A guide for configuring the macOS Catalina Kerberos Single Sign-On Extension

Contents

To follow along with this guide you will need the following:

1. Mac computer running macOS Catalina 10.15 or later.
2. Jamf Pro Sever 10.15 or later.
3. Microsoft Windows Active Directory Server 2008-R2 or later (Note: Microsoft Azure Active Directory is NOT supported.)
4. Download Scripts, LaunchAgents, and an app named sso_bundle to monitor DistributedNotifications at this link: https://hcsonline.com/images/files/KerberosSSO.zip
5. A text editor that doesn't change the contents of the text you enter. For example, we recommend BBEdit, Code Runner, Atom, or Xcode, and specifically recommend against using Microsoft Word or even TextEdit unless it's in plain text mode.

Special thanks to the people below for all their help with this guide.
• Allen Golbig • Eric Hemmeter • Francois Tiffreau • Mike Lynn

Apple documentation for more information on the Kerberos SSO Extension:
https://www.apple.com/business/docs/site/Kerberos_Single_Sign_on_Extension_User_Guide.pdf
https://developer.apple.com/documentation/devicemanagement/extensiblesinglesignonkerberos/extensiondata
https://support.apple.com/guide/mdm/single-sign-on-extensions-mdmfd9cdf845/web
https://developer.apple.com/documentation/devicemanagement/extensiblesinglesignonkerberos
https://developer.apple.com/videos/play/tech-talks/301/

What is a Kerberos Single Sign-on Extension?
The Kerberos single sign-on (SSO) extension on macOS Catalina10.15 will log users into native apps (for apps that support Kerberos authentication) and sync local user passwords with a directory service such as Microsoft Active Directory. With the Kerberos SSO extension, users do not have to provide their user name and password to access native apps, file servers, proxy servers, and URLs that support Kerberos authentication. The Kerberos SSO extension is sandboxed (this guide explains the ramifications of sandboxing) and requires a mobile device management (MDM) solution (that supports the Extensible Single Sign-on (SSO) configuration profile payload) to enable the extension. The Kerberos SSO extension is included in MacOS Catalina, iOS 13, and iPad OS. It works with local and mobile accounts and supports smart card authentication.

What is the difference between the Kerberos Single Sign-on Extension and Enterprise Connect?
Enterprise Connect was the application that inspired the creation of the Kerberos SSO Extension. The Kerberos SSO Extension is the replacement for Enterprise Connect, which will only be maintained by Apple until the fall of 2020. Apple has announced it has no plans to add any new features to Enterprise connect going forward. Some of the differences between Enterprise Connect and the Kerberos SSO Extension include the Kerberos SSO extension requires an MDM, is not an application, and is sandboxed.

Features of the Kerberos Single Sign-on Extension

Feature	Kerberos Single Sign-on Extension	Enterprise Connect
Kerberos Support	*Yes*	*Yes*
Password Changes	*Yes*	*Yes*
Password Sync	*Yes*	*Yes*
Command Line Tool	*Yes - app-sso*	*Yes - eccl*
Script Execution	*Yes*	*Yes*
Included in MDM Specification	*Yes*	*No*
Availability	*Include with macOS, iOS13, iPadOS 13*	*Requires Apple Pro Services, macOS*
Setup Notifications	*Not Needed*	*Yes*
Share Mounting	*via a script*	*Yes*
Branding	*No*	*Yes*
LDAP Proxy	*Using ldapsearch*	*Yes*

Section 1: Configure the Kerberos Single Sign-on Extension with Jamf Pro

In this section you'll create a file in the plist format that contains the settings for the Kerberos SSO extension. Below is a sample file that this guide uses as an example. A full list of Key Value Pairs for the plist can be found here:

https://developer.apple.com/documentation/devicemanagement/extensiblesinglesignonkerberos/extensiondata

1. Open a new blank document in a text editor that's designed for coding, as explained in the requirements section of this guide.

