BIFUZ – Broadcast Intent FUZzing Framework for Android

intel Software

Intel OpenSource TECHNOLOGY CENTER

FOSDEM '15

Brussels
31 Jan & 1 Feb 2015

Andreea Brînduşa Proca

Răzvan-Costin Ionescu

Agenda

Why do we need BIFUZ?

What is BIFUZ?

BIFUZ's Architecture

Walk-through

Results

Conclusions

Why do we need BIFUZ?

Android Security

Intent Fuzzing

Important Target

Android Apps

Broadcast Intents

What is BIFUZ?

Broadcast Intent FUZzing Framework for Android

Python

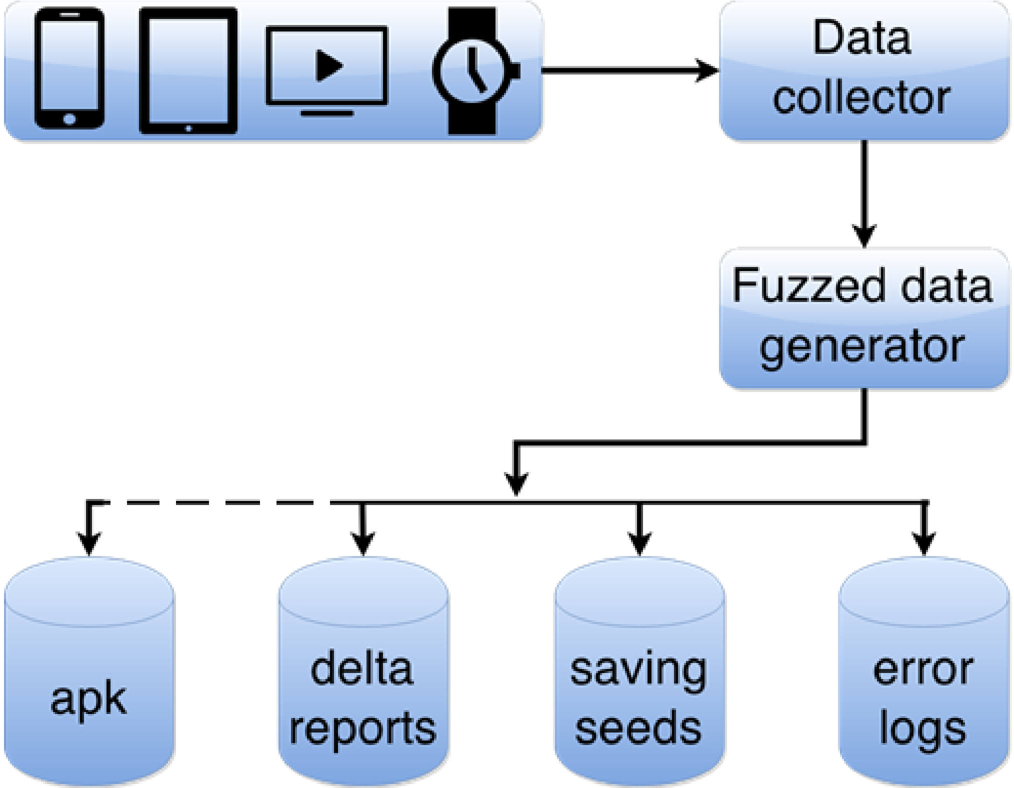
Bugs

Broadcast / Fuzzed Intents

Negative Testing

Open Source

BIFUZ's Architecture



Data collector

Fuzzed data generator

apk

delta reports

saving seeds

error logs

Walk-through

BIFUZ’s Menu Options

```
= = = = = = = = = = = = = = = = =
    ###   #  ####  #  #  ####
    #  #  #  #     #  #    ##
    ###   #  ####  #  #   ##
    #  #  #  #     #  #  #
    ###   #  #     ####  ####
= = = = = = = = = = = = = = = = =
Select one option from below
 1. Select Devices Under Test
 2. Generate Fuzzed Intent calls
 3. Generate Broadcast Intent calls for the DUT(s)
 4. Generate a delta report between 2 fuzzing sessions
 5. Run existing generated intents from file
 6. (Future) Generate apks for specific Intent calls
 Q. Quit
```

Walk-through

Fuzzed Intent Example

```
Generate broadcast intent calls for the following DUT(s): 12qw34er56ty78ui
Insert the packages wanted or type 'all' for all packages: earth, calendar
Device 12qw34er56ty78ui: Insert the name of the logs folder: FOLDER_NAME

adb -s 12qw34er56ty78ui shell am start
-a android.intent.action.VIEW
-c android.intent.category.BROWSABLE
-n com.google.earth/com.google.earth.EarthActivity
-f  0x00400000
-d http://YIV6HT9RKSNRCYDGCA6ONAX2Z0M3E3PXZI4W09VZEMA2G03KK0LNIAJ15911OAA.com
-e boolean android.intent.extra.ALARM_COUNT True
```

Walk-through

Broadcast Intent Example

```
Select one option from below
    1. Select Devices Under Test
    2. Generate Fuzzed Intent calls
    3. Generate Broadcast Intent calls for the DUT(s)
    4. Generate a delta report between 2 fuzzing sessions
    5. Run existing generated intents from file
    6. (Future) Generate apks for specific Intent calls
    Q. Quit
  Insert your choice:   3

adb -s 12qw34er56ty78ui shell am broadcast
-n com.google.earth/com.google.analytics.tracking.android.CampaignTrackingReceiver
```

Walk-through

Error Log Example

```
--------- beginning of main
F/BIFUZ_BROADCAST( 9395): adb -s 12qw34er56ty78ui shell am broadcast -n
com.google.earth/com.google.analytics.tracking.android.CampaignTrackingReceiver.
--------- beginning of system
I/ActivityManager( 3056): Start proc com.google.earth for broadcast
com.google.earth/com.google.analytics.tracking.android.CampaignTrackingReceiver: pid=9411 uid=10049 gids={50049, 9997,
3003, 1028, 1015} abi=x86
--------- beginning of crash
E/AndroidRuntime( 9411): FATAL EXCEPTION: main
E/AndroidRuntime( 9411): Process: com.google.earth, PID: 9411
E/AndroidRuntime( 9411): java.lang.RuntimeException: Unable to instantiate receiver
com.google.analytics.tracking.android.CampaignTrackingReceiver: java.lang.ClassNotFoundException: Didn't find class
"com.google.analytics.tracking.android.CampaignTrackingReceiver" on path: DexPathList[[zip file
"/system/app/GoogleEarth/GoogleEarth.apk"],nativeLibraryDirectories=[/system/app/GoogleEarth/lib/x86, /vendor/lib, /system/lib]]
```

Results

javaNullPointerException

javaClassNotFoundException

DoS attack

Buffer Overflow

SQL injection

Conclusions

- BIFUZ is an open source testing tool
- easy setup
- assess if an application is more stable than another from security perspective
- bugs might be sent to Google for verification
- reproducibility and debugging

Source code: https://github.com/fuzzing/bifuz

You may find us @:

andreea.brindusa.proca@intel.com
razvan.ionescu@intel.com