Half-Lib Writeup

Target:

> 02619b644e44542baf3a67d1264af85e libhalflib.so

Exploitation:

1. Analyze binary

Open `libhalflib.so` binary in IDA or any other disassembler.
See export table and spot functions `Java_com_sctf2019_halflib_SignInActivity_foo`, `Java_com_sctf2019_halflib_HalfLib_nativeDecrypt` and others which apply JNI naming convention.
So we are dealing with Android native library.

Decompilation and refactoring produces the following:

```
void __fastcall Java_com_sctf2019_halflib_SignInActivity_foo(JNIEnv *env,
jobject *type)
{
  struct _jmethodID *v2; // x0
  jobject v3; // [xsp+10h] [xbp-3D0h]
  jobject v4; // [xsp+38h] [xbp-3A8h]
  jclass v5; // [xsp+40h] [xbp-3A0h]
  struct _jmethodID *v6; // [xsp+50h] [xbp-390h]
  jobject v7; // [xsp+58h] [xbp-388h]
  jstring v8; // [xsp+68h] [xbp-378h]
  struct _jmethodID *v9; // [xsp+70h] [xbp-370h]
  jobject v10; // [xsp+78h] [xbp-368h]
  struct _jmethodID *v11; // [xsp+88h] [xbp-358h]
  jclass v12; // [xsp+90h] [xbp-350h]
  struct _jmethodID *v13; // [xsp+A0h] [xbp-340h]
  jclass v14; // [xsp+A8h] [xbp-338h]
  struct _jmethodID *v15; // [xsp+B8h] [xbp-328h]
  struct _jmethodID *v16; // [xsp+D0h] [xbp-310h]
  struct _jmethodID *v17; // [xsp+E8h] [xbp-2F8h]
  struct _jfieldID *v18; // [xsp+100h] [xbp-2E0h]
  struct _jmethodID *v19; // [xsp+288h] [xbp-158h]
  jclass v20; // [xsp+290h] [xbp-150h]
  char *v21; // [xsp+298h] [xbp-148h]
  jobject v22; // [xsp+2A0h] [xbp-140h]
  char *v23; // [xsp+2A8h] [xbp-138h]
  jobject v24; // [xsp+2B0h] [xbp-130h]
  struct _jmethodID *v25; // [xsp+2B8h] [xbp-128h]
  struct _jmethodID *v26; // [xsp+2C8h] [xbp-118h]
  struct _jmethodID *v27; // [xsp+2D0h] [xbp-110h]
  jclass v28; // [xsp+2D8h] [xbp-108h]
  jobject v29; // [xsp+2E0h] [xbp-100h]
  jobjectArray v30; // [xsp+2E8h] [xbp-F8h]
  struct _jmethodID *v31; // [xsp+2F0h] [xbp-F0h]
  jobject v32; // [xsp+2F8h] [xbp-E8h]
  jobject v33; // [xsp+318h] [xbp-C8h]
  jbyteArray v34; // [xsp+328h] [xbp-B8h]
  jobject v35; // [xsp+330h] [xbp-B0h]
  jclass v36; // [xsp+338h] [xbp-A8h]
  const char *v37; // [xsp+340h] [xbp-A0h]
  jobject v38; // [xsp+348h] [xbp-98h]
  jclass v39; // [xsp+350h] [xbp-90h]
  jobject v40; // [xsp+358h] [xbp-88h]
  jclass v41; // [xsp+360h] [xbp-80h]
  jobject v42; // [xsp+368h] [xbp-78h]
  jclass v43; // [xsp+370h] [xbp-70h]

  __android_log_print(3LL, "HalfLib", "foo()");
  v43 = FindClass(env, "com/sctf2019/halflib/SignInActivity");
  v18 = GetFieldID(env, v43, "loginEdit", "Landroid/widget/EditText;");
  v42 = GetObjectField(env, type, v18);
  v41 = FindClass(env, "android/widget/EditText");
  v17 = GetMethodID(env, v41, "getText", "()Landroid/text/Editable;");
