Id : fd-64

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 128,130 in following screenshot

```
126
127 @       private Settings(Cursor c) {
128           tone_url = Uri.parse(c.getString(c.getColumnIndex(
129               AlarmClockProvider.SettingsEntry.TONE_URL)));
130           tone_name = c.getString(c.getColumnIndex(
131               AlarmClockProvider.SettingsEntry.TONE_NAME));
132           snooze = c.getInt(c.getColumnIndex(
133               AlarmClockProvider.SettingsEntry.SNOOZE));
134           vibrate = c.getInt(c.getColumnIndex(
135               AlarmClockProvider.SettingsEntry.VIBRATE)) != 0;
136           volume_starting = c.getInt(c.getColumnIndex(
137               AlarmClockProvider.SettingsEntry.VOLUME_STARTING));
138           volume_ending = c.getInt(c.getColumnIndex(
139               AlarmClockProvider.SettingsEntry.VOLUME_ENDING));
140           volume_time = c.getInt(c.getColumnIndex(
141               AlarmClockProvider.SettingsEntry.VOLUME_TIME));
142       }
143
```