

Laboratorio de Sistemas Operativos y Redes: GuardianProject Haven



Integrantes:

Romero, Lisa Malena.

Mur Caramés, Lucas Christopher.

Índice.

Introduccion

Maquinas Utilizadas

Requerimientos

Instalacion

Problemas Surgidos

Introducción.

El proyecto elegido fue Haven, una aplicación que convierte un dispositivo android en un monitor de espacio físico, utilizando los sensores que posee el dispositivo, es decir, una app que te notifica de todo lo que pasa en tu móvil cuando no lo tienes encima.

En colaboración con la Freedom of the Press Foundation, de la que Snowden es miembro, han lanzado esta aplicación especialmente diseñada para mejorar la seguridad del móvil y de lo que pasa alrededor.

Instala la aplicación en un móvil barato y déjalo en la habitación de un hotel para que sus sensores capturen sonidos, vídeos o fotos de todo lo que pasa a su alrededor. Usa sus sensores para identificar cuando alguien está cerca, por lo que sirve como sistema de vigilancia.

Haven usa el acelerómetro del móvil para detectar movimientos o vibraciones. La cámara para capturar imágenes por delante y por detrás. El micrófono lo usa para detectar ruidos en el ambiente. Sensor de luminosidad para detectar cambios en la luz del lugar. El control de energía para alertar si el móvil se desconecta o se apaga.

Maquinas Utilizadas.

Características de las máquinas utilizadas para compilar el código.

Notebook 1:

- CPU: Intel Core i5-7200U @ 2.5GHz – 64 bits
- GPU: Intel HD Graphics 620
- RAM: 8 GB
- OS: Fedora workstation 28 - 64 bits

Notebook 2:

- CPU: AMD E-450 @ 1.6GHz - 64 bits
- GPU: Radeon HD Graphics 6320
- RAM: 2 GB
- OS: Ubuntu 18.04 Bionic Beaver – 64bits

Celular 1 - Moto G 3rd Gen:

- CPU: Quad-core 1.4GHz Cortex-A53
- GPU: Adreno 306
- RAM: 1 GB
- STORAGE: 8GB + 32GB MicroSD
- MAIN CAMERA: 13MP, video 1080p@30fps, HDR
- FRONTAL CAMERA: 5MP
- SENSORS: Acelerómetro, proximidad
- BATTERY: Li-Ion 2470 mAh
- OS: Android 6.0 Marshmallow

Celular 2 - Samsung S6 Edge:

- CPU: Octa-core 4X2.1 GHz Cortex-A57 & 4X1.5 GHz Cortex-A53
- GPU: Mali-T760MP8
- RAM: 3 GB
- STORAGE: 32GB
- MAIN CAMERA: 16MP, video 2160p@30fps, 1080p@60fps, 720p@120fps, HDR, dual video record
- FRONTAL CAMERA: 5MP, video 1440p@30fps
- SENSORS: Lector de huella (front-mounted), acelerómetro, giroscopio, proximidad, barómetro, pulso cardíaco, SpO2, brújula.
- BATTERY: 2600 mAh
- OS: Android 7.0 Nougat

Requerimientos.

El requerimiento general para instalar Haven es tener algún dispositivo con Android, ya sea físico o un emulador. Después, dependiendo de la forma de instalación, los requerimientos pueden variar.

En Google Play Store

Ningún requerimiento.

En F-Droid

Permitir en Ajustes->Seguridad la instalación de aplicaciones de orígenes que no sean Play Store.

En Github con los archivos APK y con el código fuente

El requerimiento principal para hacer estas instalaciones es tener un sistema operativo de base Linux.

Además, hay que habilitar en el dispositivo conectado la opción de USB Debugging para cuando se intente instalar el APK. Ir a Settings -> Developer Options-> enable USB Debugging. En el caso de tener un dispositivo con Android 4.2+, que tiene oculta la opción Developer Options, hay que habilitarla yendo a Settings -> About <device> y tocar el build number 7 veces.

Finalmente se debe instalar Android Debug Bridge (ADB), una herramienta de líneas de comandos versátil que te permite comunicarte con una instancia de un emulador o un dispositivo Android conectado. Se puede hacer corriendo este comando:

```
sudo apt-get install adb
```

En Github con el código fuente

El primer requisito para instalar Haven es tener alguna versión de JDK & JRE. Además para poder firmar la APK, es necesaria la instalación de jarsigner.

Para poder instalar las herramientas necesarias para manejarse con Android se recomienda instalar Android Studio, que incluye todas las herramientas necesarias para instalar Haven. Sus requerimientos son:

- Tener un sistema operativo de 64 bits.
- 3 GB de memoria RAM como mínimo (se recomiendan 8 GB), más 1 GB para el emulador de Android
- 2 GB de espacio en disco disponible como mínimo (se recomiendan 4 GB); 500 MB para el IDE + 1.5 GB para Android SDK y la imagen de sistema del emulador
- Resolución de pantalla mínima de 1280 x 800

Si no, una alternativa para no instalar Android Studio, es solamente instalarse Gradle (para hacer build de la aplicación) y android-tools-adb. Esto se puede hacer con los siguientes comandos:

```
sudo apt-get install android-tools-adb gradle
```

Instalación.

