VMware AirWatch App Wrapping Guide

Applying a management layer to mobile apps

AirWatch App Wrapping v5.4

Have documentation feedback? Submit a Documentation Feedback support ticket using the Support Wizard on support.air-watch.com.

Copyright © 2018 VMware, Inc. All rights reserved. This product is protected by copyright and intellectual property laws in the United States and other countries as well as by international treaties. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and other jurisdictions. All other

Table of Contents

Copyright © 2018 VMware, Inc. All rights reserved.

Copyright © 2018 VMware, Inc. All rights reserved.

Chapter 1: Overview

Copyright © 2018 VMware, Inc. All rights reserved.

Introduction to App Wrapping

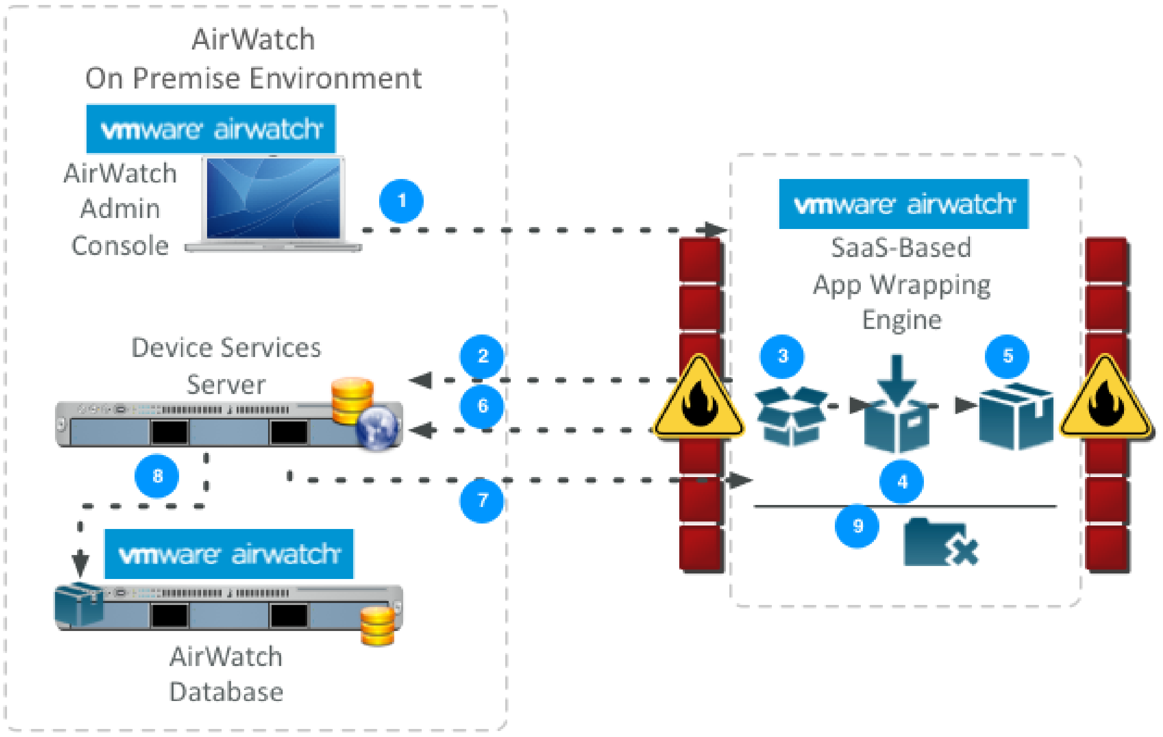
AirWatch Application Wrapping, or app wrapping, allows organizations to secure enterprise applications with little code changes. It can add an extra layer of security and data loss prevention while offering a consistent user experience. Consistency comes from using AirWatch options such as branding, single sign on (SSO), and authentication.

Modifying your internal applications with app wrapping reduces time and expenses from development on management and security. It lets you access tools already available with AirWatch by adding a layer of features over the application. Once the advanced features are applied, deploy the application to your enterprise app catalog for end-users to access.

Workflow for App Wrapping in On-Premise Environments

Review the workflow that depicts communication between the SaaS-based app wrapping engine and an on-premises deployment. AirWatch wraps and stores modified applications within the SaaS infrastructure, and it does not keep any unmodified application files.

The system securely stores and deletes internal application files and auxiliary files. All communication on port 443 is encrypted with AES-256, over SSL, and requiring HMAC token authentications.



1. The admin uploads the internal application and ancillary files, like provisioning profiles and signing certificates, to the AirWatch Console and initiates wrapping.

   The Console notifies the wrapping engine that it has a file. The console populates the download URL for the internal application file and ancillary files.

2. The wrapping engine goes to the URL on the internal network device services server and retrieves the files.

3. The wrapping engine unzips the files.

Copyright © 2018 VMware, Inc. All rights reserved.

4. The wrapping engine injects AirWatch SDK functionality.
5. The wrapping engine code-signs the application and recompresses the files.
6. The wrapping engine sends the download URL of the wrapped application to the internal network device services server.
7. The device services server downloads the wrapped application.
8. The device services server stores the wrapped application in the AirWatch database, along with auxiliary files.
9. Based on a scheduler task, the wrapping engine securely deletes original application files, provisioning profiles, and signing certificates.

The Storage of Files in the App Wrapping System

The app wrapping process deletes application binary files, provisioning profiles, and signing certificates from the app wrapping service when it completes wrapping. The system stores these files in the AirWatch database.

When adding a version of the application, the code signing files automatically populate and you can change them if needed. However, the app wrapping service does not store the files you supply.

The app wrapping service uses the application binary, signing certificate, and provisioning profile temporarily to sign the wrapped application. After wrapping is complete, the system removes the files from the wrapping service and stores them securely in the AirWatch database. If the wrapping fails or times out, the system automatically removes files from the wrapping service and stores them in the AirWatch database.

The Storage of Data in the App Wrapping System

The app wrapping system can log data about the wrapped application, but it does not store location, analytics, or telecom data.