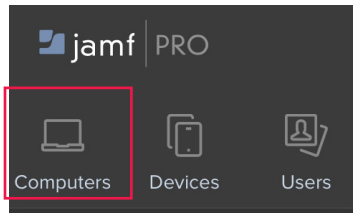
2. Copy the text below, and paste it in to your document.

```
<?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>allowAutomaticLogin</key>
  <true/>
  <key>isDefaultRealm</key>
  <false/>
  <key>pwNotificationDays</key>
  <integer>15</integer>
  <key>pwReqComplexity</key>
  <true/>
  <key>requireUserPresence</key>
  <false/>
  <key>syncLocalPassword</key>
  <true/>
  <key>useSiteAutoDiscovery</key>
  <true/>
</dict>
</plist>
```

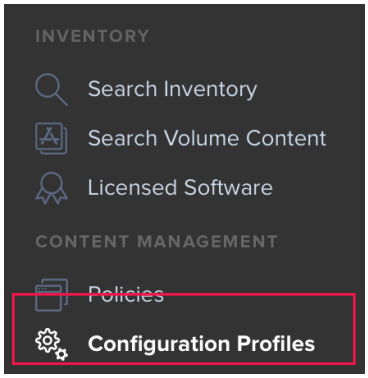
3. Save the document (this guide uses the Desktop as an example location) with a name that has the file extension of .plist.

4. With a web browser, log in to your Jamf Pro server.

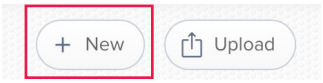
5. In the upper-left corner of the web browser, select Computers.



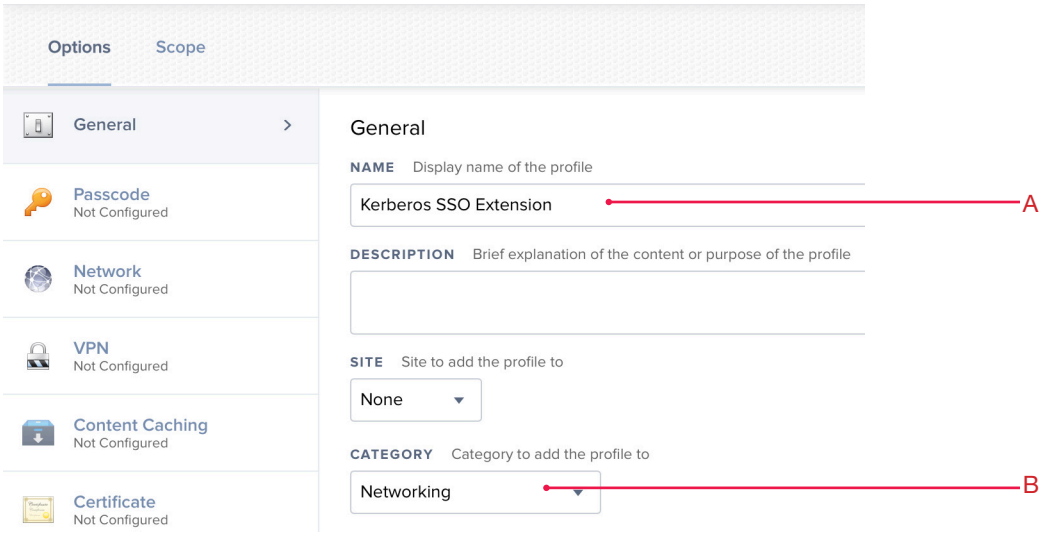
6. Select Configuration Profiles.



