Kerberos Single Sign-on Extension

User Guide

January 2020

Contents

Introduction

The Kerberos Single Sign-on (SSO) extension makes it easy to use Kerberos-based single sign-on with your organization's Apple devices.

Simplified Kerberos authentication

The Kerberos SSO extension simplifies the process of acquiring a Kerberos ticket-granting ticket (TGT) from your organization's Active Directory domain, allowing users to seamlessly authenticate to resources like websites, apps, and file servers. On macOS, the Kerberos SSO extension proactively acquires a Kerberos TGT upon network state changes to ensure that the user is ready to authenticate when needed.

Active Directory account management

The Kerberos SSO extension also helps your users manage their Active Directory accounts. On macOS, it allows users to change their Active Directory passwords and notifies them when a password is close to expiring. Users can also change their local account passwords to match their Active Directory passwords.

Active Directory support

The Kerberos SSO extension should be used with an on-premise Active Directory domain. Azure Active Directory isn't supported. To use the Kerberos SSO extension, devices don't need to be joined to an Active Directory domain. Additionally, users don't need to log in to their Mac computers with Active Directory or mobile accounts; instead, Apple recommends using local accounts.

Requirements

- iOS 13, iPadOS, or macOS Catalina.
- An Active Directory domain running Windows Server 2008 or later. The Kerberos SSO extension isn't intended for use with Azure Active Directory. It requires a traditional on-premise Active Directory domain.
- Access to the network where the Active Directory domain is hosted. This network access can be through Wi-Fi, Ethernet, or VPN.
- Devices must be managed with a mobile device management (MDM) solution with support for the Extensible Single Sign-on (SSO) configuration profile payload. Contact your MDM vendor to ask about their support for this configuration profile payload.

Enterprise Connect

The Kerberos SSO extension is intended to replace Enterprise Connect. If you're currently using Enterprise Connect and want to transition to the Kerberos SSO extension, please refer to the "Transitioning from Enterprise Connect" section in this document for more information.

Getting Started

Building and deploying a configuration profile

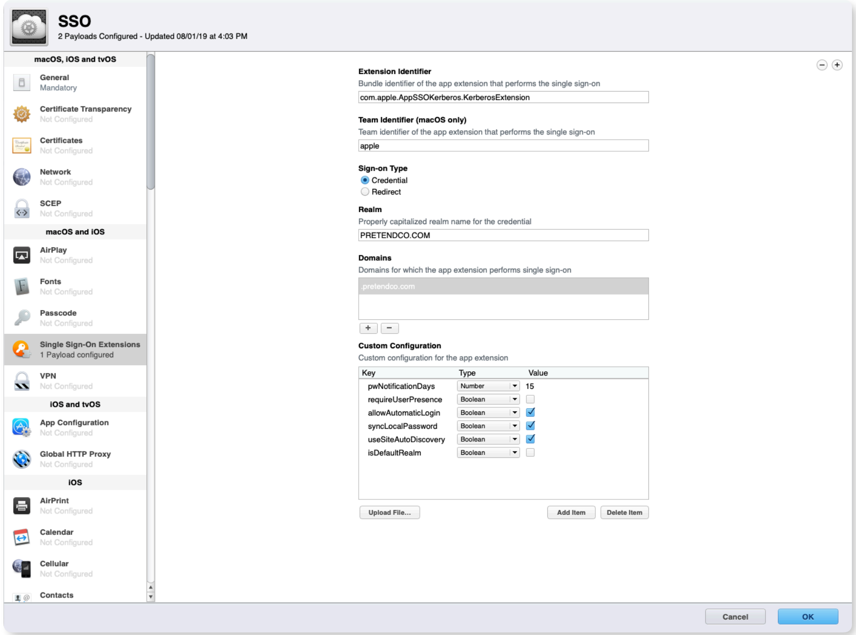
To use the Kerberos SSO extension, you must configure it using a configuration profile, delivered to the device from an MDM solution.