Logging Payload in the Wrapping Profile

To deploy a wrapped application, you assign it a wrapping profile. This process is explained in App Wrapping Profiles. You can enable the logging payload and configure the logging level. When you apply the profile to the wrapped application, the system creates an application log. See Request App Wrapping Logs and Access Log Files for Apps That Use the SDK Framework.

If you do not want the console to log data about the application, ensure this feature is disabled. Find the setting in these places:

- In the default SDK settings in Settings and Policies
- In a custom app wrapping profile

Location, Analytics, and Telecom Data

The app wrapping system does not track location, analytics, or telecom data. Although, other sections of the console do if you configure the settings.

Copyright © 2018 VMware, Inc. All rights reserved.

- The AirWatch Agent tracks location data.
- The AirWatch SDK records analytics.
- The AirWatch Telecom dashboard reports telecom data for devices.

Disable these features if you do not want to track this data.

Cluster Session Management and App Wrapping for iOS

The latest version of the app wrapping engine introduces a new mechanism called the shared keychain. This mechanism is a feature for iOS wrapped apps to communicate with other wrapped apps on the device. This approach provides benefits from both a security and a user experience perspective.

iOS applications wrapped with the following components are in the same keychain group, also called a cluster.

- Apps wrapped with signing certificates from the same developer account
- Apps that share the same AppIdentifierPrefix

These applications can share session data like an app passcode and an SSO session. By sharing this session data, they do not have to flip to the AirWatch Agent or to the anchor application every time authentication is required.

Applications wrapped with the listed components are in different keychain groups.

- Apps wrapped with signing certificates from different developer accounts
- Apps that have a different AppIdentifierPrefix

These applications cannot take advantage of passcode sharing. These scenarios require flipping to the agent or the anchor application to obtain data like the server URL. This flipping action occurs once per cluster.

Cluster Session Management and Reduced Flip Behavior for SSO with AirWatch App Wrapping v5.4

An iOS application wrapped with app wrapping engine v5.4+ only the first wrapped app flips to the anchor application on the first launch. It flips to retrieve environment information. It does not flip to retrieve account data or to lock and unlock operations. In older versions of the wrapping engine, applications had to flip to the anchor application to retrieve data and to lock and to unlock operations.

SSO Sessions and SDK-Integrated Apps

The SSO session is a time frame created at the time of SDK unlock. During this time frame the application can access allowed network resources. If you enable SSO, all SDK-integrated applications are unlocked and able to share keychain information between them.

Supported Systems and Required Components for AirWatch App Wrapping

App wrapping works on specific platforms, bit versions, architectures, AirWatch versions, and environments. Review the supported components and requirements for app wrapping to ensure the solution integrates with your mobile deployment.

Copyright © 2018 VMware, Inc. All rights reserved.

Supported Platforms and Bit Versions

The application that you wrap must be compatible with the following components. If an application was built with an AirWatch SDK older than the version listed, it is not compatible with app wrapping.

Platform	Bit Versions and Architectures	Supported AirWatch Admin Console Version
Android v4.0.3+	32-bit 64-bit	AirWatch Console v8.4+
iOS v8.0+	ARMv7 ARMv7s ARM64	AirWatch Console v8.4+

Supported Android API Levels

AirWatch app wrapping works for applications built using the Android API level15 or higher. Older versions of the Android API do not build applications that are compatible with app wrapping.

Supported Deployments and Requirements

App wrapping is available for the following deployments, using the SaaS-Hosted app wrapping engine to wrap internal applications. The feature does not wrap public or purchased applications.

Deployment	App Wrapping Engine	App Type
SaaS	AirWatch SaaS-Hosted App Wrapping Engine	Internal Applications
On-Premise	AirWatch SaaS-Hosted App Wrapping Engine	Internal Applications

Cannot Wrap Store Apps

You cannot wrap applications from app stores, even if the APK or IPA comes from the vendor directly. To incorporate the AirWatch SDK into their applications, either use ACE or work with the vendors.

Cannot Support Android Apps Built With Crosswalk Project Libraries

Crosswalk on Android provides a packaging tool and a Java wrapper layer. They can bundle Web applications into the Android Web app APKs. This Java wrapper layer calls Crosswalk runtime, and Crosswalk runtime is a full-featured Web engine mostly written in C/C++. Android platforms do not package C/C++ code into SMALI files, and the app wrapping solution cannot modify and wrap the C/C++ libraries and code.

iOS App Wrapping Requirements

- **iOS Developer Enterprise Account** – Use this account to get Xcode used to compile the application as part of the wrapping process. Go to https://developer.apple.com/xcode/ for information. This account is aimed at developing iOS applications for use internally and not for deployment to an app store.

  To develop internal applications, ensure to get auxiliary files for enterprise (internal) distribution and not app store deployment.

Copyright © 2018 VMware, Inc. All rights reserved.

- **Mobile Provisioning Profile** – Get this file from Apple's Developer Portal. Get this profile for enterprise use, because it is specific to your application and to the Code Signing Certificate. The bundle ID of the provisioning profile matches the bundle ID of the IPA file.
- **Code Signing Certificate** – Get this file from Apple's Developer Portal. Get this file for enterprise use and not app store distribution, and use it to sign the wrapped application.
- **Sign the iOS Binary** — Sign the application with the provisioning profile and the signing certificate before wrapping the application.

Updated App Wrapping Engine for iOS 9

AirWatch updated the app wrapping engine to support iOS 9. Due to the updated engine, you must rewrap applications using the updated engine. The update enables wrapped applications to work on iOS 9 devices, which are new or are upgraded to the new iOS version.

Considerations

For app wrapping to succeed, an application must use certain processes, methods, and libraries. Review the listed considerations to check that the system can wrap your application.

Android Method Limits and Multi-DEXing Support

The compiler that app wrapping uses has a limit of 62 thousand methods for applications. With the support of multi-dexing, you can now create larger APKs with each DEX limited to 65 thousand methods. However, app wrapping needs to inject functionality into the application by adding methods to the primary DEX. To ensure wrapping completes, ensure that the primary DEX has 58 thousand methods or less. This method count gives the wrapping system room to inject methods into the primary DEX.

