, only the "root" user account had full system privileges. All other users held lesser degrees of privilege, and were assigned access to files and resources based on their username and/or group membership. Access required authentication with a username and password that was typically assigned by the system administrator not the user.

The Growth of Telecommunications and Hackers

In the 1950s, the Public Switched Telephone Network (PSTN) became automated. What had before taken a human being to connect calls could now be done through the use of unique tones and codes. This innovation brought a new breed of tinkering and experimentation, known as "phreaking."

From the 1960's through the 1980's, individuals known as Phone Phreaks began experimenting with the in-band signaling tones and commands on the automated PSTN. Through the use of "blue boxes" it became possible for Phone Phreaks to send tones to phone company equipment to make free phone calls or otherwise control their equipment.

The Phone Phreaking culture would become the foundation for the computer "hacker" culture in the decades to follow.

Network Computing Changes the Game

Modems and Remote Access

In the 1980's and early 1990's, access to remote systems through the use of a telephone modem became common. Dozens of online services, such as Quantum Link (which later became America Online), and thousands of private Bulletin Board Systems (BBS) sprang up around the globe. For the first time millions of people could remotely access computer systems and servers using their modem and phone line.

Around the same time, many businesses made critical computer systems available to their employees via modem. Many of these systems lacked authentication and instead relied on "security through obscurity," a belief that if no one knew about it, it would be secure. All that was therefore required for an attacker to access the system was the phone number and some trial-and-error to determine the baud rate, parity, stop-bits, etc. The abundance of these unprotected corporate and private systems lead to a technique known as "war-dialing", where intruders would sequentially dial phone numbers to discover answering modems. This was the precursor to modern IP address range scanning (referred to by some as "war-scanning") today.

Password Insecurity

The online systems of this era used the same username and password security found on multi-user mainframes. Now with thousands of users to study, systems administrators began to see some disturbing trends. When left on their own, users often selected simple and obvious passwords, such as "password" or "qwerty". Others used passwords that were less obvious, which was good, but were nonetheless easily to guess, such as a name of a family member, a pet, or an important date.

Also, individuals often used the same password elsewhere. That meant that if one BBS or Online Service was compromised, the intruder might gained access to all of w the user's other BBS or Online Service accounts. Password compromises were limited back then – most BBS systems were considered a hobby or novelty. Very few people conducted business over online systems, and prior to the 1990s the average person did not store a large amount of personal information on remotely accessible systems. The password, even if it was the same password, was deemed "good enough" by most users and systems administrators for online use.

The Rise of the Internet

When the Internet exploded in the 1990s what was once limited and physically restricted to a very select group of researchers and academics became accessible to the masses via the dial-up Internet Service Provider. Anyone with a computer, a modem, and the willingness to shell out $19/month gained instant access to any other computer connected to the Internet. In a very short period of time, millions of people were suddenly able to access millions of other computers across the globe. And as this technology took another epic leap forward, the venerable old password did not, with little or no change from the dawn of the stand-alone computing era.

The Rise of Encryption Methods

Key Derivation

By the 1990's, passwords were also being used on the Internet as encryption keys. Encryption obfuscates data and requires a user to have a key to unlock its secrets. Encryption keys are unique and allow only the owner or the recipient of an encrypted file to unlock the data. For a variety of reasons, passwords make very poor encryption keys.

- >> Passwords used as encryption keys are susceptible to brute-force cracking, where an attack tries every possible combination of characters in order to guess the password.
- >> Encryption algorithms often have very strict requirements in format or length that a password often does not meet.
- >> A suitable encryption key is often hundreds or thousands of characters in length, and therefore not suitable for human users to memorize or know.

In order to make a password function as an encryption key, a key derivation function is used to generate an encryption key that is more secure than the original password. However, a key derivation function is intentionally computationally complex and consumes additional computing resources by design to slow or otherwise make impractical brute-force attacks.

In addition key derivation functions implemented the use of a randomly generated string of data known as “salt” that is appended to the input. The net result was that even identical passwords never result in the same key because the salt is different for each. The more obstacles you put in the way of an attack, the more secure you are likely to be. Key derivations, in addition to being computationally intensive by design, also implemented a number of iterations or rounds of computation. Passwords can be encrypted through thousands rounds, consuming a significant amount of computing resources to generate a single key. Again, this was done to further thwart efforts to brute-force keys derived from passwords by making the process of deriving the key computationally expensive and impractical.

In order to derive a key for a given password, an attacker must not only have the original password, but also the salt and the number of rounds or iterations. Changing the salt or the number of rounds used to derive a key will result in a completely different key value.

Password Hashing

If usernames and passwords are the authentication method used on multi-user systems and the Internet, then how can these be stored securely? If anyone gained access to the database of all the passwords, the password file, then all the passwords could be compromised. The passwords need to be obscured by encryption.

One solution, first implemented on System V UNIX in the 1980s, obscured the value of each stored passwords through a mathematical process called hashing. Hashing is a one-way cryptographic function performed on a value. A hash function, given the same input value, always returns the same output value and is extremely difficult to reverse to determine the original value.

Still, an attacker can brute force the original input value by generating all the possible password values and then matching the resulting hash values. If the passwords are only four characters in length, this brute force attack is relatively easy. If the passwords are 15 characters in length, the task is much, much harder, if not impossible. In general, hashing made the storage of long and complex passwords secure.

