Herramientas para el análisis dinámico de aplicaciones Android

Androl4b

Androl4b es una máquina virtual de Linux, específicamente Ubuntu que nos permite realizar análisis tanto estático como dinámico gracias a un gran número de aplicaciones, frameworks, herramientas y tutoriales que nos permitirán llevar a cabo experimentos, pruebas o análisis que se quieran realizar.
Algunas de las herramientas que incluye ésta máquina virtual son:

- Herramientas para realizar ingeniería inversa en aplicaciones: radare2, ByteCodeViewer, APKtool y MARA.
- Herramientas para realizar análisis de seguridad: Mobile Security Framework (MobSF), Drozer, BurpSuite, Qark y AndroBugs Framework.
- Aplicaciones vulnerables para realizar pruebas: DIVA (Damn Insecure and vulnerable App for Android), InsecureBankv2, Android Security Sandbox, GoatDroid y Sieve.
- Otras: FindBugs-IDEA, Wireshark y Android Studio IDE.

Para descargar ésta herramienta debemos acceder al siguiente link: https://github.com/sh4hin/Androl4b

CuckooDroid

Es un framework de análisis automatizado de malware en Android. Provee una inspección tanto estática como dinámica de un APK, asi como evadir ciertas técnicas de vm-detection, extracción de claves de cifrado, inspección SSL, rastreo de llamadas al API, entre otras utilidades. Es un framework altamente extensible y personalizable y cuenta con módulos para: análisis estático, análisis dinámico, gestores de virtualización, detección anti-análisis, análisis de tráfico, recopilación de información y firmas conductuales.

En resumen, se utiliza para ejecutar y analizar automáticamente archivos y recopilar resultados de análisis exhaustivos que describen lo que hace el malware mientras se ejecuta dentro de un sistema operativo Windows aislado.

Para descargar esta herramienta se puede ingresar a la página oficial de cuckooDroid para más información. Una manera alterna de obtener ésta herramienta es a través del siguiente link: https://github.com/idanr1986/cuckoo-droid

Android security evaluation framework

Android security toma un conjunto de aplicaciones ya sea preinstaladas o archivos APK y permite realizar pruebas en un AVD (Android Virtual Device) que ya ha sido configurado. El proceso consiste en simular todo el ciclo de vida de una aplicación Android en un dispositivo (virtual/físico) y recopilar datos mientras se activan los aspectos conductuales del mismo. En pocas palabras, se puede descargar una aplicación Android de Internet e instalarla en un dispositivo Android, ej. haga clic en diferentes botones, desplácese hacia arriba/hacia abajo, deslice el ratón, etc. Al hacerlo, se recopila un registro de actividad utilizando adb (Android debug bridge utility que está disponible como parte de un SDK de Android) y el tráfico de red utilizando tcpdump (una herramienta de captura de paquetes ampliamente utilizada).

Al analizar el comportamiento de distintas aplicaciones se encontraron resultados interesantes sobre aplicaciones que filtraban información confidencial como IMEI, IMSI, tarjeta SIM o un número de teléfono de un dispositivo. Algunas aplicaciones maliciosas pueden enviar estos datos en texto claro a través de Internet y son mucho más fáciles de capturar analizando los datos de comportamiento recopilados. Sin embargo, algunas aplicaciones maliciosas pueden ser lo suficientemente sofisticadas como para detectar la configuración predeterminada de un dispositivo Android virtual y es posible que se comporten de forma diferente en dichas configuraciones. Con el fin de superar estas limitaciones, un dispositivo virtual se puede construir a la medida ajustando el kernel y también modificando la configuración predeterminada para emular un dispositivo real o puede ser reemplazado por un dispositivo físico Android.

Esta herramienta está disponible en http://code.google.com/p/asef/

Mobile Security Framework (MobSF)

Es una herramienta para el análisis estático y dinámico de aplicaciones sin importar la plataforma (Android o iOS), es una aplicación móvil de código abierto inteligente y puede ser utilizado para un análisis de seguridad eficaz y rápido de aplicaciones Android e iOS y soporta tanto binarios (APK & IPA) como código fuente comprimido.

MobSF también puede realizar pruebas de Seguridad Web API con su API Fuzzer que puede recopilar información, analizar Cabeceras de Seguridad, identificar vulnerabilidades específicas de la API móvil como XXE, SSRF, Path Traversal, IDOR y otros problemas lógicos relacionados con la sesión y la limitación de tarifas de la API.
Esta herramienta se encuentra disponible en https://github.com/MobSF/Mobile-Security-Framework-MobSF

AVD (Android Virtual Device) con adb (Android Debug Bridge)

Esta herramienta es utilizada en muchas investigaciones acerca de malware. Consiste en poner en marcha un dispositivo virtual (emulador Android) para la instalación de aplicaciones tanto benignas como maliciosas haciendo uso de un adb que permite llevar a cabo la comunicación con el dispositivo y a través de este "puente" se puede llevar a cabo toda la gestión de aplicaciones (instalación y desinstalación), pero a diferencia de los frameworks anteriormente mencionados que también usan dispositivos virtuales, el propósito de usar un AVD en muchas ocasiones es el de crear su propio framework para realizar experimentos con las aplicaciones Android.
La documentación sobre este dispositivo se puede encontrar en la página oficial de Android Developer: https://developer.android.com/studio/command-line/adb.html

Herramientas usadas en algunos trabajos

El AVD es la herramienta más usada por los investigadores a la hora de realizar sus estudios con aplicaciones tanto benignas como maliciosas, tal es el caso de los siguientes estudios: "Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices", "A First Look at Android Malware Traffic in First Few Minutes", "DroidCollector: A High Performance Framework for High Quality Android Traffic Collection", entre otros. Debido a que para muchos investigadores es muy importante tener control de todo el proceso de captura de tráfico de aplicaciones Android y de todo su experimento, las herramientas anteriormente mencionadas excepto el AVD no son usadas por estas personas debido a que estas hacen su

propio análisis y tienen sus propios criterios para realizar el análisis y clasificación de aplicaciones benignas y maliciosas. Otra de las razones es que para los investigadores es muy importante seguir de cerca todo el proceso al igual que poder crear su propio framework.

Conclusiones

Para lograr la tarea de realizar un análisis tanto estático como dinámico de aplicaciones Android, existen muchas herramientas y frameworks que permiten llevar este proceso a cabo. Sin embargo uno de los problemas que se presenta en algunos de los entornos de virtualización mencionados anteriormente es el poco control que se tiene para realizar el análisis y ésta es una de las razones por las que todas salvo el AVD no se aplicaron en el proyecto, además del hecho de que la instalación de algunas de estas son algo complejas y sólo se permite realizar un análisis al tiempo, pero lo más importante es que los criterios que se deben usar para decidir si una aplicación es maliciosa o no deben ser definidos por las características que se investigan en uno de los objetivos del proyecto y no por un tercero.

Fuentes

https://www.redeszone.net/2017/10/15/androl4b-analisis-forense-android/
http://cuckoo-droid.readthedocs.io/en/latest/introduction/what/
https://code.google.com/archive/p/asef/
https://github.com/MobSF/Mobile-Security-Framework-MobSF/wiki/1.-Documentation