Require Device Passcode

Workspace ONE® supports use of the device passcode to unlock the mobile app user interface. The end user can enter their device passcode instead of a separate passcode to access their enterprise mobile apps.

Use of the device passcode for app unlock is controlled by security policies in the Workspace ONE Unified Endpoint Manager (UEM) console. App unlock with the device passcode is handled by the Workspace ONE mobile Software Development Kit (SDK) runtime. The policy is available for Android, iOS, and iPadOS.

Table of Contents

Background

A passcode can be used to protect the data at rest on a mobile device. A passcode can be a phrase, word, number, pattern, or other type of user secret, depending on the type of mobile device in use. In any case, a passcode can protect data at rest with some or all of these measures.

- Lock the user interface (UI) until the passcode is entered.
- Encrypt data at rest with a key derived from the passcode, known as Passcode-Based Encryption (PBE).
- Encrypt data at rest with a key that is stored on the device in a hardware security module (HSM) that is only accessible when the passcode has been entered in a trusted user interface (TUI).

Note that the last of those measures has dependencies. First, the device must have an HSM, which is the case for all recent Android, iOS, and iPadOS models. Second, the device must have a passcode. All recent devices prompt the user to set a passcode out-of-the-box (OOB). However, the user may skip setting a device passcode OOB, or may set a passcode OOB but then remove it later.

If the device doesn't have a passcode then the protection afforded by encrypting data at rest with an HSM key is reduced. Anyone with access to the device will have access to the HSM keys.

Workspace ONE Support

Workspace ONE supports protection of data at rest on mobile devices with a passcode.

Use Cases Without Require Device Passcode

Without the Require Device Passcode (RDP) policy option, Workspace ONE supports the following use cases.

- On managed devices.
  - Force the user to set a device passcode so that an HSM key adds protection.
  - Set the device UI to lock after a period of inactivity, until the device passcode is entered.
- On managed and unmanaged devices.
  - Force the user to set an app passcode to unlock their SDK apps.
  - Force the user to authenticate with their domain username and passcode when opening an SDK app.
  - Protect data at rest with a PBE key derived from the app passcode or domain username and password.
  - Lock the SDK app UI after a period of inactivity until the app passcode, or domain username and password, is entered.
  - Recover enterprise data after forgetting the app passcode.

Use Cases With Require Device Passcode

With RDP, Workspace ONE also supports the following use cases.

- On managed and unmanaged devices.
  - Lock the SDK app UI lock after a period of inactivity, until the device passcode is entered.
  - If the device passcode is removed after enrollment, block access to the SDK app UI and data until the user re-authenticates.

Additional Use Cases

Workspace ONE supports the following additional use cases that are relevant to RDP.

- Changes to policy settings in general.
  - Specifically allowing or disallowing RDP.
- App single sign-on (SSO) so that a single passcode is shared by all apps on one device.
- Multiple SDK versions in different apps on a single device, for example during roll-out of upgraded apps.
  - Specifically apps with versions of the SDK that do and don’t support RDP.

Summary of Benefits

The protection of data on smartphone and tablet devices depends to some extent on a passcode being set by the end user. See under Background, above, for a discussion of that.

Without RDP, Workspace ONE customers had only these choices to ensure that a passcode was set.

- Mobile device management (MDM) to force a device passcode to be set so that HSM protection is effective.
- SDK authentication by passcode or username and password, so that an app passcode is set.

Neither option is ideal.

- MDM can be seen as intrusive by end users, if imposed on their personal devices. End users might then request separate devices to be purchased for enterprise use, even though this is inconvenient for them and incurs additional cost for the enterprise.
- The app passcode is another secret that the end user has to remember, if they have already set a device passcode.

RDP is a third option for Workspace ONE administrators. These are the benefits.

- End users need only remember one secret, either device passcode or app passcode.
- Effective HSM protection, without the imposition of MDM.

How Workspace ONE Require Device Passcode works

Workspace ONE Require Device Passcode (RDP) works like this.

- The RDP policy is selected in the UEM console.

  See Console Configuration.