Note: The configuration profile must be delivered to the device by MDM. On macOS, that must be a user-approved MDM enrollment and installed in the System scope. Manually adding the profile is not supported.

To configure with a configuration profile, you'll use the Extensible Single Sign-on payload introduced in iOS 13, iPadOS, and macOS 10.15. Profile Manager—part of macOS Server—includes support for the Extensible Single Sign-on payload. If your MDM solution doesn't yet support this payload, you may be able to build the necessary profile in Profile Manager, then import it into your MDM solution for distribution. Contact your MDM vendor for more information.

To build a configuration profile using Profile Manager, follow these steps:

1. Sign in to Profile Manager.
2. Create a profile for a device group or a specific device.
3. Select the Single Sign-On Extensions in the Payload list, then click the Add (+) button to add a new payload.
4. In the Extension Identifier field, enter "com.apple.AppSSOKerberos.KerberosExtension."
5. In the Team Identifier field, enter "apple."



SSO
2 Payloads Configured - Updated 08/01/19 at 4:03 PM

Extension Identifier
Bundle identifier of the app extension that performs the single sign-on
com.apple.AppSSOKerberos.KerberosExtension

Team Identifier (macOS only)
Team identifier of the app extension that performs the single sign-on
apple

Sign-on Type
● Credential
○ Redirect

Realm
Properly capitalized realm name for the credential
PRETENDCO.COM

Domains
Domains for which the app extension performs single sign-on
.pretendco.com

Custom Configuration
Custom configuration for the app extension

Key	Type	Value
pwNotificationDays	Number	15
requireUserPresence	Boolean	□
allowAutomaticLogin	Boolean	☑
syncLocalPassword	Boolean	☑
useSiteAutoDiscovery	Boolean	☑
isDefaultRealm	Boolean	□

6. Select Credential under Sign-on Type.
7. In the Realm field, enter the name of your Active Directory domain where your user accounts reside, in all caps. Don't use the name of your Active Directory forest, unless your user accounts reside at the forest level.

8. Under Domains, click the Add (+) button and add domains for any resources that use Kerberos. For example, if you use Kerberos authentication with resources in us.pretendco.com, add ".us.pretendco.com." (Don't forget the leading period.)
9. Under Custom Configuration, add the following values:

Key	Type	Value
pwNotificationDays	Number	15
requireUserPresence	Boolean	Not checked
allowAutomaticLogin	Boolean	Checked
syncLocalPassword	Boolean	Checked
useSiteAutoDiscovery	Boolean	Checked
isDefaultRealm	Boolean	Not checked

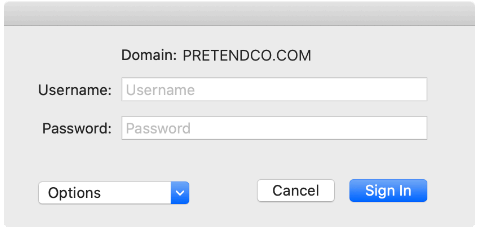
10. Click OK to save the new configuration profile. It will automatically install on the selected device or device group.

User setup—iOS and iPadOS

1. Connect your device to a network where your organization's Active Directory domain is available.
2. Do one of the following:
   - Use Safari to access a website that supports Kerberos authentication.
   - Launch an app that supports Kerberos authentication.
3. Enter your Kerberos or Active Directory user name and password.
4. You'll be asked if you want to permanently sign in automatically. Most users should tap Yes.
5. Tap Sign In. After a brief pause, your website or app will load. If you chose to sign in to the Kerberos SSO extension automatically, you'll no longer be prompted for credentials until you change your password. If you didn't choose to sign in automatically, you'll be prompted for credentials only when your Kerberos credential expires—usually in 10 hours.

