



Universidad
Nacional
de Loja

FACULTAD DE ENERGÍA LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

Ingeniería en Computación

SOFTWARE SECURITY

"Ingeniería Inversa aplicada a la App Situ de
Kradac"


Docente: Ing. Cristian Naváez Mg.Sc.

Estudiante:

Wagner Cristhoper Castillo Castro


Loja - Ecuador

2022

	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	2	de	13

Índice

1. Introducción	3
2. Objetivos	4
• Objetivo General:	4
• Objetivos específicos:	4
3. Materiales y métodos	4
Materiales	4
Métodos	4
4. Desarrollo	5
5. Conclusiones	12
6. Recomendaciones	12
7. Bibliografía	13
8. Anexos	13
• Enlace a la Sección de OWASP Mobile “Exploring the app Package”:	13
• Código en GitHub:	13
• Extensiones en Visual Estudio Code:	13
○ Extensión APKLab:	13
○ Extensión Smali2Java:	13
• Enlace a repositorios:	13
○ Repositorio de Jadx:	13
○ Descarga de Jadx en Nightly:	13

	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	3	de	13


1. Introducción

La ingeniería inversa es una actividad que se ocupa de descubrir cómo funciona un programa, función o característica de cuyo código fuente no se dispone, hasta el punto de poder modificar ese código o generar código propio que cumpla las mismas funciones. [1] El desarrollo inverso explica cómo funcionan los componentes y contribuye así a una comprensión íntegra de las funciones. Precisamente por eso es habitual que se utilice esta técnica con los productos de la competencia, pues promete la posibilidad de mejorar el propio producto o adaptarlo para mantenerse firme en el mercado.

La ingeniería inversa de código permite a los programadores invertir los procesos de desarrollo y producción de un software y, de esta forma, echar un vistazo entre las bambalinas de un programa. La deconstrucción y el desarrollo invertido de un software permite estudiar el código fuente de una aplicación. Si el código es conocido, entonces el software se convierte en un libro abierto para los expertos, que pueden cambiar, reconstruir y entenderla arquitectura del programa, el funcionamiento y las estructuras internas. [2]

Debido a los conceptos anteriormente mencionados, se entiende la importancia y relevancia de este proceso, y por tal motivo el presente documento se enfoca en establecer el procedimiento a seguir para poder obtener el código de una aplicación, siendo que la aplicación usada es Situ de la empresa Kradac. Situ es una aplicación donde se puede visualizar las rutas de transporte público en la ciudad de Loja, Ecuador.

Dentro del proceso de esta práctica se hace uso de la herramienta ApkTool y descompilador Jadx para así por medio del apk de la aplicación Situ obtener el código fuente de la aplicación, este procedimiento permite identificar cada una de las propiedades más relevantes de la aplicación mediante un posterior análisis exhaustivo de la estructura, funciones y operaciones. Además, cabe señalar que el procedimiento es más fácil debido al uso de las extensiones Smali2Java y APKLab dentro de Visual Studio Code.

 unl Universidad Nacional de Loja	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	4	de	13

2. Objetivos

- **Objetivo General:**

- Mostrar el proceso de ingeniería inversa a la aplicación Situ de Kradac.

- **Objetivos específicos:**

- Documentar el proceso de ingeniería inversa de la aplicación Situ de Kradac.
- Implementar Jadx para comprender el código de forma más clara desde Java.


3. Materiales y métodos

Materiales

- Internet
- Computadora Portátil
- Visual Estudio Code
- ApkTool y Jadx
- Extensiones Visual Estudio Code: Smali2Java y APKLab
- Sistema operativo Linux

Métodos

- **Inductivo.** - Es el proceso de conocimiento que se inicia con la observación de lo particular para llegar a lo general. Este método se lo utilizará para la formulación de objetivos.
- **Bibliográfico.** - En sentido más específico, el método de investigación bibliográfica es el conjunto de técnicas y estrategias que se emplean para localizar, identificar y acceder a aquellos documentos que contienen la información pertinente o relevante para la investigación.
- **Método experimental.** – El método experimental es un tipo de investigación cuantitativa. Se basa en un protocolo de control, la presencia de variables, la manipulación de dichas variables y la observación de resultados cuantificables.