7. Click New.

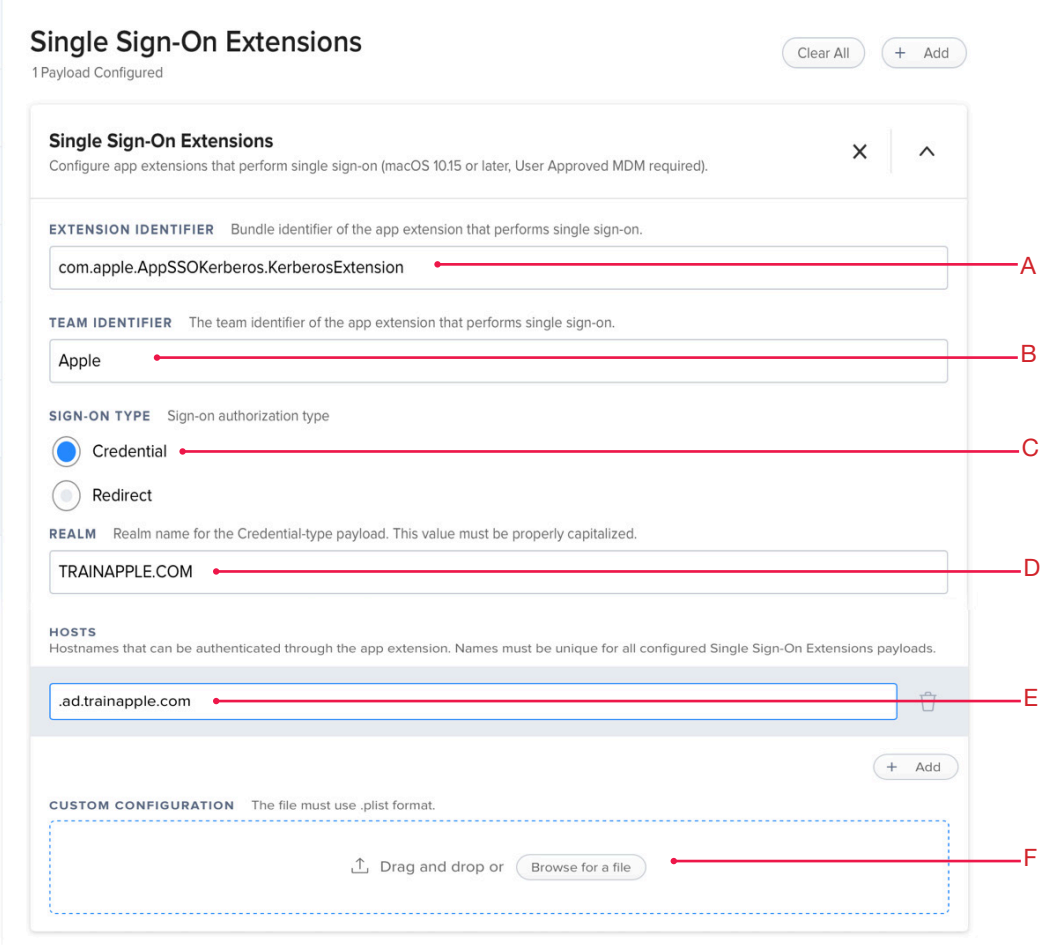


8. In the Options tab, select the General Payload.
9. Configure the following settings:
   A. Name: **Kerberos SSO Extension**
   B. (Optional): Choose a Category



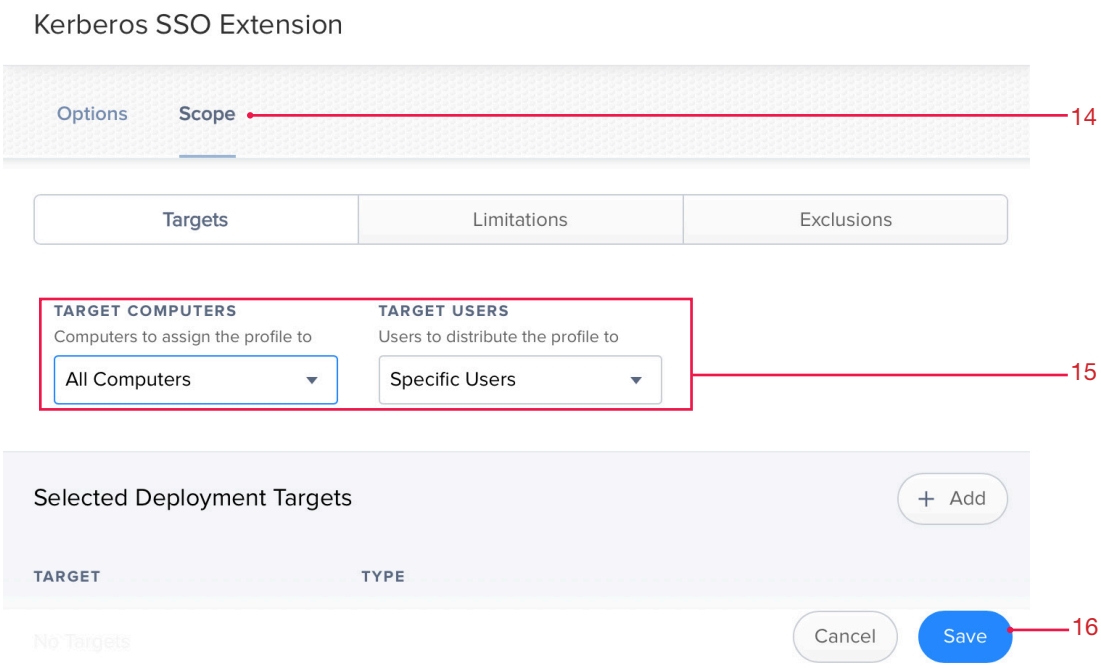
10. Select Single Sign-On Extensions.
11. Click Add.

