Mobile Application Security

Last Updated:2026.02.25

CONTENTS

Mobile Application Security

Mobile application security addresses any concerns you may have when evaluating Salesforce mobile apps for your organization.

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 the client app on the handheld mobile device. Supported operating systems are Apple iOS and Google Android.

The Salesforce client apps communicate across the wireless network to display a subset of the user's Salesforce data on the mobile device. The client app on the mobile device pulls feed data on demand to the device. This architecture provides a 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. Org admins can manage user access, even if the mobile device belongs to the user.

Permissions

Learn more about user and device permissions for the Salesforce mobile app.

User Permissions

Access to Salesforce is “default on” and doesn’t 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 mobile 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 mobile app is installed on a mobile device, the permissions requested vary for each operating system.

- Salesforce Mobile App for iOS:
  - Full Calendar Access
  - Basic Calendar Access
  - Camera
  - Contacts
  - Cross-App Tracking
  - Face ID / Biometrics
  - Always-On Location
  - Location While Using
  - Microphone
  - Photo Library (Read)
  - Photo Library (Write)
  - Speech Recognition
  - NFC Scanning

- Salesforce Mobile App for Android:
  - Take pictures and videos
  - Record audio / Use microphone
  - Precise location (GPS)
  - Approximate location (Network)
  - Read your contacts
  - Add or edit contacts
  - Read calendar events
  - View phone status and identity
  - Send notifications
  -

Full network access
- View network connections
- View Wi-Fi connections
- Control Near Field Communication
- Use fingerprint or face sensors
- Use fingerprint hardware
- Control vibration
- Change your audio settings
- Manage your device accounts
- Use account credentials
- Run tasks while the app is active
- Sync data in the background
- Run automatically at startup
- Keep phone from sleeping
- Download files without alerts
- Reorder running apps

Communication Security

The Salesforce mobile app always uses highest level of secure communications and encryption to safeguard your data.

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

Salesforce servers deny communication requests below TLS v1.2.

Mobile Application Authentication

All components of Salesforce mobile applications require user authentication at the point and time of access.

OAuth Pairing for Mobile Applications
Salesforce uses OAuth2.0 for mobile application authentication through username and password or SSO (single sign-on) credentials.

Single Sign On (SSO) for Mobile Applications
Single sign-on is a process that allows network mobile application users to access all authorized network resources without having to log in separately to each resource.

Certificates and Keys
Salesforce certificates and key pairs are used for signatures that verify a request is coming from a customer org. They're used for authenticated SSL communications with an external website, or when using a customer org as an Identity Provider.

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 only have to remember one.

Inactivity Lock

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

Session Cookie

Session cookie is only used for Visualforce pages.

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).

OAuth Pairing for Mobile Applications

Salesforce uses OAuth2.0 for mobile application authentication through username and password or SSO (single sign-on) credentials.

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’s no exchange of a password in the communication between the mobile client and the Salesforce server. For this reason, the Salesforce password isn’t stored on the device and isn’t 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 org admin enables passcode protection). The user is reprompted to enter the username and 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 or days or months).
- Refresh token expires in a defined amount of time (hours or days or 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 Mobile App for iOS: The encryption standard is AES with 256-bit key and 128-bit Initialization Vector. The encryption keys are a secure random-generated 256-bit key and 128-bit Initialization Vector. The access token is stored in a secured keychain.
- Salesforce Mobile App for Android: PBKDF2 produced AES-256 encrypted key derived from a device unique Android ID and randomly generated string. Token is stored in Android's encrypted AccountManager.

Refresh Token Storage

- Salesforce Mobile App for iOS: The encryption standard is AES with a 256-bit key and 128-bit Initialization Vector. The encryption keys are a secure random-generated 256-bit key and 128-bit Initialization Vector. The refresh token is stored in a secured keychain.
- Salesforce Mobile App for Android: PBKDF2 produced AES-256 encrypted key derived from a device unique Android ID and randomly generated string. The token is stored in Android's encrypted AccountManager.

Single Sign On (SSO) for Mobile Applications

Single sign-on is a process that allows network mobile application users to access all authorized network resources without having to log in separately to each resource.

Single sign-on allows orgs to validate username and password against their user database or other client apps rather than having separate username and 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. Federated authentication 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 doesn't 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 Salesforce to enable delegated authentication support.

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're used for authenticated SSL communications with an external website, or when using a customer org as an Identity Provider.

Customers must generate a Salesforce certificate and key pair only 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 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). The passcode 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 have their wireless connection disabled, and can't have their OAuth token revoked.

Passcode Strength and Storage

- Salesforce Mobile App 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 the required passcode length through the Salesforce Connected App.
- Salesforce Mobile App for Android: PBKDF2 produced AES-256 encrypted key derived from a device unique Android ID and randomly generated string. Tokens (access and refresh) are stored in Android's encrypted AccountManager. Admins can configure the 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.

Session Cookie

Session cookie is only used for Visualforce pages.

The session cookie 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.

Application Data Storage

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.

Remote Wipe
An org admin can restrict access to minimize the risk of information loss when a mobile device is compromised.

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 Mobile App 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 Mobile App 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.

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 Mobile App 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 Mobile App for Android: To store files offline, we require the user to enable device encryption and use the operating system's file encryption system.

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

Remote Wipe

An org admin can restrict access to minimize the risk of information loss when a mobile device is compromised.

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

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

Mobile Device Management (MDM)

The Salesforce mobile app provides an extra level of security compliance through interoperation with the most popular MDM (mobile device management) suites.

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

The Salesforce mobile app, 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.

Prerequisites for Android
There are prerequisites to implement enhanced security for the Salesforce mobile app 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.

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.

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.

Sample Property List and JSON Configurations
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. Another method of setting key-value pair assignments is through JSON.

Prerequisites for Android

There are prerequisites to implement enhanced security for the Salesforce mobile app for Android.

- First, configure Android for Work for your org. Android for Work is a program that supports enterprise use of Android devices. To learn more about the program ,see Android Enterprise. For setup information, seeGet Started with Android Enterprise.
- After 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 mobile app 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 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-

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

After 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 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. The first value in the array is the default host.

