Улучшение инструментов для отладки приложений Android и исследование возможностей их применения для анализа поведения ПО

Сидоров Владислав Олегович[1,2]

Научный руководитель: Проскурин Вадим Геннадьевич[2]

[1]МФТИ [2]ИСП РАН

17 марта 2025 г.

Мотивация

Проблема

Нужно постоянно проверять ПО на наличие вредоносного содержания и уязвимостей

Способы проверки ПО

Способы проверки ПО

Ограничение

Часто не имеем исходного кода приложения (или работа с ним является очень сложной)

Постановка задачи

Разработать инструмент для автоматизированного динамического анализа поведения ПО для выявления потенциально вредоносных действий или уязвимостей

Постановка задачи

Разработать инструмент для автоматизированного динамического анализа поведения ПО для выявления потенциально вредоносных действий или уязвимостей

Платформа

В данной работе анализируем только приложения для Android

Постановка задачи

Разработать инструмент для автоматизированного динамического анализа поведения ПО для выявления потенциально вредоносных действий или уязвимостей

Платформа

В данной работе анализируем только приложения для Android

Пример вредоносного ПО

Сканер QR-кодов, отправляющий данные с камеры пользователя в сеть

Этапы работы инструмента

- Сбор данных о работе приложения
- Анализ данных и вердикт

Сбор данных

Сбор данных

- Будем отслеживать вызовы стандартных ключевых методов

Сбор данных

- Будем отслеживать вызовы стандартных ключевых методов
- Только приложения на Java

Сбор данных

- Будем отслеживать вызовы стандартных ключевых методов
- Только приложения на Java
- Нужно фиксировать не только сам факт вызова метода, но и его аргументы, а также "внешнюю информацию": stacktrace, идентификатор потока и т.д.

Сбор данных

- Будем отслеживать вызовы стандартных ключевых методов
- Только приложения на Java
- Нужно фиксировать не только сам факт вызова метода, но и его аргументы, а также "внешнюю информацию": stacktrace, идентификатор потока и т.д.
- Работа в онлайн режиме

Сбор данных

- Будем отслеживать вызовы стандартных ключевых методов
- Только приложения на Java
- Нужно фиксировать не только сам факт вызова метода, но и его аргументы, а также "внешнюю информацию": stacktrace, идентификатор потока и т.д.
- Работа в онлайн режиме

Вывод

Хотим высокоуровневый аналог ptrace!

Frida

- Open-source фреймворк для инструментации приложений (не только под Android)
- Огромные возможности по внедрению своего кода в чужие приложения
- Встроенный режим трассировки (то, что нам нужно!)

Frida

Вроде бы все идеально, но...

Frida

Вроде бы все идеально, но... встроенный трассировщик слабо рассчитан на использование в автоматизированном режиме

Frida

Вроде бы все идеально, но... встроенный трассировщик слабо рассчитан на использование в автоматизированном режиме

План

Доработаем его под наши нужды! (проект open-source)

Список методов

Подготовили список интересующих методов для трассировки

org.apache.http.impl.client.HttpClient.execute(HttpUriRequest)
io.socket.client.Socket.connect()
java.net.InetAddress.getHostAddress()
android.media.ImageReader.getSurface()
android.hardware.Camera.open(int)
android.hardware.camera2.CameraCaptureSession.capture(CaptureRequest, CameraCaptureSession.CaptureCallback, Handler)
android.net.wifi.WifiManager.getConnectionInfo()
java.security.X509Certificate.getPublicKey()
com.google.firebase.database.DatabaseReference.child(String)
android.webkit.WebView.setWebChromeClient(WebChromeClient)
android.media.MediaRecorder.start()
android.media.projection.MediaProjectionManager.createScreenCaptureIntent()
android.media.AudioManager.setStreamVolume(int, int, int)
android.location.Location.getLatitude()
android.location.LocationManager.requestLocationUpdates(String, long, float, LocationListener)
java.security.MessageDigest.getInstance(String)
...

Всего порядка 80 функций

Анализ данных

Анализ данных

Суровая реальность

В вопросах безопасности ПО почти никогда не удается полностью обойтись без человека

Анализ данных

Только трассировочного лога недостаточно для определения вердикта о вредоносности ПО. Нужны еще базовые знания о предназначении приложения

Анализ данных

Только трассировочного лога недостаточно для определения вердикта о вредоносности ПО. Нужны еще базовые знания о предназначении приложения

Приложение	**Использует камеру?**	**Использует сеть?**	**Вредоносно?**
Видео-чат	Да	Да	
Шпионский сканер QR-кодов	Да	Да	

С точки зрения трассировки логи у обоих приложений будут похожие

Анализ данных

Только трассировочного лога недостаточно для определения вердикта о вредоносности ПО. Нужны еще базовые знания о предназначении приложения

Приложение	**Использует камеру?**	**Использует сеть?**	**Вредоносно?**
Видео-чат	Да	Да	**Нет**
Шпионский сканер QR-кодов	Да	Да	

С точки зрения трассировки логи у обоих приложений будут похожие

Анализ данных

Только трассировочного лога недостаточно для определения вердикта о вредоносности ПО. Нужны еще базовые знания о предназначении приложения

Приложение	**Использует камеру?**	**Использует сеть?**	**Вредоносно?**
Видео-чат	Да	Да	**Нет**
Шпионский сканер QR-кодов	Да	Да	**Да**

С точки зрения трассировки логи у обоих приложений будут похожие

Анализ данных

Предполагаемые способы для вынесения вердикта:

Анализ данных

Предполагаемые способы для вынесения вердикта:

- Структурирование полученных данных, визуальное оформление

Анализ данных

Предполагаемые способы для вынесения вердикта:

- Структурирование полученных данных, визуальное оформление
- Найденные вручную закономерности

Анализ данных

Предполагаемые способы для вынесения вердикта:

- Структурирование полученных данных, визуальное оформление
- Найденные вручную закономерности

Перспектива для следующей работы

Использовать LLM

Конец