User setup—macOS

1. You must authenticate to the Kerberos SSO extension. You can begin this process in several ways:
   - If your Mac is connected to the network where your Active Directory domain is available, you'll be prompted to authenticate immediately after the Extensible SSO configuration profile is installed.
   - If you use Safari to access a website that accepts Kerberos authentication, or you use an app that requires Kerberos authentication, you'll be prompted to authenticate.
   - You'll immediately be prompted to authenticate whenever you connect your Mac to a network where your Active Directory is available.
   - You can select the Kerberos SSO extension menu extra, then click Sign In.
2. You'll be prompted for Kerberos credentials. Enter your Kerberos or Active Directory user name and password.



Domain: PRETENDCO.COM

3. You'll be asked if you want to automatically sign in. Most users should click Yes.
4. Click Sign In. After a brief pause, your website or app will load. If you chose to sign in to the Kerberos SSO extension automatically, you'll no longer be prompted for credentials until you change your password. If you didn't choose to sign in automatically, you'll be prompted for credentials only when your Kerberos credential expires—usually in 10 hours.
5. If your password is close to expiring, you'll get a notification telling you how many days you have until it expires. You can click the notification and change your password.
6. If you've enabled the password sync feature, you'll be asked for your current Active Directory and local passwords. Enter both, then click OK to sync your passwords. You'll see this prompt on initial sign-in, even if your passwords are already in sync.

Password changes—macOS

You can also change your Active Directory password with the Kerberos SSO extension:

1. Ensure that you're signed in to the Kerberos SSO extension.
2. Select the Kerberos SSO menu extra and choose Change Password. You may also receive a notification that your password is expiring.
3. Enter your current password, then your new password. Make sure to use a new password that meets your organization's password requirements. Click OK.
4. After a brief pause, you'll see a dialog telling you that the password change was successful. If the password sync feature is enabled, your local account's password will be updated to match your new Active Directory password.

Using the Kerberos SSO menu extra—macOS

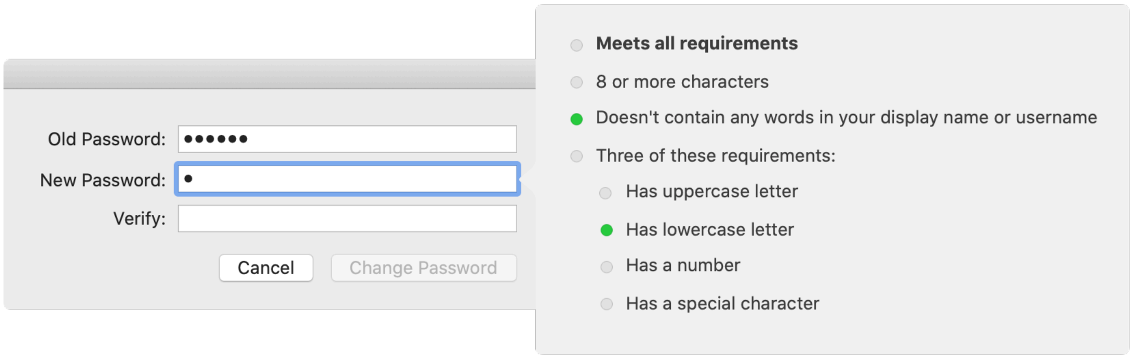
The Kerberos SSO menu extra provides easy access to useful information about your account and functions of the extension. You'll see it as a gray or black key in the menu bar on the top right.

To get status information about your account, start by looking at the Kerberos SSO menu extra icon and noting its color. If the key is gray, you're not signed in to the extension. If the key is black, you're signed in. After selecting the key, you'll see the account you're signed in with, as well as how many days you have until your password expires. The menu also allows you to sign in, sign out, and change your password.

Advanced Functions

Live password testing

In many Active Directory configurations, the Kerberos SSO extension can test new user passwords as they enter them and tell users what password requirements they must meet to change their passwords. When configured, the user will see this view when entering the new password:



Old Password: ••••••

New Password: •

Verify:

○ **Meets all requirements**

○ 8 or more characters

● Doesn't contain any words in your display name or username

○ Three of these requirements:

- ○ Has uppercase letter
- ● Has lowercase letter
- ○ Has a number
- ○ Has a special character

To use this feature, your Active Directory domain must use only standard Active Directory password policies. By default, Active Directory allows an administrator to require that a password be complex and a certain length. To learn about what constitutes a complex password, see technet.microsoft.com/en-us/library/cc786468(v=ws.10).aspx.

