Dynamic analysis system apps on Android Workshop

Last updated Jun 04, 2022

Vitor Ventura @_vventura

cisco | TALOS

Who am I?

CyberSecurity Researcher at Cisco Talos

- Mobile malware lover
- APT hunter
- Reverse engineer

Located in Portugal

Vitor Ventura

@_vventura

Talos
Cisco Security Research

Agenda

Agenda

1. The problem
2. Workshop setup
3. Hands on
4. Final remarks

The problem

The problem

Android Operating system

- Open operating system used by the majority of the phones
- Multiple types of applications
- Multiple permissions levels

Two types of applications

- System
  - Pre-installed with the OS.
  - Don’t need to be signed by Google
- Non-system
- User-level

System applications

- Especifically with the system.img
- Added by vendor, telecoms, Google or others at device creation

System Apps dynamic analysis

- Most don’t have a UI to be launched
- Can share process or UserID
- Can have signature level permissions
- Cannot be deleted by the user

cisco | TALOS

The objective

Perform dynamic analysis

Dynamic analysis of system applications

Instrument the target application

To perform dynamic analysis we will use Frida to instrument the target application.

We can do both dynamic analysis for reverse engineering

fuzzing of the application inputs for offensive researach

Use stock images

Keep the images as pristine as possible

Keep vendor kernel tweaks

Workshop setup

What you need

- Computer that can run a Virtual Machine with 4GB of RAM (8 would be better)
- Apple silicon might be a problem, can’t help with it
- Some knowledge of Android applications.
- Will to experiment
- The password for the VM is : password

Setup

Overall

V	Virtual Machine	This is a Linux XFCE Ubuntu machine, preloaded with everything you need. Android emulator, Frida, scripts for repacking, jadx, etc. PASSWORD IS: password
E	Emulator	Android emulator is already loaded and patched with Magisk. The image is an Android S API level 30. On x86 architecture, to make it easier and faster for everyone.
T	Target Application	In the workshop we will target the MtpService application. Which is responsible to handle USB device connection events on the Android Operating System.
S	Scripts	Scripts have been created to make it easier to restart the emulator, repack the application, etc.

Setup

Directories

T	~/tools	This contains the software you will use. You shouldn't need to get here has all the tools are in the $PATH
B	~/tools/bin	If you are curious about any of the scripts you can find them here including the environment settings.
W	~/workdir	This is where all the workshop should be done
L	~/workdir/lib_gadget	This is where the Frida gadget is stored so that you can inject it into your target application

Setup

Scripts/Alias

R	rebuild_package	Takes directory as argument repackages the application and signs it. Cert pwd is 000000
B	unpack_package	Takes an apk as argument. Unpacks the apk using apktool. (Does not unpack resources)
W	list-avds	List all emulator virtual devices available. In this case there only one Pixel_4_XL_API_30
L	start-avds	Starts a virtual device. Takes one argument which is the avd name as displayed by the list-avds command.
J	jadx	This is the decompiler we will use to analyze the apk. Current version 1.4.0.

Setup

Other useful commands

#1	adb pull	Allows the download of a file from the emulator into the local system. Ex. adb pull /system/priv-app/MtpService/MtpService.apk
#2	adb push	Allows the upload of a file from the local system to the emulator. Ex. adb push MtpService.apk /system/priv-app/MtpService/
#3	Download all apks from a path	for F in $(adb shell /bin/ls <THE PATH>); do adb pull <THE PATH>/$F/$F.apk; done
#4	Unpack multiple apk	for F in $(/bin/ls); do apktool d $F; done (in the same location where the apk's are)

Setup

Other useful commands

#5	Recursive grep with file name	find . -maxdepth 2 -name AndroidManifest.xml -exec grep -H 'sharedUserId="android.media"' {} \; | cut -d: -f1
#6	Uninstall a package	adb uninstall <package name> Ex: adb uninstall com.android.MtpService

cisco | Talos

Hands on

Steps

1. Use adb shell and find MtpService.apk
2.
3.
4.
5.
6.

cisco | Talos

Use adb pull to get the file out

```
generic_x86:/system/priv-app # ls -la MtpService/
total 1560
drwxr-xr-x  3 root root    4096 2021-03-08 22:14 .
drwxr-xr-x 39 root root    4096 2021-03-08 22:14 ..
-rw-r--r--  1 root root 1585098 2021-03-08 22:11 MtpService.apk
drwxr-xr-x  3 root root    4096 2021-03-08 22:14 oat
```

Steps

1. Use adb shell and find MtpService.apk
2. Unpack it and lets look around an choose.
3.
4.
5.
6.

cisco | Talos

How can we start the app?

