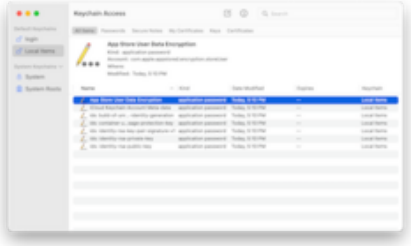
Keychain (software)

By Contributors to Wikimedia projects

Published: 2005-11-01 · Archived: 2026-04-06 00:48:15 UTC

From Wikipedia, the free encyclopedia

Keychain	
	
Developer	Apple
Initial release	1999
Operating system	Mac OS 9, macOS
Successor	Passwords
Type	system utility
License	APSL-2.0
Website	Keychain Services
Keychain Access	
	
 Screenshot of Keychain Access on macOS 12.	
Developer	Apple Inc.

Stable release	11.0 (55314) / 2022
Operating system	Mac OS 9, macOS
Successor	Passwords
Type	password manager
Website	Keychain Access Help

Keychain is a password management system developed by Apple for macOS. It was introduced with Mac OS 8.6, and was included in all subsequent versions of the operating system, as well as in iOS. A keychain can contain various types of data: passwords (for websites, FTP servers, SSH accounts, network shares, wireless networks, groupware applications, encrypted disk images), private keys, certificates, and secure notes. Some data, primarily passwords, in the Keychain are visible and editable using a user-friendly interface in Passwords, a built in app in macOS Sequoia and iOS 18 and available in System Settings/Settings in earlier versions of Apple's operating systems.

Keychains were initially developed for Apple's e-mail system, PowerTalk, in the early 1990s. Among its many features, PowerTalk used plug-ins that allowed mail to be retrieved from a wide variety of mail servers and online services. The keychain concept naturally "fell out" of this code, and was used in PowerTalk to manage all of a user's various login credentials for the various e-mail systems PowerTalk could connect to.

The passwords were not easily retrievable due to the encryption, yet the simplicity of the interface allowed the user to select a different password for every system without fear of forgetting them, as a single password would open the file and return them all. At the time, implementations of this concept were not available on other platforms. Keychain was one of the few parts of PowerTalk that was obviously useful "on its own", which suggested it should be promoted to become a part of the basic Mac OS. But due to internal politics, it was kept inside the PowerTalk system and, therefore, available to very few Mac users.[*citation needed*]

It was not until the return of Steve Jobs in 1997 that Keychain concept was revived from the now-discontinued PowerTalk. By this point in time the concept was no longer so unusual, but it was still rare to see a keychain system that was not associated with a particular piece of application software, typically a web browser. Keychain was later made a standard part of Mac OS 9, and was included in Mac OS X in the first commercial versions.

In macOS, keychain files are stored in `~/Library/Keychains/` (and subdirectories), `/Library/Keychains/`, and `/Network/Library/Keychains/`, and the Keychain Access GUI application is located in the Utilities folder in the Applications folder.[1][2] It is free, open source software released under the terms of the APSL-2.0.[3] The command line equivalent of Keychain Access is `/usr/bin/security`.

The keychain database is encrypted per-table and per-row with AES-256-GCM. The time at which each credential is decrypted, how long it will remain decrypted, and whether the encrypted credential will be synced to iCloud varies depending on the type of data stored, and is documented on the Apple support website.[4]

Locking and unlocking

[edit]

The default keychain file is the `login` keychain, typically unlocked on login by the user's login password, although the password for this keychain can instead be different from a user's login password, adding security at the expense of some convenience.[5] The Keychain Access application does not permit setting an empty password on a keychain.

The keychain may be set to be automatically "locked" if the computer has been idle for a time,[6] and can be locked manually from the Keychain Access application. When locked, the password has to be re-entered next time the keychain is accessed, to unlock it. Overwriting the file in `~/Library/Keychains/` with a new one (e.g. as part of a restore operation) also causes the keychain to lock and a password is required at next access.

Password synchronization

[edit]

If the login keychain is protected by the login password, then the keychain's password will be changed whenever the login password is changed from within a logged-in session on macOS. On a shared Mac/non-Mac network, it is possible for the login keychain's password to lose synchronization if the user's login password is changed from a non-Mac system. Also, if the password is changed from a directory service like Active Directory or Open Directory, or if the password is changed from another admin account e.g. using the System Preferences. Some network administrators react to this by deleting the keychain file on logout, so that a new one will be created next time the user logs in. This means keychain passwords will not be remembered from one session to the next, even if the login password has not been changed. If this happens, the user can restore the keychain file in `~/Library/Keychains/` from a backup, but doing so will lock the keychain, which will then need to be unlocked at next use.

Third-party software for keychain synchronization

[edit]

There was a 3rd party software application developed, that enabled synchronization of personal keychains generated using keychain access in Mac OS X, these standard keychain access - generated users keychains could then be synchronised between devices (iPhones - desktop Apple computers), using a pair of keychain synchronization apps developed by Patrick Stein of Jinx Software, one for Mac OS X and another for iOS called Keychain2Go. Keychain2Go could not be successfully updated by the developer to account for restrictions that Apple made to Keychain and access to Keychain in Mac OS X Sierra 10.12.[7]

Keychain is distributed with both iOS and macOS. The iOS version is simpler because applications that run on mobile devices typically need only very basic Keychain features. For example, features such as ACLs (Access Control Lists) and sharing Keychain items between different apps are not present. Thus, iOS Keychain items are only accessible to the app that created them.

As Mac users' default storage for sensitive information, Keychain is a prime target for security attacks.

In 2019, 18-year-old German security researcher Linus Henze demonstrated his hack, dubbed KeySteal, that grabs passwords from the Keychain. Initially, he withheld details of the hack, demanding Apple set up a bug bounty for macOS. Apple had however not done so when Henze subsequently revealed the hack. It utilized Safari's access to security services, disguised as a utility in macOS that enables IT administrators to manipulate keychains.[8]

- List of password managers

1. ^ *"Mac OS X 10.5 Help - Changing your keychain password"*. *Docs.info.apple.com. Archived from the original on May 31, 2012. Retrieved March 28, 2016.*
2. ^ *"Mac OS X 10.4 Help - Changing your keychain password"*. *Docs.info.apple.com. Archived from the original on May 31, 2012. Retrieved March 28, 2016.*
3. ^ *Apple Inc. "Source Browser". opensource.apple.com. Archived from the original on March 7, 2012. Retrieved February 26, 2012.*
4. ^ *"Keychain data protection". Apple Inc. May 17, 2021. Archived from the original on December 20, 2021. Retrieved December 20, 2021.*
5. ^ *"Mac OS X 10.5 Help: Changing your keychain password". Docs.info.apple.com. Archived from the original on June 13, 2011. Retrieved February 26, 2012.*
6. ^ *"Mac OS X 10.4 Help: Locking and unlocking your keychain". Docs.info.apple.com. Archived from the original on June 13, 2011. Retrieved February 26, 2012.*
7. ^ *Stein, Patrick. "Keychain2go keychain synhcronisation software". Jinx Software. Archived from the original on May 31, 2023. Retrieved March 22, 2023.*
8. ^ *Newman, Lily Hay (June 1, 2019). "The Tricky Shenanigans Behind a Stealthy Apple Keychain Attack". Wired. Retrieved July 9, 2021.*