Key	Data Type	Platform	Description
			Android: Requires https:// in the host URL. **iOS:** Doesn't require https:// in the host URL.
`AppServiceHostLabels`	String, String Array	Android, iOS	Labels for the hosts. The number of `AppServiceHostLabels` entries must match the number of `AppServiceHosts` entries.
`OnlyShowAuthorizedHosts`	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 is 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 configure key-value pair assignments through your MDM suite.

Here’s the supported key:

Key	Data Type	Platform	Description
`ClearClipboardOnBackground`	Boolean	iOS	If true, the contents of the iOS clipboard are cleared when the mobile app is backgrounded. This

Key	Data Type	Platform	Description
			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 when the user starts and backgrounds the mobile app.

This security functionality is available 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 and JSON Configurations

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. Another method of setting key-value pair assignments is through JSON.

Sample Property List Configuration for iOS

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

The plist contains the key-value pair assignments that an MDM provider sends to a mobile app to enforce security configurations.

Here’s a sample plist.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTD
s/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>
```

Sample JSON Configuration for Android

On Android, you can set key-value pair assignments through JSON. Like a plist, JSON contains the key-value pair assignments that an MDM provider sends to a mobile app to enforce security configurations.

Here are a few JSON configuration samples.

Note The exact shape of the JSON depends on the MDM provider being used. For example, the values outside of the `managedProperty` key vary by MDM solution.

For a single host:

```
{
    "kind": "androidenterprise#managedConfiguration",
    "productId": "app:com.salesforce.chatter",
    "managedProperty": [
        {
            "key": "OnlyShowAuthorizedHosts",
            "valueBool": false
        },
        {
            "key": "AppServiceHostLabels",
            "valueString": "HostLabel"
        },
        {
            "key": "AppServiceHosts",
            "valueString": "https://host"
        }
    ]
}
```

For multiple hosts using comma-separated values:

```
{
    "kind": "androidenterprise#managedConfiguration",
    "productId": "app:com.salesforce.chatter",
    "managedProperty": [
        {
            "key": "OnlyShowAuthorizedHosts",
            "valueBool": false
        },
        {
            "key": "AppServiceHostLabels",
            "valueString": "HostLabel1,HostLabel2"
        },
        {
            "key": "AppServiceHosts",
            "valueString": "https://host1,https://host2"
        }
    ]
}
```

For multiple hosts using string arrays:

```
{
    "kind": "androidenterprise#managedConfiguration",
    "productId": "app:com.salesforce.chatter",
    "managedProperty": [
        {
            "key": "OnlyShowAuthorizedHosts",
            "valueBool": false
        },
        {
            "key": "AppServiceHostLabels",
            "valueString": ["HostLabel1,"HostLabel2"]
        },
        {
            "key": "AppServiceHosts",
            "valueString": ["https://host1","https://host2"]
        }
    ]
}
```

Mobile Application Management (MAM) with Enhanced Mobile App Security

Salesforce offers Enhanced Mobile App Security, a paid mobile application management (MAM) add-on designed to meet high security and compliance needs for the Salesforce mobile app.

REQUIRED EDITIONS

Setup for Enhanced Mobile Security available in: Lightning Experience
Setup for Enhanced Mobile Security available in: Production Orgs and Sandbox

USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

Note Your organization must license Enhanced Mobile App Security in order to use the feature. Contact your Salesforce sales rep for more information.

Unlike external mobile device management (MDM) solutions, Enhanced Mobile App Security protects at the app level, so it doesn't need to manage users' entire devices. Using the convenient Mobile Security Setup UI, you can configure a range of security policies to limit users' access and actions, and you can specify the severity of violations. You can also monitor user actions required for your compliance checks.

Important Starting with the Winter '24 release (October 2023), Salesforce will end support for setting and enforcing mobile security policies via Connected App custom attributes. If your org has configured Connected App custom attributes to set mobile security policies, we strongly encourage you to migrate your existing security policies to the Mobile Security Setup UI as soon as possible.

Enable Enhanced Mobile App Security
Before you can configure mobile app security policies, there are a couple steps to get Enhanced Mobile App Security enabled.

Enable and Configure Mobile App Security Policies
Use the convenient Setup UI to enable, configure, and enforce mobile security policies.

What Your Users See
Users with the Enforce Enhanced Mobile App Security user permission see a dedicated user interface in the Salesforce mobile app.

Enable Enhanced Mobile App Security

Before you can configure mobile app security policies, there are a couple steps to get Enhanced Mobile App Security enabled.

REQUIRED EDITIONS

Setup for Enhanced Mobile Security available in: Lightning Experience
Setup for Enhanced Mobile Security available in: Production Orgs and Sandbox

USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

Note Your organization must license Enhanced Mobile App Security in order to use the feature. Contact your Salesforce sales rep for more information.

1. **Install the Metadata (Optional)**
   Metadata is the information that describes the configuration of your org.
2. **Install the Salesforce Connected Apps (Optional)**
   This step is required if your org is new and there are no connected apps installed yet. Alternatively, you can log into your org using the Salesforce mobile app to trigger an installation of the required connected apps.
3. **Enable Users for Security Enforcement**
   You can assign permission set or assign a custom profile to enable user permissions.

Install the Metadata (Optional)

Metadata is the information that describes the configuration of your org.

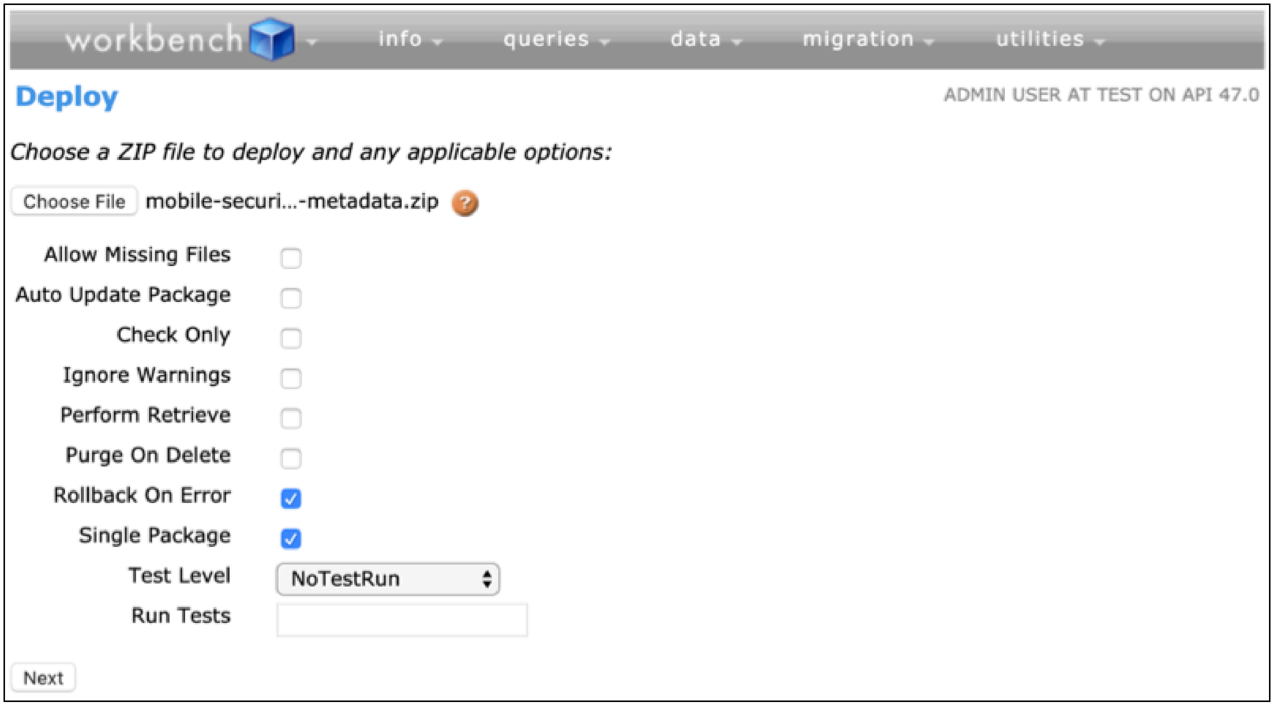
REQUIRED EDITIONS

Setup for Enhanced Mobile Security available in: Lightning Experience
Setup for Enhanced Mobile Security available in: Production Orgs and Sandbox

USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

Important Workbench (https://workbench.developerforce.com) is an open-source tool for interacting with your org. However, because Salesforce doesn't maintain Workbench, we can't address issues or bugs related to using it in Spring '24. For more information, see Replace Workbench With Salesforce Developer Tools