- What is the way to start this application from the UI?
  - Short answer you can’t
- So how can we start it with Frida?
  - Short answer we can’t
- What is the alternative?
  - Load a gadget upon an event.
  - Lets do it right at the device boot!

cisco | Talos

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionCodename="11" android:sharedUserId="and
roid.media" package="com.android.mtp" platformBuildVersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label" android:process="android.process.medi
a" android:usesCleartextTraffic="true" android:usesNonSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true" android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" android:permission="android.permission.MA
NAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
            </intent-filter>
        </provider>
        <service android:name="com.android.mtp.MtpDocumentsService"/>
        <activity android:excludeFromRecents="true" android:icon="@mipmap/ic_launcher_download" android:label="@string/downloads_app_label" android:name="com.android.mtp.ReceiverActivity" android:theme="@android:style/
Theme.NoDisplay">
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"/>
            </intent-filter>
            <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" android:resource="@xml/device_filter"/>
        </activity>
        <receiver android:exported="true" android:name="com.android.mtp.UsbIntentReceiver">
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"/>
                <action android:name="android.hardware.usb.action.USB_DEVICE_DETACHED"/>
            </intent-filter>
            <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" android:resource="@xml/device_filter"/>
        </receiver>
        <receiver android:name="com.android.mtp.MtpReceiver">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
            </intent-filter>
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_STATE"/>
            </intent-filter>
        </receiver>
        <service android:name="com.android.mtp.MtpService"/>
    </application>
</manifest>
```

Steps

1. Use adb shell and find MtpService.apk
2. Unpack it and lets look around an choose the right place
3. Add the gadget and repack it
4.
5.
6.

cisco | Talos

How to add the Frida Gadget?

Add the Library load instructions on :

MtpReceiver.smali

In the :

onReceive()

method

```
# virtual methods
.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V
    .locals 3

    .line 35
    invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

    move-result-object v0

    const-string v1, "android.intent.action.BOOT_COMPLETED"

    .line 36
    invoke-virtual {v1, v0}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

    move-result v1

    const-string v2, "android.hardware.usb.action.USB_STATE"

    if-eqz v1, :cond_0

    const/4 p2, 0x0

    .line 37
    new-instance v0, Landroid/content/IntentFilter;

    invoke-direct {v0, v2}, Landroid/content/IntentFilter;-><init>(Ljava/lang/String;)V
```

How to add the Frida Gadget?

Add the Library load instructions on :

MtpReceiver.smali

In the :

onReceive()

method

```
# virtual methods
.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V
    .locals 3

    const-string v0, "gadget"
    invoke-static {v0}, Ljava/lang/System;->loadLibrary(Ljava/lang/String;)V

    const-string v0, "VV-Talos"
    const-string v1, "Mtp: On receive"
    invoke-static {v0, v1}, Landroid/util/Log;->v(Ljava/lang/String;Ljava/lang/String;)I

    .line 35
    invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

    move-result-object v0

    const-string v1, "android.intent.action.BOOT_COMPLETED"

    .line 36
    invoke-virtual {v1, v0}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

    move-result v1

    const-string v2, "android.hardware.usb.action.USB_STATE"
```

How to add the Frida Gadget?

This will load the frida gadget

Add the Library load instructions on :

MtpReceiver.smali

In the :

onReceive()

method

```
# virtual methods
.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V
    .locals 3

    const-string v0, "gadget"
    invoke-static {v0}, Ljava/lang/System;->loadLibrary(Ljava/lang/String;)V

    const-string v0, "VV-Talos"
    const-string v1, "Mtp: On receive"
    invoke-static {v0, v1}, Landroid/util/Log;->v(Ljava/lang/String;Ljava/lang/String;)I

    .line 35
    invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

    move-result-object v0

    const-string v1, "android.intent.action.BOOT_COMPLETED"

