

## Id : fd-43

Result : False Positive

Sink statement : Line 60 in below screenshot is the sink statement.

```
54 @ public static long newAlarm(Context c, int secondsPastMidnight) {
55
56     ContentValues v = new ContentValues();
57     v.put(AlarmClockProvider.AlarmEntry.TIME, secondsPastMidnight);
58     Uri u = c.getContentResolver().insert(AlarmClockProvider.ALARMS_URI, v);
59     long alarmid = ContentUris.parseId(u);
60     Log.i(TAG, msg: "New alarm: " + alarmid + " (" + u + ")");
61
62     // Inserted entry is ENABLED by default with no options. Schedule the
63     // first occurrence.
64     Calendar ts = TimeUtil.nextOccurrence(secondsPastMidnight, repeat: 0);
65     scheduleAlarmTrigger(c, alarmid, ts.getTimeInMillis());
66
67     return alarmid;
68 }
69
```

Both u and alarmid variables to the sink statement are generated at line 58 and 58 respectively and hence are not coming from the given source statement.

Source Statement : Line 54 in following screenshot

```
49
50 @ public Alarm(Cursor c) {
51     time = c.getInt(c.getColumnIndex(AlarmClockProvider.AlarmEntry.TIME));
52     enabled = c.getInt(c.getColumnIndex(
53         AlarmClockProvider.AlarmEntry.ENABLED)) != 0;
54     label = c.getString(c.getColumnIndex(AlarmClockProvider.AlarmEntry.NAME));
55     repeat = c.getInt(c.getColumnIndex(
56         AlarmClockProvider.AlarmEntry.DAY_OF_WEEK));
57     next_snooze = c.getLong(c.getColumnIndex(
58         AlarmClockProvider.AlarmEntry.NEXT_SNOOZE));
59 }
60
```