1. Go to your email and verify your account.
2. In a new browser tab, go to Workbench (https://workbench.developerforce.com/login.php).
3. Review the terms of service and check the **I agree to the terms of service** box.
4. Click **Login with Salesforce** and log in with your credentials.
5. Click **Allow** for the Workbench Connected app.
6. From the header dropdown menu, go to **Migration | Deploy**.
7. Download the zip file located here.
8. Check **Rollback On Error** and **Single Package**.



Deploy

ADMIN USER AT TEST ON API 47.0

Choose a ZIP file to deploy and any applicable options:

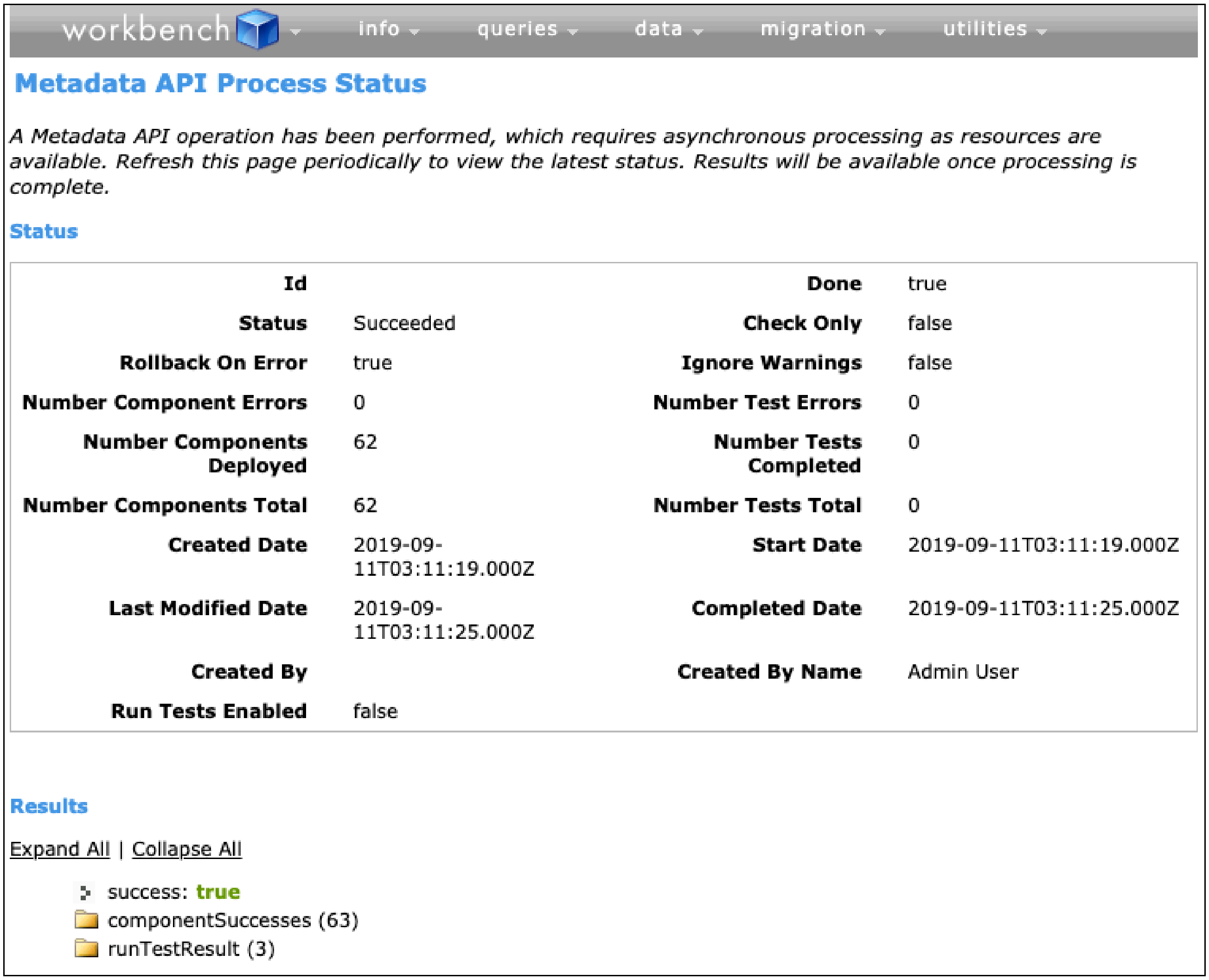
Choose File mobile-securi...-metadata.zip

Allow Missing Files	□
Auto Update Package	□
Check Only	□
Ignore Warnings	□
Perform Retrieve	□
Purge On Delete	□
Rollback On Error	☑
Single Package	☑
Test Level	NoTestRun
Run Tests	

Next

9. Click **Next** then **Deploy**.

Upon completion, your Workbench Metadata API Process Status should look like this image.



Metadata API Process Status

A Metadata API operation has been performed, which requires asynchronous processing as resources are available. Refresh this page periodically to view the latest status. Results will be available once processing is complete.

Status

Id		**Done**	true
Status	Succeeded	**Check Only**	false
Rollback On Error	true	**Ignore Warnings**	false
Number Component Errors	0	**Number Test Errors**	0
Number Components Deployed	62	**Number Tests Completed**	0
Number Components Total	62	**Number Tests Total**	0
Created Date	2019-09-11T03:11:19.000Z	**Start Date**	2019-09-11T03:11:19.000Z
Last Modified Date	2019-09-11T03:11:25.000Z	**Completed Date**	2019-09-11T03:11:25.000Z
Created By		**Created By Name**	Admin User
Run Tests Enabled	false		

Results

Expand All | Collapse All

success: **true**
componentSuccesses (63)
runTestResult (3)

Verify the Deployment

- Log in to your org as the Admin and go to Setup.
- From Setup enter `Apex Triggers` in the Quick Find box, then select **Apex Triggers**.
- Verify the four new Apex triggers.
- From Setup enter `Object Manager` in the Quick Find box, then select **Object Manager**.
- Verify the four new objects.
- From Setup enter `Tabs` in the Quick Find box, then select **Tabs**.
- Verify the four new tabs.

To avoid test failures, remove any metadata deployed from the step if your license expires.

Install the Salesforce Connected Apps (Optional)

This step is required if your org is new and there are no connected apps installed yet. Alternatively, you can log into your org using the Salesforce mobile app to trigger an installation of the required connected apps.

REQUIRED EDITIONS
Setup for Enhanced Mobile Security available in: Lightning Experience
Setup for Enhanced Mobile Security available in: Production Orgs and Sandbox

USER PERMISSIONS NEEDED	
To install and uninstall connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps
To install and uninstall packaged connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps AND Download AppExchange Packages
To read, create, update, or delete connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

1. Go to the Salesforce and Chatter Connected Apps Administration page (https://appexchange.salesforce.com/listingDetail?listingId=a0N3000000B4cUuEAJ).
2. Click **Get It Now**.
3. Log in with your org admin credentials.
4. Click **Install in Production**.
5. Review and agree to the Terms and Conditions and click **Confirm & Install**.
6. Select **Install for Admins Only** and click **Install**.
7. After the installation is complete, click **Done**.

Enable Users for Security Enforcement

You can assign permission set or assign a custom profile to enable user permissions.

REQUIRED EDITIONS

Setup for Enhanced Mobile Security available in: Lightning Experience
Setup for Enhanced Mobile Security available in: Production Orgs and Sandbox

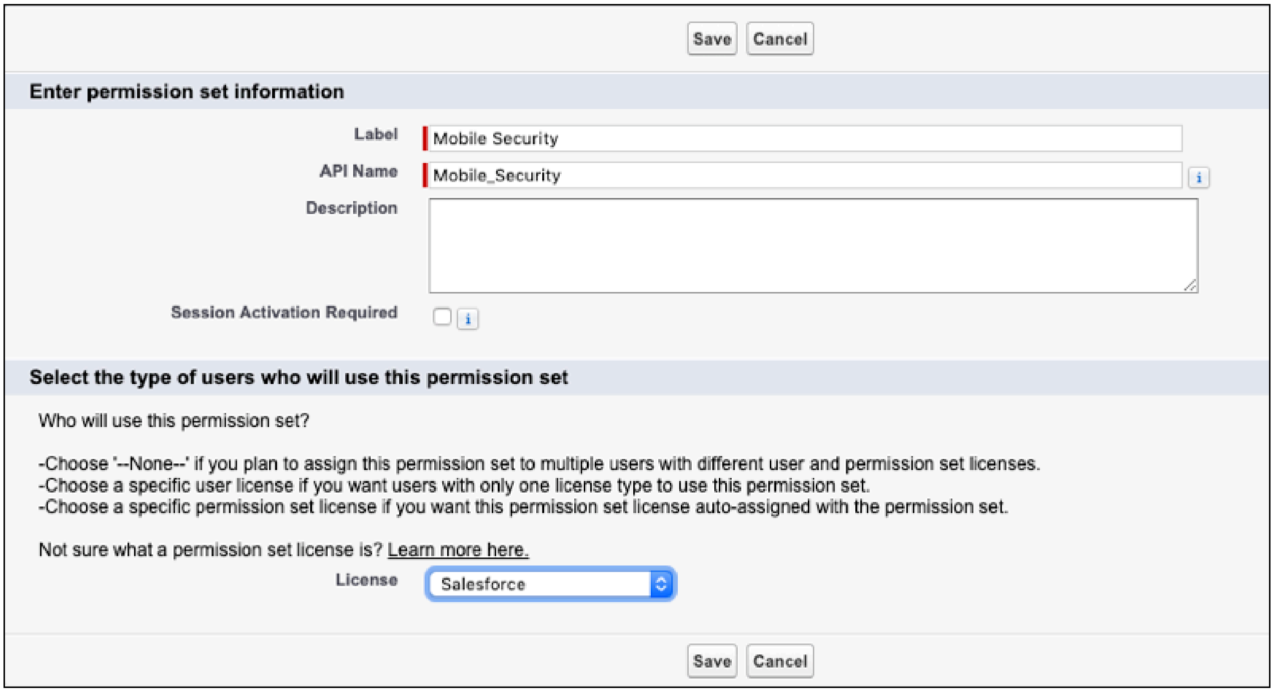
USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata
To create permission sets:	Manage Profiles and Permission Sets
To assign permission sets:	Assign Permission Sets

Assign Custom Profile

1. From Setup, enter Profiles in the Quick Find box, then select Profiles.
2. Find the Standard User profile and click **Clone**.
3. Set the Profile Name *MobileSecurity*.
4. On the MobileSecurity detail page, click **Edit**.
5. Check the **Enforce Enhanced Mobile App Security** and **New Salesforce Mobile App** boxes.
6. Click **Save**.

Assign Permission Set

- From Setup, enter *Permission* in the Quick Find box, then select **Permission Sets**.
- To create a permission set, click **New**.
- Set the fields:
  - Label: Mobile_Security
  - API Label: Mobile_Security



Enter permission set information

Label: Mobile Security

API Name: Mobile_Security

Description:

Session Activation Required □

Select the type of users who will use this permission set

Who will use this permission set?

-Choose '--None--' if you plan to assign this permission set to multiple users with different user and permission set licenses.
-Choose a specific user license if you want users with only one license type to use this permission set.
-Choose a specific permission set license if you want this permission set license auto-assigned with the permission set.

Not sure what a permission set license is? Learn more here.

License: Salesforce

- Set the License to **Salesforce**.
- Click **Save**.
- Scroll down to System and click **System Permissions**.
- Click **Edit**.
- Check the **Enforce Enhanced Mobile App Security** checkbox.
- Click **Save**.
- Go back to the Mobile Security Permission Set Overview page and click **Assigned Connected Apps**.
- Assign the Android and iOS connected apps to the permission set.
- Go back to the Mobile Security Permission Set Overview page and click **Manage Assignments**.
- Click **Add Assignments**.
- Select the users you want to have this permission set and click **Assign**.

Enable and Configure Mobile App Security Policies

Use the convenient Setup UI to enable, configure, and enforce mobile security policies.

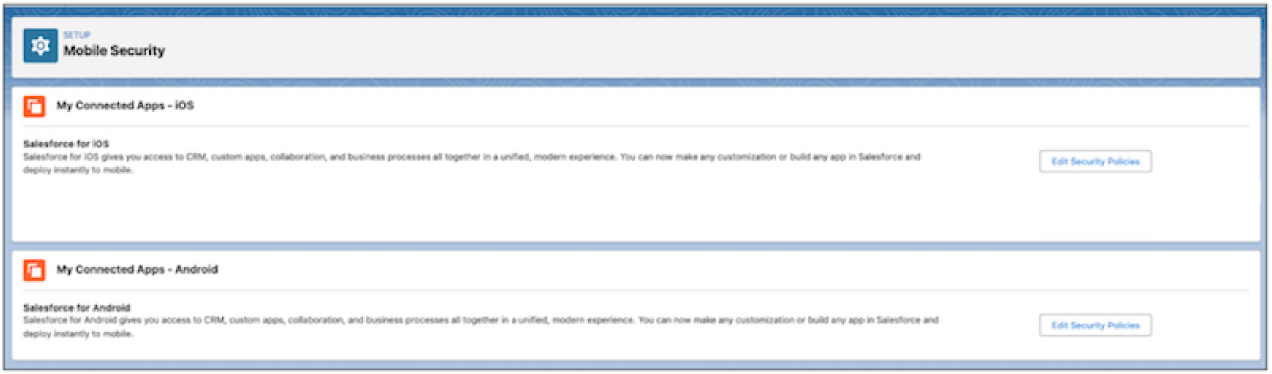
REQUIRED EDITIONS

USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

Note If you already have mobile security policies configured via connected app attributes, you can migrate your existing security policies to the Setup UI.

To configure your policies:

1. From Setup, in the Quick Find box, enter `Mobile Security`, and then select **Mobile Security**.
2. Select the **iOS** or **Android** tab.



My Connected Apps - iOS

Salesforce for iOS

Salesforce for iOS gives you access to CRM, custom apps, collaboration, and business processes all together in a unified, modern experience. You can now make any customization or build any app in Salesforce and deploy instantly to mobile.

My Connected Apps - Android

Salesforce for Android

Salesforce for Android gives you access to CRM, custom apps, collaboration, and business processes all together in a unified, modern experience. You can now make any customization or build any app in Salesforce and deploy instantly to mobile.

Each severity level represents the actions enforced in the event of a violation.

Severity Level	Actions Enforced
`critical`	Wipes app data and logs user out
`error`	Blocks access to the app until the issue is resolved, but doesn't log user out
`warn`	Notifies the user of the violation and recommends how to resolve, but user is able to continue using the app
`info`	Blocks prohibited action or logs user action and informs user

At cold starts and when a user's access token has expired, Enhanced Mobile App Security checks policies and enforces actions. Users can also manually recheck their policies.

Tip Mobile security policies take effect when users force quit the Salesforce mobile 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 mobile app so everyone is required to log in again.We also recommend that you warn users about the changes that you intend to make, especially if you restrict previously available activities.