Example of Method Limiting in the Gradle File

```
afterEvaluate {
    tasks.matching {
        it.name.startsWith('dex')
    }.each { dx ->
        if (dx.additionalParameters == null) {
            dx.additionalParameters = []
        }
        dx.additionalParameters += "--set-max-idx-number=58000"
    }
}
```

Find information on how to limit methods on the Web from the listed site as January 2017.

- https://developer.android.com/studio/build/shrink-code.html

Copyright © 2018 VMware, Inc. All rights reserved.

Standard Processes

App wrapping works with Android and iOS applications developed using standard Android and iOS SDK processes.

Standard and C/C++ Libraries

App wrapping works with applications using standard Android and iOS Java/Objective-C layer libraries. If an application uses low-level C/C++ libraries, then some app wrapping features may not work or the application may not wrap properly.

Native Libraries in Android Apps

App Wrapping cannot fully support native libraries inside Android applications because the wrapping engine cannot interpret the processes these libraries invoke. Applications may wrap but these applications may not behave as expected after you install them on devices. Problems can arise with core functionalities, wrapping restrictions, tunneling, encryption, single sign-on, and other application processes.

Android Library Dependencies

Ensure that the listed libraries are not obfuscated in the original version of the application or wrapping fails.

- com.android.support:multidex:1.0.1
- com.android.support:support-v13:23.1.1
- com.android.support:appcompat-v7:23.1.1
- com.android.support:cardview-v7:23.1.1
- com.android.support:design:22.2.1
- com.google.guava:guava:18.0
- org.apache.commons:codec:1.7
- org.apache.commons:io:2.4
- org.apache.commons:lang3:3.1
- com.google.zxing:zxing:3.2.1
- com.sqlcipher:3.3.1-2
- com.squareup.okhttp:2.0.0-RC1
- org.apache.commons:codec:1.7
- org.apache.commons:io:2.4"
- org.apache.commons:lang3:3.1
- com.google.gson:gson:2.4
- libdatabase_sqlcipher.so
- libecjni.so
- libf5apptun.so

Copyright © 2018 VMware, Inc. All rights reserved.

- libiocipher.so
- libsqlcipher_android.so
- libstlport_shared.so

Using Apps Developed in Swift

AirWatch application settings and policies support applications developed in Swift. Ensure that the application continues to use Objective-C runtime and libraries. As of now, iOS is continuing to use Objective-C libraries and APIs by default.

Tampering Protection

Remove tampering protection from the application you want to wrap. App wrapping involves altering the application so app wrapping cannot work with this protection enabled.

Entitlements for iOS Apps

Enable the keychain-access-groups permission in the entitlements of iOS applications before wrapping. This permission allows AirWatch to store Secure Channel Certificates in the iOS keychain of the application because AirWatch uses Secure Channel Certificates to communicate.

If you do not enable this permission, AirWatch automatically enables the permission. If your mobile provisioning profile does not have the keychain-access-groups listed in the entitlements, you might have a wrapping issue . The wrapped application might not behave as expected when installed on devices.

Mobile Provisioning Profile for iOS Apps

Ensure you use a mobile provisioning profile that matches the bundle ID of the application. Wildcard provisioning profiles might not allow the use of certain entitlements, like iCloud.

Synchronous Calls and iOS Apps

Avoid synchronous calls, if possible. Instead, consider using asynchronous methods or putting synchronous calls in their own threads. Synchronous logic can negatively impact the ability of the feature to intercept preventable calls.

Xamarin Requirements for Android

AirWatch is certified to wrap applications built using Xamarin, but you must override all methods by the super class. To override method() from the super class, call super.method() in the method(). This process requires the addition of code to all applicable classes.

Code to Add

```
@Override
Public void onCreate(Bundle param ){
Super.onCreate(param);        // make sure you have this call in order for App Wrapping to be
```

Copyright © 2018 VMware, Inc. All rights reserved.

```
supported with Xamarin apps
}
```

Add this code to all classes extending to the listed classes.

- Application.class
- Activity.class
- AppCompatActivity.clas,
- AccountAuthenticatorActivity.class
- ExpandableListActivity.class
- FragmentActivity.class
- ListActivity.class
- NativeActivity.class
- LauncherActivity.class
- PreferenceActivity.class
- Webview.class
- WebviewClient.class

Retrieve User Name Data

Retrieve usernames for AirWatch enrolled users from a secure location in the wrapping layer. Use this information to authorize these users to access your other systems and to check against their user roles. This procedure is optional when using the app wrapping feature and it works for only Android.

Place this code in the application before you upload it to the AirWatch Console for wrapping. Code the application to expect the user name value during the wrapping process using the listed string.

```
String username= java.lang.System.getProperty("aw-username");
```

Get the user name directly from the system property **aw-username**.

Copyright © 2018 VMware, Inc. All rights reserved.

Chapter 2: Enable App Wrapping for On-Premise Deployments

Copyright © 2018 VMware, Inc. All rights reserved.

Enabling the App Wrapping Engine

Enable the AirWatch app wrapping feature to communicate with your network server in your on-premises environment. SaaS deployments do not configure this option because it is already configured.

1. Navigate to **Groups & Settings > All Settings > System > Advanced > Site URLs**.
2. Select **Enable App Wrapping** in the app wrapping section.
3. Complete the entry for your platform.

   SaaS Wrapping Endpoint

   - **iOS** – Enter the URL for the AirWatch SaaS-Hosted App Wrapping Server for iOS, **https://appwrap04.awmdm.com**.
   - **Android** – Enter the URL for the AirWatch SaaS-Hosted App Wrapping Server for Android, **https://appwrapandroid.awmdm.com**.

Enabling Cloud Services

Enable or disable Cloud Services depending on your AirWatch environment. You can disable this setting to troubleshoot app wrapping issues, but this action reduces security because it bypasses HMAC authentication.

SaaS deployments do not configure this setting because it is already configured.