  v40 = CallObjectMethodV(env, v42, v17);
  v39 = FindClass(env, "java/lang/CharSequence");
  v16 = GetMethodID(env, v39, "toString", "()Ljava/lang/String;");
  v38 = CallObjectMethodV(env, v40, v16);
  v37 = GetStringUTFChars(env, v38, 0LL);
  v36 = FindClass(env, "android/content/Context");
  v15 = GetMethodID(env, v36, "getClassLoader", "()Ljava/lang/ClassLoader;");
  v35 = CallObjectMethodV(env, type, v15);
  v34 = NewByteArray(env, 4252);
  SetByteArrayRegion(env, v34, 0, 4252, PAYLOAD);
  v14 = FindClass(env, "java/nio/ByteBuffer");
  v13 = GetStaticMethodID(env, v14, "wrap", "([B)Ljava/nio/ByteBuffer;");
  v33 = CallStaticObjectMethodV(env, v14, v13, v34);
  v12 = FindClass(env, "dalvik/system/InMemoryDexClassLoader");
  v11 = GetMethodID(env, v12, "<init>", "(Ljava/nio/ByteBuffer;Ljava/lang/ClassLoader;)V");
  v10 = NewObjectV(env, v12, v11, v33, v35);
  v9 = GetMethodID(env, v12, "loadClass", "(Ljava/lang/String;)Ljava/lang/Class;");
  v8 = NewStringUTF(env, "com.sctf2019.halflib.UsersProvider");
  v7 = CallObjectMethodV(env, v10, v9, v8);
  v6 = GetMethodID(env, v7, "<init>", "(Landroid/content/Context;)V");
  v32 = NewObjectV(env, v7, v6, type);
  v31 = GetMethodID(
          env,
          v7,
          "query",
          "(Landroid/net/Uri;[Ljava/lang/String;Ljava/lang/String;[Ljava/lang/String;Ljava/lang/String;)Landroid/database/Cursor;");
  v5 = FindClass(env, "java/lang/String");
  v4 = NewStringUTF(env, v37);
  v30 = NewObjectArray(env, 1, v5, v4);
  v29 = CallObjectMethodV(env, v32, v31, 0LL, 0LL, 0LL, v30, 0LL);
  if ( v29 != 0LL )
  {
    v28 = FindClass(env, "android/database/Cursor");
    v27 = GetMethodID(env, v28, "getCount", "()I");
    v26 = GetMethodID(env, v28, "moveToFirst", "()Z");
    if ( CallIntMethodV(env, v29, v27) == 1 && CallBooleanMethodV(env, v29, v26) != 0 )
    {
      v25 = GetMethodID(env, v28, "getString", "(I)Ljava/lang/String;");
      v24 = CallObjectMethodV(env, v29, v25, 1LL);
      v23 = (char *)GetStringUTFChars(env, v24, 0LL);
      v22 = CallObjectMethodV(env, v29, v25, 2LL);
      v21 = (char *)GetStringUTFChars(env, v22, 0LL);
      v20 = FindClass(env, "android/widget/Toast");
      v19 = GetStaticMethodID(
              env,
              v20,
              "makeText",
              "
(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/Toast;");
      v3 = CallStaticObjectMethodV(env, v20, v19, type, v22, 1LL);
      v2 = GetMethodID(env, v20, "show", "()V");
      CallVoidMethodV(env, v3, v2);
      __android_log_print(3LL, "HalfLib", &unk_2B79F);
      ReleaseStringUTFChars(env, v24, v23);
      ReleaseStringUTFChars(env, v22, v21);
      ReleaseStringUTFChars(env, v38, v37);
    }
  }
}
```