Policy Name	Description	Severity Level
Allowed Device List	A device allowlist is a list of devices that a user can use. Specify allowed devices as a	Severity:

Policy Name	Description	Severity Level
	semicolon separated list.	- `critical` - `error` - `warn` Examples: - `iPhone11,8` (allows iPhone XR)
Authentication Server Certificate Pinning	Certificate pinning for the authentication server URLs where the user provides credentials to log in. For more info, see Configure Authentication Server Certificate Pin.	Severity: `info`
Block 3D Touch (iOS only)	3D touch or long press is when a user presses and holds an app icon to perform tasks without having to open the app first.	Severity: `info`
Block Calendar	Block calendar access on a user's device.	Severity: `info`
Block Camera (iOS only)	Block camera access on a user's device.	Severity: `info`
Block Contacts	Block access to contacts on a user's device.	Severity: `info`
Block Custom Keyboard (iOS only)	A custom keyboard replaces the built-in keyboard on a mobile device with a third-party alternative. Specify if you want to block custom keyboards.	Severity: `info`
Block File Backups (iOS only)	A file backup, such as iCloud, syncs files and photos from a user's mobile device onto cloud storage. The policy blocks files saved	Severity: `info`

Policy Name	Description	Severity Level
	within the Salesforce mobile app from syncing to a file backup.	
Block Jailbroken Device	A jailbroken or rooted mobile device can access system files to install unapproved apps or to modify settings.	Severity: - `critical` - `error` - `warn`
Block Man In The Middle Attack	A man-in-the-middle attack allows attackers to secretly intercept communications between two systems, client and server. This policy relies on network connectivity. If a user's device has low connectivity or the device is in airplane mode without an internet connection, the policy can get triggered.	Severity: - `critical` - `error` - `warn`
Block Microphone (iOS only)	Block microphone access on a user's device.	Severity: `info`
Block OS Share Actions (iOS only)	A user can perform specific tasks such as copying a link or saving an image with operating system (OS) share actions.	Severity: `info`
Block Screenshot (Android only)	A screenshot captures what's displayed on a user's mobile device. Specify if you want to block screenshots.	Severity: `info`
Blocked Device List	A device blocklist is a list of devices that a user is blocked from using. Specify blocked devices as a semicolon separated list.	Severity: - `critical` - `error` - `warn` Examples:

Policy Name	Description	Severity Level
		- `iPhone11,8` (blocks iPhone XR) - `Google` (blocks all Google devices)
Check Biometric Login Data	Validate biometric login data every time a user opens the app. The Check Biometric Login Data and Require Device Passcode policies can't be enabled at the same time.	Severity: - `critical` - `error` - `warn`
Disable URL Caching (iOS only)	URL cache saves some information from visited websites. Specify if you want to disable URL caching.	Severity: `info`
Enable Strict Data Leak Protection Controls (iOS only)	Enabling strict data leak protection blocks access to the context menu in iOS that allows a user to copy, web search, and look-up on selected text. Note: In the mobile app, some pages are native and some are hybrid. This policy works only on native and non-editable hybrid pages.	Severity: `info`
Log Email	Log the event when a user emails a contact from the app.	Severity: `info`
Log Phone Call	Log the event when a user makes a phone call from the app.	Severity: `info`
Log Screenshot (iOS only)	A screenshot captures what's displayed on a user's mobile device. An event is logged when a user takes a screenshot. The event is sent to the org's event stream and can be viewed any even streaming integration,	Severity: `info`

Policy Name	Description	Severity Level
	such as Splunk or Fairwarning. The event doesn't log the screenshot image itself. It only logs the event of a user taking a screenshot.	