Configure this setting at the **Global** organization group (OG) using system administrator credentials.

1. Navigate to **Groups & Settings > All Settings > Admin > Cloud Services**.
2. Select **App Wrapping Secure Communication Enabled**.

   If the application fails to wrap, you can disable the checkbox and try to wrap again. However, clearing the checkbox bypasses the HMAC token authentication check, making this option less secure.

   Check the **Auto Discovery AirWatch Id** entry in the AirWatch ID section (on the same page as the Cloud Services section). This ID is your AirWatch credentials and provides a secure connection with the AirWatch Cloud.

Copyright © 2018 VMware, Inc. All rights reserved.

Chapter 3 : App Wrapping Profiles

Copyright © 2018 VMware, Inc. All rights reserved.

MAM Functionality With Settings and Policies and the AirWatch SDK

The Settings and Policies section of the AirWatch Console contains settings that can control security, behaviors, and the data retrieval of specific applications. The settings are sometimes called SDK settings because they run on the AirWatch SDK framework.

You can apply these SDK features to applications built with the AirWatch SDK, to supported AirWatch applications, and to applications wrapped by the AirWatch app wrapping engine because the AirWatch SDK framework processes the functionality.

Types of Options for SDK Settings

AirWatch has two types of the SDK settings, default and custom. To choose the type of SDK setting, determine the scope of deployment.

- Default settings work well across organization groups, applying to large numbers of devices.
- Custom settings work with individual devices or for small numbers of devices with applications that require special mobile application management (MAM) features.

Default Settings

Find the default settings in **Groups & Settings > All Settings > Apps > Settings And Policies** and then select **Security Policies** or **Settings**. You can apply these options across all the AirWatch applications in an organization group. Shared options easier to manage and configure because they are in a single location.

View the matrices for information on which default settings apply to specific AirWatch applications or the AirWatch SDK and app wrapping.

Custom Settings

Find the custom settings in **Groups & Settings > All Settings > Apps > Settings And Policies > Profiles**. Custom settings for profiles offer granular control for specific applications and the ability to override default settings. However, they also require separate input and maintenance.

Assign the Default or Custom Profile

To apply AirWatch features built with the AirWatch SDK, you must apply the applicable default or custom profile to an application. Apply the profile when you upload or edit the application to the AirWatch Console.

1. Navigate to **Apps & Books > Applications > List View > Internal**.
2. Add or edit an application.
3. Select a profile on the **SDK** tab:
   - **Default Settings Profile**
     - For Android applications, select the **Android Default Settings @ <Organization Group>**.
     - For Apple iOS applications, select the **iOS Default Settings @ <Organization Group>**.
   - **Custom Settings Profile** – For Android and Apple iOS applications, select the applicable legacy or custom profile.
4. Make other configurations and then save the application and create assignments for its deployment.

Copyright © 2018 VMware, Inc. All rights reserved.

Changes to Default and Custom Profiles

When you make changes to the default or custom profile, AirWatch applies these edits when you select **Save**.

Changes can take a few minutes to push to end-user devices. Users can close and restart AirWatch applications to receive updated settings.

Configure Custom App Wrapping Profile Settings

App wrapping applies functionality with the AirWatch SDK framework. You configure these features in the Settings and Policies section of the console. These settings are the default settings. If you want to create exceptions to the default settings, use custom settings for your wrapped applications.

Add a profile with the desired configurations and apply that profile to the application.

1. Navigate to **Groups & Settings > All Settings** > **Apps** > **Settings and Policies**> **Profiles**.
2. Select **Add Profile** and choose **App Wrapping Profile** and the applicable platform.
3. Configure **General** settings and then complete the settings for the desired payload. Most of the options available in the default settings sections are available in the custom payload profiles. You can view the default explanations for descriptions of what the payloads do.

Supported Settings and Policies Options By Component and AirWatch App

Use the default settings profile to apply an AirWatch SDK feature to an SDK application, an AirWatch application, or a wrapped application by setting the configurations in **Policies and Settings** and then applying the profile. View compatibility information to know what features AirWatch supports for your application.

Scope of Matrices

The data in these tables describes the behaviors and support of the specific application.

Settings and Policies Supported Options for SDK and App Wrapping

UI Label	App Wrapping	
	iOS	Android
Passcode: Authentication Timeout	✓	✓
Passcode: Maximum Number Of Failed Attempts	✓	✓
Passcode: Passcode Mode Numeric	✓	✓
Passcode: Passcode Mode Alphanumeric	✓	✓
Passcode: Allow Simple Value	✓	✓
Passcode: Minimum Passcode Length	✓	✓

Copyright © 2018 VMware, Inc. All rights reserved.

UI Label	App Wrapping	
	iOS	Android
Passcode: Minimum Number Complex Characters	✓	✓
Passcode: Maximum Passcode Age	✓	✓
Passcode: Passcode History	✓	✓
Passcode: Biometric Mode	✓	x
Username and Password: Authentication Timeout	✓	✓
Username and Password: Maximum Number of Failed Attempts	✓	✓
Single Sign On: Enable	✓	✓
Integrated Authentication: Enable Kerberos	x	x
Integrated Authentication: Use Enrollment Credentials	✓	* ✓
Integrated Authentication: Use Certificate	x	x
Integrated Authentication: Use NAPPS Authentication	x	x
Offline Access: Enable	✓	✓
Compromised Detection: Enable	✓	✓
AirWatch App Tunnel: Mode	✓	✓
AirWatch App Tunnel: URLs (Domains)	✓	✓
Geofencing: Area	x	x
DLP: Bluetooth	x	✓
DLP: Camera	✓	✓
DLP: Composing Email	✓	✓
DLP: Copy and Paste Out	✓	✓
DLP: Copy and Paste Into	✓	✓
DLP: Data Backup	✓	x
DLP: Location Services	✓	✓
DLP: Printing	✓	✓
DLP: Screenshot	x	✓
DLP: Third Party Keyboards	✓	✓

Copyright © 2018 VMware, Inc. All rights reserved.