En Google Play Store

Desde el dispositivo elegido con android instalar la aplicación haciendo [click aquí](#).

En F-Droid

F-Droid es un repositorio de software para las aplicaciones de Android, funciona de manera similar a la tienda de Google Play, pero sólo contiene software libre y de código abierto. Se la puede descargar haciendo [click aquí](#).

Para instalarla se debe agregar el repositorio donde está Haven a F-Droid. O se hace manualmente en Settings->Repositorios agregando [“Haven Nightly Bleeding Edge”](#), o escaneado el código QR:



Una vez agregado, buscar en Categorías el repositorio “nightly” y descargar su única aplicación.

Desde Github con los archivos APK

Desde el repositorio de Haven en Github, yendo a la parte de releases, se puede descargar el APK ya firmado, o si no haciendo [click aquí](#). Ya descargado se conecta el dispositivo a la computadora y se lo instala usando este comando:

```
adb install Haven-0.2.0-alpha-5-release.apk
```

Desde Github con el código fuente

Se puede descargar el código fuente de Haven haciendo click aquí y después descomprimiendo el zip. O si no clonando el repositorio desde git:

```
git clone https://github.com/guardianproject/haven.git
```

Una vez descargado el repositorio, abrir Android Studio e importa Haven como proyecto Gradle. Esperar a que importe el proyecto y las dependencias.

Desde una terminal, ubicarse en la carpeta de haven y buildear el proyecto con Gradle:

```
./gradlew build
```

Esto genera en ~/build/output/apk dos archivos APK, el de debug, que sirve para testeado y el de release, que es el listo para lanzar a producción.

El siguiente paso es instalar el APK de release en el dispositivo. Pero antes se la debe firmar. Este proceso no es más que dar una clave privada generada por el desarrollador y con esto se logra indicar quien es el responsable de la aplicación. No es un requisito para instalarlo a un dispositivo, pero si, si se lo desea subir a Google Play.

Entonces el primer paso es crear una keystore en ~/build/output/apk/release , la cuál va a ser nuestro “signer tool”.

```
sudo keytool -keygen -v -keystore name.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias anAlias
```

****IMPORTANTE NO PERDER ACCESO A LA KEYSTORE EN CASO DE TENER UNA APK PUBLICADA, YA QUE NO VAMOS A PODER ACTUALIZAR LA VERSIÓN QUE ESTÉ EN LA STORE!****

En este ejemplo se te solicitan contraseñas para el keystore y la clave, y para otorgar los campos de nombre distinguido de tu clave. Luego, se genera el keystore como un archivo llamado name.keystore, que se guarda en el directorio actual (puedes moverlo cuando lo desees). El keystore contiene una clave única válida por 10 000 días.

```
[lucas@lucas-fedorawork bin]$ sudo keytool -genkey -v -keystore labohaven.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias havenlaboapp
[sudo] password for lucas:
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]: Labo Haven Lucas Lisa
What is the name of your organizational unit?
  [Unknown]: UNQ LABO
What is the name of your organization?
  [Unknown]: UNQ
What is the name of your City or Locality?
  [Unknown]: Quilmes
What is the name of your State or Province?
  [Unknown]: Bs As
What is the two-letter country code for this unit?
  [Unknown]: AR
Is CN=Labo Haven Lucas Lisa, OU=UNQ LABO, O=UNQ, L=Quilmes, ST=Bs As, C=AR correct?
  [No]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
    for: CN=Labo Haven Lucas Lisa, OU=UNQ LABO, O=UNQ, L=Quilmes, ST=Bs As, C=AR
Enter key password for <havenlaboapp>
    (RETURN if same as keystore password):
[Storing labohaven.keystore]
```

El siguiente paso es firmar el APK con el siguiente comando:

```
sudo jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore name.keystore apk_route_and_name.apk anAlias
```

Donde va a figurar los siguientes outputs:

```
[lucas@lucas-fedorawork bin]$ sudo jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore Labohaven.keystore ~/StudioProjects/haven/build/ou
tputs/apk/release/Haven-labo-lisa-lucas.apk havenlaboapp
Enter Passphrase for keystore:
updating: META-INF/MANIFEST.MF
  adding: META-INF/HAVENLAB.SF
  adding: META-INF/HAVENLAB.RSA
signing: AndroidManifest.xml
signing: androidsupportmultidexversion.txt
signing: assets/fonts/Roboto-Medium.ttf
signing: assets/sugar_upgrades/2.sql
signing: be/tarsos/dsp/AudioDispatcher.java
```

