Salesforce Mobile App Security Guide

Version 3, 0

@salesforcedocs

Last updated: October 11, 2018

CONTENTS

CHAPTER 1 Introduction

This document describes the Salesforce mobile app (Salesforce for Android, Salesforce for iOS, and the mobile web), and addresses security concerns an enterprise may have when evaluating Salesforce for their organization.

CHAPTER 2 Salesforce App Architecture Overview

Salesforce uses the Lightning Platform, with app logic and database storage provided by Salesforce's hosted app servers and client apps. The Salesforce solution consists of the Salesforce app server, and either the client app or mobile web on the handheld mobile device. Supported operating systems are Apple iOS and Google Android.

The Salesforce client app communicates across the wireless network to display a subset of the user's Salesforce data on the handheld device. The client app or browser on the handheld device pulls feed data on demand to the device. This architecture provides a very high quality of service and a productive working experience for the end user.

Salesforce provides a sandboxed environment for a user to access Salesforce data from a mobile device, while an org admin can manage user access, even if the mobile device belongs to the user.

CHAPTER 3 Permissions

User Permissions

Access to Salesforce is "default on" and does not require an org admin to grant permission to use the app. Admins can edit profile and permission sets to revoke access to any user through the admin console . The Salesforce app provides access to data and functions based on the core permissions and rights defined for each user by their Salesforce admin. Mobile users are never able to view or access more than their permissions allow.

Mobile Device Permissions

When the Salesforce app is installed on a mobile device, the permissions requested vary for each OS.

- **Salesforce for Android:** At the time of installation, Android requires permission for:
  - Read-Only Contacts
  - Access Network State (WiFi or Cellular)
  - Authentication of Accounts
  - Internet
  - Read Phone State
  - Vibrate, Wake, and Lock
  - Write External Storage
  - GPS Location (Coarse and Fine)
  - Push Messaging
- **Salesforce for iOS:** After installation, iOS requests permission when the app needs access to each item listed. The user can approve or deny the request.
  - Contacts
  - Location
  - Photo Library
  - Camera
  - Microphone
  - Speech Recognition
  - Calendars

CHAPTER 4 Communication Security

The Salesforce app uses SSL/TLS v1.2 for Over-The-Air (OTA) communication encryption. All Salesforce OAuth authorization endpoints are HTTPS only.

Communication requests over HTTP or HTTPS below TLS v1.2 are denied by Salesforce servers, unless the org admin opts out of and unchecks “Require secure connections (HTTPS)” in the administration console.

CHAPTER 5 Authentication

In this chapter ...

All components of Salesforce require user authentication at the point and time of access. Salesforce utilizes OAuth2.0 for authentication through username/password or SSO (single sign-on) credentials.

OAuth Pairing

During the initial login, the device is uniquely identified and paired with the mobile user's account using the OAuth 2.0 protocol (http://tools.ietf.org/html/rfc6749). All requests to the Salesforce service are made using the OAuth token established through the pairing created during activation.

After initial login, there is no exchange of a password in the communication between the mobile client and the Salesforce server. For this reason, the Salesforce password is not stored on the device and is not required even when the password is changed or has expired.

A user obtains an access token and refresh token after successfully completing the OAuth User-Agent authentication. A user can use the refresh token to get a new access token (session ID). Upon logout, the OAuth access and refresh tokens are revoked, and the user set passcode is wiped (if passcode is enabled by org admin). The user is re-prompted to enter the username/password and reset the passcode.

The available refresh token expiration policies:

- Refresh token never expires.
- Refresh token expires immediately (for example, the refresh token is never valid).
- Refresh token expires if it isn't used for an amount of defined time (hours/days/months).
- Refresh token expires in defined amount of time (hours/days/months), regardless of use.

The default access token expiration schedule is set at 2 hours, but can be as short as 15 minutes or as long as 24 hours

Access Token Storage

- **Salesforce for iOS:** The encryption standard is AES with 256-bit key and 128-bit Initialization Vector. As for the encryption keys, these are secure random-generated 256-bit key and 128-bit Initialization Vector. The access token is stored in a secured keychain.
- **Salesforce for Android:** PBKDF2 produced AES-256 encrypted key derived from device unique Android ID and randomly generated string. Token is stored in Android's encrypted AccountManager.
- **Salesforce Mobile Web:** Access token is never stored on the mobile device. The mobile browser app requires a user to re-enter the username/password to obtain a new access token.