U I Label	App Wrapping	
	iO S	Android
DLP: Watermark	✓	✓
DLP: Limit Documents to Open Only in Approved Applications	✓	✓
DLP: Allowed Applications List	✓	✓
N AC: Cellular Connection	✓	X
N AC: Wi-Fi Connection	✓	✓
N AC: Allowed SSIDs	✓	✓
Branding: Toolbar Color	✓	X
Branding: Toolbar Text Color	✓	X
Branding: Primary Color	✓	X
Branding: Primary Text Color	✓	X
Branding: Secondary Color	✓	X
Branding: Secondary Text Color	✓	X
Branding: Organization Name	X	X
Branding: Background Image iPhone and iPhone (Retina)	X	X
Branding: Background Image iPhone 5 (Retina)	X	X
Branding: Background Image iPad and iPad (Retina)	X	X
Branding: Background Small, Medium, Large, and XLarge	X	X
Branding: Company Logo Phone	X	X
Branding: Company Logo Phone High Res	X	X
Branding: Company Logo Tablet	X	X
Branding: Company Logo Tablet High Res	X	X
Logging: Logging Level	✓	✓
Logging: Send Logs Over Wi-Fi	✓	✓
Analytics: Enable	X	X
Custom Settings	X	X

* ✓ This option is supported only on Android apps that use Webview.

Copyright © 2018 VMware, Inc. All rights reserved.

Chapter 4 : Trou blesh ooting Tip s

Copyright © 2018 VMware, Inc. All rights reserved.

Troubleshoot App Wrapping

If you have problems wrapping an internal application, try general troubleshooting steps to find and fix the issue.

Actions to try include to remove AirWatch from the process, to check the communication with VMware Tunnel, and to review app wrapping logs.

With the complexity of mobile networks, it might be necessary to involve Professional Services, if this level of support is part of the services agreed upon by the organization and AirWatch. Pass on to them any data gathered from performing the listed troubleshooting steps.

- Side-load the application, unwrapped, and watch the behavior. This step takes AirWatch out of the process and ensures that the application works as expected.
- Download app wrapping logs to look for issues. See the Request App Wrapping Logs topic for details on how to access these logs.
- If the AirWatch Admin Console reports that wrapping failed, Professional Services can access and review app wrapping engine logs to find issues.
- For iOS platforms, resign applications to see if the provisioning profile and signing certificate work. Side-load the resigned application and see if it works as expected.
- For environments that use the VMware Tunnel, test the Tunnel. Access a secure site with the AirWatch Browser through the Tunnel to make sure that the Tunnel directs traffic as expected.
- If the application developer used Mobile App Development Platforms, or MADPs, to build the applications, ensure that the wrapping engine supports it. See the Mobile App Development Platform, MADP Support topic.
- Ensure that the application developer used supported methods and libraries to build the application. See the Developer Resources topic.

Troubleshoot No Static Method Error

If a wrapped Android application crashes on the device, check ADB device logs for a No Static Method error message.

```
java.lang.NoSuchMethodError: No static method
addAccessibilityStateChangeListener in class
Landroid/support/v4/view/accessibility/AccessibilityManagerCompat;
```

This error displays when a developer built the original application with a dependency or library not supported by the app wrapping system. Refer to the Developer Resources for a list of supported dependencies and versions for Android.

The app wrapping system can wrap applications built with non-supported dependencies and libraries but those applications crash on devices.

Re-wrapping Applications

Applications require re-wrapping for several reasons that include app wrapping engine updates, operating system changes, and system fixes. The console identifies wrapping issues in the console so you know to re-wrap an application.

Copyright © 2018 VMware, Inc. All rights reserved.

The re-wrap process follows the same steps as the original wrapping process except you must build a new version of the application before you upload it to the console.

1. Build a new version of the app so that it has a version number that is greater than the currently deployed version in AirWatch.
2. Upload the rebuilt app to AirWatch in **Apps & Books > Applications > Native > Internal**.
3. Use the app wrapping tab to re-wrap the application.

Wrapping Success and the Engine

AirWatch does not push a wrapped app to devices until the wrapping engine reports wrapping success. Find the success status in the AirWatch Console at **Apps & Books > Applications > Native > Internal** and view the **Wrap Status** column.

If wrapping fails, use the **Queue App For Wrapping** check box on the **App Wrapping** tab, which only displays upon failure. The wrapping engine re-wraps the application after you select **Save & Publish** from the flexible deployment page. When the wrapping engine reports success, AirWatch pushes the application to devices. This work flow prevents pushing failed wrapped applications to devices.

Request App Wrapping Logs

You can request logs for your wrapped applications with the request logs feature. When requested, the system writes an application log.

Another type of log for wrapped apps is the crash log. The system automatically writes this data when the wrapped application crashes.

To request application logs for wrapped applications, use this process.

1. Navigate to **Devices > List View** and select the device.
2. Select the **Apps** tab and choose **Request Logs** from the actions menu.
3. Navigate to **Apps & Books** > **Applications > Analytics > App Logs**.
4. Find the log by the bundle ID in the **App Name** column and select **Download** from the actions menu.

Access Log Files

Access logs that are sent by the AirWatch SDK.

Access SDK and Wrapped App Logs by Log File

Access SDK application logs from the App Logs page.

1. Navigate to **Apps & Books** > **Applications > Analytics > App Logs**.
2. Download or delete logs using the actions menu.

Copyright © 2018 VMware, Inc. All rights reserved.

Access Logs by the View Logs Page

Use the V iew Logs feature from the actions menu to quickly access available log files pertaining to a single application.

1. Navigate to **Apps & Book s > Applications > N ative** and select the **I nternal** tab.
2. Select the application.
3. Select the **V iew Logs** option from the actions menu.

Use the AirWatch App Tunnel and Per-App V PN as a Wrapping Alternative

If you use app wrapping to tunnel to network resources, an alternative solution is to use the App Tunnel and Per-App V PN. This alternative does not require the maintenance associated with rewrapping applications after a wrapping engine update.