	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	5	de	13

4. Desarrollo

Lo primero es conseguir el apk de playstore y luego compartir el archivo mediante un administrador de carpetas como Files de Google, este archivo se puede compartir por cualquier medio de comunicación. En IOS este proceso se puede obviar debido a que se puede realizar sin el uso de una aplicación intermediaria.

Luego de obtener el aplicativo, el primer paso es establecer un nuevo nombre al apk, el establecido dentro de la aplicación es “*kradac.situ.apk*”.

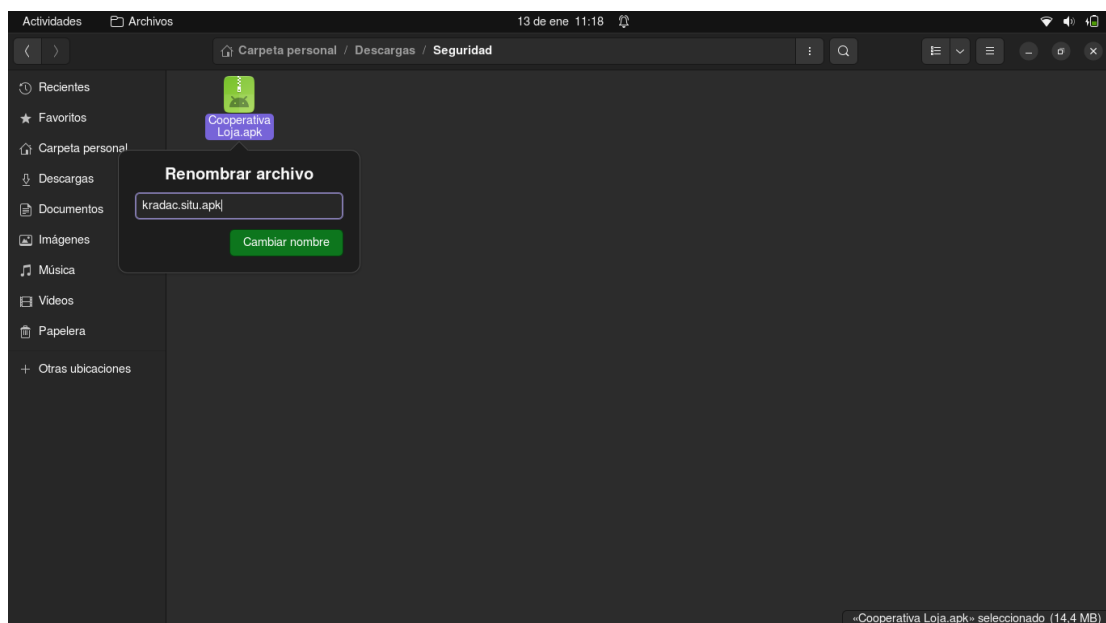



Ilustración 1: Renombramiento al aplicativo

En el siguiente paso se prosigue a instalar por medio de la consola la herramienta apktool, es aconsejable antes actualizar todos los paquetes por medio del comando “*sudo apt-get update*” y posterior a esto escribir el comando “*sudo apt-get install apktool*”. ApkTool es una herramienta que permite la ingeniería inversa dentro de Android. Esta herramienta puede descifrar los recursos de un aplicativo de forma casi original y reconstruirlos después de algunas modificaciones.

 <div> <div>unl</div> <div>Universidad Nacional de Loja</div> </div>	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	7	de	13

Code, además permite volver a compilar el archivo a un punto APK cuando sea necesario, pudiendo de esta forma crear archivos crakeados.