Refresh Token Storage

- **Salesforce for iOS:** The encryption standard is AES with 256-bit key and 128-bit Initialization Vector. As for the encryption keys, these are secure random-generated 256-bit key and 128-bit Initialization Vector. The refresh token is stored in a secured keychain.
- **Salesforce for Android:** PBKDF2 produced AES-256 encrypted key derived from device unique Android ID and randomly generated string. Token is stored in Android's encrypted AccountManager.
- **Salesforce Mobile Web:** The web server authentication flow for the mobile browser app doesn't use or store a refresh token on the device. The mobile browser app requires a user to re-enter the username/password to obtain a new access token.

Single Sign On (SSO)

Single sign-on is a process that allows network users to access all authorized network resources without having to log in separately to each resource. Single sign-on allows orgs to validate username/password against their user database or other client apps rather than having separate username/password managed by Salesforce.

Federated Authentication Support

When federated authentication is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce verifies an assertion in the HTTP POST request, and allows single sign-on if the assertion is true. This is the default form of single sign-on.

See "Configuring SSO for Mobile and Desktop Apps Using SAML and OAuth" for more information.

Delegated Authentication Support

When delegated authentication is enabled, Salesforce does not validate a user's password. Instead, Salesforce makes a Web services call to a customer org to establish authentication credentials for the user. Admins must request delegated authentication support to be enabled by Salesforce.

See "Understanding Delegated Authentication Single Sign-On" for more information.

Certificates and Keys

Salesforce certificates and key pairs are used for signatures that verify a request is coming from a customer org. They are used for authenticated SSL communications with an external web site, or when using a customer org as an Identity Provider. Customers only need to generate a Salesforce certificate and key pair if they're working with an external website that wants verification that a request is coming from a Salesforce org.

Salesforce offers two types of certificates:

- **Self-Signed:** A self-signed certificate is signed by Salesforce. Not all external websites accept self-signed certificates.
- **CA-Signed:** A CA-signed certificate is signed by an external certificate authority (CA). Most external websites accept CA-signed certificates. Customers must first generate the certificate signing request to send to a CA, and then import the signed version of the certificate before they can use it.

See "About Salesforce Certificates and Keys" for more information.

Identity Providers and Service Providers

An identity provider is a trusted provider that enables a customer to use single sign-on to access other websites. A service provider is a website that hosts apps. Customers can enable Salesforce as an identity provider, then define one or more service providers, so their users can access other apps directly from Salesforce using single sign-on. This can be a great help to users: instead of having to remember many passwords, they will only have to remember one.

Salesforce is automatically enabled as an identity provider when a domain is created. After a domain is deployed, admins can add or change identity providers and increase security for their organization by customizing their domain's login policy.

Enabling Salesforce as an identity provider requires a Salesforce certificate and key pair that is signed by an external certificate authority (CA-signed) or self-signed. If customers haven't generated a Salesforce certificate and key pair, one is automatically created for them when they enable Salesforce as an identity provider. They also have the option of picking an already generated certificate, or creating one.

Salesforce uses the SAML 2.0 standard for single sign-on and generates SAML assertions when configured as an identity provider.

See "About Identity Providers and Service Providers" for more information.

Inactivity Lock

Upon initial activation, Salesforce prompts the user to create an arbitrary passcode (if required by the org admin), which is used to unlock the app after reboot, or an admin defined period of inactivity (1, 5, 10, or 30 minutes).

The passcode lock protects lost or stolen devices that may have their wireless connection disabled, and can't have their OAuth token revoked.

Passcode Strength and Storage

- **Salesforce for iOS:** A PBKDF2 hash of the passcode is stored in the secure keychain, for passcode validation. The hashed passcode can be accessed only while the device is unlocked by the user (kSecAttrAccessibleWhenUnlockedThisDeviceOnly). The passcode is also used as a source of entropy for encryption operations within the app. Admins can configure required passcode length through the Salesforce Connected App.
- **Salesforce for Android:** PBKDF2 produced AES-256 encrypted key derived from device unique Android ID and randomly generated string. Tokens (access and refresh) are stored in Android's encrypted AccountManager. Admins can configure required passcode length through the Salesforce Connected App.

An extra layer of security is provided when an admin enables passcode lock. Locally stored data is erased after 10 failed attempts at entering the passcode. Users are required to log in again to continue using the app.

Salesforce Mobile Web: Users are prompted to re-enter username/password after 30 minutes of inactivity, or if they navigate to a different site or close the mobile browser.

Session Cookie

Session cookie is only used for Visualforce pages. It is derived from the OAuth Access Token and is scoped to the Visualforce page. The WKWebView/WebView stores it in the cache.

Restrict Device Platforms

Admins can restrict Salesforce app access through the admin console by blocking the Salesforce Connected App for either platform (iOS or Android).

