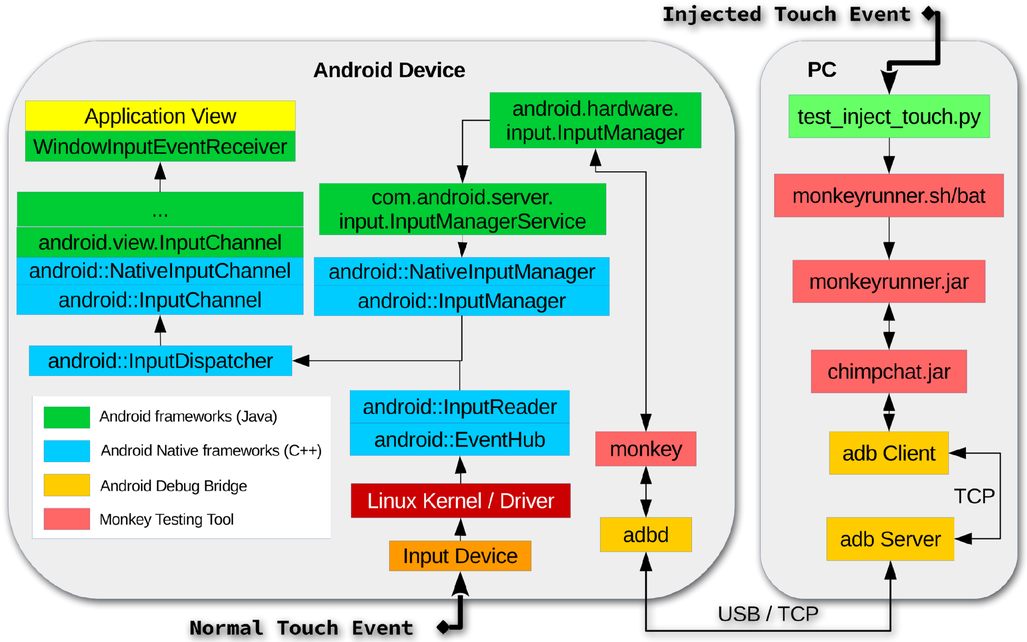
Fuzz Testing on Android

Guanqing Yan, Tianshuo Su

Intro

Fuzz testing was originally presented as a class project of CS736 in 1988. The idea is to test the reliability of applications by feeding them random generated inputs. The methodology not only has potentials to reveal bugs that may not be discovered from normal testing, but also is easy to be setup and run.

Ever since fuzz testing has first been proposed, it has been conducted on Unix shells, X-Window, MacOS and Windows NT. In this study, we experimented fuzz testing on Android, one of the dominating mobile platforms today.



Fg. 1 The path for normal and injected touch event

Discussion

Fuzz testing presented a number of challenges on Android.

1. Many apps have social features that allows a user to connect to other users. Traditional fuzz testing may send garbage and cause disturbances to real users.
2. The apps are usually a lot more stateful and less deterministic than traditional command line utilities. This reduces the reproducibility of test results.
3. All the apps we tested are proprietary, this limits the amount of analysis we can do on the cause of bugs.

Mechanism

We selected 18 popular applications from the Google Play Store, For each application, 20 runs of 500 random touch events were conducted.

In each test run, we injected a random touch event into the application, and then queried the top most activity.

We used a rather simple criteria to detect if the app has crashed, which is when the top most activity becomes the home page.

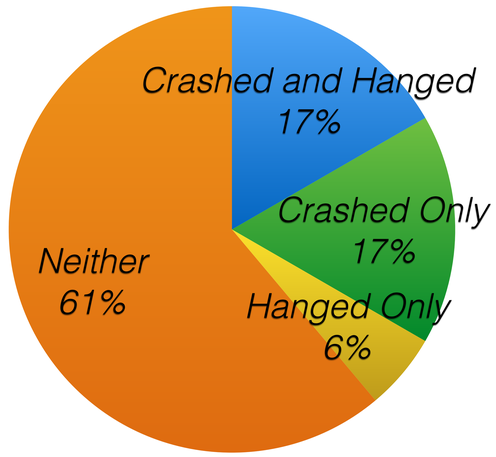
If the top most activity was not the homepage nor the app we were testing, we assumed that a redirection has occurred (e.g tapping on a link may take you to Chrome). In that case, we simply simulated a touch on the 'back' button.

If the injected touch event blocked for more than a predefined threshold (4.5s in our experiment), we categorized it as App Not Responding (ANR, equivalent of hanging).

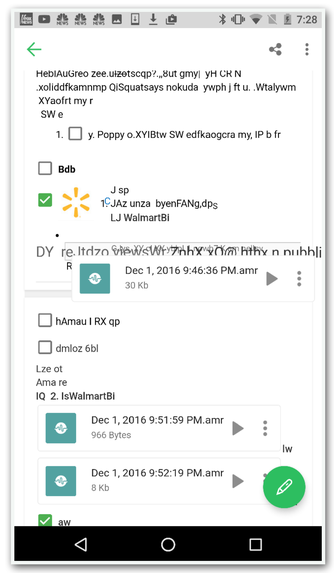
```
for i in range(NUM_RUNS):
  start_app()
  seed_random(i)
  last_touch_time = current_time
  for j in range(NUM_TOUCHES):
    last_touch_time = current_time()
    # may block
    device.generate_random_touch_event()

    if current_time() - last_touch_time > ANR_LIMIT:
      handle_anr()
      break
    if top_activity() == HOME_PAGE:
      handle_crash()
      break
    if top_activity() != app_being_tested:
      # redirection happened, get back to
      # the app we are testing
      device.press_back_button()
  stop_app()
```

Result



Fg. 2 Experiment Result



Fg.3 Evernote under fuzz testing