

Id : fd-63

Result : False Positive

Sink statement : Line 327 in below screenshot is the sink statement.

```
318 private void refreshNotifyBar() {
319     final NotificationManager manager =
320         (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
321     // Recursive trigger.
322     final PendingIntent tick = PendingIntent.getService(
323         context: this, requestCode: 0, new Intent( packageContext: this, AlarmNotificationService.class)
324         .putExtra(AlarmNotificationService.COMMAND,
325             AlarmNotificationService.REFRESH), flags: 0);
326
327     final Cursor c = getContentResolver().query(
328         AlarmClockProvider.ALARMS_URI,
329         new String[] { AlarmClockProvider.AlarmEntry.TIME,
330             AlarmClockProvider.AlarmEntry.ENABLED,
331             AlarmClockProvider.AlarmEntry.NAME,
332             AlarmClockProvider.AlarmEntry.DAY_OF_WEEK,
333             AlarmClockProvider.AlarmEntry.NEXT_SNOOZE },
334         selection: AlarmClockProvider.AlarmEntry.ENABLED + " == 1",
335         selectionArgs: null, sortOrder: null);
336 }
```

All the variables to the sink statement are constants.

```
234
235     public static final String TIME = "time";
236     public static final String ENABLED = "enabled";
237     public static final String NAME = "name";
238     public static final String DAY_OF_WEEK = "dow";
239     public static final String NEXT_SNOOZE = "next_snooze";
240 }
```

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 }
```