El enlace a esta extensión se puede visualizar dentro del punto 8. Anexos

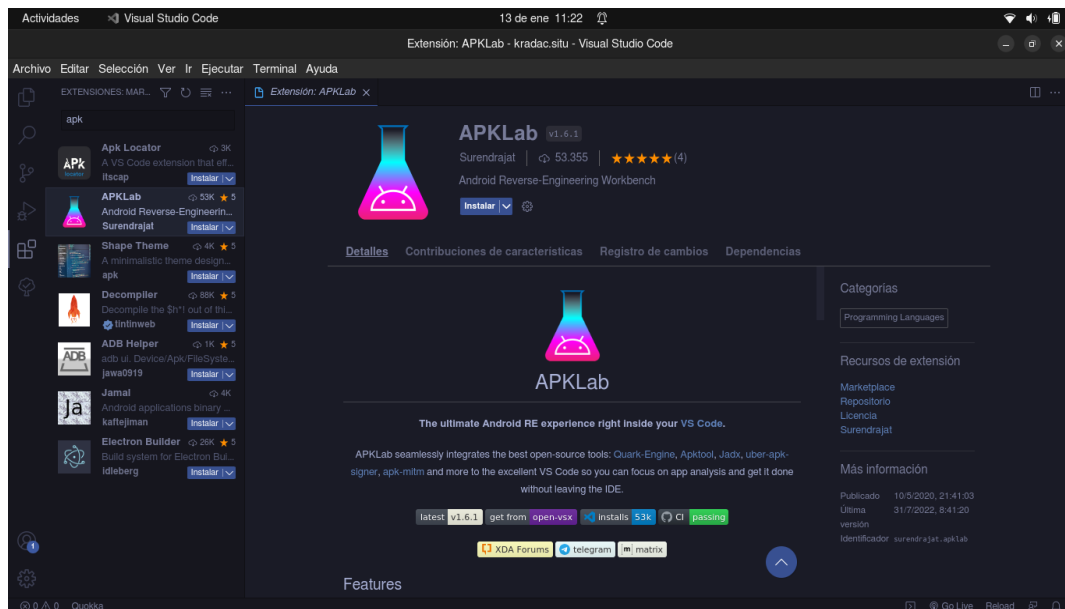



Ilustración 4: Instalación de extensión APKLab

Smali2Java es otra extensión importante dentro del proceso, esta herramienta permite usar las funcionalidades de Jadx directamente desde Visual Estudio Code, la herramienta permite convertir el código seleccionado a java para una comprensión mas clara. El enlace a esta extensión se puede visualizar dentro del punto 8. Anexos



Ilustración 5: Instalación de extensión Smali2Java

 <div> <div>unl</div> <div>Universidad Nacional de Loja</div> </div>	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	8	de	13

Jadx es una herramienta de línea de comandos para producir código fuente de java desde Android. El enlace de este repositorio esta adjuntado dentro de punto 8 Anexos como **“Repositorio de Jadx”**.

