

ANÁLISIS DE FLUJO DE INFORMACIÓN EN APLICACIONES ANDROID

Lina Marcela Jiménez Becerra

UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Junio 9, 2015

Descripción del Problema

Manipulación de información del usuario

- El desarrollador Android no tiene cómo definir políticas de seguridad para regular el flujo de información de sus aplicaciones.
- Complejidad para prevenir fugas de información del usuario.

Descripción del Problema

Manipulación de información del usuario

- El desarrollador Android no tiene cómo definir políticas de seguridad para regular el flujo de información de sus aplicaciones.
- Complejidad para prevenir fugas de información del usuario.

Reporte McAfee

- Aplicaciones Android invasivas de la privacidad del usuario.
- No toda aplicación invasiva contiene malware.
- De las aplicaciones que más vulneran la privacidad del usuario 35 % contienen malware.

Descripción del Problema

Limitaciones de la API

- Políticas de control de acceso de la API.
- Regular el acceso a recursos protegidos.
- No hacen seguimiento al flujo de información.

Propuestas existentes

- Análisis estático y análisis dinámico.
- Análisis dinámico: actuales caminos de ejecución.
- Análisis estático: es posible incluir todos los caminos de ejecución.

Descripción del Problema

Propuestas existentes

Data-Flow con técnicas de análisis tainting.

- Se hace seguimiento a los datos marcados.
- No incluye todos los posibles caminos de ejecución.
- Ejemplo: FlowDroid

Descripción del Problema

Propuestas existentes

Flujo de información con técnicas Program Dependence Graphs(PDG).

- Los PDG proveen una representación del programa que se analiza.
- Análisis de flujos de información del programa de principio a fin.
- Incluye todos los posibles caminos de ejecución.
- Ejemplo: Joana.

Descripción del Problema

Propuestas existentes

Enfoque de las propuestas existentes:

- Identificar fugas de información en aplicativos ya implementados.
- FlowDroid: no incluye todos los posibles caminos de ejecución.
- Joana: no permite definir las políticas de seguridad a evaluar.

Propuesta de solución

El desarrollador requiere

- Definir políticas de seguridad desde la implementación de sus aplicativos.
- Una herramienta que verifique las políticas definidas.
- Garantizarle al usuario que la aplicación respeta determinadas políticas de seguridad.

Características de Jif

- Lenguaje tipado de seguridad.
- Extensiones de seguridad al lenguaje java.
- Restricciones para uso de la información.
- Análisis de flujo de información mediante chequeo de etiquetas.

Características sobresalientes de Jif

- Anotar propiedades de seguridad.
- Verificar las propiedades de seguridad.
- Cubrir todas las posibles ramas de ejecución en el análisis.
- Diseñado para aplicativos Java.

Flujos: explícitos - implícitos

```
int x,y;  
x = 1;  
y = 4 + x;
```

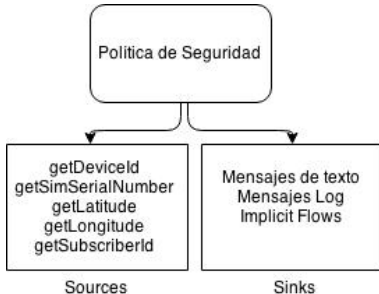
```
void foo(a){  
  int x;  
  if(a > 10)  
    x = 1;  
  else  
    x = 2;  
  printf(x);  
}
```

Política de Seguridad

Anotaciones Propuestas

Anotaciones a la API

Política de Seguridad



Flujos de información
entre: información con
nivel de seguridad
alto(sources) e
información con nivel
de seguridad
bajo(sinks).

Autoridad y Labels de Anotación

Autoridad Máxima



Nivel de Seguridad Alto:



Sólo el principal *Alice*
dueño de la política
podrá leer la información.

Nivel de Seguridad Bajo:



No se define un principal,
todos pueden leer
la información.

Anotaciones a la API

Controlar canales

- Mensajes de texto (SmsManager)
- Mensajes log (Log)

Anotaciones a la API

Controlar canales

- Mensajes de texto (SmsManager)
- Mensajes log (Log)

Clases adicionales requeridas

- Clases para los sources (TelephonyManager)
- Clases para métodos de sobresscritura (Activity)

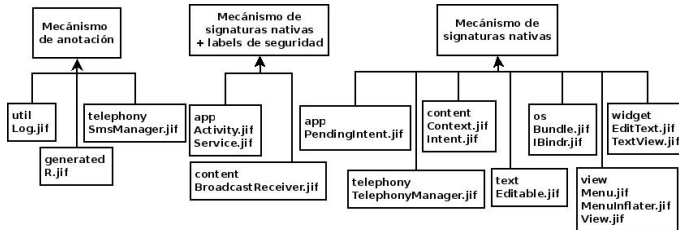
Anotaciones a la API

Controlar canales

```
sendTextMessage{Alice:} (  
String{Alice:} destinationAddress ,  
String{Alice:} sourceAddress ,  
String{}text ,  
PendingIntent{Alice:} sentIntent ,  
PendingIntent{Alice:} deliveryIntent  
){}  

```

Anotaciones a la API



Evaluación

- Conjunto de evaluación.
- DroidBench benchmark.

Evaluación

- Conjunto de evaluación.
- DroidBench benchmark.

	FlowDroid	JoDroid	Prototipo
Precisión	78,57 %	78,57 %	73,68 %
Recall	78,57	78,57 %	100 %
Detección Flujos Implícitos	No	Si	Si

Cuadro comparativo

Item	Prototipo vs FlowDroid				Prototipo vs JoDroid			
	ventaja	desvent	similit	diff	ventaja	desvent	similit	diff
Menor Precisión		✓				✓		
Mayor Recall	✓				✓			
Menor costo en desempeño					✓			
Bajo costo en desempeño			✓					
Detección de flujos implícitos	✓						✓	
No detección automática de sources y sinks		✓					✓	
No soporte para Análisis interApp		✓					✓	
Tipo de análisis(flujo de información; flujo de datos)				✓				
Tipo de análisis IFC							✓	
Técnica de análisis: PDG, slicing								✓

Conclusiones

- Herramienta de análisis mediante el sistema de anotaciones de Jif.

Conclusiones

- Herramienta de análisis mediante el sistema de anotaciones de Jif.
- Análisis de flujos implícitos.

Conclusiones

- Herramienta de análisis mediante el sistema de anotaciones de Jif.
- Análisis de flujos implícitos.
- Desempeño y completitud en el análisis.

Conclusiones

- Herramienta de análisis mediante el sistema de anotaciones de Jif.
- Análisis de flujos implícitos.
- Desempeño y completitud en el análisis.
- Retos para el análisis de aplicaciones Android mediante el sistema de anotaciones de Jif.

Trabajo Futuro

- Extensiones al esquema de anotación.
- Análisis de políticas de integridad.
- Mecanismos adicionales: declasificación y endorsement.

Preguntas