  The RDP policy setting will be available if SDK Authentication Type is set to Passcode, or to Username and Password. RDP won't be available if SDK Authentication Type is set to Disabled.

- Each SDK app will receive the RDP policy setting during enrollment, and after an update, same as other Workspace ONE security settings.

  If the RDP policy is set and the device has a passcode then, after enrollment or authentication, the SDK app will switch to being unlocked by the device passcode.

  See End User Experience.

See also the Question and Answer section for more information.

Console Configuration

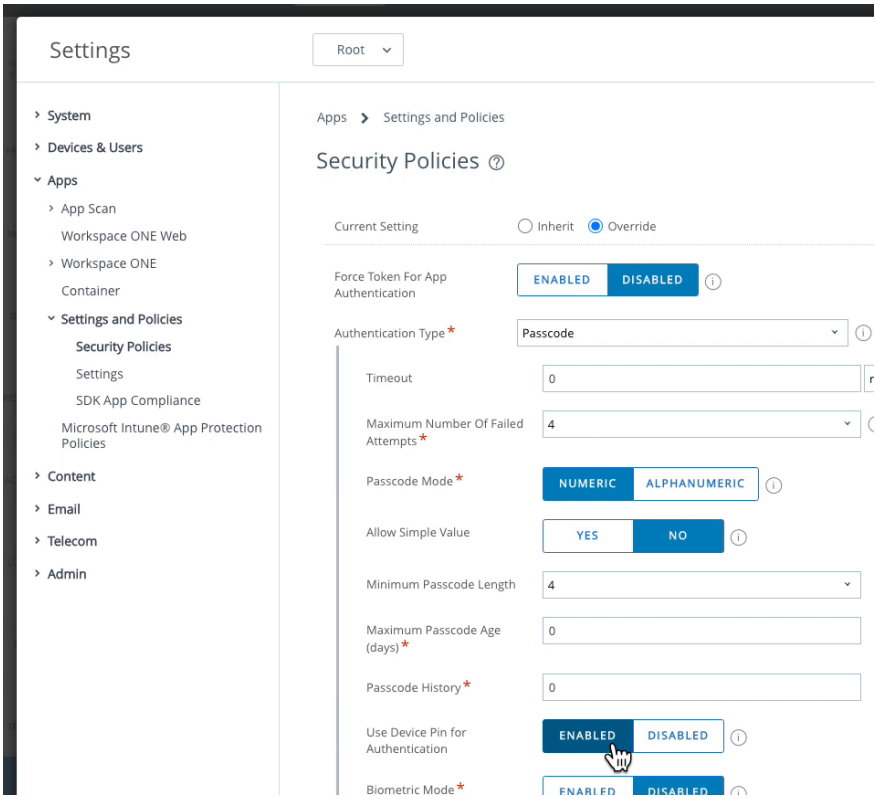
This feature can be configured in the Workspace ONE management console. The following instructions are intended for application developers or other users wishing to try out the feature quickly. Full documentation can be found in the online help.

1. Log in to the management console and select an organization group.
2. Navigate to: Groups & Settings, All Settings, Apps, Settings and Policies, Security Policies.

   This opens the Security Policies configuration screen, on which a number of settings can be switched on and off, and configured.
3. For the Authentication Type setting, select either Passcode or Username and Password.

   When either of these is selected, further controls will be displayed.
4. For the Use Device Pin for Authentication setting, select Enabled.