    .line 36
    invoke-virtual {v1, v0}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

    move-result v1

    const-string v2, "android.hardware.usb.action.USB_STATE"
```

How to add the Frida Gadget?

This will load the frida gadget

Add the Library load instructions on :

MtpReceiver.smali

In the :

onReceive()

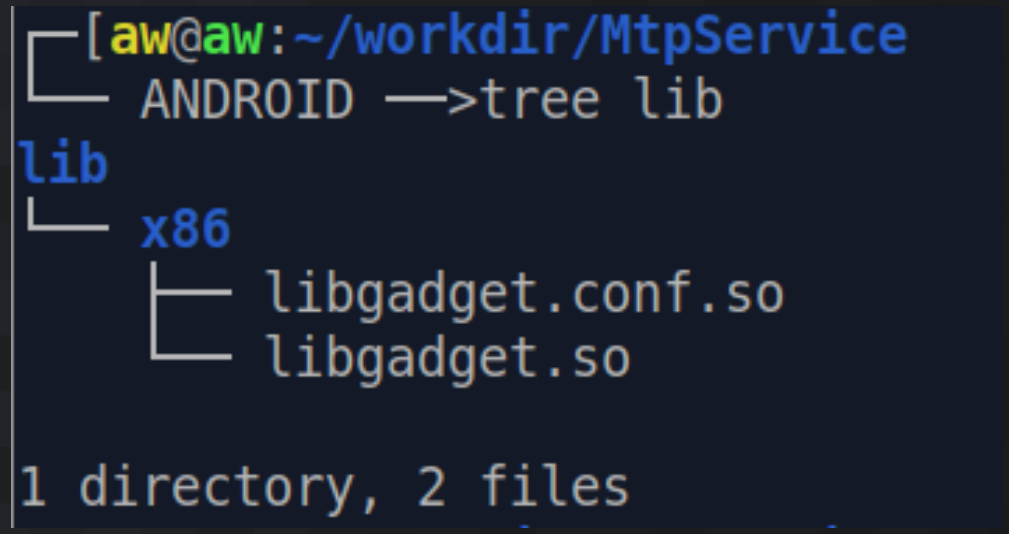
method

Good old println debug

```
# virtual methods
.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V
    .locals 3

    const-string v0, "gadget"
    invoke-static {v0}, Ljava/lang/System;->loadLibrary(Ljava/lang/String;)V

    const-string v0, "VV-Talos"
    const-string v1, "Mtp: On receive"
    invoke-static {v0, v1}, Landroid/util/Log;->v(Ljava/lang/String;Ljava/lang/String;)I

    .line 35
    invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

    move-result-object v0

    const-string v1, "android.intent.action.BOOT_COMPLETED"

    .line 36
    invoke-virtual {v1, v0}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

    move-result v1

    const-string v2, "android.hardware.usb.action.USB_STATE"
```

Create the patched package

1. In the pkg directory create the directory lib/x86
2. Copy the contents of $WORKDIR/lib_gadget into it.
   - libgadget.so is the gadget
   - libgadget.conf.so is the gadget configuration
3. Outside the package directory run rebuild_package.sh <dir_name>



Gadget configuration

- Frida gadget will execute a java script
- Push the file script.js to /data/local/tmp
- For the purposes of the workshop the script will simply log a message to logcat

```
┌─[aw@aw:~/workdir/MtpService/lib/x86
└── ANDROID ──>cat libgadget.conf.so
{
    interaction": {
      "type": "script",
      "path": "/data/local/tmp/script.js",
      "on_change": "reload"
    }
}
```

```
┌─[aw@aw:~/workdir
└── ANDROID ──>cat script.js
'use strict';

console.log("Waiting for Java..")

Java.perform(function () {
  var Log = Java.use("android.util.Log");
  Log.v("VV-Talos", "Inspect away :D ");
});
```

Install the package

- Finally we install the patched package

adb install –r MtpService_patched.apk

- What do you see?

Install the package

```
└─ ANDROID ──>adb install -r MtpService_patched.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package com.android.mtp signatures do not match previously installed vers
ion; ignoring!]
Performing Streamed Install
adb: failed to install MtpService_patched.apk: Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package com.android.mtp si
gnatures do not match previously installed version; ignoring!]
```

- Finally we install the patched package

Install the package