It works as an alternative if you only want the application to tunnel into the internal network to access resources. If you do not need advanced management features for the application, then consider using tunneling and per-app V PN.

Component Ex planations and Configurations

The Per App Tunnel component and V Mware Tunnel apps for iOS, Android, Windows Desktop, and macOS allow both internal and public applications to access corporate resources that reside in your secure internal network. They allow this functionality using per app tunneling capabilities. Per app tunneling lets certain applications access internal resources on an app-by-app basis. This restriction means that you can enable some apps to access internal resources while you leave others unable to communicate with your back-end systems.

This alternative solution is different from app tunneling with app wrapping because it supports both TCP and HTTP(S) traffic. It works for both public and internally developed apps. However, for internal apps, the V Mware Tunnel app acts as an alternative option only if the sole requirement is tunneling into the internal network. Otherwise, you must use app wrapping to take advantage of features including integrated authentication, geofencing, offline access control, and so on.

After configuring and installing V Mware Tunnel with the Per-App Tunnel component, the workflow to enable and use per app tunneling in AirWatch includes:

1. Creating a V PN profile for your end-user devices. These profiles depend on your device platform.

   If your platform uses user profiles and device profiles, such as Windows Desktop and Android, you must create user profiles.

2. After creating a V PN profile, push the profiles and the apps to the devices.

   For iOS and Android platforms, you must enable the Use V PN check box on the Deployment tab of the Add Application page to use app tunneling.

Windows Desktop devices use the native Per-App V PN functionality. Add the apps to the V PN profile to enable Per-App Tunnel functionality.

N ote: V Mware Tunnel does not support Per-App V PN functionality for macOS devices. You can restrict access to domains through the Safari Domains feature of the Network Traffic rules. For more information, see Network Traffic Rules for Per-App Tunnel.

Copyright © 2018 V Mware, Inc. All rights reserved.

Additional Details

An on-demand feature lets you configure apps to connect automatically using VMware Tunnel when launched. The connection remains active until a time-out period of receiving no traffic, then it is disconnected. When using VMware Tunnel, no IP address is assigned to the device, so you do not need to configure the network or assign a subnet to connected devices.

In addition, iOS apps can use the iOS DNS Service to send DNS queries through the VMware Tunnel server to the DNS server on a corporate network. This service allows applications such as Web browsers to use your corporate DNS server to look up the IP address of your internal Web servers.

See the **VMware Tunnel Admin Guide** using the AirWatch Resources for information.

Review App Wrapping and Tunnel Support for iOS APIs to see what iOS APIs are supported for app wrapping and the AirWatch App Tunnel.

Known Issues

Review known issues and their workarounds. Also, view explanations for the issue in the app wrapping feature.

Known Issue – Browsing Web Sites and Accessing HTTP Endpoints, iOS

Browsing Web sites and accessing HTTP endpoints is slow when you use the VMware Tunnel. This behavior occurs only on iOS.

Explanation

When accessing a Web site or an HTTP/S endpoint using the VMware Tunnel, every request is signed for VMware Tunnel validation. This signing can add significant overhead for Web sites that have many requests.

A Web page that contains many resources (images, css, and javascript files) exhibits delays because each resource that is downloaded is signed. For example, a page with 50 images and many javascript files sees delays much greater than a Web page with only 5 resources.

AirWatch is developing new VMware Tunnel functionality to resolve this architectural issue.

Note: The known issue does not affect Android.

Workaround

The app wrapping version deployed with AirWatch v7.1 improves the performance of browsing in Web sites using HTTPS.

The latest app wrapping version does not improve the slow behavior with Web sites that use HTTP.

Consider creating a self-signed SSL certificate for the Web site or endpoint and test the browsing speed with the new app wrapping implementation.

Known Issue – DAR, Data at Rest, Encryption

AirWatch v7.1+ and the app wrapping feature does not support DAR encryption for the app wrapping engine for iOS. However, it does support DAR encryption for the app wrapping engine for Android. AirWatch uses the Advanced Encryption Standard, AES-256, with encrypted keys for encryption and decryption.

Copyright © 2018 VMware, Inc. All rights reserved.

Ex planation, Android

When you enable DAR in app wrapping, the app wrapping engine injects an alternative file system into the application. It securely stores all the data in the application. The application uses the alternative file system to store all files in an encrypted storage section instead of storing files in disk.

DAR encryption helps protect data in case the device is compromised because the encrypted files created during the lifetime of the application are difficult to access by an attacker. This protection applies to any local SQ Lite database, because all local data is encrypted in a separate storage system.

Ex planation, iO S

Although AirWatch v7.1+ and the app wrapping feature do not currently support DAR for iOS, review the following information on data protection when developing iOS applications.

Data Protection in iO S 7

iOS 7 includes data protection for all third-party applications. This data protection requires no action by a developer to enable the DAR encryption. However, it requires the device user to set a passcode.

The data protection level that is enabled by default is the same as the **Complete until first login** mode. The local files are encrypted from the time the device restarts to the time the end-user unlocks the device.

Known Issue – Incorrect Parameter Error for iOS Applications

Save Failed error displays after uploading a wrapped iOS application to the AirWatch Console.

Ex planation

When uploading iOS applications to AirWatch, you also upload the corresponding certificates and provisioning profile. A corrupted certificate can cause the following error when wrapping an iOS application in AirWatch.

Work Around

Check the validity of the certificate using these processes.

- V alidate the bundle ID of the application to the corresponding certificate and provisioning profile.
- V alidate the certificate on a Mac device by double-clicking the certificate file and adding it to the Keychain. If the certificate fails to add to the Keychain, the certificate does not work.
- V alidate the certificate on Windows by double-selecting the certificate to import it to the local machine. If the import wizard displays an error at any time, the certificate does not work.
- V alidate that the certificate has the P12 file extension. If it does not, the certificate does not work.

Known Issue – Wrapped App Run Failure

Wrapped apps loop continuously when starting from the AirWatch Agent or the AirWatch Container application.

Ex planation

A possible cause for the loop is a setting in the wrapped app that forces the application to close when you send it to the background.

