Id : ds-152

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 59 respectively and hence are not coming from the given source statement.

Source statement : Line 149 in following screenshot

```
147
148  @      public static String until(Context c, Calendar from, Calendar to) {
149             long minutes = (to.getTimeInMillis() - from.getTimeInMillis()) / 1000 / 60;
150             long days = minutes / 1440;
151             minutes -= (days * 1440);
152             long hours = minutes / 60;
153             minutes -= (hours * 60);
154
155             String s = "";
156             if (days > 0)
157               s += days > 1 ? "{days} days" + " " :
158                  "{days} day" + " ";
159             if (hours > 0)
160               s += hours > 1 ? "{hours} hours" + " " :
161                  "{hours} hour" + " ";
162             if (minutes > 0)
163               s += minutes > 1 ? "{minutes} minutes" + " " :
164                  "{minutes} minute" + " ";
165             return s;
166      }
167
```