Function retrieves login from UI `EditText` element, creates `InMemoryDexClassLoader` and loads bytes at offset `.rodata:000000000002A3F7 PAYLOAD         DCB 0x64, 0x65, 0x78, 0xA, 0x30, 0x33, 0x35, 0, 0x1F, 0xAA, ...`

Then it invokes `query` function on loaded class, receives `Cursor` object,
retrieves username and password and, finally, makes popup notification with the password in text.

2. Payload

Extract `PAYLOAD` data and decompile it (e.g. JADx https://github.com/skylot/jadx):

```
package com.sctf2019.halflib;

import android.content.ContentProvider;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.CursorWrapper;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.net.Uri;
import android.util.Log;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

public class UsersProvider extends ContentProvider {
    /* access modifiers changed from: private */
    public static final String TAG = UsersProvider.class.getSimpleName();

    /* renamed from: db */
    private SQLiteDatabase f0db;

    public UsersProvider(Context context) {
        C00001 r0 = new SQLiteOpenHelper(context,
HalfLib.getInstance().getTableName(), null, 1) {
            public void onCreate(SQLiteDatabase db) {
                Log.i(UsersProvider.TAG, "DbHelper: onCreate");
                HalfLib.getInstance().onCreate(db);
            }

            public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
                Log.i(UsersProvider.TAG, "DbHelper: onUpgrade");
                HalfLib.getInstance().onUpgrade(db);
            }
        };
        this.f0db = r0.getWritableDatabase();
    }

    public boolean onCreate() {
        return true;
    }

    public Cursor query(Uri uri, String[] projection, String selection, String[]
args, String sortOrder) {
        String str = TAG;
        StringBuilder sb = new StringBuilder();
        sb.append("query: ");
        sb.append(args[0]);
        Log.i(str, sb.toString());
        final Cursor cursor = HalfLib.getInstance().query(this.f0db, args[0]);
        return new CursorWrapper(cursor) {
            public String[] getColumnNames() {
                return new String[]{"id", "username", "password"};
            }

            public String getString(int column) {
                if (column != 2) {
                    return cursor.getString(column);
                }
                return HalfLib.getInstance().decrypt(cursor.getString(1),
cursor.getString(2));
            }
        };
    }

    @Nullable
    public String getType(@NonNull Uri uri) {
        return null;
    }

    public Uri insert(Uri uri, ContentValues values) {
        String str = TAG;
        StringBuilder sb = new StringBuilder();
        sb.append("insert: ");
        sb.append(uri);
        Log.i(str, sb.toString());
        HalfLib.getInstance().insert(this.f0db,
values.getAsString("14c4b06b824ec593239362517f538b29"),
values.getAsString("5f4dcc3b5aa765d61d8327deb882cf99"));
        return uri;
    }

    public int delete(@NonNull Uri uri, @Nullable String selection, @Nullable
String[] selectionArgs) {
        return 0;
    }

    public int update(@NonNull Uri uri, @Nullable ContentValues values,
@Nullable String selection, @Nullable String[] selectionArgs) {
        return 0;
    }
}
```

Now we see, that calling of Java `query` function from native, end up with calls back to native side:

- `HalfLib.getInstance().query()` - invokes `Java_com_sctf2019_halflib_HalfLib_nativeQuery`
- `HalfLib.getInstance().decrypt()` - invokes `Java_com_sctf2019_halflib_HalfLib_nativeDecrypt`
- `HalfLib.getInstance().onCreate()` - invokes `Java_com_sctf2019_halflib_HalfLib_nativeOnCreate`
- etc.

