

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

Сидоров Владислав Олегович^{1,2}

Научный руководитель: Проскурин Вадим Геннадьевич²

¹МФТИ ²ИСП РАН

20 мая 2025 г.

1. Введение
2. Постановка задачи
3. Прodelанная работа
4. Результаты

Мотивация

Хотим понимать поведение программы во время ее исполнения

Мотивация

Хотим понимать поведение программы во время ее исполнения

Зачем это может быть нужно:

Мотивация

Хотим понимать поведение программы во время ее исполнения

Зачем это может быть нужно:

- Отладка

Мотивация

Хотим понимать поведение программы во время ее исполнения

Зачем это может быть нужно:

- Отладка
- Обнаружение подозрительной активности

Мотивация

Хотим понимать поведение программы во время ее исполнения

Зачем это может быть нужно:

- Отладка
- Обнаружение подозрительной активности
- Проверка соблюдения требований к ПО

Одним из подходов для решения этой задачи является отслеживание вызовов некоторого набора методов во время исполнения программы (call monitoring)

Широко известный ptrace

- Классический системный вызов на Unix-подобных системах
- Позволяет одному процессу на низком уровне контролировать работу другого процесса
- Чаще всего используется для внедрения своего кода в чужой процесс (в т.ч. для call monitoring)

Почему платформой выбран Android?

Почему платформой выбран Android?

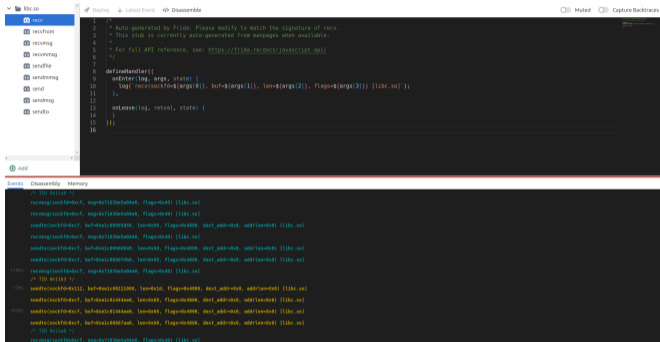
- Огромное количество приложений. Потенциальный доступ к чувствительной информации пользователя

Почему платформой выбран Android?

- Огромное количество приложений. Потенциальный доступ к чувствительной информации пользователя
- Основная масса ПО на Java \Rightarrow имеем одинаковую основную структуру всех приложений (пакеты)

Разработать инструмент общего назначения для автоматизированного динамического анализа поведения ПО на Android методом трассировки вызовов

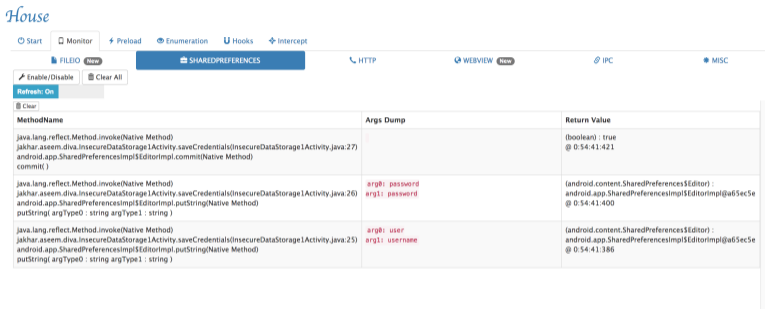
Готовые инструменты есть, но у них всех присутствуют критически важные для использования недостатки



<https://github.com/frida/frida>

Недостатки:

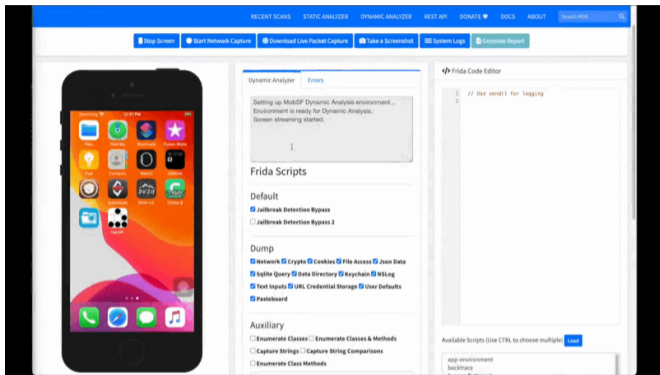
- Результаты анализа не сохраняются, отсутствует возможность загрузить отчет
- Во время работы инструмента нельзя никак менять список отслеживаемых методов
- При каждом запуске нужно указывать каждый метод/пакет отдельно



<https://github.com/nccgroup/house>

Недостатки:

- Отсутствие поддержки (последнее изменение — 5 лет назад)
- Результаты анализа не сохраняются, отсутствует возможность загрузить отчет
- Перехватываемые методы Android API вшиты в код инструмента + их мало



<https://github.com/MobSF/Mobile-Security-Framework-MobSF>

Недостатки:

- Динамический анализ только после полного статического анализа
- Отсутствие возможности использовать с версией Android > 11
- Перехватываемые методы Android API вшиты в код инструмента

FRIDA

- Open-source фреймворк для инструментации приложений (не только под Android)
- Огромные возможности по внедрению своего кода в чужие приложения
- Работает через скрипты на JavaScript, есть свой API для Java

- Работаем только с Java-методами
- За основу берем фреймворк для динамической инструментации Frida
- Устраняем недостатки существующих решений

Был подготовлен 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

Был разработан полноценный прототип инструмента

Главная

Редактор скриптов

Вывод

Редактор

Список

Сохранено

ID конфига: 38

```

74+ overloads: []
75+   arguments: []
76+     "var_name": "url",
77+     "var_type": "java.lang.String"
78+   },
79+   "return_value": {
80+     "var_type": "void"
81+   }
82+ },
83+ {
84+   "arguments": [
85+     {
86+       "var_name": "url",
87+       "var_type": "java.lang.String"
88+     },
89+     {
90+       "var_name": "additionalHttpHeaders",
91+       "var_type": "java.util.Map"
92+     }
93+   ],
94+   "return_value": {
95+     "var_type": "void"
96+   }
97+ },
98+ {
99+   "module_name": "android.webkit.WebView",
100+   "method_name": "setWebChromeClient",
101+   "overloads": [
102+     {
103+       "arguments": [
104+         {
105+           "var_name": "client",
106+           "var_type": "android.webkit.WebChromeClient"
107+         }
108+       ],
109+       "return_value": {
110+         "var_type": "void"
111+       }
112+     }
113+   ],
114+   "module_name": "android.net.NetworkInfo",
115+   "method_name": "isConnected",
116+   "overloads": [
117+     {
118+       "arguments": [
119+       ],
120+       "return_value": {
121+         "var_type": "boolean"
122+       }
123+     }
124+   ],
125+   "return_value": {
126+     "var_type": "boolean"
127+   }
128+ }

```

Net

Сохранить

Сохранение перезапишет существующий конфиг с таким именем

Сканировать метод

Имя модуля

Имя метода

Сканировать

Редактор

Список

Сохранено

ID скрипта: 12

```

3
4 = Java.perform(function() {
5   let clazz;
6   let method;
7   let overload;
8   let overloads;
9
10  clazz = Java.use("android.net.Uri");
11  method = clazz.parse;
12
13
14  overload = method.overload("java.lang.String");
15  overload.implementation = function(arg_uriString) {
16    let retval = this.parse(arg_uriString);
17
18    send({
19      event: "Intercept_log",
20      data: {
21        module_name: "android.net.Uri",
22        method_name: "parse",
23        signature: "(Ljava.lang.String;) android.net.Uri",
24        args: {
25          uriString: arg_uriString
26        },
27        ret: retval,
28      }
29    });
30    return retval;
31  }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46  clazz = Java.use("java.net.URL");
47  method = clazz.openConnection;
48
49
50  overload = method.overload();
51  overload.implementation = function() {
52    let retval = this.openConnection();
53  }
54

```

Net

Сохранить

Сохранение перезапишет существующий скрипт с таким именем

☒ Привязать конфиг

Конфиг будет сохранен под именем: Net

Загрузить скрипт

16 / 21

Что умеет прототип?