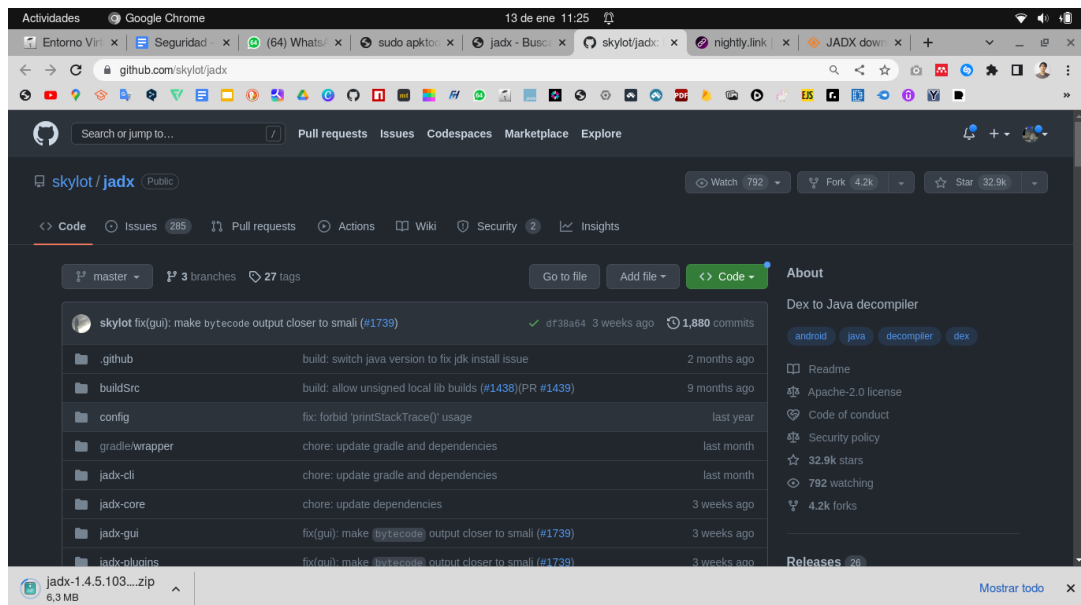
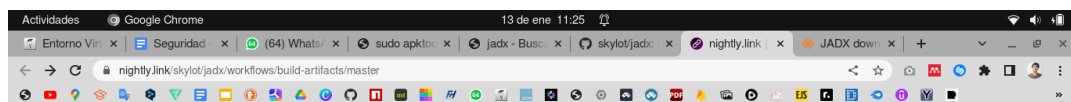


Ilustración 6: Repositorio Jadx

En la siguiente imagen se puede apreciar la pagina de descarga para jadx. El enlace a escoger es el señalado por el cuadro rojo. El enlace de esta página esta adjuntado dentro de punto 8. Anexos como **“Descarga de Jadx en Nightly”**.



nightly.link

Repository skylot/jadx


Workflow build-artifacts.yml | Branch master

Choose one of the artifacts:

jadx-1.4.5.103-df38a642	https://nightly.link/skylot/jadx/workflows/build-artifacts/master/jadx-1.4.5.103-df38a642.zip
jadx-gui-1.4.5.103-df38a642-no-jre-win.exe	https://nightly.link/skylot/jadx/workflows/build-artifacts/master/jadx-gui-1.4.5.103-df38a642-no-jre-win.exe.zip
jadx-gui-1.4.5.103-df38a642-with-jre-win	https://nightly.link/skylot/jadx/workflows/build-artifacts/master/jadx-gui-1.4.5.103-df38a642-with-jre-win.zip

nightly.link/skylot/jadx/workflows/_jadx-1.4.5.103-df38a642.zip

Ilustración 7: Descarga de Jadx dentro de Nightly

 <div> <div>unl</div> <div>Universidad Nacional de Loja</div> </div>	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	9	de	13

Luego de la descarga del anterior archivo se prosigue a extraer el archivo con el fin de obtener la ruta binaria de la carpeta resultante.