Password-Based Key Derivation Function

One of the most widely implemented password key derivation functions in use today is PBKDF2 (also known as RSA's PKCS #5 v2.0). PBKDF2 requires five elements:

- A pseudo random number function (PNF)
- A password from which the key is derived
- A randomly generated cryptographic salt
- A number of iterations (for example, 10,000)

The desired length of resulting derived key

Complementing Passwords with Additional Authentication Methods

Multi-Factor Authentication

While hashing solved a primary issue around password security, other problems were not addressed. Users still used the same password for multiple accounts, and unencrypted passwords could be intercepted while in transit over the Internet. Given that some username and passwords might fall into the wrong hands, additional means of authentication were needed.

To mitigate this, some organizations deploy Two-Factor or Multi-Factor Authentication systems. This requires the user to incorporate two or more authentication factors from a list that includes: Something you have (a card or token), Something you know (a PIN or password), and Something you are (biometrics).

The most common form of Two-Factor Authentication has been in use more than forty years. It can be found at the local ATM. Here the user has a physical card from your bank (Something you have) and a Password Identification Number or PIN (Something you know). And in some Asian countries, banks also require a user's fingerprint (Something you are).

Most enterprise implementations of multi-factor authentication use a token that expires after a short period of time, usually 1-2 minutes. These include:

>> **On-Demand Tokens / One-Time Password**

Upon successful authentication with a username and password, the short-lived multi-factor token is delivered via out-of-band delivery, such as SMS text, e-mail or voice call to the user device

>> **Time-Based Token**

A new token is generated each minute from an algorithm that computes the token based on the current time and a pre-shared key.

While multi-factor authentication does provide an additional layer of security for online accounts, it is not bulletproof. It can be compromised in several ways. For the token, an attacker could gain access to the email account or device to which the token is delivered. Or, using social engineering, the attacker could change the delivery email address or destination phone number for the token. For the time-based token, an attacker only needs to have access to the seed key used by the multi-factor code generator, which happened in the RSA SecurID breach in 2011.

Biometrics - A Simple Convenience

If a password, a passphrase, an encryption key, a multi-factor token are all subject to compromise when transmitted over the internet, then surely, a thumbprint, such as one from Apple TouchID, or a retina scan is more secure? Or is it?

Biometric authentication is ultimately just more data. An aspect of the user's biometric profile, such as a thumbprint, is converted into electronic data. This thumbprint data is then compared to a stored data set—kind of like comparing a password hash. If a match is found, the device authenticates the user.

These data keys are stored on the device - which are, theoretically, only accessible via biometric authentication. However, in order to authenticate an online account with a biometric profile, that data key must inevitably be transmitted over the Internet and to the online service. And of course, that biometrically accessible data key must be stored on the end-user device, and a matching key or key hash, stored on the server in order to determine a match.

If that biometric key is compromised at any point, just like a compromised password, that key can be used by unauthorized persons to gain access to the online account, encrypted data, or any other sensitive information safe-guarded by the biometric data key.

Not to mention, most biometric authentication systems, such as Apple's TouchID, are still vulnerable to a variety of low-tech attacks, such as lifting latent prints with talcum powder and transparent tape.Unless you wear gloves, you are constantly leaving your fingerprints on everything that you touch, including your mobile device's touchscreen. Everything that a hacker needs to access your biometrically-protected phone is on the outside of your phone or for that matter around your home or office.

Given this, biometrics, while a convenience for the average user and possibly more secure than other screen-lock options on a mobile device, should not be relied upon to provide security.

Conclusion

Password Management Best Practices

Whether online user accounts are protected with a simple password, a key, multi-factor authentication, or biometric authentication, one fact remains: Online authentication is performed by transmitting a pattern of bits across the internet that is then compared to an expected result by the authenticating party. Any pattern of bits can, theoretically, be recorded and replicated by a third party.

Processes, such as multi-factor authentication or biometric authentication, can be applied to the authentication process to complicate or provide extra layers of protection to online authentication, but the fact remains that a sequence or sequences of bits matched to a known and expected result provides authentication to online accounts today. Authentication online remains almost unchanged in the nearly 50 years that they have been used to provide security to computing systems.

Until new technology addresses the security issues inherent in online authentication, there are three techniques users can implement to increase the chances of keeping online accounts secure from hackers.

- >> **Use a randomly generated complex and unique password**
  Users should use a unique password for every online account. The password should be complex and of the greatest length allowed. 12 characters should be a minimum, 16 characters or more are better. Use upper and lower case characters, numbers, and special characters.

- >> **Enable multi-factor authentication**
  While not the perfect solution, multi-factor authentication can greatly help prevent unauthorized access of your online accounts. Even if your password is compromised, multi-factor authentication can possibly alert you to the fact that your password has been compromised and buy you time to change your passwords and secure your account.

- >> **Change passwords regularly**
  Has your password been compromised? There is no way to know for sure. In light of the recent incidents of data loss, users only find out weeks or months later that their data or passwords have been compromised. The only way to stay ahead of the game is to change your passwords often. Passwords should, ideally, be changed every 60-90 days while remaining unique and complex at all times.

The best way to generate and keep track of complex and unique passwords is to **use a secure password manager.**

About Keeper Security

Keeper Security is transforming the way businesses and individuals protect their passwords and sensitive digital assets to significantly reduce cyber theft. As the leading password manager and digital vault, Keeper helps millions of people and thousands of businesses substantially mitigate the risk of a data breach. Keeper is SOC 2 Certified and utilizes best-in-class encryption to safeguard its customers. Keeper protects industry-leading companies including Sony, Chipotle, and The University of Alabama at Birmingham. Keeper partners with global OEMs and mobile operators to preload Keeper on smartphones and tablets.

Keeper Security, Inc.
850 W Jackson Blvd Suite 500,
Chicago, IL 60607-3025
312.226.5544
keepersecurity.com