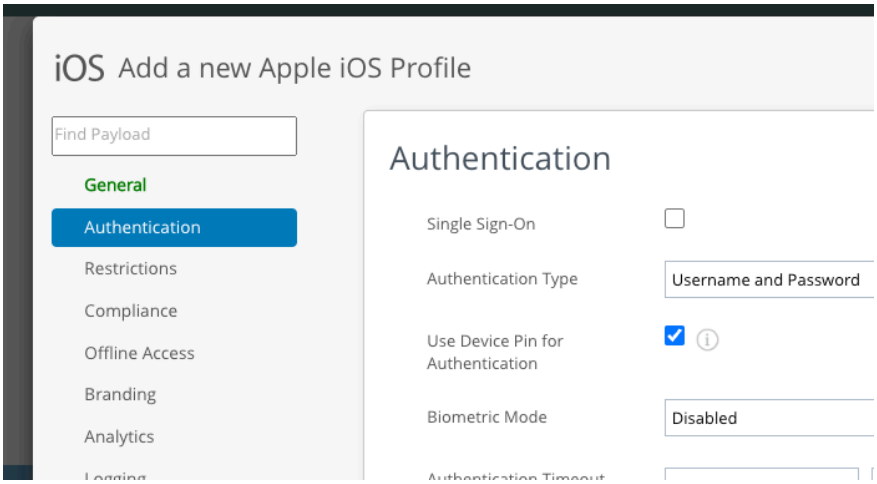
The following screen capture shows the location of the setting in the console user interface.



Screen Capture: UEM Require Device Passcode setting

In that screen capture Authentication Type: Passcode has been selected. The appearance would be similar if Authentication Type: Username and Password had been selected.

RDP can also be configured in custom SDK profiles, in the Authentication section. The following screen capture shows the setting in the console user interface.



Screen Capture: UEM Require Device Passcode in custom SDK profile

In that screen capture Authentication Type: Username and Password has been selected in a custom profile for iOS.

See also the Question and Answer section for more information.

End User Experience

After this feature is configured in the console, it will be become available

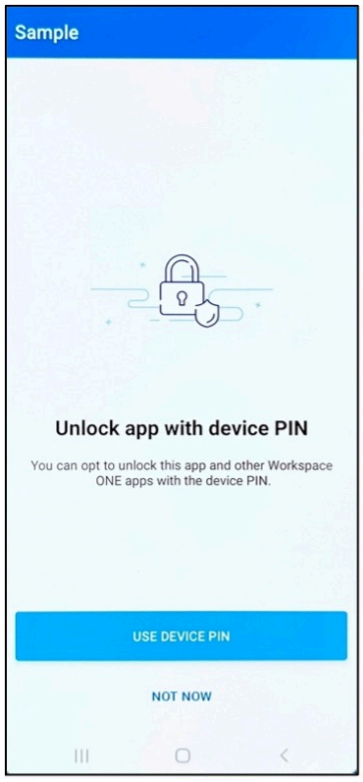
- at new enrollments of SDK apps.
- when SDK apps receive updated security settings, if already enrolled.

Even if the feature is available at enrollment time, the user will still have to set an app passcode, or authenticate once with their username and password, before starting to use the device passcode to unlock their SDK apps.

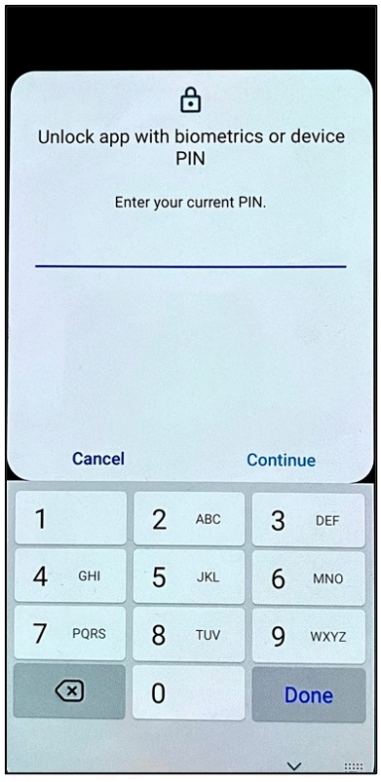
In the 22.10 SDK for Android the switch to device passcode takes place after an explicit choice by the user. In the 22.10 SDK for iOS the switch is automatic.

Subsequently, when the user is next unlocking an SDK app, a device passcode entry screen will be superimposed in front of the app UI. The passcode entry UI is part of the device operating system.

These screen captures show how the option is offered to Android end users, and how the device passcode entry screen could appear when used to unlock an SDK app.



Screen Capture: SDK option to use device passcode



Screen Capture: Device passcode entry screen

The device passcode entry UI will have a cancel option. If the user selects that option then they will be presented with the SDK authentication UI. They can then enter their app passcode, or username and password, according to security policies

See also the Question and Answer section for more information.