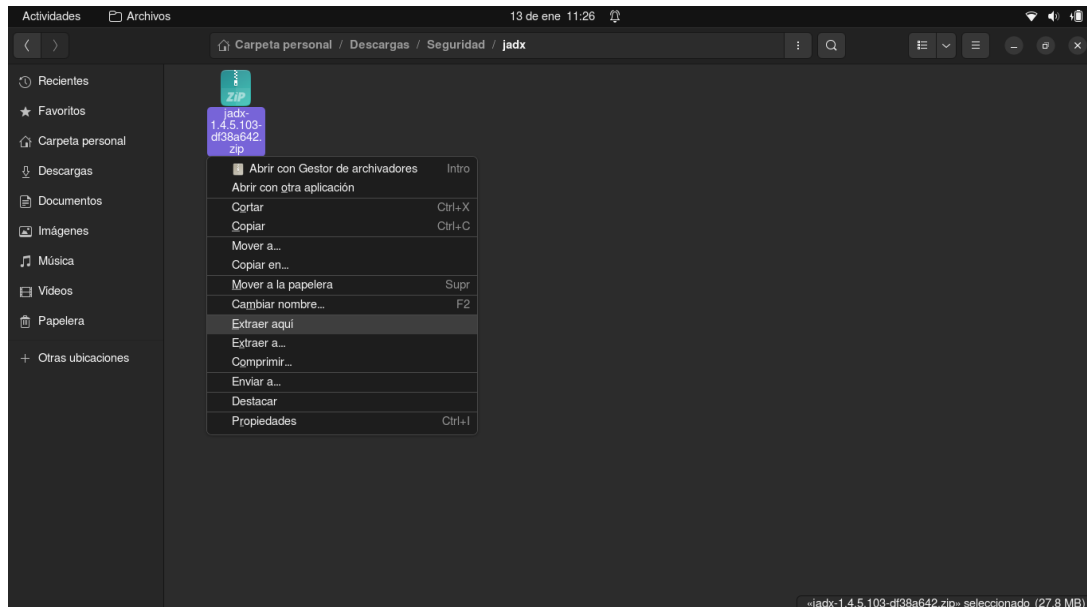


Ilustración 8: Extracción del archivo Jadx

El siguiente paso es configurar la extensión Smali2Java con el fin de establecer la ruta de Jadx. La cual es una ruta bin a Jadx, que permite producir el código java desde Android.

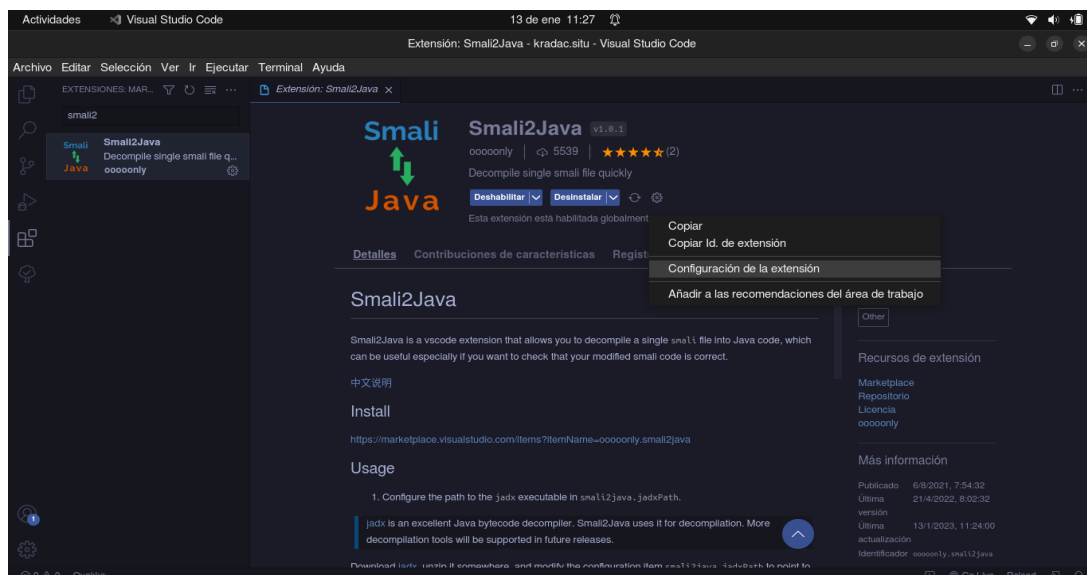



Ilustración 9: Configuración de la extensión Smile2Java

En la siguiente imagen se puede observar la configuración dentro de Smali2Java a Jadx Path, en la imagen se puede apreciar una ruta Linux debido a que no hubo