12. Configure the following settings:
    A. Extension Identifier: **com.apple.AppSSOKerberos.KerberosExtension**
    B. Team Identifier: Apple
    C. Sign-On Type: Credential
    D. Realm: Enter your Kerberos realm in all CAPS.
    E. Hosts: Enter the domain that hosts resources that use Kerberos. NOTE: the value you enter must start with a leading period. For example, this guide uses .ad.trainapple.com.
    F. Custom Configuration: Drag the plist file you created earlier into the "Custom Configuration" field, or click "Browse for a file" then navigate to the plist file you created.

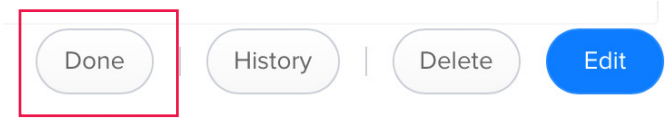


13. Confirm that the Custom Configuration section displays the values from your plist file.

14. Select the Scope tab.
15. Configure the scope according to your needs (this guide uses All Computers because it's not a production environment and only test computers are enrolled).
16. Click Save.

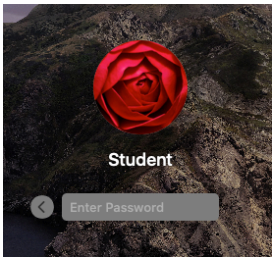


17. Click Done.



Section 2: Test the Kerberos SSO Extension with a Mac Computer
This section requires a Mac Computer enrolled in a Jamf Pro server (Jamf Pro 10.15 or later) with the Kerberos SSO extension configuration profile applied to the Mac from Jamf Pro.

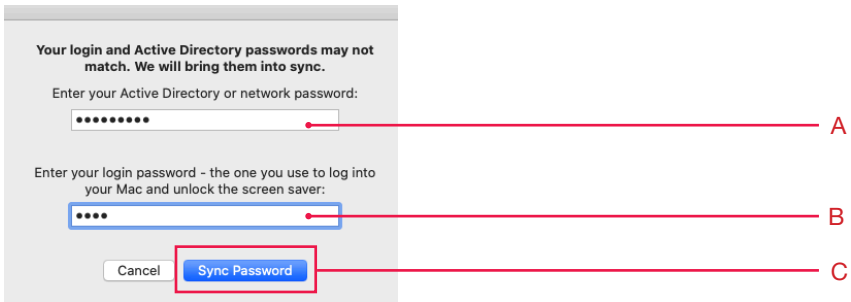
1. Log in to the Mac Computer.



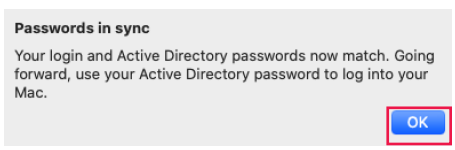
2. Confirm that your Mac displays the dialog below, and perform the following tasks:
   A. Enter your Active Directory credentials.
   B. Click Sign In.

3. At the message below, select Yes.

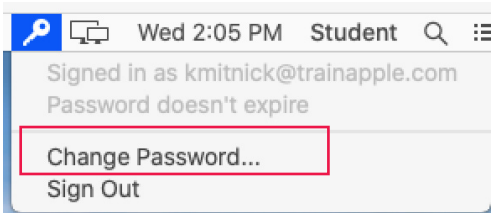
4. At the password sync message:
   A. Enter your Active Directory password.
   B. Enter your local login password.
   C. Click Sync Password.