```
└─ ANDROID ──>adb install -r MtpService_patched.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package com.android.mtp signatures do not match previously installed vers
ion; ignoring!]
Performing Streamed Install
adb: failed to install MtpService_patched.apk: Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package com.android.mtp si
gnatures do not match previously installed version; ignoring!]
```

- Finally we install the patched package
- Signature on the existent package is different from the patched package

Steps

1. Use adb shell and find MtpService.apk
2. Unpack it and lets look around an choose the right place
3. Add the gadget and repack it
4. Doesn’t install because of the signatures don’t match
5.
6.

Lets remove the original package

- The issue is that we have a pre-installed package with a different signature
- Try to uninstll the package

Try using command
adb uninstall com.android.MtpService
adb root
adb uninstall com.android.MtpService

Lets remove the original package

- The issue is that we have a pre-installed package with a different signature

- Try to uninstall the package
  - Doesn't work because it's a system package

```
┌─[aw@aw:~/workdir
└── ANDROID ──>adb uninstall com.android.MtpService
Failure [DELETE_FAILED_INTERNAL_ERROR]
┌─[aw@aw:~/workdir
└── ANDROID ──>adb root
┌─[aw@aw:~/workdir
└── ANDROID ──>adb uninstall com.android.MtpService
Failure [DELETE_FAILED_INTERNAL_ERROR]
```

Magisk to the rescue

- With Magisk we can create a module to hide and/or replace a previous package.
- Basically you can patch the file system with any content upon boot.
- Including simply making empty dirs

```
generic_x86:/system/priv-app # ls -la MtpService/
total 1560
drwxr-xr-x  3 root root    4096 2021-03-08 22:14 .
drwxr-xr-x 39 root root    4096 2021-03-08 22:14 ..
-rw-r--r--  1 root root 1585098 2021-03-08 22:11 MtpService.apk
drwxr-xr-x  3 root root    4096 2021-03-08 22:14 oat
```

Magisk to the rescue

- Magisk modules root is at:
  - /data/adb/modules
- Create a directory with name of your module
- Replicate the FS structure to the directory you want to replace
- For an empty create a file called .replace inside the path

```
generic_x86:/data/adb/modules #
    .
    |-vv-talos
    |---system
    |-----priv-app
    |-------MtpService
```

Magisk to the rescue

- To make it easier you have a prebuilt module
- Push vv-talos_1.tar file in the workdir to /data/adb/modules
- Unpack it and everything is ready
- reboot

```
generic_x86:/data/adb/modules #
    .
    |-vv-talos
    |---system
    |-----priv-app
    |-------MtpService
```

Magisk to the rescue

```
generic_x86:/ $ ls /system/priv-app/MtpService/
MtpService.apk  oat
```

```
generic_x86:/ # ls -la /system/priv-app/MtpService/
total 16
drwxr-xr-x  2 root   root   4096 2022-05-27 09:32 .
drwxr-xr-x 39 root   root   4096 2021-03-08 22:14 ..
-rw-rw-r--  1 system system    0 2022-05-27 09:32 .replace
```

Install the package

- Finally we install the patched package

adb install –r MtpService_patched.apk

- What do you see?

Install the package

```
┌─[aw@aw:~/workdir
└── ANDROID ──>adb install -r MtpService_patched.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Failure [INSTALL_FAILED_SHARED_USER_INCOMPATIBLE: Reconciliation failed...: Reconcile failed: Pac
kage com.android.mtp has no signatures that match those in shared user android.media; ignoring!]
Performing Streamed Install
adb: failed to install MtpService_patched.apk: Failure [INSTALL_FAILED_SHARED_USER_INCOMPATIBLE:
Reconciliation failed...: Reconcile failed: Package com.android.mtp has no signatures that match
those in shared user android.media; ignoring!]
```

- Finally we install the patched package
- Similar but different error

Install the package

