

Id : fd-114

Result : False Positive

Sink statement : Line 504 in below screenshot is the sink statement.

```
502 public void release() {
503     if (!alarmids.isEmpty())
504         Log.w(TAG, msg: "Releasing wake lock with active alarms! (" +
505             alarmids.size() + ")");
506     Log.i(TAG, msg: "Releasing wake lock");
507     wakelock.release();
508     wakelock = null;
509
510     handler.removeCallbacks(timeout);
511     handler.removeMessages(ActiveAlarms.TRIGGER_INC);
512     handler.sendMessage(ActiveAlarms.RESET_VOLUME);
513     handler = null;
514
515     if (player.isPlaying())
516         player.stop();
517     player.reset();
518     player.release();
519     player = null;
520 }
521 }
```

The sink statement gets variable `alarmids.size()`. Though the source results in adding to the `alarmids` list, the sink statement is getting the size of the list and thus does not contain the data.

Source Statement : Line 83 in following screenshot

```
80
81 @ private List<Alarm> enabledAlarms(ContentResolver r) {
82     LinkedList<Alarm> ids = new LinkedList<>();
83     Cursor c = r.query(
84         AlarmClockProvider.ALARMS_URI,
85         new String[] {
86             AlarmClockProvider.AlarmEntry._ID,
87             AlarmClockProvider.AlarmEntry.TIME,
88             AlarmClockProvider.AlarmEntry.DAY_OF_WEEK,
89             AlarmClockProvider.AlarmEntry.NEXT_SNOOZE },
90         selection: AlarmClockProvider.AlarmEntry.ENABLED + " == 1",
91         selectionArgs: null, sortOrder: null);
92     while (c.moveToNext())
93         ids.add(new Alarm(c));
94     c.close();
95     return ids;
96 }
```