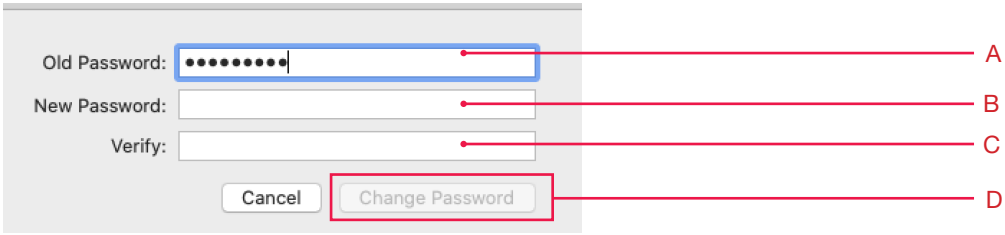
5. Click OK.

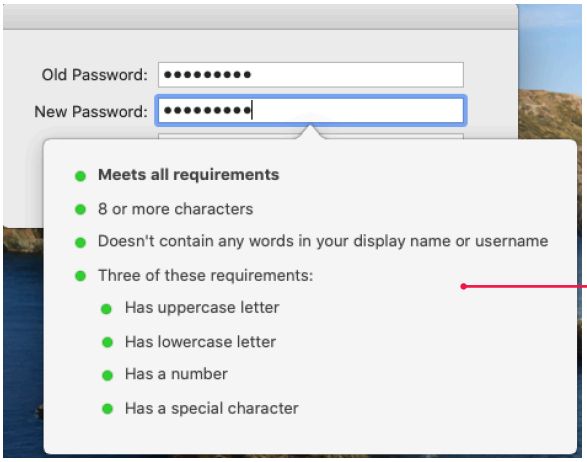


6. Click Kerberos SSO menu item (looks like a key in the menu bar). This displays the name of the user who is signed in to Active Directory, their password expiration, and allow you to change the Active Directory Users Password.
7. Choose Change Password.



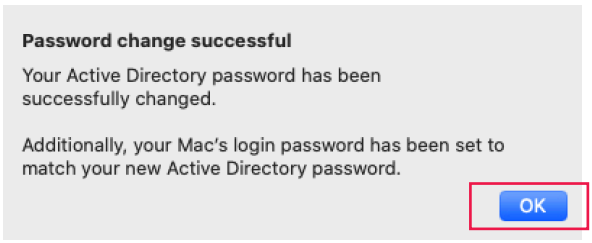
7. Enter the following:
   A. Old Password: Enter your current Active Directory Password.
   B. New Password: Enter a new Active Directory Password.
   C. Verify: Enter the new Active Directory Password again.
   D. Click Change Password.





NOTE: When you enter a new password, a window displays all the requirements for the new password, and displays a green status indicator for each requirement that the new password satisfies.

8. At the password change successful message, Click OK.
   NOTE: This changes the local Mac password in order to keep it in sync with the Active Directory password.



Section 3: Introduction to Using Command Line Tools for the Kerbereos SSO Extension

This section covers the following command-line tools to you can use for troubleshooting the Kerbreos SSO extension and Kerberos in general:

- app-sso (explained next)
- klist (list Kerberos tickets)
- kdestroy (destroy Kerberos tickets)
- kinit (obtain a Kerberos ticket)

app-sso is the command line tool for the for the Kerberos SSO extension. There is no **man** page for **app-so** however if you open Terminal and enter **app-sso -h** it will return all the options for the tool. Example below:

```
app-sso -h

Usage:
app-sso [options]

Options:
-a, --authenticate REALM [opts] Displays the login dialog for the specified realm,
      or end with success if already authenticated.
-u, --username USERNAME   The username for authentication.
-f, --force  Display the login screen regardless if already authenticated.
-q, --quiet  Supress the realm information after login.
-d, --logout REALM   Performs a logout for the specified realm.
-c, --changepassword REALM   Displays the change password dialog for the specified
realm.
-l, --listrealms   Displays the list of configured realms.
-i, --realminfo REALM   Displays the information for the specified realm.
-v, --verbose   List the complete site code cache in the results.
-s, --sitecode REALM   Performes a site lookup for the specified realm.
-v, --verbose   List the complete site code cache in the results.
-r, --reset {REALM}   Resets the cache for all realms or the specified realm.
-k, --keychainoption REALM   Resets the save to keychain option for the specified
realm.
-j, --json   Formats the output as json rather than plist format.
-h, --help   Prints this help.
```

NOTE: You can use each option with a single dash and a letter, or two dashes and a complete word. For example, -h or --help.