 <div> <div>unl</div> <div>Universidad Nacional de Loja</div> </div>	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	10	de	13

inconvenientes con Smali2Java, sin embargo, si la presente configuración no funciona el siguiente paso se puede realizar desde Windows o Mac Os transfiriendo los archivos a ese sistema operativo, instalando de nuevo las extensiones y cambiando la ruta hasta el archivo jadx.bat, en este caso seria:

“C:\Users\Wagner Castillo\Downloads\Seguridad Kradac\jadx\jadx-1.4.5.103-df38a642\bin\jadx.bat”

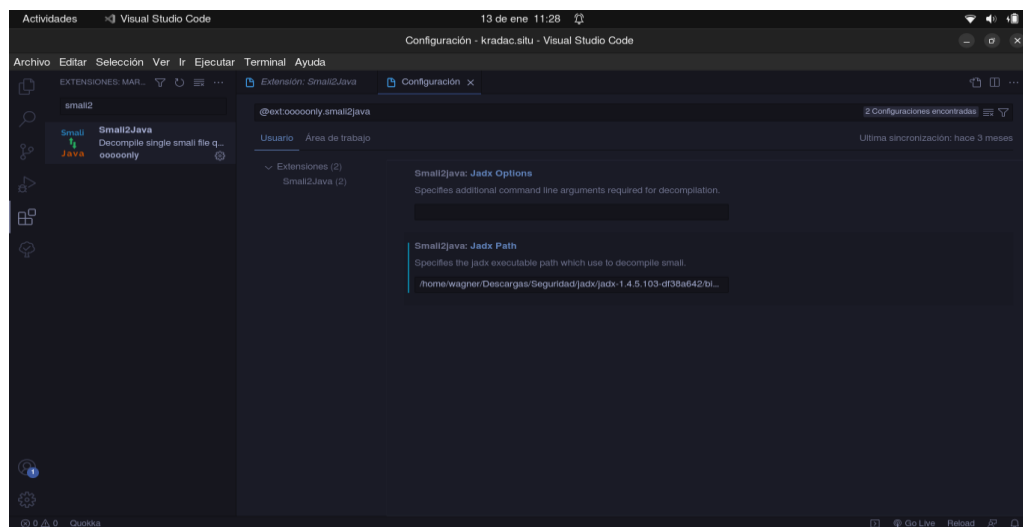


Ilustración 10: Configuración de la Ruta

Jadx permite descompilar los archivos a Java cuando los mismos tienen una extensión Smali. En la siguiente imagen se observa que esta opción se encuentra presente luego de dar clic derecho sobre un código con .smali.