```
┌─[aw@aw:~/workdir
└── ANDROID ──>adb install -r MtpService_patched.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Failure [INSTALL_FAILED_SHARED_USER_INCOMPATIBLE: Reconciliation failed...: Reconcile failed: Pac
kage com.android.mtp has no signatures that match those in shared user android.media; ignoring!]
Performing Streamed Install
adb: failed to install MtpService_patched.apk: Failure [INSTALL_FAILED_SHARED_USER_INCOMPATIBLE:
Reconciliation failed...: Reconcile failed: Package com.android.mtp has no signatures that match
those in shared user android.media; ignoring!]
```

- Finally we install the patched package
- Similar but different error
- The signature of the patched package is different from the one on packages sharing the UID

sharedUserId

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://sche
mas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionC
odename="11" android:sharedUserId="android.media" package="com.android.mtp" platformBuildV
ersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.ap
p.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label"
 android:process="android.process.media" android:usesCleartextTraffic="true" android:usesN
onSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true"
android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" and
roid:permission="android.permission.MANAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
            </intent-filter>
                                                                    1,1           Top
```

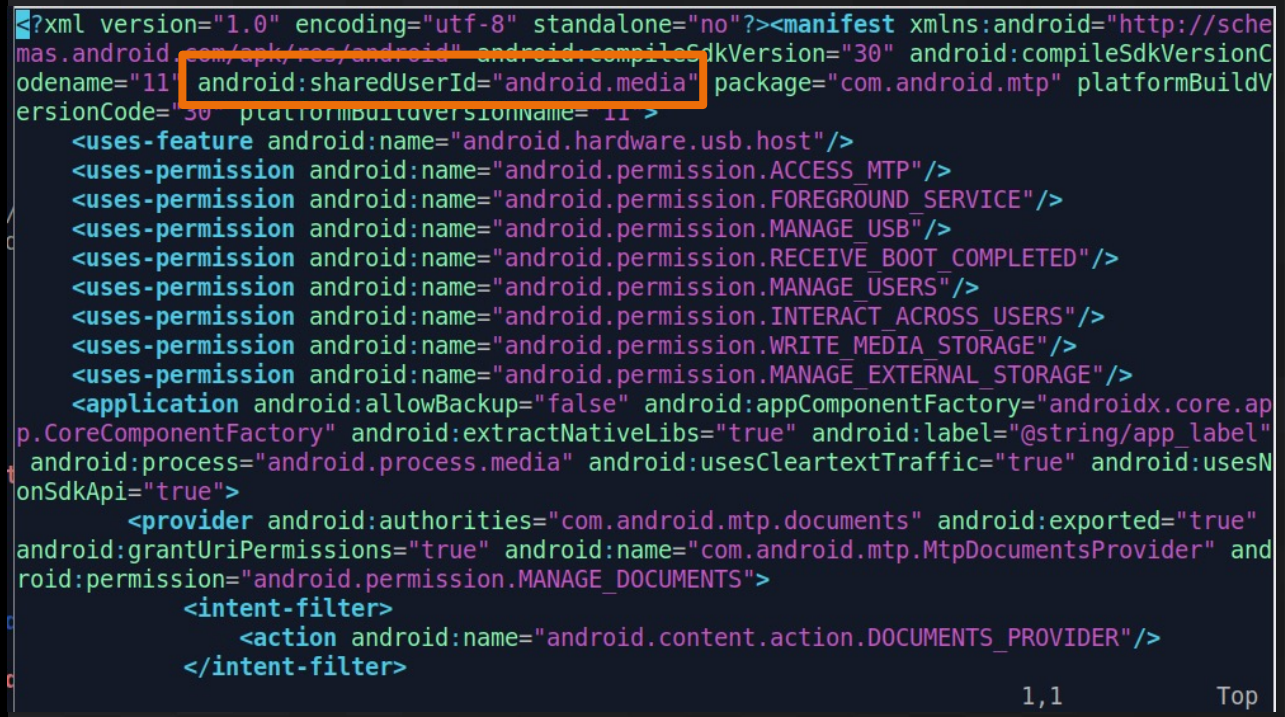
Its declared in the manifest.

Steps

1. Use adb shell and find MtpService.apk
2. Unpack it and lets look around an choose the right place
3. Add the gadget and repack it
4. Doesn’t install because of the signatures don’t match
5. Doesn’t install because there is shared user ID
6.

How to solve the sharedUserId issue?