In this section, most steps contain the following:

- a command that you should run by entering the command then pressing Return.
- an explanation of what to expect in the results of the command.
- example results from running the command ; your results will differ from the example results.

1. If Terminal is not already open, open Terminal. You can use Spotlight to open it.

2. Run the following command:

```
app-sso -i trainapple.com
```

The output is in XML format by default.

```
<?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>gss_cred_uuid</key>
  <string>5E414EBD-5B3B-6151-EFC1-11B1A54FBC7F</string>
  <key>home_directory</key>
  <string>\\AD\UserHomes\kmitnick</string>
  <key>login_date</key>
  <string>2020-01-24T23:22:50Z</string>
  <key>realm</key>
  <string>TRAINAPPLE.COM</string>
  <key>site_code</key>
  <string>Default-First-Site-Name</string>
  <key>upn</key>
  <string>kmitnick@TRAINAPPLE.COM</string>
  <key>user_name</key>
  <string>kmitnick</string>
</dict>

</plist>
```

3. Run the following command (using your domain instead of the sample domain):

```
app-sso -i trainapple.com -j
```

The output is in JSON format and a litte easier to read. All you need to do is add the **-j** to the end of the command to get the JSON format.

```
{
 "site_code" : "Default-First-Site-Name",
 "home_directory" : "\\\\AD\\UserHomes\\kmitnick",
 "upn" : "kmitnick@TRAINAPPLE.COM",
 "realm" : "TRAINAPPLE.COM",
 "user_name" : "kmitnick",
 "gss_cred_uuid" : "5E414EBD-5B3B-6151-EFC1-11B1A54FBC7F",
 "login_date" : "2020-01-24T23:22:50Z"
}
```

The **app-sso** command line tool can be very useful for gathering information and parsing it out to be used in scripts or Extension Attributes in Jamf Pro.

4. Run the following command:

```
klist
```

The **klist** command lists the Kerberos ticket information for a user.

```
Credentials cache: API:57774A53-9DFA-455C-8FA5-B417BDB9BEF5
    Principal: kmitnick@TRAINAPPLE.COM

 Issued          Expires         Principal
Jan 24 18:22:51 2020 Jan 25 04:22:50 2020 krbtgt/TRAINAPPLE.COM@TRAINAPPLE.COM
Jan 24 18:22:51 2020 Jan 25 04:22:50 2020 ldap/ad.trainapple.com@TRAINAPPLE.COM
```

5. Run the following command:

   ```
   kdestroy
   ```

   This command removes all Kerberos tickets. There is no output unless there is an error.

6. Run the **klist** command again.

   ```
   klist
   ```

   Because there is no more Kerberos tickets to destroy, you'll see the following result:

   ```
   klist: krb5_cc_get_principal: No credentials cache file found
   ```

7. Run the following command (and use your account name and Kerberos realm in ALL CAPS):

   ```
   kinit kmitnick@TRAINAPPLE.COM
   ```

   The **kinit** command will get a new Kerberos ticket granting ticket (TGT) for a user. Enter the password for the user when prompted. There will be no output unless there's an error.

8. Run the **klist** command.

   ```
   klist
   ```

   Provide the password when prompted.

   ```
   Credentials cache: API:055F15DF-9171-4162-8239-56F68A1B2AFB
       Principal: kmitnick@TRAINAPPLE.COM
    Issued          Expires         Principal
   Jan 24 18:53:42 2020 Jan 25 04:53:34 2020 krbtgt/TRAINAPPLE.COM@TRAINAPPLE.COM
   kmitnick@TRAINAPPLE.COM's password:
   ```

Section 4: Use Debug Logs

Use the following commands to enable the debug logs. If you need to troubleshoot an issue with the Kerberos SSO extension, enable the debug logs, reproduce the issue, then review the logs to identify the issue. To maintain optimal performance, and to reduce the amount of storage consumed by logging information, be sure to disable debug mode when you're done testing.