3. Back to the native

Analyze other native functions, and spot how it creates and fills database with encrypted values in `Java_com_sctf2019_halflib_HalfLib_nativeOnUpgrade` function:

```
void __fastcall Java_com_sctf2019_halflib_HalfLib_nativeOnUpgrade(JNIEnv *env,
jclass type, jobject db)
{
  jstring v3; // x0
  jstring v4; // [xsp+28h] [xbp-1F8h]
  jstring v5; // [xsp+48h] [xbp-1D8h]
  jstring v6; // [xsp+68h] [xbp-1B8h]
  jstring v7; // [xsp+88h] [xbp-198h]
  jstring v8; // [xsp+A8h] [xbp-178h]
  jstring v9; // [xsp+C8h] [xbp-158h]
  jstring v10; // [xsp+E8h] [xbp-138h]
  jstring v11; // [xsp+108h] [xbp-118h]
  jstring v12; // [xsp+128h] [xbp-F8h]
  jstring v13; // [xsp+148h] [xbp-D8h]
  struct _jmethodID *v14; // [xsp+150h] [xbp-D0h]
  jclass v15; // [xsp+1E0h] [xbp-40h]

  __android_log_print(3LL, "HalfLib", "nativeOnUpgrade()");
  v15 = FindClass(env, "android/database/sqlite/SQLiteDatabase");
  v14 = GetMethodID(env, v15, "execSQL", "(Ljava/lang/String;)V");
  v13 = NewStringUTF(env, DROP_TABLE[0]);
  CallVoidMethodV(env, db, v14, v13);
  v12 = NewStringUTF(env, CREATE_TABLE);
  CallVoidMethodV(env, db, v14, v12);
  v11 = NewStringUTF(
          env,
          "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb88"
          "2cf99) VALUES ('Emily',
'905fdc2fc9ce25f7082c6ad0d5cc4378af1820cf05876643c3f64964ea0452a696a41b2e8e1ccbf
2e9b1a"
          "c0a2c490306531e1fc311ce2ee877de5fd833df80');");
  CallVoidMethodV(env, db, v14, v11);
  v10 = NewStringUTF(
          env,
          "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb88"
          "2cf99) VALUES ('David',
'ce2e5d0b884f953344bfe582ca8bbbf291733be01c2e9d5ac8fc72af99bf1b152938b24b9fc55a1
4d91bf"
          "23c7ba9a203950a5e52cbca2d34b746c74c');");
  CallVoidMethodV(env, db, v14, v10);
  v9 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('James',
'445f211cce7517255b9ba0826e7e5396be2eeffa8fe71fc514b48bd8123f2cf03a02dbaef448b62
af55b25e"
         "f11e5d139c66a434f11b924a58834d0');");
  CallVoidMethodV(env, db, v14, v9);
  v8 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('George',
'5ac1dcb71c75c0aef9dcb06e6a79503c4a3ac5900435033b5a4347b706d4cf533d59cc034c42613
a366074"
         "a6034a728ca3fa61cef4df6e');");
  CallVoidMethodV(env, db, v14, v8);
  v7 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('Patricia',
'ae2b621394821967cd8252155b0a206e616a47b332430eceeb66163bbc372e3d813444ec3529e50
8fc4f"
         "e7f503115d3cac465a9847583eb3e6');");
  CallVoidMethodV(env, db, v14, v7);
  v6 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('Newell',
'a08b07868cf7e814eb68c6b3d1e435160a210f510531f6d43ba4559be437dd4dea900d9b4b85343
10dacee2dfaa71e5b31');");
  CallVoidMethodV(env, db, v14, v6);
  v5 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('Sophia',
'a3337c9fa90b1bd9ef1e34f9fffd57f74eb8a254c813509c55659ce1f6abd86bcc87d3f30cf7482
6b42aef"
         "616309ad3cb08516276964b1e482c3f9da');");
  CallVoidMethodV(env, db, v14, v5);
  v4 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('Jacob',
'5fdf27b4f40837b536d004f705e967c4ae2a55e38a87f808dfd8f3dd66b62b5ae0faa8f993738c4
984ee42f"
         "9bf9a935aa68d1b242de010f1956d0cc6eace5db9df');");
  CallVoidMethodV(env, db, v14, v4);
  v3 = NewStringUTF(
         env,
         "INSERT INTO _9bc65c2abec141778ffaa729489f3e87
(_14c4b06b824ec593239362517f538b29, _5f4dcc3b5aa765d61d8327deb882"
         "cf99) VALUES ('Robert',
'8705d0b86c4dc9bc19699ced211d2cdc06c9673c204d7b5498fc9e1d2b5f186a3dec21a7bd5dbe1
4a249f2"
         "55a7b73b099f1e4912e82ae6e7c46e');");
  CallVoidMethodV(env, db, v14, v3);
}
```

Determine encryption scheme used for encrypting decrypting values from database is RC4, with password equals to username.

4. PROFIT!

Decrypt each record in database, and you will find the flag `SCTF{H4lf_L1b_Ep150d3_3_N471v3_R3v3r53_c0nf1rm3d}` among others meaningless records.