

**Id : fd-0**

Result : False Positive

Reason :

The source statement is **line number 33** in file DBUtil.java shown in following screenshot

```
30
31 @ public static Alarm get(Context context, long id) {
32     Alarm s = null;
33     final Cursor c = context.getContentResolver().query(
34         ContentUris.withAppendedId(AlarmClockProvider.ALARMS_URI, id),
35         new String[] {
36             AlarmClockProvider.AlarmEntry.TIME,
37             AlarmClockProvider.AlarmEntry.ENABLED,
38             AlarmClockProvider.AlarmEntry.NAME,
39             AlarmClockProvider.AlarmEntry.DAY_OF_WEEK,
40             AlarmClockProvider.AlarmEntry.NEXT_SNOOZE },
41         selection: null, selectionArgs: null, sortOrder: null);
42     if (c.moveToFirst())
43         s = new Alarm(c);
44     else
45         s = new Alarm();
46     c.close();
47     return s;
48 }
```

The sink statement is **line number 107** in file AlarmClockActivity.java as shown in the following screenshot

```
99
100 @Override
101 public void onItemClick(AdapterView<?> parent, View v, int x, long id) {
102     boolean check = ((CheckBox)v.findViewById(R.id.enabled)).isChecked();
103
104     ContentValues val = new ContentValues();
105     val.put(AlarmClockProvider.AlarmEntry.ENABLED, !check);
106     val.put(AlarmClockProvider.AlarmEntry.NEXT_SNOOZE, 0);
107     getContentResolver().update(
108         ContentUris.withAppendedId(AlarmClockProvider.ALARMS_URI, id),
109         val, where: null, selectionArgs: null);
110
111     if (check) {
112         AlarmNotificationService.removeAlarmTrigger(
113             getApplicationContext(), id);
114     } else {
115         DbUtil.Alarm a = DbUtil.Alarm.get(getApplicationContext(), id);
116         long nextUTC = TimeUtil.nextOccurrence(a.time, a.repeat)
117             .getTimeInMillis();
118         AlarmNotificationService.scheduleAlarmTrigger(
119             getApplicationContext(), id, nextUTC);
120     }
121 }
122 };
```

The sink statement (line 107) takes 2 variables, id and val.

The variable val is generated locally in line numbers 104,105 and 106 which use variable check that is a user input and a constant value 0. Hence the variable val is not tainted.

The variable id is an input to the function onItemClick.

Attached file fd-0\_sink.draw.io shows that the variable id is read from the database and ALARMS-TABLE\_OPERATIONS.draw.io shows that the source statement does not write the id column to the database.