Upgrading Widevine DRM to Android P Release

Version 1.1

Overview

This document describes the steps required to integrate Widevine DRM on Android P devices. This document is an addendum to “Widevine Security Integration Guide for CENC: Android Supplement”. A separate document is needed for devices that update from Android O to Android P because several data files have changed location. Most Android versions do not need a separate document.

The steps are: adding the Widevine service to the device’s build files, updating the device manifest, setting SELinux permissions and supporting a data migration operation for existing devices upgrading to P release.

Adding Widevine to device build files

The <device>/device.mk file for the device must include the following product packages:

```
PRODUCT_PACKAGES += \
    android.hardware.drm@1.0-impl \
    android.hardware.drm@1.0-service \
    android.hardware.drm@1.1-service.widevine \
    android.hardware.drm@1.1-service.clearkey
```

Updating the device manifest

The vendor manifest.xml file for the device must include the following entries:

```
   <hal format="hidl">
        <name>android.hardware.drm</name>
        <transport>hwbinder</transport>
        <version>1.0</version>
        <interface>
            <name>ICryptoFactory</name>
            <instance>default</instance>
        </interface>
        <interface>
            <name>IDrmFactory</name>
            <instance>default</instance>
        </interface>
        <fqname>@1.1::ICryptoFactory/clearkey</fqname>
        <fqname>@1.1::IDrmFactory/clearkey</fqname>
        <fqname>@1.1::ICryptoFactory/widevine</fqname>
        <fqname>@1.1::IDrmFactory/widevine</fqname>
    </hal>
```

Setting SELinux permissions

1. Add to <device>/sepolicy/vendor/file.te

```
type mediadrm_vendor_data_file, file_type, data_file_type;
```

2. Add to <device>/sepolicy/vendor/file_contexts

```
/vendor/bin/hw/android\.hardware\.drm@1\.1-service\.widevine
u:object_r:hal_drm_widevine_exec:s0

/data/vendor/mediadrm(/.*)? u:object_r:mediadrm_vendor_data_file:s0
```

3. Add to <device>/sepolicy/vendor/hal_drm_widevine.te

```
allow hal_drm_widevine mediadrm_vendor_data_file:dir create_dir_perms;
allow hal_drm_widevine mediadrm_vendor_data_file:file create_file_perms;
```

Data File Migration

Prior to P release, Widevine DRM stored data in /data/mediadrm. Starting with P release, HALs are not allowed to access files on /data outside of /data/vendor. Therefore, for **existing devices that are running Widevine DRM prior to P release**, data stored in /data/mediadrm must be moved to /data/vendor/mediadrm. Furthermore, for devices that are running Android N and are upgrading to P directly and not through Android O, Level 3 data has been migrated in O from /data/mediadrm/ to /data/mediadrm/L3. So, Level 3 data stored in /data/mediadrm/ in Android N needs to be transferred to /data/vendor/mediadrm/L3 in P. New devices releasing with P, or devices that have not included Widevine DRM prior to P but are upgrading to P do not need to install and run this script.

Widevine provides a script in /vendor/widevine/libwvdrmengine/move_widevine_data.sh that copies the data files from /data to /data/vendor. Some configuration steps are required to enable the device to run this script.

Adding move_widevine_data.sh dependency to device.mk

To install and run this script, vendors must add move_widevine_data.sh as a dependency to the DRM HAL's PRODUCT_PACKAGES in the device-specific makefile, <device>/device.mk

```
# DRM HAL Data Migration
PRODUCT_PACKAGES += move_widevine_data.sh
```

Changing SELinux Policy Files

Add the following lines in the device SELinux policy files as shown below:

1. Add to <device>/sepolicy/vendor/file_contexts

```
/system/bin/move_widevine_data\.sh u:object_r:move-widevine-data-sh_exec:s0
```

2. Create <device>/sepolicy/vendor/move-widevine-data-sh.te

```
type move-widevine-data-sh, domain, coredomain;
type move-widevine-data-sh_exec, exec_type, file_type;
init_daemon_domain(move-widevine-data-sh);

typeattribute move-widevine-data-sh data_between_core_and_vendor_violators;
```

```
allow move-widevine-data-sh shell_exec:file rx_file_perms;
allow move-widevine-data-sh toolbox_exec:file rx_file_perms;

allow move-widevine-data-sh file_contexts_file:file { read getattr open };

allow move-widevine-data-sh media_data_file:file { getattr setattr relabelfrom
rename };
allow move-widevine-data-sh media_data_file:dir  { create reparent rename
rmdir setattr rw_dir_perms relabelfrom };

allow move-widevine-data-sh mediadrm_vendor_data_file:dir { create_dir_perms
relabelto };

# for writing files_moved so we only execute the move once
allow move-widevine-data-sh mediadrm_vendor_data_file:file { create open write
getattr relabelto };
```

Testing Changes

The following tests provide the minimum testing for data file migration. More robust testing is recommended.

1. Verify the build has successfully installed the script

From $ANDROID_BUILD_TOP, look for the script in /system/bin.

`find $OUT -name move_widevine_data.sh`

Expect: file is found in $OUT/system/bin if file migration is implemented.
Expect: no file if file migration is NOT implemented.

2. Verify with adb shell

Using “adb shell” to verify the script is in /system/bin and Widevine data is moved.

Verify move_widevine_data.sh is installed in /system/bin:

```
adb shell
su 0
cat /system/bin/move_widevine_data.sh
```

Expect: file is found if file migration is implemented.
Expect: no file if file migration is NOT implemented.

Verify Widevine drm HAL data is moved from /data/mediadrm to /data/vendor/mediadrm:

```
adb shell
su 0
ls /data/mediadrm
```

Expect: other drm directories, but IDM*/L1 or IDM*/L3 do not exist

```
adb shell
su 0
ls -lR /data/vendor/mediadrm
```

Here, vendor can compare the files created in /data/mediadrm before the upgrade to /data/vendor/mediadrm files after the upgrade. The size should be identical. The group is changed from mediadrm to media.

Expect: /data/vendor/mediadrm folder is created, and IDM*/L1 or IDM*/L3 as well as other data files exist

3. Testing with applications that can save offline movie, e.g. Play Movies & TV and Netflix.

Install O-MR1, purchase a movie and download it to the device for offline playback and verify playback succeeds.

Upgrade to P, verify the offline movie plays back successfully.

4. Checking SELinux policy

Verify there are no “avc: denied” errors related to the change.

```
adb pull /sys/fs/selinux/policy
adb logcat -b all -d | audit2allow -p policy | tee mvdata.txt
```

Expect: no “avc: denied” logging related to move-widevine-data-sh context

```
adb logcat -d > foo.txt
```

Expect: no “avc: denied” logging related to move-widevine-data-sh context

```
adb shell
su 0
dmesg | grep avc
```

Expect: no “avc: denied” logging related to move-widevine-data-sh context