1. Run the following three commands to enable debug logging:

```
sudo log config --mode "level:debug,persist:debug" --subsystem com.apple.AppSSO
sudo log config --mode "level:debug,persist:debug" --subsystem com.apple.Heimdal
sudo log config --mode "level:debug,persist:debug" --subsystem org.h5l.gss
```

2. After enabling the debug logs, kill the KerberosExtension process. Run the command below:

```
pkill -9 KerberosExtension
```

3. Now you can reproduce your problem.
4. After you've reproduced your problem, run the following command to gather the log files.

```
sudo sysdiagnose
```

   After you press Enter, the **sysdiagnose** command shows you the progress of gathering logs. After the **sysdiagnose** process completes, it indicates the location where the output from the command is available (in a new folder inside /var/tmp).

5. Have a look at the log files to identify your issue. The Finder automatically displays the folder with the log files from the sysdiagnose command.
6. Run the three commands below to disable debug logging:

```
sudo log config --subsystem com.apple.AppSSO --reset
sudo log config --subsystem com.apple.Heimdal --reset
sudo log config --subsystem org.h5l.gss --reset
```

Section 5: Use Scripts

This section covers running scripts based on monitoring DistributedNotifications. Any language that can hook into the native DistributedNotifications API can monitor DistributedNotifications. You'll use sample scripts and a Swift app named sso_bundle to monitor DistributedNotifications.

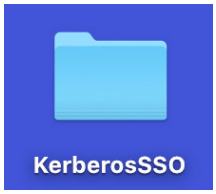
NOTE: The Scripts, LaunchAgents, and sso_bundle app that you downloaded at the beginning of this guide are required to compete this section. The scripts used in this guide are proof-of-concept scripts that display a dialog to illustrate that it worked.

Because the Kerberos SSO extension is sandboxed, it can't directly run scripts. Instead, it posts a distributed notification when an event occurs, which another process can listen for. When the other process notices the notification, it can then run a script. This guide coves three DistributedNotifications:

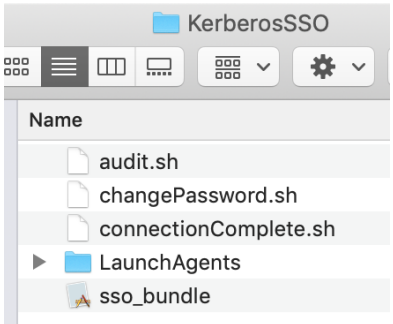
1. com.apple.KerberosPlugin.InternalNetworkAvailable - Use to run audit scripts.
2. com.apple.KerberosPlugin.ConnectionCompleted - Use to run scripts on a successful connection.
3. com.apple.KerberosPlugin.ADPasswordChanged - Use to run scripts on a successful password change.

You need a separate LaunchAgent for each Distributed Notification that you want to monitor. The single instance of the sso_bundle app can monitor each LaunchAgent.

1. In the Finder, open the KerberosSSO folder that you downloaded at the beginning of this guide.

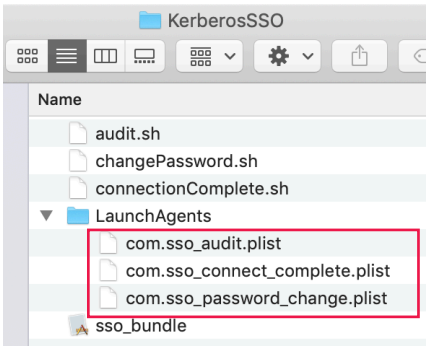


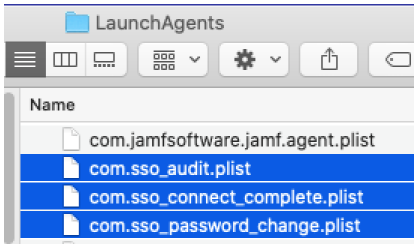
2. Confirm that inside the folder are scripts, LaunchAgents, and an app named sso_bundle.



3. Open the LaunchAgents folder, then move all the .plist files to the /Library/LaunchAgents folder. Enter your administrative credentials when prompted. Once done, delete the LaunchAgents folder from the KerberosSSO folder.

   NOTE: We recommend that you take the time to open one of the LauchAgents in a plain text editor of your choice to examine its contents for a better understanding of what they are doing.