- Подключение к Android устройству и определенному процессу на нем

Что умеет прототип?

- Подключение к Android устройству и определенному процессу на нем
- Сохранение скриптов перехвата для будущего использования

Что умеет прототип?

- Подключение к Android устройству и определенному процессу на нем
- Сохранение скриптов перехвата для будущего использования
- Динамическая загрузка любого количества скриптов

Что умеет прототип?

- Подключение к Android устройству и определенному процессу на нем
- Сохранение скриптов перехвата для будущего использования
- Динамическая загрузка любого количества скриптов
- Сохранение логов перехвата в ожидаемом формате в БД, формирование отчета

```
21:48:20      Uri.parse                                {'uriString': 'https://api19-va.tiktokv.com/tiktok/v1/realtime/feedback/?device_platform=android&os...  <instance: android.net.Uri, $className:
20.04.25      (string): uri                                android.net.Uri$StringUri>
21:48:21      Uri.parse                                {'uriString': 'https://v16m.tiktokcdn-us.com/368ae65247aa1527ee9ac2dae5c195b/68184a27/video/tos/us...  <instance: android.net.Uri, $className:
20.04.25      (string): uri                                android.net.Uri$StringUri>
21:48:24      Random.nextLong                             {}  3231619530287989228
20.04.25      (): long
21:48:24      WifiManager.getConnectionInfo               {}  SSID: <unknown ssid>, BSSID:
20.04.25      (): WifiInfo                                02:00:00:00:00:00, MAC:
                                                02:00:00:00:00:00, IP: /10.0.2.16,
                                                Security type: 0, Supplicant state:
```

Что умеет прототип?

- Подключение к Android устройству и определенному процессу на нем
- Сохранение скриптов перехвата для будущего использования
- Динамическая загрузка любого количества скриптов
- Сохранение логов перехвата в ожидаемом формате в БД, формирование отчета

```
21:48:20      Uri.parse                                {'uriString': 'https://api19-va.tiktokv.com/tiktok/v1/realtime/feedback/?device_platform=android&os...  <instance: android.net.Uri, $className:
18.04.25      (String): uri                                android.net.Uri$StringUri>
21:48:21      Uri.parse                                {'uriString': 'https://v16m.tiktokcdn-us.com/368ae65247aa1527ee9ac2dae5c195b/68184a27/video/tos/us...  <instance: android.net.Uri, $className:
18.04.25      (String): uri                                android.net.Uri$StringUri>
21:48:24      Random.nextLong                                {}  3231619530287989228
18.04.25      () : long
21:48:24      WifiManager.getConnectionInfo                {}  SSID: <unknown ssid>, BSSID:
18.04.25      () : WifiInfo                                02:00:00:00:00:00, MAC:
                                                02:00:00:00:00:00, IP: /10.0.2.16,
                                                Security type: 0, Supplicant state:
```

Учли недостатки существующих инструментов

Инструмент был протестирован на реальном вредоносном приложении - AndroidRAT

Данная программа позволяет получить удаленный доступ к устройству через консоль. В консоли можно выполнять заранее заготовленные команды (получить фото с камеры, записать аудио)

Разработанный инструмент успешно перехватывает методы Android API, соответствующие командам

Разработан прототип инструмента для динамического анализа поведения ПО
Тестирование показало, что инструмент корректно работает и пригоден для
практического применения
В прототипе были устранены недостатки существующих решений

- Расширение функционала по сбору данных
- Внедрение автоматизированного анализа данных
- Защита от обнаружения инструментации

Конец