Esperamos a que termine...

```
signing: res/mipmap-xxxhdpi-v4/ic_launcher.png
signing: res/raw/channel.bc
signing: res/raw/uvencode.bc
signing: res/xml/file_provider_paths.xml
signing: res/xml/settings.xml
signing: resources.arsc
jar signed.
```

Para verificar que se haya creado la firma correctamente, corremos el siguiente comando:

```
sudo jarsigner -verify -verbose apk_route_and_name.apk
```

Esperamos a que termine...

```
sm      4264 Fri Nov 30 00:00:00 ART 1979 res/xml/settings.xml
sm      2144576 Fri Nov 30 00:00:00 ART 1979 resources.arsc

s = signature was verified
m = entry is listed in manifest
k = at least one certificate was found in keystore
i = at least one certificate was found in identity scope

- Signed by "CN=Labo Haven Lucas Lisa, OU=UNQ LABO, O=UNQ, L=Quilmes, ST=Bs As, C=AR"
  Digest algorithm: SHA1
  Signature algorithm: SHA1withRSA, 2048-bit key

jar verified.
```

```
[lucas@lucas-fedorawork bin]$ sudo jarsigner -verify -verbose ~/StudioProjects/haven/build/outputs/apk/release/Haven-labo-lisa-lucas.apk
s      194125 Thu Nov 29 20:42:12 ART 2018 META-INF/MANIFEST.MF
s      194220 Thu Nov 29 20:42:12 ART 2018 META-INF/HAVENLAB.SF
s      1374 Thu Nov 29 20:42:12 ART 2018 META-INF/HAVENLAB.RSA
sm     12 Fri Nov 30 00:00:00 ART 1979 META-INF/android.support.design_material.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.appcompat.appcompat.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.arch.core_core_runtime.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.asynclayoutinflater.asynclayoutinflater.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.cardview.cardview.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.coordinatorlayout_coordinatorlayout.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.core_core.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.cursoradapter.cursoradapter.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.customview_customview.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.documentfile.documentfile.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.drawerlayout.drawerlayout.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.fragment_fragment.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.interpolator.interpolator.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.legacy_legacy-support-core-ui.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.legacy_legacy-support-core-utils.version
sm     11 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.legacy_legacy-support-v13.version
sm     11 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.legacy_legacy-support-v4.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.lifecycle_lifecycle-livedata-core.version
sm     6 Fri Nov 30 00:00:00 ART 1979 META-INF/androidx.lifecycle_lifecycle-livedata.version
```

Ya verificado la firma digital del APK, el siguiente paso es instalarlo en el dispositivo. Para ello se lo conecta a la computadora y se ingresa el siguiente comando:

```
adb -d install apk_route_and_name.apk
```

Este comando reconoce el dispositivo conectado y le instala el APK.


```
lucas@lucas-fedorawork release]$ adb -d install Haven-labo-lisa-lucas.apk
Haven-labo-lisa-lucas.apk: 1 file pushed. 3.4 MB/s (14522239 bytes in 4.110s)
  pkg: /data/local/tmp/Haven-labo-lisa-lucas.apk
Success
```

Problemas Surgidos.

APK Sin firma digital en la Notebook 1:

```
Haven-0.2.0-alpha-5-release-unsigned.a...d. 3.3 MB/s (14324078 bytes in 4.173s)
  pkg: /data/local/tmp/Haven-0.2.0-alpha-5-release-unsigned.apk
Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES]
```

Solucionado firmando digitalmente el apk.

Notebook 2:

Hubo dos problemas al hacer el build con Gradle.

El primero fue que al tener con otros nombres las variables de ambiente de Java, Gradle no podia encontrarla y no corria. Se soluciono cambiando los nombres de las carpetas de jdk-11.0.1 a java-11 en /etc/profile.d/jdk.sh.

El segundo fue que al ser muchas dependencias a instalarse, se colgó la computadora y dio build fallido. En el segundo intento, Se soluciono solo.

No poder probar la instalación con el emulador con Android Studio porque se colgaba la computadora. El problema surgía por no tener suficiente RAM para el IDE.

Cuando se intentó firmar la APK con jarsigner, tiro error porque no estaba el archivo de las keys en la misma carpeta del APK sin firmar.

Referencias.

- <https://github.com/guardianproject/haven>
- <https://clipset.20minutos.es/hacen-app-android-edward-snowden/>
- <https://developer.android.com/studio/?hl=es-419>
- <https://stackoverflow.com/questions/10930331/how-to-sign-an-already-compiled-apk&sa=D&source=hangouts&ust=1543622379278000&usg=AFQjCNH09quZgpvm7FnTtKI92iC38Zc8Rw>
- <https://es.wikipedia.org/wiki/F-Droid>
- <https://developer.android.com/studio/command-line/adb?hl=es-419>