Admins can also enable/disable mobile web through admin console. If the mobile web experience is disabled, the user is taken to the full Salesforce site from the mobile browser.

CHAPTER 6 Storage Security

In this chapter ...

A mobile device may be lost or stolen at any time. Since mobile devices are small and designed to be highly portable, they may not remain under the physical control of a trusted person. Therefore, Salesforce provides methods to secure the device data if it passes out of control of the user or the user's organization.

Salesforce has multiple levels of security at the handheld device level. First, device vendors provide the ability to enforce OS-level password access restrictions on any device apps or data. Users must be required to use the device protection in accordance with the owning enterprise's security policy. If the device is locked by a strong password, it is difficult for unauthorized persons to do anything with it.

Local Data Protection

The data stored locally on the device is saved in the device's embedded memory and never on an external memory card.

Mobile platforms don't generally allow data extraction from a local database. To make the system more secure, Salesforce does provide encryption on the device database.

Feed Database Encryption

Feeds are made up of feed items. A feed item is a piece of information posted by a user (for example, a poll) or by an automated process (for example, when a tracked field is updated on a record).

- **Salesforce for iOS:** Database encrypted via SQLCipher using 256-bit AES (CBC mode/PBKDF2 key derivation)

  Records pertaining to inactive feed item data are evicted from the database after 5 days have elapsed. Temporary files (such as viewed image attachments) are stored only in memory while used.

- **Salesforce for Android:** Database encrypted via SQLCipher using 256-bit AES (CBC mode/PBKDF2 key derivation)

  Records pertaining to inactive feed item data are evicted from the database after 5 days have elapsed. Temporary files (such as viewed image attachments) are stored only in memory while used.

- **Salesforce Mobile Web:** No encryption required. No data is stored locally on the device when using the mobile web.

Files and Attachments

A file or attachment is any file that a user uploads, shares, or attaches to posts, comments, or records. All file types are supported: documents, presentations, spreadsheets, PDFs, images, audio files, and video files.

- **Salesforce for iOS:** Files and attachments are stored on the device's file system in a double-encrypted format. We use the device's hardware encryption capability to encrypt the files while the device is locked and in addition we perform our own encryption using AES algorithm (128-bit block size and 256-bit key size). When the file is being viewed, there's a temporary unencrypted copy kept on the file system (removed when the 'viewing' operation is complete).
- **Salesforce for Android:** To store files offline, we require the user to enable device encryption and use the OS's file encryption system. If enabled, a passcode 5 or more alphanumeric characters adds an extra layer of file encryption. This allows the app to securely store local files.
- **Salesforce Mobile Web:**No encryption required. No data is stored locally on the device when using the mobile web.

Offline Sync

If Salesforce users lose their wireless connection, they can enable offline sync to navigate within the app and view most recent items.

- **Salesforce for iOS:** Database encrypted via SQLCipher using 256-bit AES (CBC mode/PBKDF2 key derivation).
- **Salesforce for Android:** Database encrypted via SQLCipher using 256-bit AES (CBC mode/PBKDF2 key derivation).
- **Salesforce Mobile Web:** Offline sync functionality isn't available in mobile web.

Remote Wipe

To minimize the risk of information loss when a device is compromised, an org admin can:

1. Disable a user completely (for example, termination of an employee) to remove access and wipe the data from the apps.

2. View the Connected Apps OAuth Usage report in the administration console to revoke the OAuth refresh token and associated access tokens. This wipes the app, which forces the user to reauthenticate (e.g. employee loses a phone).

CHAPTER 7 Mobile Device Management (MDM)

In this chapter ...

Both Salesforce for Android and Salesforce for iOS provide an extra level of security compliance through interoperation with the most popular MDM (mobile device management) suites. Salesforce for Android and iOS, with an MDM, give you enhanced functionality for distribution and control over your users' devices. The enhanced security functions when you combine Salesforce with an MDM include certificate-based authentication and automatic custom host provisioning.

Note: SAML 2.0 (security assertion markup language) must be enabled and configured for your organization.

There are prerequisites to implement enhanced security for Salesforce for Android.

- First, configure Android for Work for your org. Android for Work is a program that supports enterprise use of Android devices. See Android for Work to learn more about the program and Android for Work Help for setup information.
- Once Android for Work is set up, the next step is to configure your Mobile Device Management (MDM) suite. There are a multitude of MDM solutions in the market place. When you decide on the right product, work with your MDM provider to complete the configuration for your org.
- After you have Android for Work and your MDM suite up and running in your org, you're ready to implement the enhanced security features of Salesforce for Android.