Note: You may not be able to use this feature if your domain uses third-party tools or DLLs to extend standard Active Directory password policy. For example, if you're not allowed to use certain words other than your user name in your password or you must use a specific amount of special characters in your password, you might be using third-party password policy extensions. If you're unsure, ask your Active Directory administrator for more information.

If your organization's Active Directory domain meets the requirements, you can enable live password testing. In your Kerberos SSO extension configuration profile, set the following parameters:

Parameter	Key	Type	Value	Optional
Require complex passwords	pwReqComplexity	Boolean	YES	No
Required password length	pwReqLength	Integer	Number	Yes
Reuse previous password limit	pwReqHistory	Integer	Number	Yes
Minimum password age	pwReqMinAge	Integer	Number	Yes

Live password testing has some limitations. It can't test if a password has already been used. It's also unable to test if your password contains your Active Directory display name if you don't already have a Kerberos TGT. This may happen if you're setting your password for the first time or if your password has expired. All other tests will work normally.

Password requirements display

If you can’t use live password testing, you can configure the Kerberos SSO extension to display a text string with your organization’s password requirements as users enter their new passwords. In your Kerberos SSO extension configuration profile, set “pwReqText” to a string containing the text you want to display to a user during password changes.

Changing or disabling password functionality

Some organizations may not be able to use the standard password change functionality of the Kerberos SSO extension, since they don’t allow password changes against Active Directory. In your Kerberos SSO extension configuration profile, set “allowPasswordChanges” to FALSE to disable this functionality.

Password change website support—macOS

The Kerberos SSO extension can be configured to open a password change website in the default browser when the user selects “Change password” or acknowledges a password expiration notice. Apple recommends using this feature only when using a local account, as mobile accounts are not supported.

In your Kerberos SSO extension configuration profile, set “pwChangeURL” to the URL of your password change website. Once users have changed their passwords, they must sign out of the Kerberos extension, then sign back in with their updated passwords. If local password sync is enabled, users are guided through bringing their passwords back in sync.

Password sync—macOS

The Kerberos SSO extension can set the local account password to match a user’s Active Directory password. Enable this feature by setting “syncLocalPassword” to TRUE in the Custom Configuration section of your Kerberos SSO extension configuration profile.

Password sync encompasses two basic functions. First, when the user uses the Kerberos SSO extension to change passwords, this feature sets their local password to match their Active Directory password. Should the local and Active Directory passwords fall out of sync, the Kerberos SSO extension brings them back in sync using the following:

- Upon enabling password sync, and upon every subsequent connection attempt by the Kerberos SSO extension, the dates that users last changed their local and Active Directory passwords are compared to cached values. If the values match, the passwords are in sync and no action is needed. If they don’t match, the Kerberos SSO extension will prompt users for their local and Active Directory passwords. Once users supply their local passwords, the Kerberos SSO extension sets their local password to match their Active Directory password.
- Password changes work in a similar fashion. When users perform a password change with the Kerberos SSO extension, their old Active Directory passwords will be checked against the local accounts. If an old Active Directory password and the local password match, the Kerberos SSO extension changes both passwords. If they don’t match, only the Active Directory password is changed. Users are then prompted for their local passwords during the next connection attempt.

This feature has the following requirements:

- If users are logged in to their Mac computers with Active Directory—not local—accounts, password sync is disabled. This feature is intended for use only with local accounts; if users are logged in to their Mac computers with Active Directory accounts, this feature is unnecessary.
- If a password policy is being enforced on local accounts—for example, using a configuration profile or using the pwpolicy command—make sure the local password policy matches or is less strict than the Active Directory password policy. If local password policy is more strict than Active Directory policy, the Kerberos SSO extension may accept a password that meets Active Directory requirements but fails to set the local password, since the password doesn't meet local password requirements. If local password policy must be more strict than Active Directory password policy, you shouldn't use this feature.
- The local user name is different from the Active Directory user name—only passwords are set to match.

