Universitas
Muhammadiyah
Sidoarjo

IMPLEMENTASI FRIDA FRAMEWOROK UNTUK MANIPULASI ALUR KERJA PADA APLIKASI ANDROID

Oleh :
ALDO REGHAN RAMADHAN
Progam Studi Informatika
Universitas Muhammadiyah Sidoarjo
2020

Latar Belakang

Android adalah sistem operasi mobile yang dikembangkan oleh Google dan pertama kali dirilis pada tahun 2008. Saat ini, Android merupakan sistem operasi mobile terpopuler di dunia dengan pangsa pasar lebih dari 70%.

SSL Pinning adalah sebuah teknik keamanan pada SSL/TLS yang digunakan untuk memastikan bahwa koneksi yang terjadi antara client dan server menggunakan sertifikat yang telah ditentukan sebelumnya, dan bukan sertifikat dari otoritas sertifikasi yang tidak dikenal atau dipercayai.

Root Android adalah proses untuk mendapatkan akses penuh atau hak akses superuser ke sistem operasi Android. Dalam kondisi normal, pengguna tidak memiliki akses penuh ke sistem Android dan hanya dapat melakukan operasi yang diizinkan oleh sistem operasi.

Frida adalah sebuah framework open-source yang digunakan untuk melakukan dynamic instrumentation (pemantauan dan modifikasi pada runtime) pada aplikasi di berbagai platform seperti Android, iOS, Windows, macOS, dan Linux.

Rumusan Dan Batasan Masalah

Rumusan Masalah :

1. Bagaimana cara bypass anti root pada aplikasi android menggunakan Frida framework?
2. Bagaimana cara implementasi Frida framework menggunakan javascript?

Batasan Masalah

1. Aplikasi yang digunakan dalam penelitian ini sudah melalui proses reverse engineering.
2. Memanipulasi function yang terdapat di aplikasi dengan javascript menggunakan Frida Framework.
3. Penelitian ini hanya menggunakan satu activity dalam penerapannya.
4. Penelitian ini hanya memanipulasi function yang menghasilkan nilai integer dan string.
5. Media untuk menjalankan program adalah menggunakan emulator

Tujuan Dan Manfaat

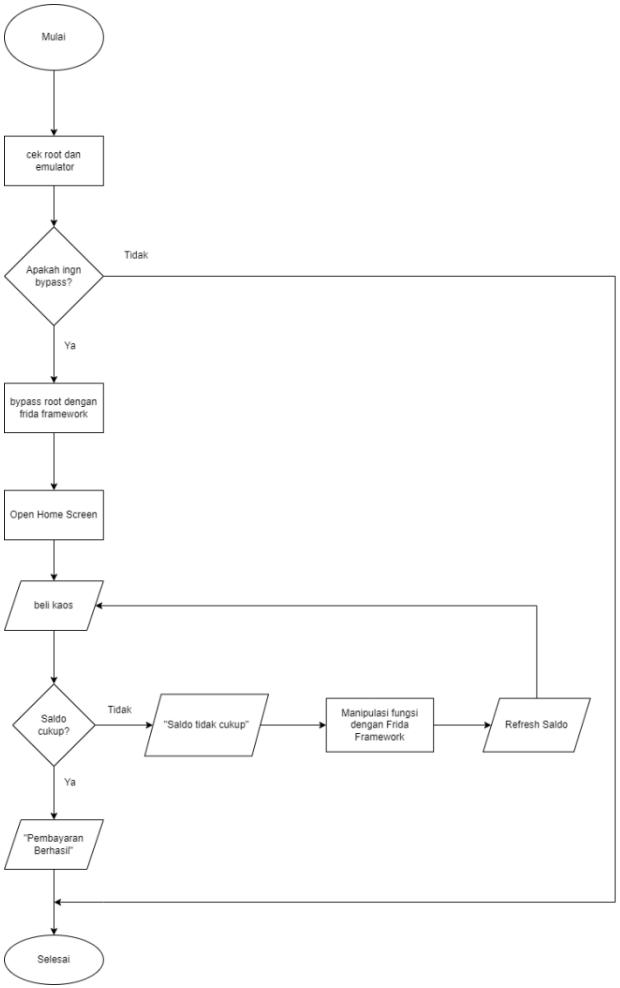
Tujuan Penelitian

1. Memanipulasi fungsi kotlin yang terdapat di aplikasi menggunakan javascript dengan Frida Framework.
2. Sebagai bentuk tambahan pengujian terhadap aplikasi android.