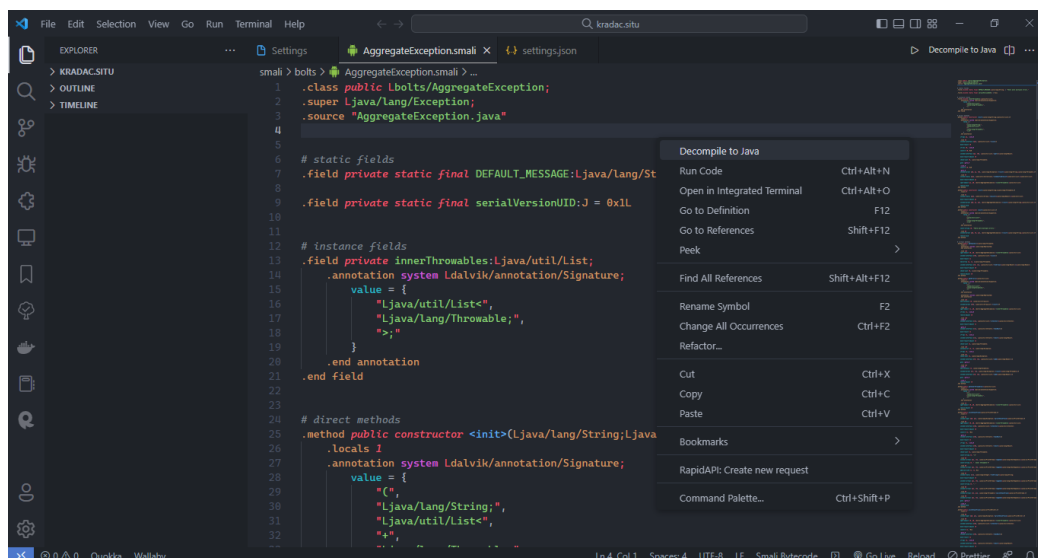

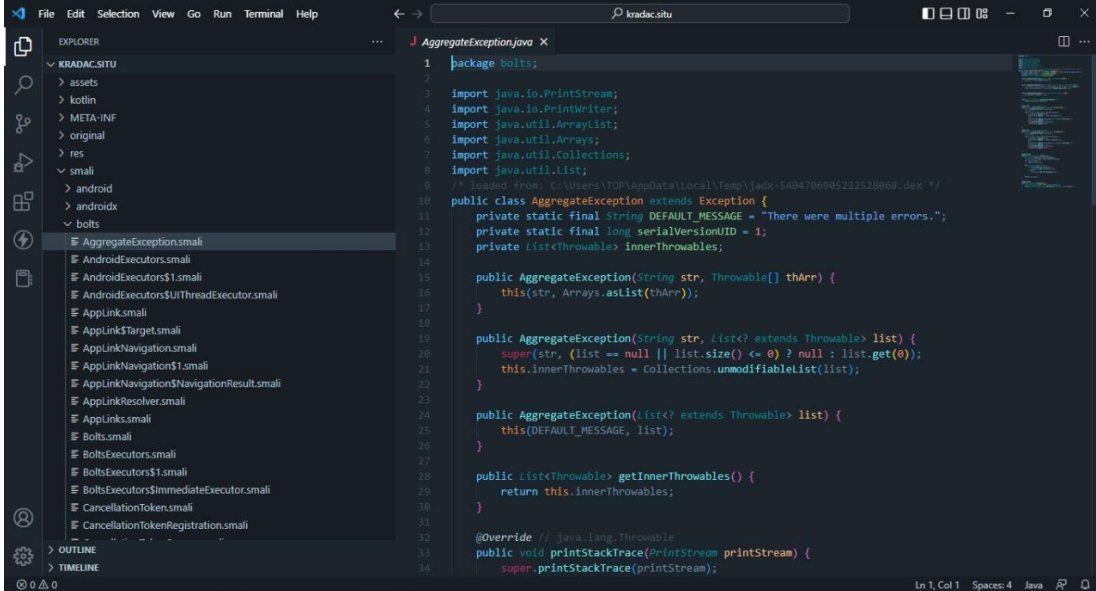


Ilustración 11: Producción de Smile a Java

 <div> <div>unl</div> <div>Universidad Nacional de Loja</div> </div>	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	11	de	13

En la siguiente imagen se puede observar como el archivo cambio de .smali a .java. De esta forma, el código puede ser mejor comprendido y es una gran ayuda para aquellos que deseen observar el funcionamiento de una aplicación de forma más clara y entre más posibles usos.




```

1 package bolts;
2
3 import java.io.PrintStream;
4 import java.io.PrintWriter;
5 import java.util.ArrayList;
6 import java.util.Arrays;
7 import java.util.Collections;
8 import java.util.List;
9
10 // Generated from C:\Users\TOP\AppData\Local\Temp\jadx-548478690522528068.dex */
11 public class AggregateException extends Exception {
12     private static final String DEFAULT_MESSAGE = "There were multiple errors.";
13     private static final long serialVersionUID = 1;
14     private List<Throwable> innerThrowables;
15
16     public AggregateException(String str, Throwable[] thArr) {
17         this(str, Arrays.asList(thArr));
18     }
19
20     public AggregateException(String str, List<? extends Throwable> list) {
21         super(str, (list == null || list.size() <= 0) ? null : list.get(0));
22         this.innerThrowables = Collections.unmodifiableList(list);
23     }
24
25     public AggregateException(List<? extends Throwable> list) {
26         this(DEFAULT_MESSAGE, list);
27     }
28
29     public List<Throwable> getInnerThrowables() {
30         return this.innerThrowables;
31     }
32
33     @Override // java.lang.Throwable
34     public void printStackTrace(PrintStream printStream) {
35         super.printStackTrace(printStream);
36     }
37 }