Smart card support—macOS

The Kerberos SSO extension supports the use of smart card–based identities for authentication. Smart cards must have a CryptoTokenKit driver available; tokend-based drivers aren't supported. macOS 10.15 includes support for the PIV standard, which is widely used by the U.S. government.

Before beginning, make sure your Active Directory domain is configured to support smart card authentication. The process for enabling smart card authentication to Active Directory is out of the scope of this document. Refer to Microsoft's documentation for additional details.

To sign in to the Kerberos SSO extension with a smart card, follow these steps:

1. Click the Options menu, then select "Use a smart card."
2. When you see the Identity button, insert your smart card and click the button.
3. Choose the identity you want to authenticate with, click OK, then click Sign In.
4. Enter your PIN when prompted.

If the Kerberos SSO extension needs to acquire a Kerberos TGT, you'll be asked to insert your smart card and enter your PIN. More information about smart card support in macOS is available by running "man SmartCardServices" in the Terminal.

Distributed notifications—macOS

The Kerberos SSO extension posts distributed notifications when various events occur. Apps and services in macOS use distributed notifications to tell other apps and services that an event has occurred. An app or service listening for this event can take some action when it occurs.

An administrator can use this functionality to perform some action when certain events occur. For example, an administrator may want to run a script every time the Kerberos SSO extension acquires a new Kerberos credential.

The Kerberos SSO extension simply posts distributed notifications when specified events occur. It doesn’t run any actions when those events occur. The administrator must provide a tool to listen for these notifications and run actions when they occur.

The appendix contains an example of a script and launchd property list (.plist) that can listen for notifications and run actions. Modify this example as needed for your deployment.

Below are the distributed notifications posted by the Kerberos SSO extension:

Name	When posted
com.apple.KerberosPlugin.ConnectionCompleted	The Kerberos SSO extension has run its connection process.
com.apple.KerberosPlugin.ADPasswordChanged	The user has changed the Active Directory password with the extension.
com.apple.KerberosPlugin.LocalPasswordSynced	The user has brought the Active Directory and local passwords in sync.
com.apple.KerberosPlugin.InternalNetworkAvailable	The user has connected to a network where the configured Active Directory domain is available.
com.apple.KerberosPlugin.InternalNetworkNotAvailable	The user has connected to a network where the configured Active Directory domain is not available.
com.apple.KerberosExtension.gotNewCredential	The user has acquired a new Kerberos TGT.
com.apple.KerberosExtension.passwordChangedWithPasswordSync	The user has changed the Active Directory password, and the local password has been updated to match the new Active Directory password.

Command line support—macOS

Administrators can use a command line tool called *app-sso* to control the Kerberos SSO extension and access useful information. For example, they can use the tool to initiate sign-in, password changes, and sign-out. It also can print useful information, like the currently signed-in user, the computer's current Active Directory site, the user's network home share, when the user's password expires, and a variety of other useful information in property list or JSON format. This information can be parsed and uploaded to a Mac management solution for inventory and other purposes.

For more information on using app-sso, run "app-sso -h" in the Terminal app.

Mobile accounts—macOS

The Kerberos SSO extension doesn't require that your Mac be bound to Active Directory or that the user be logged in to the Mac with a mobile account. Apple suggests you use the Kerberos SSO extension with a local account. Local accounts work best with the recommended deployment model for macOS and are the best choice for today's Mac users, who may intermittently connect to your organization's network. The Kerberos SSO extension was specifically created to enhance Active Directory integration from a local account.