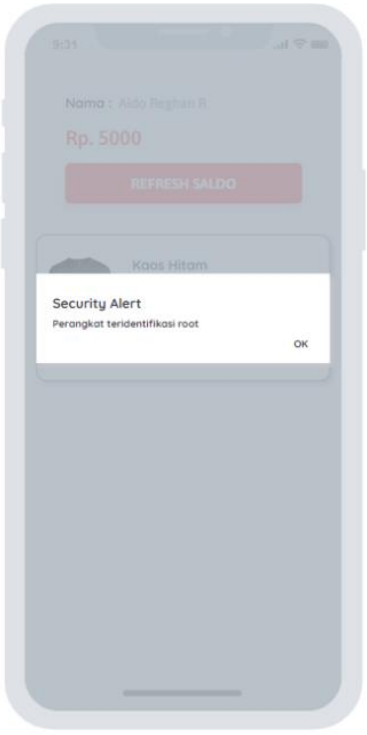
Manfaat Penelitian

1. Sebagai acuan untuk developer pentingnya dalam implementasi anti reverse pada saat mengcompile source code menjadi sebuah APK.
2. Sebagai pengingat untuk developer bahwa no system is safe tidak ada sistem yang aman pada setiap aplikasi yang telah dibuat.
3. Sebagai pengetesan lebih lanjut dalam security.

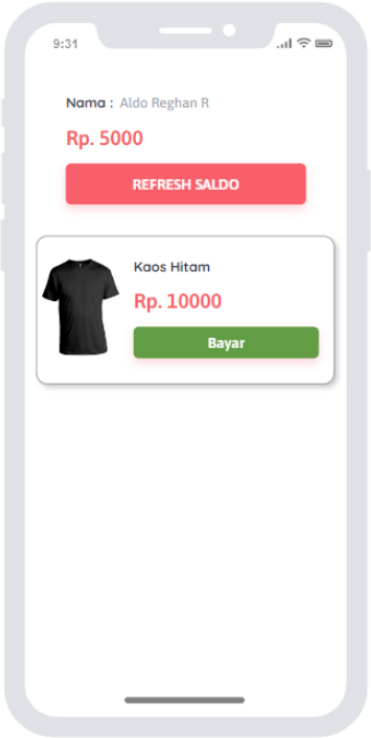
Flowchart Alur Penelitian



Desain Aplikasi



Aplikasi Terdeteksi root

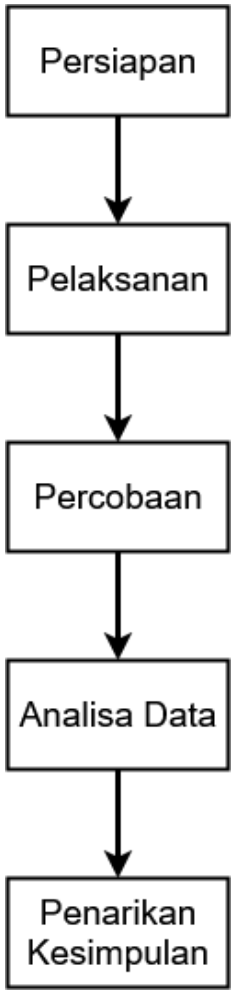


Home Screen Aplikasi

Pop Up Gagal Transaksi

Pop Up Sukses Transaksi

Metode

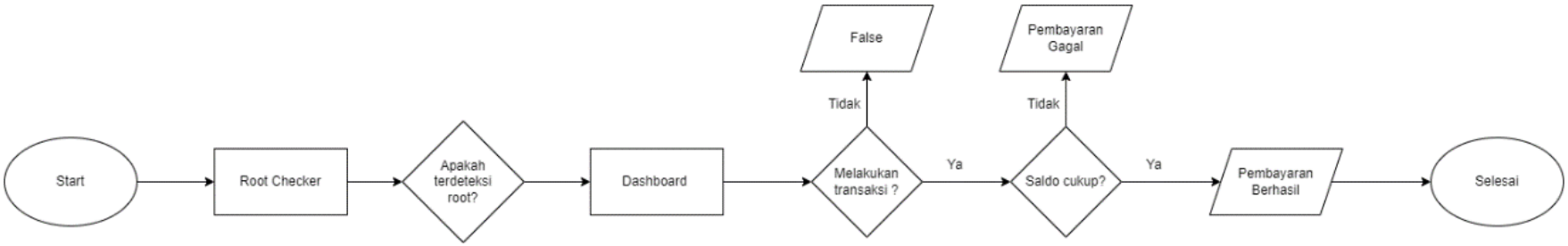

Persiapan
Pelaksanan
Percobaan
Analisa Data
Penarikan Kesimpulan

Tahapan Persiapan

- Penulis menyiapkan aplikasi hasil reverse engineering
- Penulis menulis exploit untuk mengubah alur aplikasi

