

**Id : ds-1501**

Cross ref : 1499

Result : Unknown

Reason :

Sink statement : Line 590 in the screenshot below

```
577  @Override
578  public void onTextChanged(CharSequence s, int start, int before, int count) {
579      String number = s.toString();
580      if (start == 0 && before == 0 && count == 0) {
581          return;
582      } else if (TextUtils.isEmpty(s) && before > 0) {
583          setCallLogMode();
584          contactsEntryAdapter.resetFilter();
585          list.setSelection(0);
586      } else if (number.equals("#06#")) {
587          showDeviceId();
588      } else if (number.startsWith("#*#") && number.endsWith("#*#")) {
589          String secretCode = new StringBuilder(number).substring(4, number.length()-4);
590          sendBroadcast(new Intent( action: "android.provider.Telephony.SECRET_CODE",
591                                  Uri.parse("android_secret_code://" + secretCode)));
592      } else {
593          setContactsMode();
594          contactsEntryAdapter.getFilter().filter(s);
595      }
596  }
```

Source Statement : Line 114 in the below screenshot

```
108
109  @Override
110  public boolean onItemLongClick(AdapterView<?> adapterView, View view, int position, long id) {
111      if (position == 0) {
112          return false;
113      }
114      String order = ((TextView)view.findViewById(R.id.entry_order)).getText().toString();
115      if (SpeedDial.getNumber( context: this, order).length() != 0) {
116          String speedDialData = ((TextView)view.findViewById(R.id.entry_title)).getText().toString();
117          clearSpeedDialSlotDialog(order, speedDialData);
118          return true;
119      }
120      return false;
121  }
```

The graph ds-1499\_sink.drawio shows the sources of variables passed to the sink statement, the graph comes to point where there is unknown end to predict the exact source of the variable phone number.