However, should you choose to continue using mobile accounts, you can still use the Kerberos SSO extension. This feature has the following requirements:

- Password sync doesn't work with mobile accounts. If you use the Kerberos SSO extension to change your Active Directory password and you're logged in to your Mac with the same user account you're using with the Kerberos SSO extension, password changes function as they do from the Users & Groups preference pane. But if you perform an external password change—meaning you change your password on a website, or your help desk resets it—the Kerberos SSO extension can't bring your mobile account password back in sync with your Active Directory password.
- Using a password change URL with the Kerberos extension and a mobile account is unsupported.

Domain-realm mapping

An administrator may need to define a custom domain-realm mapping for Kerberos. For example, an organization may have a Kerberos realm named "ad.pretendco.com," but may need to use Kerberos authentication for resources in the "fakecompany.com" domain.

Note: The Kerberos implementation on Apple operating systems can automatically determine domain-realm mapping in almost all situations. It is very rare for an administrator to customize these settings.

Domain-realm mapping can be configured for the Kerberos SSO extension by following these steps:

1. In the Custom Configuration section of the Extensible SSO profile, add an object called domainRealmMapping. The object type should be Dictionary.
2. Set the key of this dictionary to the name of your realm in caps.
3. Set the value of this dictionary to be of type Array. The first value should be the name of your Kerberos realm in lowercase, beginning with a period. The second value should be the name of the domain needing to authenticate against this realm, again starting with a period. Add arrays as needed.

For further information, refer to the Kerberos documentation.

Transitioning from Enterprise Connect

Overview

The Kerberos SSO extension is intended to replace Enterprise Connect, a similar tool that many organizations already use. Most organizations transitioning from Enterprise Connect to the Kerberos SSO extension will follow these steps:

1. Build a configuration profile for the Kerberos SSO extension that provides similar functionality to your current Enterprise Connect profile.
2. Uninstall Enterprise Connect.
3. Deploy the new Kerberos SSO extension configuration profile.
4. Have users sign in to the Kerberos SSO extension.

Transitioning to the Kerberos SSO extension isn't required to upgrade your organization's Mac computers to macOS 10.15. Enterprise Connect functions as expected with macOS 10.15, but organizations should still plan on an eventual transition from Enterprise Connect.

Who shouldn't transition

The Kerberos SSO extension will meet the needs of the vast majority of organizations using Enterprise Connect. An organization that meets the following criteria, however, may not be able to transition from Enterprise Connect or may be able to only partially transition:

- An organization that currently has Mac computers running macOS 10.14 or earlier should leave Enterprise Connect running on these systems, and transition only Mac computers running macOS 10.15 to the Kerberos SSO extension. The Kerberos SSO extension and its associated configuration profile will function only on Mac computers running macOS 10.15. Upgrade these systems to macOS 10.15 to take advantage of the Kerberos SSO extension.
- An organization that uses a Mac management tool that doesn't support user-approved MDM enrollment.
- An organization that isn't using a management tool.
- An organization that uses an Active Directory functional level of Windows Server 2003 or earlier.

Building a Kerberos SSO extension configuration profile

You'll need to build a configuration profile for the Kerberos SSO extension that's similar to your Enterprise Connect configuration profile. Many preference keys in your current Enterprise Connect configuration profile have equivalents in a Kerberos SSO extension profile. Start by reviewing the table below, which contains a map of Kerberos SSO extension equivalents to Enterprise Connect preference keys:

Enterprise Connect	Kerberos SSO extension	Notes
adRealm	Realm	Realm should be in all caps.
Automatic login (enabled by default)	allowAutomaticLogin	Add to Custom Configuration section. It must be set to True for automatic login to work.
disablePasswordFunctions	allowPasswordChange	Add to Custom Configuration section. Set to False to disable password changes.
passwordChangeURL	pwChangeURL	Add to Custom Configuration section.
passwordExpireOverride	pwExpireOverride	Add to Custom Configuration section.
passwordNotificationDays	pwNotificationDays	Add to Custom Configuration section.
prepopulatedUsername	principalName	Add to Custom Configuration section.
pwReqComplexity	pwReqComplexity	Add to Custom Configuration section.
pwReqHistory	pwReqHistory	Add to Custom Configuration section.
pwReqLength	pwReqLength	Add to Custom Configuration section.
pwReqMinimumPasswordAge	pwReqMinAge	Add to Custom Configuration section.
pwReqText	pwReqText	Add to Custom Configuration section. Supply a string of text to display instead of a path to a RTF file.
syncLocalPassword	syncLocalPassword	Add to Custom Configuration section.