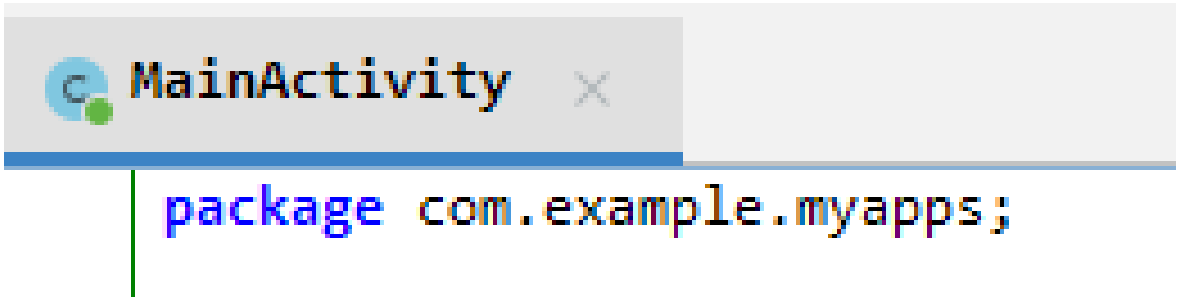
Tahapan Pelaksanaan

Identifikasi normal flow



Tahapan Pelaksanaan

Identifikasi package



Tahapan Pelaksanaan

Identifikasi library anti root

MainActivity

```
            inflate = null;
        }
        setContentView(inflate.getRoot());
22      RootBeer rootBeer = new RootBeer(this);
23      boolean isRooted = rootBeer.isRooted();
24      String PRODUCT = Build.PRODUCT;
        Intrinsics.checkNotNullExpressionValue(PRODUCT, "PRODUCT");
        boolean isEmulator = StringsKt.contains$default((CharSequence) PRODUCT, (CharSequence) "sdk", false, 2, (Object) null);
25      AlertDialog.Builder builder = new AlertDialog.Builder(this);
26      builder.setTitle("Security Alert");
27      builder.setCancelable(false);
28      builder.setPositiveButton("OK", new DialogInterface.OnClickListener() { // from class: com.example.myapps.MainActivity$$ExternalSyntheticLambda0
            @Override // android.content.DialogInterface.OnClickListener
            public final void onClick(DialogInterface dialogInterface, int i) {
                MainActivity.m26onCreate$lambda0(MainActivity.this, dialogInterface, i);
            }
        });
29      if (!isRooted || !isEmulator) {
20          if (!Intrinsics.areEqual("Aldo", "Aldo")) {
39              builder.setMessage("Aplikasi sudah dimodif");
40              builder.show();
            }
30      } else if (isRooted) {
31          builder.setMessage("Perangkat teridetifikasi root");
32          builder.show();
        } else {
34          builder.setMessage("Perangkat teridetifikasi emulator");
35          builder.show();
        }
```

Tahapan Pelaksanaan

Identifikasi function

MainActivity

```
47      public static final void m27onCreate$lambda1(MainActivity this$0, View it) {
            Intrinsics.checkNotNullParameter(this$0, "this$0");
48          ActivityMainBinding activityMainBinding = this$0.binding;
            ActivityMainBinding activityMainBinding2 = null;
            if (activityMainBinding == null) {
                Intrinsics.throwUninitializedPropertyAccessException("binding");
                activityMainBinding = null;
            }
            activityMainBinding.tvVersion.setText(this$0.showString("Aldo"));
49          ActivityMainBinding activityMainBinding3 = this$0.binding;
            if (activityMainBinding3 == null) {
                Intrinsics.throwUninitializedPropertyAccessException("binding");
            } else {
                activityMainBinding2 = activityMainBinding3;
            }
            activityMainBinding2.tvNumber.setText(String.valueOf(this$0.numberData(2500, 2500)));
        }
```

Tahapan Percobaan

Menjalankan frida server yang terdapat di emulator

```
E:\>adb root
adbd is already running as root

E:\>adb shell
generic_x86:/ # cd /data/local/tmp
generic_x86:/data/local/tmp # ./frida-server
```