```

Ilustración 12: Obtención de código Java


	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	12	de	13

5. Conclusiones

OBJETIVOS	CONCLUSIONES
Mostrar el proceso de ingeniería inversa a la aplicación Situ de Kradac.	El proceso de ingeniería inversa permite obtener casi cualquier código de forma original este procedimiento puede resultar sumamente fácil si no se establecen medidas de control como la ofuscación de código, protección frente a la manipulación o verificaciones del entorno con el fin de evitar la exposición del código frente a hackers o la competencia.
Documentar el proceso de ingeniería inversa de la aplicación Situ de Kradac.	El procedimiento de la documentación se trabajó desde la consideración de tratar de ser claro y conciso, además se puso a disposición cada uno de los enlaces ocupados y relevantes para la presente práctica.
Implementar Jadx para comprender el código de forma más clara desde Java.	Jadx es un descompilador que facilito el proceso de obtención de código, además cabe señalar que ofrece una interfaz sencilla la cual hacer simple el proceso de obtención de código Java a partir de los archivos Android Dex y Apk.

6. Recomendaciones

- Es importante recordar que muchos de los comandos que fueron expuestos en la presente practica se establecen gracias al uso de librerías que se encuentran completamente disponibles debido al uso de librerías gratuitas. Además, la documentación de estas librerías ofrece muchas más posibilidades con respecto a otras opciones y soluciones.
- La presente practica surge del uso de una las herramientas mencionadas por la guía de OWASP Mobile Application Security, esta guía ofrece mucha mas información dentro la sección “Exploring the app Package”. Además, se pueden encontrar muchas más secciones interesantes dentro de la guía. El enlace a esta sección se encuentra dentro de la sección 8. Anexos.

	Software Security		Práctica / Taller Nro.1			
			Fecha:	14/01/2023		
	Docente: Cristian Narváez		Página:	13	de	13

7. Bibliografía

- [1]. Celer, Victor (30 de mayo de 2021). «Mitigación de Ingeniería Inversa en Apps Android» (PDF). Revista CelerSMS (Colombia) (1): 5-6. ISSN 2745-2336. OCLC 1261377002.
- [2]. *La ingeniera inversa de software*. (2020, March 18). IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/ingenieria-inversa-de-software/>

8. Anexos

- **Enlace a la Sección de OWASP Mobile “Exploring the app Package”:**
https://mas.owasp.org/MASTG/Android/0x05b-Basic-Security_Testing/#exploring-the-app-package
- **Código en GitHub:** <https://github.com/wagnercastillo/IngenieriaInversa-AppSitu-Kradac>
- **Extensiones en Visual Estudio Code:**
 - **Extensión APKLab:**
<https://marketplace.visualstudio.com/items?itemName=Surendrajat.apklab>
 - **Extensión Smali2Java:**
<https://marketplace.visualstudio.com/items?itemName=ooooonly.smali2java>
- **Enlace a repositorios:**
 - **Repositorio de Jadx:** <https://github.com/skylot/jadx>
 - **Descarga de Jadx en Nightly:**
<https://nightly.link/skylot/jadx/workflows/build-artifacts/master>