Note: Some preference keys in your Enterprise Connect configuration profile may not be listed here. They may refer to functionality that's no longer needed in the Kerberos SSO extension or that's no longer supported.

Uninstalling Enterprise Connect

Running the Kerberos SSO extension and Enterprise Connect concurrently on the same computer isn't supported. After you transition to the Kerberos SSO extension, uninstall Enterprise Connect. You'll need administrative rights to perform the uninstall. To uninstall Enterprise Connect, follow the steps below:

Enterprise Connect 2.0 and later

1. Unload the Enterprise Connect agent launching the Terminal app and running "launchctl unload /Library/LaunchAgents/com.apple.ecAgent" as the currently logged-in user.
2. Quit the Enterprise Connect menu extra by launching the Terminal app and entering "killall Enterprise\ Connect\ Menu" in the Terminal app.
3. Delete the Enterprise Connect app from the Applications folder.
4. Delete the Enterprise Connect launchd .plist at /Library/LaunchAgents/com.apple.ecAgent.plist.

Enterprise Connect 1.9.5 and earlier

1. Quit Enterprise Connect by entering "killall Enterprise\ Connect" in the Terminal app.
2. Delete the Enterprise Connect app from the Applications folder.

The appendix provides a sample script that removes any version of Enterprise Connect.

Enterprise Connect script triggers

Enterprise Connect can run scripts when certain events occur. For example, Enterprise Connect can run a script when it completes its connection process or when the user performs a password change. The Kerberos SSO Extension handles scripts differently from Enterprise Connect. It doesn't directly run scripts. Instead, it posts a distributed notification when an event occurs, which another process can listen for, then run a script. See the "Advanced Functions" section of this document for details.

Below are references to Enterprise Connect's script triggers and their equivalent distributed notifications in the Kerberos SSO extension:

Enterprise Connect	Kerberos SSO extension
auditScriptPath	com.apple.KerberosPlugin.InternalNetworkAvailable
connectionCompletedScriptPath	com.apple.KerberosPlugin.ConnectionCompleted
passwordChangeScriptPath	com.apple.KerberosPlugin.ADPasswordChanged

Network shares

The Kerberos SSO extension doesn't support the handling of network shares, like the user's network home directory. You can replace much of this functionality with scripts.

Appendix

Device Management Profile: ExtensibleSingleSignOnKerberos

developer.apple.com/documentation/devicemanagement/extensiblesinglesignonkerberos?language=objc

Mobile Device Management Protocol Reference

developer.apple.com/business/documentation/MDM-Protocol-Reference.pdf

Device Management Profile: ExtensibleSingleSignOnKerberos.ExtensionData

developer.apple.com/documentation/devicemanagement/extensiblesinglesignonkerberos/extensiondata?language=objc

Sample script—Processing distributed notifications

The Kerberos SSO extension posts a variety of distributed notifications when different events occur, like when the user changes a password or the corporate network goes online. As an administrator, you can use a script or app to listen for these notifications and take some action when they're posted, like running a script or shell command.

Below is a sample script that can run scripts or commands when notifications are posted. It should be executed as a LaunchAgent to run as the logged-in user or LaunchDaemon to run as root. The script takes two required parameters:

- **-notification** is the name of the distributed notification you want to listen for. See page 11 for examples.
- **-action** is the action you want to execute when the distributed notification is posted. An example is "sh /path/to/script.sh."