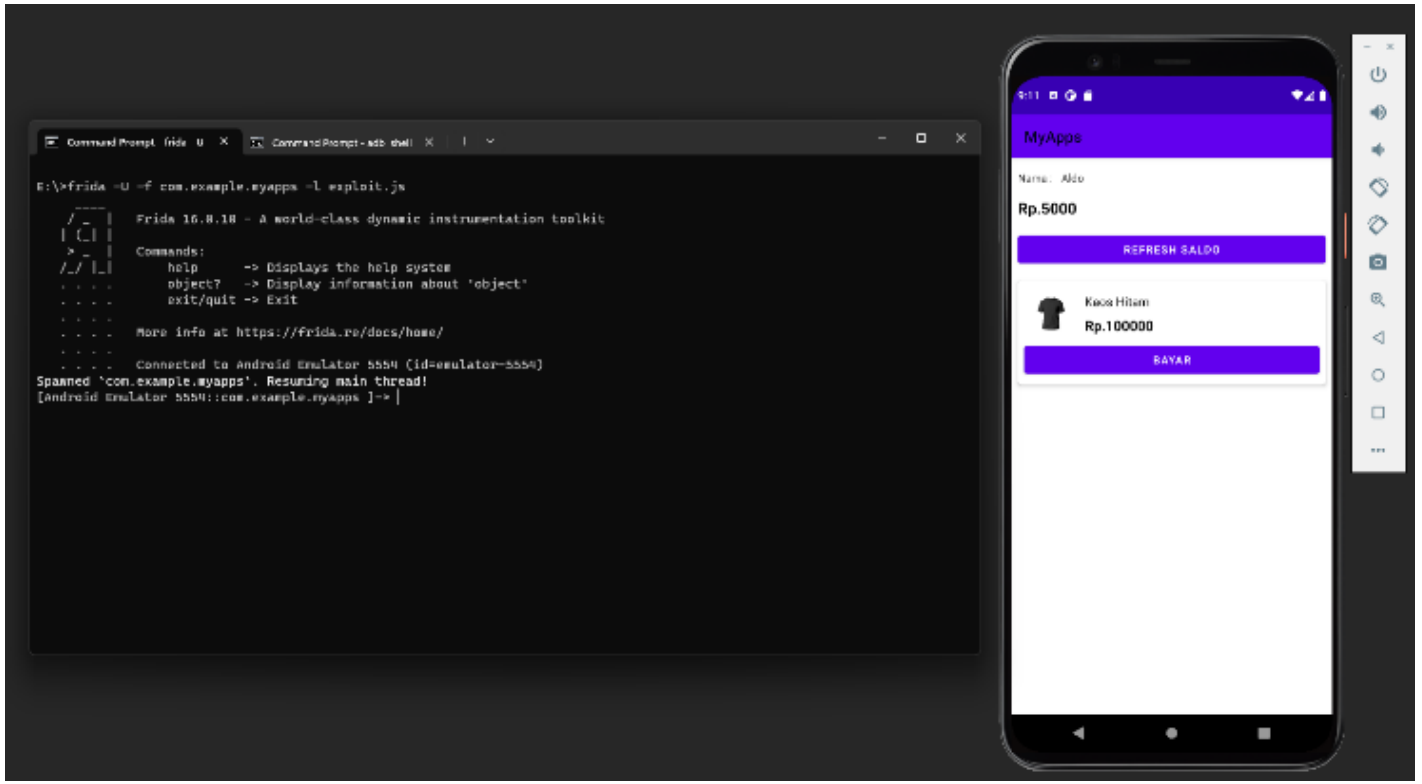
Tahapan Percobaan

Melihat list aplikasi

```
E:\>frida-ps -U
 PID  Name
----  ------------------------------------------------
8339  Calendar
8387  Clock
2303  Gallery
8205  MyApps
1865  Phone
```

Tahapan Percobaan

Menjalankan exploit



Hasil dan Pembahasan

- Jurnal ini menginvestigasi penggunaan Frida Framework dalam memanipulasi alur kerja pada aplikasi Android. Hasil penelitian menunjukkan bahwa implementasi anti root tidak cukup untuk mengamankan sebuah aplikasi android itu dibuktikan, uji coba peneliti berhasil membypass library anti root.
- Pembahasan jurnal ini menekankan pentingnya pengujian lebih lanjut pada berbagai jenis aplikasi untuk mengukur ketersediaan dan kehandalan Frida Framework. Selain itu, keterbatasan potensi risiko dalam penggunaan Frida juga dibahas, termasuk potensi penyalahgunaan dan dampaknya pada pengalaman end user.

Kesimpulan

Dari hasil penelitian dan pembahasan tentang “Implementasi Frida Framework untuk Manipulasi Alur Kerja pada Aplikasi Android”, maka dapat diambil kesimpulan bahwa :

1. Pada versi os android terbaru tidak menjamin bahwa aplikasi akan aman terhadap Frida.
2. Frida Framework merupakan alat yang kuat dan efektif dalam melakukan manipulasi alur kerja pada aplikasi Android secara dinamis pada saat runtime
3. Implementasi Frida memungkinkan pengguna untuk melakukan intersepsi dan modifikasi pada kode aplikasi tanpa perlu melakukan reverse engineering atau memodifikasi file APK secara manual.
4. Dengan menggunakan Frida, peneliti dapat memanipulasi input, output, atau perilaku fungsi tertentu pada aplikasi Android, sehingga memungkinkan pengembangan fitur tambahan atau pengujian keamanan yang lebih efektif.
5. Developer lebih memperhatikan penggunaaan library anti root yang menjadi fokus agar aplikasi tidak bisa dimodifikasi tetap bisa di bypass


UMSIDA
DARI SINI PENCERAHAN BERSEMI