Log Security Policy Evaluation Result	A security policy evaluation assesses whether users are meeting security requirements. Log the results of a security policy evaluation.	Severity: `info`
Log SMS	Log the event when a user sends a text message from the app.	Severity: `info`
Log Out User After Changing Biometric Login Data (iOS only)	Biometric login uses facial or fingerprint recognition to unlock devices and apps.	Severity: `info`
Log Out User After Device Restart	Specify if you want to log out a user after a device restart.	Severity: `info`
Maximum Application Version	Specify the maximum app version that can be installed on your user's mobile device.	Severity: - `critical` - `error` - `warn` Example: `220.6`
Maximum Days Offline Without Policy Refresh	We perform a security policy refresh when a user opens the app. Specify the maximum number of days a user can go without a security policy refresh.	Severity: - `critical` - `error` - `warn` Example: `30`
Maximum OS Version	Specify the maximum operating system (OS) version your user's mobile device can't exceed.	Severity: - `critical` - `error`

Policy Name	Description	Severity Level
		- `warn` Example: `12.1.9`
Minimum Application Version	Specify the minimum app version that must be installed on your user's mobile device.	Severity: - `critical` - `error` - `warn` Example: `1.0`
Minimum OS Version	Specify the minimum operating system (OS) version your user's mobile device must meet.	The number of the minimum OS version. Severity: - `critical` - `error` - `warn` Example: `11.9`
Minimum Security Patch Version (Android only)	A security patch helps protect a user's mobile device from vulnerabilities. Specify the required minimum security patch version.	The date of the minimum security patch version. Severity: - `critical` - `error` - `warn` Example: `2027-05-18`
Mobile Browser URI Scheme	Specify the mobile browser URI scheme for opening links on a user's device.	Severity: `info` Example for Chrome on iOS: `googlechromes://`
Phone Call Application Handler	Specify an app to use for making a phone call on a user's device.	Severity: `info` The value must be configured as

Policy Name	Description	Severity Level
		a `https://` link to the phone call app. The app is also required to Universal Links (iOS) and AsssetLinks (Android).
Require Device Passcode	A device passcode adds a layer of security for your user’s mobile device. Specify if you want to require a device passcode. The Require Device Passcode and Check Biometric Login Data policies can’t be enabled at the same time.	Severity: • `critical` • `error` • `warn`
Resource Certificate Pinning	Certificate pinning for the resource URLs used by the app to fetch data for the user.	Severity: `info`

Tip Use the Security Center app to define and deploy mobile app security policies to selected tenants from the Mobile App Security Policy option in Security Center. For more information on using Security Center to deploy security policies, see Define and Deploy Security Policies.

Configure Authentication Server Certificate Pin
Provide domain name and certificate fingerprint to configure authentication server certificate pin.

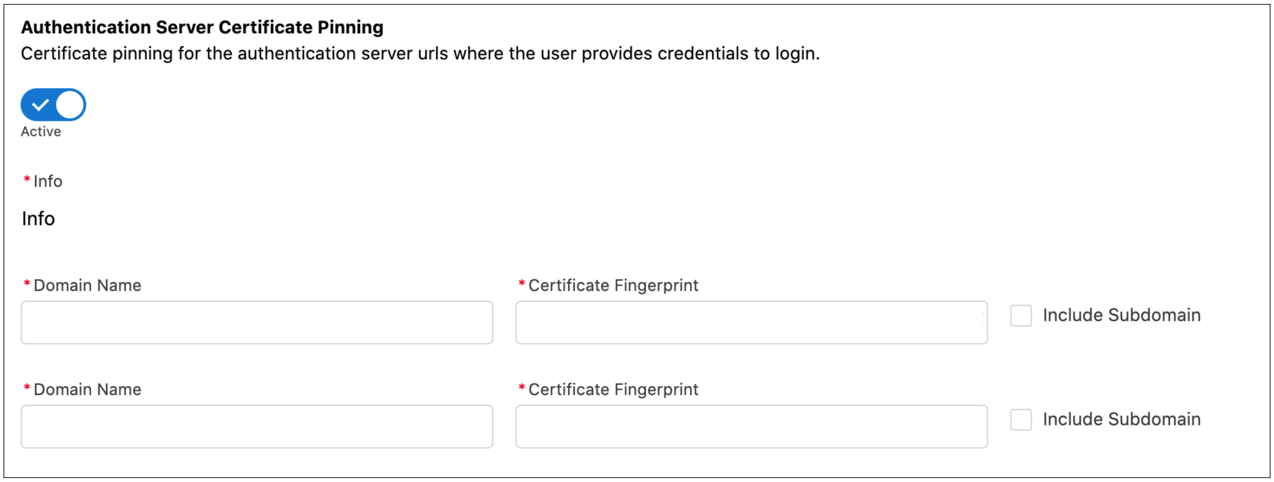
Configure Authentication Server Certificate Pin

Provide domain name and certificate fingerprint to configure authentication server certificate pin.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED	
To create and modify Enhanced Mobile App Security settings:	Manage Enhanced Mobile App Security AND Modify Metadata

1. In the Mobile Security Setup UI, enable **Authentication Server Certificate Pinning**.



Authentication Server Certificate Pinning

Certificate pinning for the authentication server urls where the user provides credentials to login.

Active

* Info

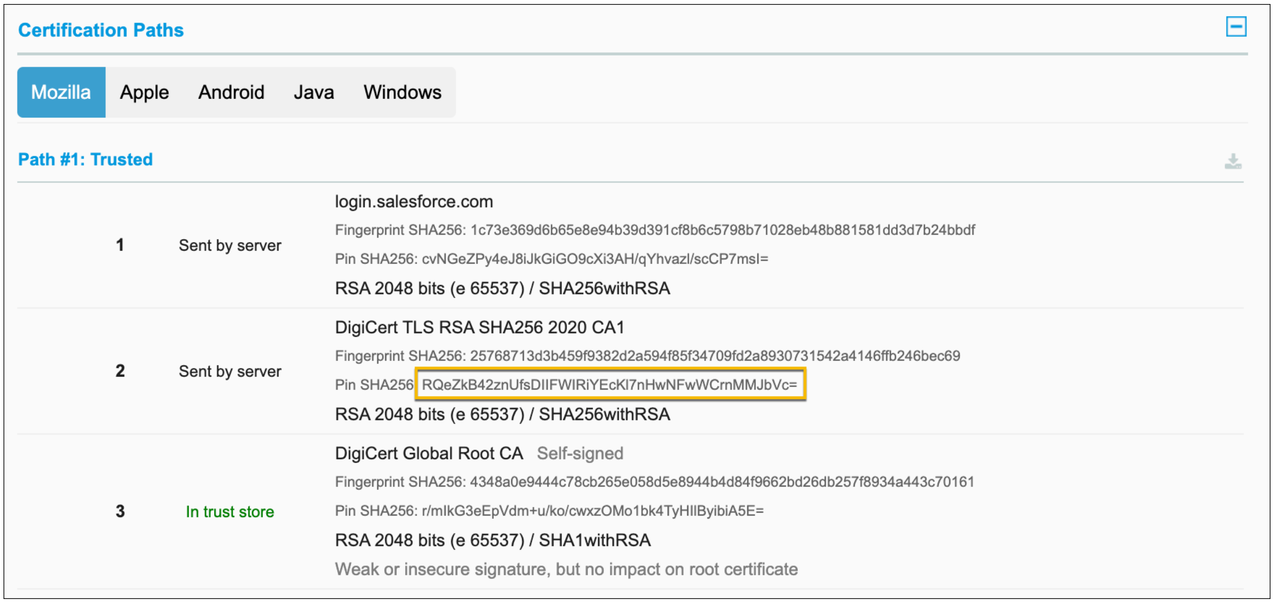
Info

* Domain Name	* Certificate Fingerprint	□ Include Subdomain
* Domain Name	* Certificate Fingerprint	□ Include Subdomain

2. Enter the domain name for the server that you want to pin the certificate for in the **Domain Name** field. For example, `https://login.salesforce.com`.