Question and Answer

Further information about the RDP policy option is presented as questions and answers here.

Why is this feature only available with some SDK authentication types

The RDP option is available if SDK Authentication Type is set to Passcode, or to Username and Password. It isn't available if SDK Authentication Type is set to Disabled.

The SDK uses multiple keys to protect its data, and the data of the application into which it is integrated. This approach is often referred to as a key hierarchy.

Limiting the availability of RDP also limits the amount of code change to the SDK key hierarchy that was necessary to deliver the feature. This has made it possible to deliver the feature in a reliable manner and with quality assured.

The availability of RDP could be expanded in future.

Can the SDK derive an encryption key from the device passcode

The SDK cannot derive an encryption key from the device passcode.

When the user has to enter their device passcode, the SDK sends a request to the operating system (OS) to launch a suitable UI. When the passcode has been entered the OS closes the UI and returns a success code to the SDK. Or, if the user cancels or fails to enter the correct passcode, the OS returns a failure code.

In neither case does the OS expose the passcode value to the requester, to preserve the device's security.

The OS UI can be opened by any app; privileges aren't required. OS support is similar in Android, iOS, and iPadOS.

What security features of the mobile operating system are used

The SDK uses the built-in security of the device OS as a base for protection of data at rest. Specifically for RDP, built-in security features are used as follows.

RDP is only compatible with SDK authentication types in which there is a passcode supplied by the end user, either an app passcode created for the purpose, or their domain username and password. This means that SDK PBE, see Background, is in effect in all RDP cases.

What differs if PBE:on is set is that a copy of the SDK PBE key is stored on the device. The PBE key copy is protected by another key, referred to here as the key encryption key (KEK).

- The SDK for Android creates the KEK with key use authorization: user authentication.

See the Android website for details, for example here.
https://developer.android.com/training/articles/keystore#KeyUseAuthorizations

- The SDK for iOS creates the KEK in the keychain and flags it for access only when the device is unlocked, and if a passcode is set on the device.

  See the Apple website for details, for example here.
  https://support.apple.com/guide/security/secb0694df1a/web
  The flag used is `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly`.

Those protections means that the OS will do one of the following, depending on OS version and make.

- Delete the KEK securely at the moment that the device passcode is removed.
- Make the KEK inaccessible if the device doesn't have a passcode.

In either case the PBE key then cannot be recovered from the copy encrypted by the KEK. The user must either enter their app passcode, so that the SDK can re-derive the PBE key, or go through Workspace ONE lost passcode recovery, or re-authenticate with their domain username and password.

If the OS makes the KEK inaccessible then setting a new device passcode will restore access to the KEK. The user could then resume using the device passcode to unlock their SDK apps without re-authenticating.

Is this feature compatible with Mobile Device Management

MDM can be used to force users to set a device passcode, and to impose type and complexity conditions on the passcode. RDP can be applied whether or not the device passcode was forced by MDM.

This means that RDP is compatible with both Mobile Device Management (MDM) and Mobile Application Management (MAM).

What are the limitations

These are some known limitations of RDP.

- RDP isn't supported by Workspace ONE Intelligent Hub for Android at time of writing.
- RDP is an SDK profile setting so it isn't directly compatible with UEM Smart Groups at time of writing. Smart Groups are also referred to as App Groups and Application Groups.

  This limitation can be worked around in an organization group (OG) as follows.

  - Create a device profile in the OG that forces a device passcode, or an Android work profile passcode.
  - Create a Smart Group for managed devices in the OG.
  - Assign the device profile to the Smart Group.
  - Allow managed, work profile, and unmanaged devices to enroll in the OG.
  - Set RDP:on in the SDK Profile of the OG.

The results will be like this.

- Users of managed devices will be forced to set a device or profile passcode when the profile is applied.

  They will also have to set an app passcode, or authenticate with username and password, when they enrol an SDK app. But they will immediately switch to using the device passcode to unlock their SDK apps.

  Users of managed devices won't be able to remove the device passcode.