Copyright © 2018 V M ware, Inc. All rights reserved.

Work Around

Check the PLIST file for the setting **UIApplicationExitsOnSuspend**. If this option is enabled, remove the setting a rewrap the application.

Known Issue – Issues Wrapping With Apple iOS 8

Applications are not wrapping successfully or are not loading on to devices running Apple iOS 8 after wrapping.

Explanation – Compatibility

Applications developed to run on Apple iOS 8 are not functioning as expected when tunneling through VMware Tunnel or using other application settings and policies.

Work Around – Compatibility

Validate the date the app was wrapped and the app wrapping engine version to ensure that the engine was compatible with the iOS version. If the wrapping date or engine version is different than what is listed, rewrap the application.

Find this information on the **Wrapping** tab by navigating to **Apps & Books > Applications > Native > Internal**. Select **Edit** from the actions menu of the wrapped app to view the Wrapping tab.

- Use the **Wrapped Engine version 3.2.1+**. If the engine version was lower than 3.2.1, the older engine version might have caused an issue with wrapping.
- Check that the date the app was wrapped is after **September 15, 2014**. If it was before this date, the app wrapping engine was not compatible with Apple iOS 8 at the time.

Explanation – Code Signing Signature

Applications developed to run on Apple iOS 8 are not functioning as expected. The application cannot find the code signing signature as recorded in this MMAP error.

```
[deny-mmap] mapped file has no team identifier and is not a platform binary:
/private/var/mobile/Containers/Bundle/Application/………..//…………/libappwrap.dylib
```

Work Around – Code Signing Signature

Regenerate the signing certificate and the mobile provisioning file and rewrap the application. Reupload the application and the regenerated auxiliary files in the AirWatch Console.

Copyright © 2018 VMware, Inc. All rights reserved.

Chapter 5: App Wrapping Resources

Copyright © 2018 VMware, Inc. All rights reserved.

Developer Resources

Review the developer resources supported by AirWatch. Use the tables to identify what methods and libraries to use with app wrapping and application management. Additional comments list partial support, suggest how to use the resource, or suggest other informational sites.

These lists are not comprehensive.

iOS Wrapped Apps

Features	Options	Supported by AirWatch	Comments
Project Template Designates the method to implement iOS applications	Storyboards	Yes	
	NIBs Only	Yes	
	Master/Detail Template	Yes	
	Tabbed Template	Yes	
Networking Redirects HTTP and HTTPS traffic using the VMware Tunnel or other proxies	NSURLConnection	Yes	
	NSURLSession	Yes	Configure NSURLSession using **[NSURLSession sharedSession]**. Not Supported: **NSURLSession Download Task**. See App Wrapping Support for iOS APIs for more information.
	AFNetworking Version 1	Yes	
	AFNetworking Version 2	Partially	Supported: • AFHTTPRequestOperation • AFHTTPRequestOperationManager • AFURLConnectionOperation Not Supported: • AFURLSessionManager • AFHTTPSessionManager
	ASIHTTPRequest	No	

Copyright © 2018 VMware, Inc. All rights reserved.

F eatures	O ptions	Supported by AirWatch	Comments
Email Composing Prevents an application from using the native email client to send emails using data loss prevention settings	MFMailComposeV iewController	Yes	Check the **canSendEmail** property before use. See the following site for more information: https://developer.apple.com/library /ios/ documentation/M essageUI /Reference/ M FM ailC omposeV iewC ontroller_ class/Reference/Reference.html# //apple_ ref/doc/uid/TP4 0 0 0 8 2 0 0
Copy and Paste Prevents users from copying content from the wrapped application into other applications using data loss prevention settings	UITextField	Yes	
	UITextV iew	Yes	
	UIWebV iew	Yes	AirWatch cannot block certain HTML input options in a Web page.
Integrated Authentication Authenticates a user automatically against NTLM and basic Web sites or Web services	Web Service/Website Authentication	Yes	Ensure the endpoint uses NTLM or basic authentication.
Camera Blocks access to the camera within the wrapped application	UIImagePickerController	Yes	Currently does not block picking from the photo roll.
	AV CaptureSession	Yes	
iCloud Data Back up Blocks data from syncing with iTunes	NSFileManager	Yes	Blocks the property **ubiq uityI dentityT ok en** used to synchronize with iCloud.
Opening Documents in Other Apps Blocks wrapped applications from opening documents in other applications	UIDocumentInteractionController	Yes	
	UIActivityV iewController	Yes	

Copyright © 2018 VMware, Inc. All rights reserved.

F eatures	O ptions	Supported by AirWatch	Comments
Location Services Blocks wrapped applications from using location services to retrieve the current location of the device	CLLocationManager	Yes	Use the properties to check if services are available before use. See the following site for more information: https://developer.apple.com/library /mac/ documentation/C oreLocation/Reference/ C LLocationM anager_ C lass/C LLocationM anager/ C LLocationM anager.html

Android Wrapped Apps

F eatures	O ptions	Supported by AirWatch	Comments
Networking Redirects HTTP and HTTPS traffic using the V Mware Tunnel or other proxies	F5	Yes	Covers all App level http/https communications.
	V Mware Tunnel	Yes	Works at the following component levels: • android/webkit/WebV iew • android/webkit/WebV iewClient • com/squareup/okhttp/OkHttpClient • java/net/URL • org/apache/http/impl/client/AbstractHttpClient • org/apache/http/impl/client/DefaultHttpClient • org/apache/http/impl/client/ HttpClientAndroidLib • org/xwalk/core/XWalkV iew The V Mware Tunnel supports only HTTP and HTTPS traffic, so you cannot use classes such as Socket().

Copyright © 2018 V M ware, Inc. All rights reserved.