3. For the **Certificate Fingerprint**, you have two options to obtain the value.

   You can obtain the Certificate Fingerprint value from your org's internal team that owns the certificate.

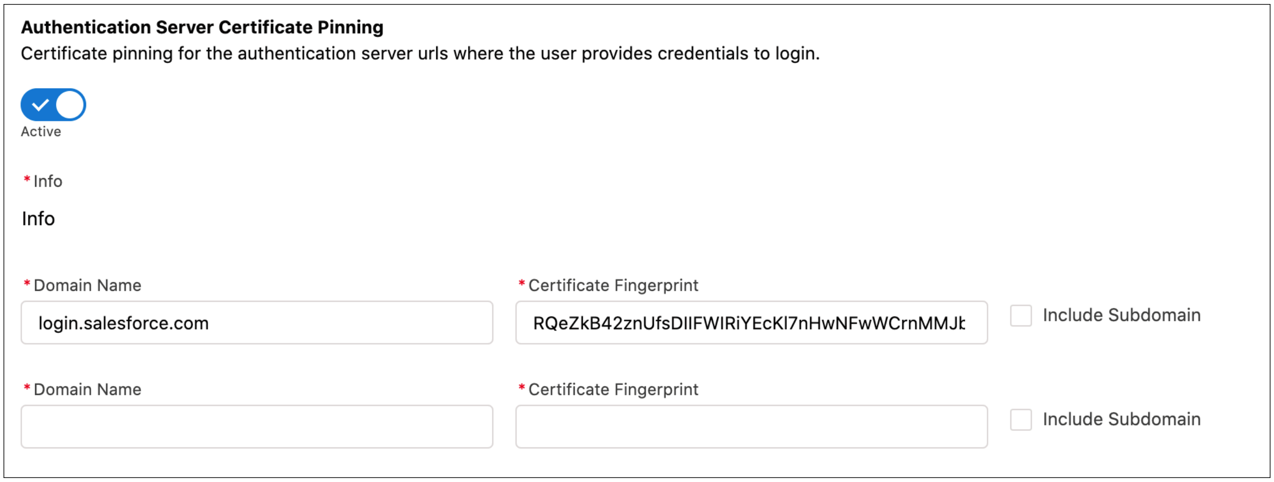
   You can use a third-party application such as www.ssllabs.com to obtain the Certificate Fingerprint value.



Path #1: Trusted

1	Sent by server	login.salesforce.com Fingerprint SHA256: 1c73e369d6b65e8e94b39d391cf8b6c5798b71028eb48b881581dd3d7b24bbdf Pin SHA256: cvNGeZPy4eJ8iJkGiGO9cXi3AH/qYhvazl/scCP7msI= RSA 2048 bits (e 65537) / SHA256withRSA
2	Sent by server	DigiCert TLS RSA SHA256 2020 CA1 Fingerprint SHA256: 25768713d3b459f9382d2a594f85f34709fd2a8930731542a4146ffb246bec69 Pin SHA256 RQeZkB42znUfsDIIFWIRiYEcKI7nHwNFwWCrnMMJbVc= RSA 2048 bits (e 65537) / SHA256withRSA
3	In trust store	DigiCert Global Root CA Self-signed Fingerprint SHA256: 4348a0e9444c78cb265e058d5e8944b4d84f9662bd26db257f8934a443c70161 Pin SHA256: r/mIkG3eEpVdm+u/ko/cwxzOMo1bk4TyHIIByibiA5E= RSA 2048 bits (e 65537) / SHA1withRSA Weak or insecure signature, but no impact on root certificate

4. Copy and paste the value into the **Certificate Fingerprint** field.



Authentication Server Certificate Pinning

Certificate pinning for the authentication server urls where the user provides credentials to login.

Active

* Info

Info

* Domain Name	* Certificate Fingerprint	
login.salesforce.com	RQeZkB42znUfsDIlFWIRiYEcKl7nHwNFwWCrnMMJb	□ Include Subdomain
* Domain Name	* Certificate Fingerprint	
		□ Include Subdomain

5. Click **Save**.

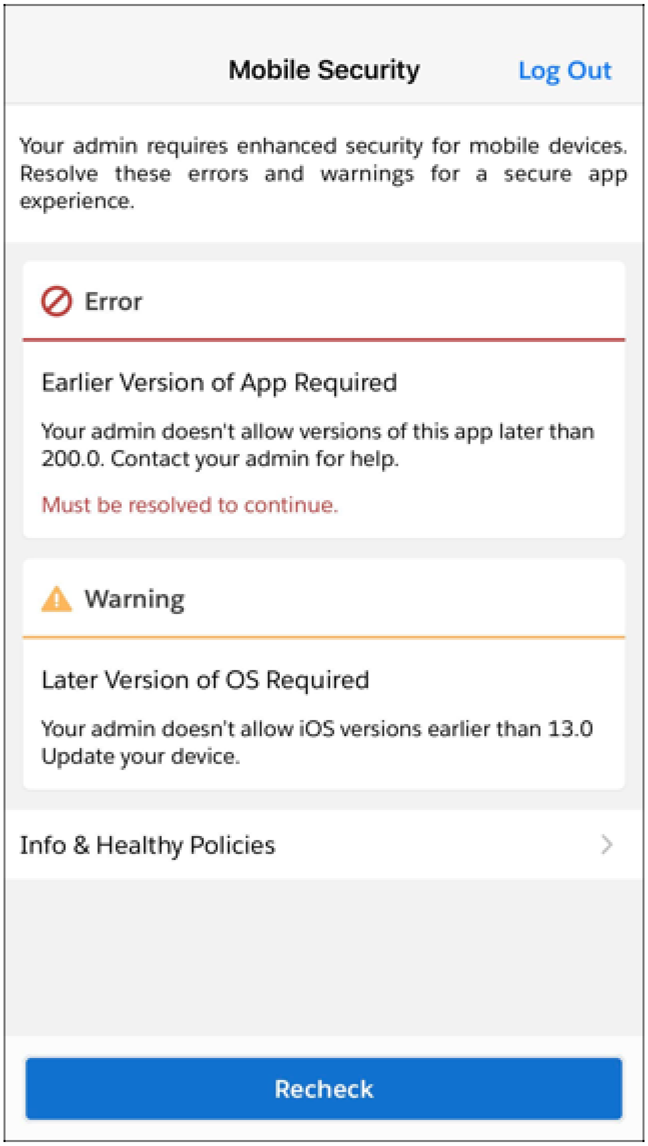
Here are a few things to keep in mind about authentication server certificate pins.

- The test of the server certificate happens when the user logs in. This policy doesn't replace the Man-in-the-Middle security policy.
- The certificate is set after the user logs in for the first time, and is enforced for subsequent logins on that server. If the policy is updated or removed, then the updated policy is picked up by the app the next time the user logs in.
- The authentication server pins aren't enforced if the user is using advanced authentication in Safari.
- If the pinned server certificate is rotated, but the app isn't updated, then the user can't log in because of the mismatched pins. A workaround is to ask the user to reinstall the app to obtain the new certificate pin value.
- It's recommended that you use the pin of the intermediate certificate, since that provides a good balance between how frequently it expires and security value.

What Your Users See

Users with the Enforce Enhanced Mobile App Security user permission see a dedicated user interface in the Salesforce mobile app.

When policies are violated, the UI notifies users of the specific policy violation and the severity of the violation. It also recommends how to resolve the violation. When users take steps to resolve violations, they can manually recheck their policies.



Mobile Security

Your admin requires enhanced security for mobile devices. Resolve these errors and warnings for a secure app experience.

Error

Earlier Version of App Required

Your admin doesn't allow versions of this app later than 200.0. Contact your admin for help.

Must be resolved to continue.

Warning

Later Version of OS Required

Your admin doesn't allow iOS versions earlier than 13.0 Update your device.

The interface is also transparent about all of the policies enforced. In addition to policy violations, it allows users to view healthy policies and the actions that are monitored and blocked in the app.

Best Practices and Troubleshooting

This information helps you implement best practices and troubleshoot when configuring mobile app security.

Notes

- iOS: Before 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 isn’t a binary (on or off), but implemented at different levels.
- Salesforce provides multiple levels of security; however, there’s no application that can guarantee a

completely secure system.