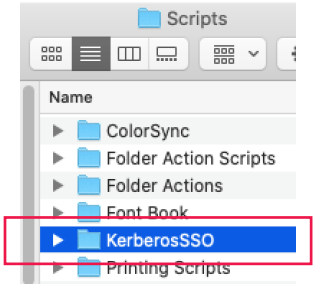
4. macOS requires that each LaunchAgent has root as its owner, and wheel as its primary group. In a following command you'll also set the file permissions to be 644, which corresponds to the following:
   - Read and write for the owner
   - Read only for the primary group
   - Read only for everyone else

   In Terminal run the two following commands:

```
sudo chown root:wheel \
/Library/LaunchAgents/com.sso_audit.plist \
/Library/ LaunchAgents/com.sso_connect_complete.plist \
/Library/LaunchAgents/com.sso_ password_change.plist

sudo chmod 644 \
/Library/LaunchAgents/com.sso_audit.plist \
/Library/LaunchAgents/com.sso_connect_complete.plist \
/Library/LaunchAgents/com.sso_password_change. plist
```

5. In the Finder, move the KerberosSSO folder to /Library/Scripts folder. Enter your administrative credentials when prompted.



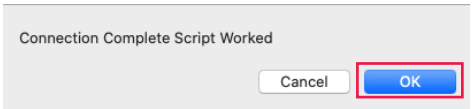
6. macOS requires that each of the scripts has root as its owner and wheel as its primary group. In a following command you'll also set the file permissions to be 755, which corresponds to the following:
   - Read, write, and execute for the owner
   - Read and execute for the primary group
   - Read and execute for everyone else

   Open Terminal and run the following two commands:

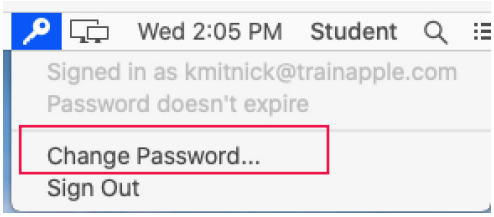
```
sudo chown -R root:wheel /Library/Scripts/KerberosSSO
sudo chmod -R 755 /Library/Scripts/KerberosSSO
```

7. Log out of the Mac and log back in as the same user. This is an easy way to load the LaunchAgents you just installed.
8. If you had the Kerberos SSO Extension set to auto login from a prior section, your Mac displays the “Connection Complete Script Worked” message below. Click OK. This message is shown because the sso_bundle app got a DistributedNotification that said the connection was completed successfully. That triggered the connection complete script to run.

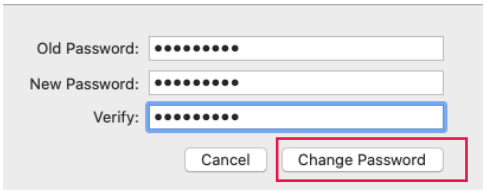
   NOTE: If you were not signed into the Kerberos Extension, sign in with your Active Directory credentials to see the message below.



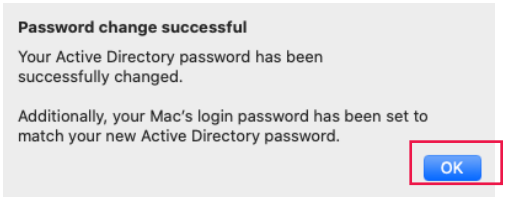
9. Click the Kerberos SSO menu item then choose Change Password.



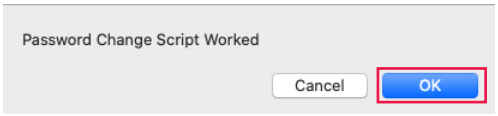
10. Enter your old Active Directory Password, then enter and verify your new password. Click Change Password.



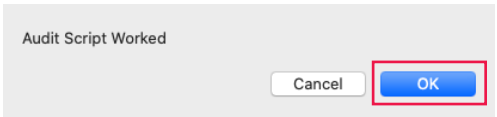
11. Click OK at the message below.



12. The password change script has successfully run. Click OK.



13. To test the audit script, simulate a successful network connection. There are a few ways to accomplish this. While the Kerberos SSO Extension is successfully connected to Active Directory, remove the Ethernet cable or turn off Wi-Fi then plug the Ethernet cable back in or turn on the Wi-Fi. Once the network connection is established the audit script will run. Click OK at the message below.



This completes the guide.