1. We need to find out which packages use the same UserId
2. Either remove them or repackage them with your own certficate
3. Reinstall them



Find out the packages

It must be a system application

1. Fetch all the system applications
2. Unpack all
1. Check the manifest to see which ones declare the same sharedUserId

#3	Download all apks from a path	for F in $(adb shell /bin/ls <THE PATH>); do adb pull <THE PATH>/$F/$F.apk; done
#4	Unpack multiple apk	for F in $(/bin/ls); do apktool d $F; done (in the same location where the apk's are)
#5	Recursive grep with file name	find . -maxdepth 2 -name AndroidManifest.xml -exec grep -H 'sharedUserId="android.media"' {} \; | cut -d: -f1

cisco | Talos

Find out the packages

It must be a system application

1. Fetch all the system applications

2. Unpack all

1. Check the manifest to see which ones declare the same sharedUserId

```
┌─[aw@aw:~/workdir/t
└── ANDROID ──>find . -maxdepth 2 -name AndroidManifest.xml -exec
 grep -H 'sharedUserId="android.media"' {} \; | cut -d: -f1
./DownloadProviderUi/AndroidManifest.xml
./SoundPicker/AndroidManifest.xml
./MediaProviderLegacy/AndroidManifest.xml
./DownloadProvider/AndroidManifest.xml
```

Again Magisk to the rescue

- As before the modules are already done
- Push vv-talos_2.tar file in the workdir to /data/adb/modules
- Unpack it, copy the contents of system to vv-talos/system and everything is ready
- reboot

```
.
|-system
|---priv-app
|-----DownloadProvider
|-----DownloadProviderUi
|-----MediaProviderLegacy
|-----MtpService
|-----SoundPicker
```

Again Magisk to the rescue

- As before the modules are already done
- Push vv-talos_2.tar file in the workdir to /data/adb/modules
- Unpack it, copy the contents of system to vv-talos/system and everything is ready
- reboot

```
.
|-system
|---priv-app
|-----DownloadProvider
|-----DownloadProviderUi
|-----MediaProviderLegacy
|-----MtpService
|-----SoundPicker
```

Install the package

- Finally we install the patched package no errors
- We added our code to the receiver of the BOOT_COMPLETED action
- Our debug messages are on logcat

```
┌─[aw@aw:~/workdir
└── ANDROID ──>adb install MtpService_patched.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Success
Install command complete in 561 ms
```

Install the package

- Finally we install the patched package no errors
- We added our code to the receiver of the BOOT_COMPLETED action
- Our debug messages are on logcat
- Lets reboot and check logcat

```
┌─[aw@aw:/tmp
└── ANDROID ──>adb logcat | egrep -i '(vv-|frida)'
05-27 12:06:08.052   258   258 I Magisk  : vv-talos: exec [post-fs-data.sh]
05-27 12:06:08.058   251   253 I Magisk  : vv-talos: loading mount files
05-27 12:06:57.238  1865  1896 F Frida   : Failed to start: Could not listen on address 127.0.0.1,
 port 27042: Unable to create socket: Operation not permitted
05-27 12:06:57.238  1865  1865 V VV-Talos: Mtp: On receive
```

cisco | Talos

Install the package

- Still have some problem we cant create the socket for Frida to contact.

```
┌─[aw@aw:/tmp
└── ANDROID ──>adb logcat | egrep -i '(vv-|frida)'
05-27 12:06:08.052   258   258 I Magisk  : vv-talos: exec [post-fs-data.sh]
05-27 12:06:08.058   251   253 I Magisk  : vv-talos: loading mount files
05-27 12:06:57.238  1865  1896 F Frida   : Failed to start: Could not listen on address 127.0.0.1,
 port 27042: Unable to create socket: Operation not permitted
05-27 12:06:57.238  1865  1865 V VV-Talos: Mtp: On receive
```

Steps

1. Use adb shell and find MtpService.apk
2. Unpack it and lets look around an choose the right place
3. Add the gadget and repack it
4. Doesn’t install because of the signatures don’t match
5. Doesn’t install because there is shared user ID
6. No permissions

cisco | Talos

Permissions

- Keep in mind that the gadget is runing under the application context.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://sche
mas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionC
odename="11" android:sharedUserId="android.media" package="com.android.mtp" platformBuildV
ersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.ap
p.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label"
 android:process="android.process.media" android:usesCleartextTraffic="true" android:usesN
onSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true"
android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" and
roid:permission="android.permission.MANAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
            </intent-filter>
                                                                          1,1           Top
```

cisco | Talos

Permissions

- Keep in mind that the gadget is runing under the application context.
- In order for an application to open a port it needs the permission
  - android.permission.INTERNET

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://sche
mas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionC
odename="11" android:sharedUserId="android.media" package="com.android.mtp" platformBuildV
ersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.ap
p.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label"
 android:process="android.process.media" android:usesCleartextTraffic="true" android:usesN
onSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true"
android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" and
roid:permission="android.permission.MANAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
            </intent-filter>
                                                                          1,1           Top
```