Certificate-Based Authentication

Using certificates to authenticate simplifies provisioning your mobile users, and your day-to-day mobile administration tasks by eliminating usernames and passwords. Salesforce uses X.509 certificates to authenticate users more efficiently, or as a second factor in the login process.

MDM Settings for Certificate-Based Authentication

To enable certificate-based authentication for your mobile users, you need to configure key-value pair assignments through your MDM suite. Here are the supported keys:

Key	Data Type	Platform	Description
`RequireCertAuth`	Boolean	Android, iOS	If true, the certificate-based authentication flow initiates. **Android:** Uses the user certificate on the device for

Key	Data Type	Platform	Description
			authentication inside a webview. **iOS:** Redirects the user to Safari for all authentication requests.
`ManagedApp CertAlias`	String	Android	Alias of the certificate deployed on the device picked by the app for user authentication. Required for Android only.

Note: There's a minimum device OS version requirement to use certificate-based authentication. For Android, the minimum supported version is 5.0. For iOS, the minimum supported version is 7.0.

Once you save your key-value pair assignments, you can push the mobile app with the updated certificate-based authentication flow to your users via your MDM suite.

Automatic Custom Host Provisioning

You can now push custom login host settings to your mobile users. This spares your mobile users from having to manually type long URLs for login hosts—typically a frustrating and error-prone activity. You can configure key-value pair assignments through your MDM to define multiple custom login hosts for your mobile users.

MDM Settings for Automatic Custom Host Provisioning

To push custom login host configurations to your mobile users, you need to configure key-value pair assignments through your MDM suite. Here are the supported keys:

Key	Data Type	Platform	Description
`AppServiceHosts`	String, String Array	Android, iOS	Login hosts. First value in the array is the default host. **Android:** Requires https:// in the host URL. **iOS:** Doesn't require https:// in the host URL.

Key	Data Type	Platform	Description
`AppService HostLabels`	String, String Array	Android, iOS	Labels for the hosts. The number of `AppServiceHostLabels` entries must match the number of `AppServiceHosts` entries.
`OnlyShow AuthorizedHosts`	Boolean	Android, iOS	If true, prevents users from modifying the list of hosts that Salesforce can connect to.

Additional Security Enhancements

You can add an extra layer of security for your iOS users by clearing the contents of their clipboard whenever the mobile app is in the background. Users may copy and paste sensitive data as a part of their day-to-day operations, and this enhancement ensures any data they copy onto their clipboards are cleared whenever they background the app.

MDM Settings for More Security Enhancements

To clear the clipboards of your iOS users when the mobile app is in the background, you need to configure key-value pair assignments through your MDM suite. Here is the supported key:

Key	Data Type	Platform	Description
`ClearClipboard OnBackground`	Boolean	iOS	If true, the contents of the iOS clipboard are cleared when the mobile app is backgrounded. This prevents the user from accidentally copying and pasting sensitive data outside of the app.

Note: If the mobile app stops working unexpectedly, the copied data can remain on the clipboard. The contents of the clipboard are cleared once the user starts and backgrounds the mobile app.

This security functionality is available through Android for Android devices running OS 5.0 and greater, and that have Android for Work set up. Contact your MDM provider to configure this functionality for your Android users.

Sample Property List Configuration

Note: Setting key-value pair assignments through a plist is only available on iOS.

One method of setting key-value pair assignments is through an XML property list, or plist. The plist contains the key-value pair assignments that an MDM provider sends to a mobile app to enforce security configurations.

Here is a sample plist:

```
<?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>AppServiceHosts</key>
 <array>
  <string>host1</string>
  <string>host2</string>
 </array>
 <key>AppServiceHostLabels</key>
 <array>
  <string>Production</string>
  <string>Sandbox</string>
 </array>
 <key>RequireCertAuth</key>
 <true/>
 <key>ClearClipboardOnBackground</key>
 <false/>
 <key>OnlyShowAuthorizedHosts</key>
 <false/>
</dict>
</plist>
```

CHAPTER 8 Salesforce Connected App Security Attributes

Salesforce for Android and Salesforce for iOS provide an extra level of security compliance without the use of an MDM (mobile device management) provider. This lets you configure security attributes, such as disabling copy and paste or disabling sharing files outside of your org, for your users from Setup in the full Salesforce site.

Salesforce for Android and Salesforce for iOS are connected apps. As a result, you can control the users who have access to the apps, as well as other security policies. By default, all users in your organization can log in to Salesforce for Android and Salesforce for iOS.