F eatures	O ptions	Supported by AirWatch	Comments
Data At Rest Encryption Encrypts data stored on the application	Java File I/O System	Yes	Supports the following classes: • java/io/FileInputStream • java/io/FileReader • java/io/FileOutputStream • java/io/FileWriter • Context > openFileInput • Context > openFileOutput • android/os/ParcelFileDescriptor (specific to the shared input stream through the Content provider) • java/io/File
	Database Support (SQ L Lite)	Yes	Net.sql.cipher.SQ LiteDatabase.openOrCreateDatabase (databaseFile, password, null)
Camera Blocks access to the camera within the wrapped application	android.hardware.Camera	Yes	Restricts at the API level.
	MediaStore.ACTION_ IMAGE_CAPTURE Intent	Yes	Restricts at a device level.
Opening Documents in Other Apps Blocks wrapped applications from opening documents in other applications	Intent.ACTION_V IEW	Yes	Controls “ Open File with” using the Intent approach start Activity.
File Sharing with BluetoothControls sharing files with Bluetooth	Intent.ACTION_SEND Intent.ACTION_CHOOSER	Yes	Controls file sharing using the Intent approach start Activity.

Copyright © 2018 VMware, Inc. All rights reserved.

F eatures	O ptions	Supported by AirWatch	Comments
Stream Sharing with Bluetooth Controls application in-built Bluetooth (point to point) communication	BluetoothDevice.ACTION_ ACL_CONNECTED BluetoothAdapter.ACTION_ DISCOV ERY_STARTED	Yes	
Location Access Controls application location change listening capability	LocationListener > OnLocation Changed (Location loc)	Yes	In restricted mode, it blocks location update callbacks.

Mob ile App Development Platform, MADP Support

A mobile app development platform (MADP) is a system that attempts to reduce the development effort for creating mobile applications. AirWatch has partnered with various MADP vendors to ensure wrapping functionality on applications developed with the platform.

Vendor	Certification Status	Certification N otes
Adobe Phonegap	Certified	Wrapping functions with applications developed using Phonegap.
Appcelerator	Certified	Wrapping functions with applications developed using Appcelerator.
Cordova	Certified	Wrapping functions with applications developed using Cordova-based platforms.
IBM Worklight	Certified	Wrapping succeeds with applications developed using IBM Worklight.
Kony	Certified	Wrapping functions with applications developed using Kony.
MicroStrategy		Use supported configurations from the AppConfig Community or use the AirWatch SDK instead of wrapping. Due to partner integration with Microstrategy, the app developer manually includes the AirWatch SDK into the project for AirWatch specific functionalities. For example, App Tunneling and App Authentication (Passcode and Username/Password).
Oracle MAF Mobile	Certified	Wrapping functions with applications developed using Oracle MAF Mobile.
Pegasystems Antenna	Certified	Wrapping functions with applications developed using Pegasystems Antenna.

Copyright © 2018 VMware, Inc. All rights reserved.

Vendor	Certification Status	Certification Notes
Salesforce Touch Platform	Certified	Wrapping functions with applications developed using the Salesforce Touch Platform.
SAP	Certified	Wrapping succeeds with applications developed using SAP.
Sencha	Certified	Wrapping functions with applications developed using Sencha.
Telerik	Certified	Wrapping functions with applications developed using Telerik.
Xamarin	Certified with caveats (see Xamarin Support for Android)	Wrapping functions with applications developed using Xamarin. Implement the ModernHttpClient library. See https://github.com/paulcbetts/ModernHttpClient for more information. **Caution**: For iOS apps, wrapping is not supported if Xamarin Insights is used because this addition can cause a crash upon starting. The Insights library is used for crash reporting and the AirWatch SDK. The AirWatch SDK also has a code for crash reporting that the SDK system injects into the app during the time of wrapping. The two together can conflict and cause a crash.

App Wrapping and Tunnel Support for iOS APIs

The listed APIs are compatible with app wrapping or with the AirWatch App Tunnel. Identify APIs that you use and see if the Tunnel can meet your needs as an app wrapping alternative.

This list is not comprehensive.

Key

Option	Description
✓	Supports using the API.
X	Does not support using the API.
Researching	Researching compatibility.
Partial support	Supports using the API but not with on-demand features.

iOS API	iOS 7.X + App Wrapping	iOS 7.X + AW Tunnel	iOS 8.X + App Wrapping	iOS 8.X + AW Tunnel	iOS 9.X + App Wrapping	iOS 9.X + AW Tunnel
NSURLSession – Data Task	✓	✓	✓	Researching	✓	✓
NSURLSession – Download Task	X	✓	X	✓	X	✓

Copyright © 2018 VMware, Inc. All rights reserved.

iOS API	iOS 7.X + App Wrapping	iOS 7.X + AW Tunnel	iOS 8.X + App Wrapping	iOS 8.X + AW Tunnel	iOS 9.X + App Wrapping	iOS 9.X + AW Tunnel
NSURLConnection	✓	✓	✓	Researching	✓	✓
CFHTTP	✓	✓	✓	✓	✓	✓
CFSocket (TCP)	Researching	Researching	Researching	Researching	Researching	✓
CFSocket (UDP)	X	X	X	X	X	X
BSD networking (TCP)	X	Partial support	X	Partial support	X	Partial support
BSD networking (UDP)	X	X	X	X	X	X
BSD networking (DNS)	X	Partial support	X	Partial support	X	Partial support
WKWebView	Researching	Researching	X	✓	X	✓
UIWebView	✓	✓	✓	Researching	✓	Researching
Background tasks	Researching	✓	Researching	✓	Researching	✓

Copyright © 2018 VMware, Inc. All rights reserved.

Accessing Other Documents

While reading this documentation you may encounter references to documents that are not included here.

The quickest and easiest way to find a particular document is to navigate to https://my.air-watch.com/help/9.2/en/Content/Release_Notes/Doc_List_PDFs.htm and search for the document you need. Each release-specific document has a link to its PDF copy on AirWatch Resources.

Alternatively, you can navigate to AirWatch Resources on myAirWatch (resources.air-watch.com) and search. When searching for documentation on Resources, be sure to select your AirWatch version. You can use the filters to sort by PDF file type and AirWatch App Wrapping v5.4.

Copyright © 2018 VMware, Inc. All rights reserved.