- Users of unmanaged devices won't be forced to set a device passcode.

  They will have to set an app passcode, or authenticate with username and password, when they enrol an SDK app.

In summary, data on every device will be protected by a passcode, no user will have to remember more than one passcode, and users can choose whether to accept device management.

What happens if security policies change

Changes in security policies, including RDP, are first made at the UEM console. Changes will be retrieved and applied by the SDK runtime the next time the app starts, or after an interval. In some cases it will be necessary for the end user to authenticate to the app before the SDK can communicate with the UEM.

What happens depends on the transition.

- Transition from RDP:off to RDP:on.

  The user might have to authenticate one last time with their app passcode or username and password depending on the SDK Authentication setting. They will then start using their device passcode to unlock their apps, see End User Experience.

- Transition from RDP:on to RDP:off.

  The user might have to authenticate one last time with their device passcode, after which they will stop using their device passcode to unlock their apps. Instead they will use their app passcode. If they have forgotten their app passcode they can recover in the usual way, by re-authenticating in the app and then setting a new app passcode.

- Transition from RDP:on to SDK Authentication:Disabled.

  The user might have to authenticate one last time with their device passcode, after which they won't have to authenticate to unlock their apps. SDK and app data won't be protected by Workspace ONE PBE after the transition.

- Transition from SDK Authentication:Disabled to SDK Authentication:Passcode and RDP:on.

  The user will first have to set an app passcode. They will then start using their device passcode to unlock their apps, see End User Experience.

- Transition from SDK Authentication:Disabled to SDK Authentication:Username and Password and RDP:on.

  The user will first have to enter their username and password. They will then start using their device passcode to unlock their apps, see End User Experience.

What happens when an app is upgraded from an old version of the SDK

Each mobile app that integrates the SDK is built with a particular version, chosen by the app developer. When the developer is working on a new version of their app they may choose to upgrade to a newer version of the SDK.

In case an app is upgraded from an SDK version that doesn't support RDP to a version that does, and security policies are configured with RDP:on, the user experience will be the same as during the transition from RDP:off to RDP:on, see under What happens if security policies change above.

Which time-out applies to this feature

The time-out specified in the UEM security policies applies to SDK apps, regardless of which passcode is in use. After the interval specified in the UEM, the SDK will time out and lock the app UI.

The device may itself monitor user activity and lock the screen after a period of idleness. Device-level monitoring may be configurable in the device settings by the end user. The setting could be referred to as an auto-lock or screen time-out, for example, and is separate to the SDK app time-out.

What happens if the device passcode is changed or removed

Whenever the device passcode is being used to unlock an SDK app, the current device passcode will be the one required. After the device passcode has been changed the new passcode immediately becomes the one used to unlock SDK apps.

If the device passcode was being used to unlock SDK apps, but has been removed, the user will have to authenticate with their app passcode, or username and password, according to security policies.

See also the description of What security features of the mobile operating system are used.

What happens in Android Work Profile mode

Android devices can be managed in Work Managed mode or Work Profile mode. Those are also referred to as the Device Owner (DO) and Profile Owner (PO) modes.

In Work Managed mode the device can have only one passcode, or in theory no passcode. In Work Profile mode there can be up to two passcodes, one for the device and another for the profile.

Work Profile devices may have a setting like Use One Lock by which the user can specify that the profile delegates to the device for its passcode. In that case the device again has only one passcode.

These rules apply.

- If the SDK app is installed in the Work Profile then the passcode of the profile will be used for RDP. Note that the profile passcode could actually be the device passcode in the case of one-lock delegation, see above.
- If the SDK app is installed outside the Work Profile then the passcode of the device will be used for RDP.

What happens if there are different SDK versions in the apps on one device

Each mobile app that integrates the SDK is built with a particular version, chosen by the app developer. The SDK apps on one device could have been built with different versions of the SDK. Some of the apps on a device might therefore support RDP while others don't.

If RDP:on is set in the UEM configuration and there are multiple SDK versions in the apps on a device, this is what happens.

- The user will use the device passcode to unlock apps that support RDP, see End User Experience.
- Apps that don't support RDP will continue to require the app passcode, or username and password, to unlock.