You can control security and access policies for Salesforce for Android and Salesforce for iOS using settings components that are installed from the managed Salesforce connected apps package. These components need to be installed in Salesforce:

- Salesforce for Android
- Salesforce for iOS

These components are automatically installed when one of your users installs Salesforce from the App Store or Google Play on a mobile device and authenticates with your organization by logging in to the mobile app.

Alternatively, you can manually install the Salesforce and Chatter Apps connected apps package so you can review and modify the default security and access settings before rolling out Salesforce for Android and Salesforce for iOS to your users.

When the Salesforce connected apps components are installed, they're added to the Connected Apps page. (From Setup, enter *Connected Apps* in the Quick Find box, then select the option for managing connected apps.) Here, you can view and edit the settings for each of the apps, including controlling user access with profiles, permissions, and IP range restrictions. An error message is displayed if a restricted user attempts to log in to Salesforce for Android or Salesforce for iOS.

Several of the Salesforce app custom attributes have a default value that automatically applies when a user logs in to Salesforce for Android or Salesforce for iOS. If the default values are appropriate for your org, you're all set.

To change a default value, or configure an attribute that doesn't have a default setting, go to Setup in the full Salesforce site. Enter *Connected Apps* in the Quick Find box, select **Connected Apps**, then click **Salesforce for Android** or **Salesforce for iOS**. In the Custom Attributes section on the connected app page, click **New** and enter the attribute name and value.

To configure a security attribute, click **Salesforce for Android** or **Salesforce for iOS** from the Connected Apps page. In the Custom Attributes section on the connected app page, click **New** and enter the details for the attribute.

Important: Remember to wrap attribute values in quotation marks.

The following custom attributes are available for Salesforce for Android and Salesforce for iOS, which are also connected apps.

Attribute Key	Attribute Value	Platform	Description
`DISABLE_EXTERNAL_PASTE`	• `TRUE` • `FALSE`	Android, iOS	• If set to `TRUE`, lets users copy and paste within the Salesforce app, but disables copying within and pasting outside of the Salesforce app. • If set to `FALSE` (default if attribute value isn't defined), lets users copy and paste within and outside of the Salesforce app. • Note: The `DISABLE_EXTERNAL_PASTE` attribute doesn't affect Share extensions on iOS.

`FORCE_EMAIL_CLIENT_TO`	The email app's URI scheme.
	Can differ by platform. For example, here's an Android URI scheme example for Blue Mail, and an iOS URI scheme example for Gmail.
	Android:
	`https://play.google.com/store/apps/details?id=me.bluemail.mail&hl`
	iOS:
	`googlegmail:///co?to=`
		Android, iOS	If a user taps on an email action in the Salesforce app, the user is directed to the email app specified in the attribute value.
			You can specify one email app only.
			The attribute value you enter depends on the email app and the device platform.
			• For Android, use the URI listed in the Google Play Store for the desired email app.
			• For iOS, do an Internet search to locate the URI scheme for the desired email app. For example,

Attribute Key	Attribute Value	Platform	Description
			search for *iOS Mail URI scheme*.
SHOW_ONBOARDING_CAROUSEL	• TRUE • FALSE	iOS	• If set to TRUE, onboarding screens appear when users log into the Salesforce app. • If set to FALSE, disables onboarding screens when users log into the Salesforce app.
SHOW_OPEN_IN	• TRUE • FALSE	Android, iOS	• If set to TRUE, lets users share a file from the Salesforce app via a link to the file, or open a Salesforce file in a third-party app. • If set to FALSE, disables users from sharing a file from the Salesforce app or opening a Salesforce file in a third-party app.
SHOW_PRINT	• TRUE • FALSE	iOS	• If set to TRUE, lets users print from the Salesforce app. • If set to FALSE, disables printing from the Salesforce app.

Tip: Connected app attribute changes take effect when users force quit the Salesforce app or when they log in to a new session. To ensure that new or modified settings take effect for all users, we recommend that you revoke access to the Salesforce app so everyone is required to log in again.

We also recommend that you warn users about the changes you intend to make, especially if you're going to restrict activities that were previously available. The Salesforce app doesn't display messages or indicators that connected app settings have changed.

CHAPTER 9 Notes

- iOS: Prior to entering applicationDidEnterBackground, a benign splash screen is displayed to protect sensitive data from automatic iOS snapshotting (iOS uses automatic snapshotting for transition animations). The application prevents any snapshots of customer data during backgrounding.
- Security is not a binary (on/off), but implemented at different levels.
- Salesforce provides multiple levels of security; however, there's no application that can guarantee a completely secure system.