To run the script, you must install the developer command line tools. An installer package for these tools is available on the Apple Developer site.

```
#!/usr/bin/swift

import Foundation
import AppKit

class NotifyHandler {
    var notification: String
    var action: String
    init(notification: String, action: String) {
        self.notification = notification
        self.action = action
    }
    func observe() {
        DistributedNotificationCenter.default.addObserver(
            forName: Notification.Name(notification),
            object: nil,
            queue: nil,
            using: self.gotNotification(notification:)
        )
    }
    func gotNotification(notification: Notification) {
       let task = Process()
        task.launchPath = "/bin/zsh"
        task.arguments = ["-c", self.action]
        task.launch()
    }
}

let app = NSApplication.shared
```

```
class AppDelegate: NSObject, NSApplicationDelegate {
    func applicationDidFinishLaunching(_ notification: Notification) {
        let scriptPath: String = CommandLine.arguments.first!
        guard let notificationName = UserDefaults.standard.string(forKey: "notification") else {
            print("\(scriptPath): No notification passed, exiting...")
            exit(1)
        }
        guard let actionPath = UserDefaults.standard.string(forKey: "action") else {
            print("\(scriptPath): No action passed, exiting...")
            exit(1)
        }
        let nh = NotifyHandler.init(notification: notificationName, action: actionPath)
        nh.observe()
    }
}

let delegate = AppDelegate()
app.delegate = delegate
app.run()
```

Below is a sample LaunchAgent property list. Replace the values in “<—- text —->” format with the specified values:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>KeepAlive</key>
        <true/>
        <key>Label</key>
        <string><--replace with unique label--></string>
        <key>RunAtLoad</key>
        <true/>
        <key>ProgramArguments</key>
        <array>
                <string><--replace with path to sample distributed notification script--></string>
                <string>-notification</string>
                <string><--replace with notification name--></string>
                <string>-action</string>
                <string><--replace with path to script to execute when notification is posted--></string>
        </array>
</dict>
</plist>
```

Sample script—Uninstalling Enterprise Connect

This sample script removes any version of Enterprise Connect. Execute it from a Mac management solution or manually. The script must run with root privileges.

```
#!/bin/zsh

# Unload the Kerberos helper from versions of EC prior to 1.6
if [[ -e /Library/LaunchDaemons/com.apple.Enterprise-Connect.kerbHelper.plist ]]; then
  launchctl bootout system /Library/LaunchDaemons/com.apple.Enterprise-Connect.kerbHelper.plist
  rm /Library/LaunchDaemons/com.apple.Enterprise-Connect.kerbHelper.plist
fi

# Remove the privileged helper from versions of EC prior to 1.6
if [[ -e /Library/PrivilegedHelperTools/com.apple.Enterprise-Connect.kerbHelper ]]; then
  rm /Library/PrivilegedHelperTools/com.apple.Enterprise-Connect.kerbHelper
fi

# Remove the authorization db entry from versions of EC prior to 1.6
/usr/bin/security authorizationdb read com.apple.Enterprise-Connect.writeKDCs > /dev/null

if [[ $? -eq 0 ]]; then
  security authorizationdb remove com.apple.Enterprise-Connect.writeKDCs
fi

if [[ -e /Library/LaunchAgents/com.apple.ecAgent.plist ]]; then
  # Enterprise Connect 2.0 or greater is installed
  # Unload ecAgent for logged in user and remove from launchd

  loggedInUser=$(scutil <<< "show State:/Users/ConsoleUser" | awk '/Name :/ && ! /loginwindow/ { print $3 }')
  loggedInUID=$(id -u $loggedInUser)

  launchctl bootout gui/$loggedInUID /Library/LaunchAgents/com.apple.ecAgent.plist
  rm /Library/LaunchAgents/com.apple.ecAgent.plist

  # Quit the menu extra
  killall "Enterprise Connect Menu"
fi

# Finally, remove the Enterprise Connect app bundle
rm -rf /Applications/Enterprise\ Connect.app
```