cisco | Talos

Permissions

- In order for an application to open a port it needs the permission
- In order for an application to open a port it needs the permission
  - android.permission.INTERNET
- I need to change the AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionCodename="11" android:sharedUserId="android.media" package="com.android.mtp" platformBuildVersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label"
     android:process="android.process.media" android:usesCleartextTraffic="true" android:usesNonSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true"
android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" android:permission="android.permission.MANAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
```

cisco | Talos

Permissions

- Rebuild the package, reinstall the package.
- Reboot
- Check the logs:
  - adb logcat | egrep –i ‘(vv-|Frida)’

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionCodename="11" android:sharedUserId="android.media" package="com.android.mtp" platformBuildVersionCode="30" platformBuildVersionName="11">
    <uses-feature android:name="android.hardware.usb.host"/>
    <uses-permission android:name="android.permission.ACCESS_MTP"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.MANAGE_USB"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.MANAGE_USERS"/>
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS"/>
    <uses-permission android:name="android.permission.WRITE_MEDIA_STORAGE"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:extractNativeLibs="true" android:label="@string/app_label"
 android:process="android.process.media" android:usesCleartextTraffic="true" android:usesNonSdkApi="true">
        <provider android:authorities="com.android.mtp.documents" android:exported="true"
android:grantUriPermissions="true" android:name="com.android.mtp.MtpDocumentsProvider" android:permission="android.permission.MANAGE_DOCUMENTS">
            <intent-filter>
                <action android:name="android.content.action.DOCUMENTS_PROVIDER"/>
```

SUCCESS ?

```
┌─[aw@aw:~/workdir
└── ANDROID ──>cat script.js
'use strict';

console.log("Waiting for Java..")

Java.perform(function () {
  var Log = Java.use("android.util.Log");
  Log.v("VV-Talos", "Inspect away :D ");
});
```

SUCCESS !

```
┌─[aw@aw:~/workdir
└── ANDROID ──>cat script.js
'use strict';

console.log("Waiting for Java..")

Java.perform(function () {
  var Log = Java.use("android.util.Log");
  Log.v("VV-Talos", "Inspect away :D ");
});
```

```
┌─[aw@aw:~/workdir
└── ANDROID ──>adb logcat | egrep -i '(vv-|frida)'
- waiting for device -
05-27 15:18:35.972   258   258 I Magisk  : vv-talos: exec [post-fs-data.sh]
05-27 15:18:35.981   251   253 I Magisk  : vv-talos: loading mount files
05-27 15:19:25.134  1980  1980 V VV-Talos: Inspect away :D
05-27 15:19:25.216  1980  1980 V VV-Talos: Mtp: On receive
```

cisco | Talos

Wrap up

Wrap - up

- Having a system rooted is not the same has having a custom image
- On a rooted system you can have adb running has root
- You can’t bypass kernel level definitions
- Using Magisk you have a door into the kernel and the system

Limitations

- Choose carefully the gadget loading location
- Some system apps will simply schedule tasks.
  - These tasks run on different processes address spaces
  - You need to load the gadget inside the task that gets scheduled
- Keep in mind that after loading the gadget the app will continue to run, and may exit
- Avoid interactive analysis with Frida. Use scripts as much as possible.
- Keep in mind that you can patch what ever you want on the target app to make your analysis easier.

References

- A good tutorial on Frida Gadget use:
  - https://lief-project.github.io/doc/stable/tutorials/09_frida_lief.html
- Dalvik opcodes:
  - https://source.android.com/devices/tech/dalvik/dalvik-bytecode
- Smali code examples:
  - https://github.com/JesusFreke/smali/tree/master/examples
- Magisk Modules documentation
  - https://topjohnwu.github.io/Magisk/guides.html

Thank you!

TALOSINTELLIGENCE.COM

blog.talosintelligence.com

@talossecurity


CISCO
TALOS
TALOSINTELLIGENCE.COM