What happens if mobile app single sign-on is in use

Workspace ONE supports single sign-on (SSO) for mobile apps. Use of SSO is configured in the UEM security policies, same as RDP. If SSO is in use then the SDK apps on the device are unlocked with the same app passcode. SSO is compatible with RDP.

SSO without RDP works like this.

- When the user authenticates in a first app, an authentication session starts that is shared by other apps.
- When the user opens a second app the user doesn't have to authenticate, because the second app becomes part of the authentication session.
- The authentication session ends when the user is inactive in SDK apps for an extended period, or when the device is switched off or rebooted for example.

Note that SSO can be switched off for particular apps. An app for which SSO is switched off doesn't become part of any existing authentication session, and doesn't start an authentication session when it is unlocked.

SSO and RDP work together like this.

- If the first app supports RDP, and a device has been set, the user will unlock the app by entering their device passcode, see End User Experience. That starts the SSO authentication session, if the SSO is in use for that app.
- If the first app doesn't support RDP the user will unlock the app by entering their app passcode, or domain username and password. That starts the SSO authentication session, if the SSO is in use for that app.

- The second app will become part of the authentication session started by the first app, regardless of its support for RDP, if SSO is in use for that app.

What happens if biometric authentication is in use

Workspace ONE supports the use of fingerprint recognition, face recognition, and biometric authentication in general for mobile apps. Use of biometric authentication is configured in the UEM security policies, same as RDP.

To use biometric authentication for SDK apps the user must already have registered a biometric credential such as their fingerprint or face on their mobile device. The user can then unlock their SDK apps by presenting the biometric credential.

If RDP is in use then the SDK launches an OS UI in which the device passcode can be entered. If biometric authentication is also in use then the option to use a biometric credential will be offered in the OS UI. In effect, the SDK configures the launched OS UI according to the policy settings received from the UEM.

Registered biometric credentials aren't always recognised by mobile devices. When the device fails to recognise a credential as registered then biometric authentication doesn't succeed. The user then has the option to fall back to another form of authentication. If RDP is in use then the device passcode is a possible fallback. Otherwise the fallback will be the app passcode, or username and password, depending on security policy.

Impact to security

The impact to security of RDP can be assessed in several aspects.

- There is a theoretical reduction in security in the case that there was both an app passcode and a device passcode.

  In principle, both those passcodes would have to be known or bypassed in order to access enterprise app data on the device. Removing one of those passcodes removes one protective barrier.

  Note that both passcodes would be known to the legitimate user so the app passcode can't be considered a second authentication factor, unlike a biometric credential.

- More reliance is placed on the built-in protection of the device.

  If RDP is in use then the effectiveness of Workspace ONE PBE is reduced, because the key derived from the app passcode would be accessible after the device passcode had been entered or bypassed.

  There have been a number of advances in the security of mobile devices in recent years. For example, an HSM and Trusted Execution Environment (TEE) is now a feature of all mainstream devices. The additional protection offered by Workspace ONE PBE isn't always worth the inconvenience to users of having to remember and enter another passcode.

- The usability of the security solution is improved.

  The worker at the modern enterprise will get mobile access to the enterprise data and services on which their success depends. If the secure mobility solution that their employer offers is too difficult to use, or too intrusive on their privacy, those workers will find their own solution. Their own solution won't be managed by the enterprise and won't necessarily be secure.

  Making the security solution easier to use increases its adoption and therefore its effectiveness.

In conclusion, requiring a device passcode to unlock the mobile app user interface can increase usability at only a small cost to security.

Reference

See also the Security Policies Profiles for the SDK pages in the product documentation, for example here.
https://docs.omnissa.com/bundle/SDKandAppMgmtVSaaS/page/UEMSDKSecurityPolicies.html

See also the Require Device Passcode (RDP) page in the UEM product documentation, for example here.
https://docs.omnissa.com/bundle/SDKandAppMgmtVSaaS/page/UEMSDKRequireDevicePasscode.html

Document Information

Revision History

13dec2022 Initial Publication.

12Feb2025 Updated License

License