

Fuzzing remote interfaces for system services in Android

Alexandru Blanda
Information Security Engineer

DEEPSEC

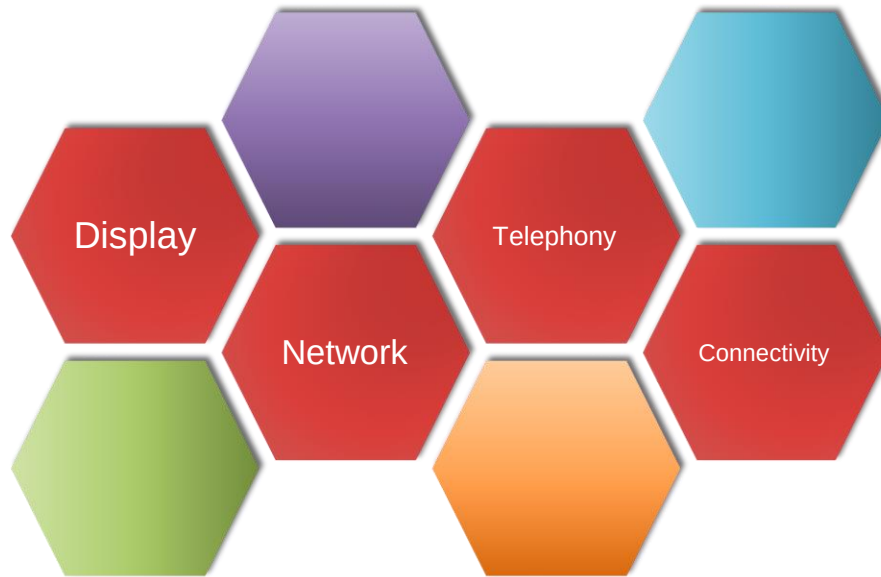
Agenda

Context

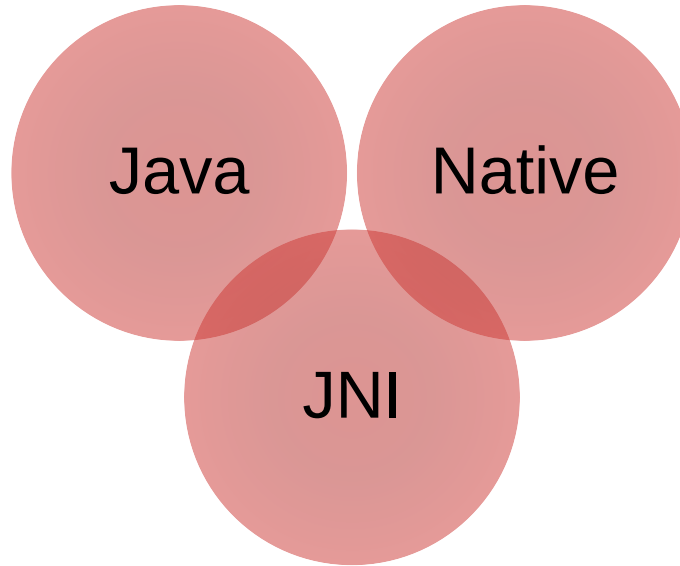
Fuzzing tool implementation

Results and vulnerabilities

- System services – the core of Android
- Implement many of the fundamental Android features



- System services – expose remote interfaces



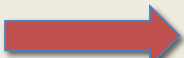
- Android IPC mechanism – Binder
- Android controls access to Binder objects
- Some Binder objects (**system services**) need to be universally accessible

```
root@flounder:/ # service
```

```
Usage: service [-h|-?]
```

```
    service list
```

```
    service check SERVICE
```



```
    service call SERVICE CODE [i32 N | i64 N | f N | d N | s16
```

Options:

i32: Write the 32-bit integer N into the send parcel.

i64: Write the 64-bit integer N into the send parcel.

f: Write the 32-bit single-precision number N into the parcel

d: Write the 64-bit double-precision number N into the parce

s16: Write the UTF-16 string STR into the send parcel.

Context

Service CLI

Android service list

```
flounder:/ $ service list
Found 118 services:
0      nfc: [android.nfc.INfcAdapter]
1      phone: [com.android.internal.telephony.ITelephony]
2      isms: [com.android.internal.telephony.ISms]
...
18     media_session: [android.media.session.ISessionManager]
19     restrictions: [android.content.IRestrictionsManager]
...
45     notification: [android.app.INotificationManager]
46     recovery: [android.os.IRecoverySystem]
47     updatelock: [android.os.IUpdateLock]
```

- Each service has a number of methods → can be called using **service call**

```
$ service call SERVICE CODE [i32 N | i64 N | f N | d N | s16 STR]
```

```
$ service call meminfo 13
```

```
i32 12 i32 43 s16 "string_example"
```

- Each service has a remote interface that defines the methods, implemented in **Java** or **C/C++**

```
...  
101 SurfaceFlinger: [android.ui.ISurfaceComposer]  
  
$ find /path/to/tree -name ISurfaceComposer.h / ISurfaceComposer.aidl  
...
```

Remote interface – example

```
class IAudioPolicyService : public Iinterface
{
public:
    DECLARE_META_INTERFACE(AudioPolicyService);

1 → virtual status_t registerEffect(const effect_descriptor_t *desc,
                                   audio_io_handle_t io,
                                   uint32_t strategy,
                                   int session,
                                   int id) = 0;

2 → virtual status_t unregisterEffect(int id) = 0;

3 → virtual status_t setEffectEnabled(int id, bool enabled) = 0;
```

Context

Service CLI

Remote interface – transaction codes

```
class IAudioPolicyService : public Iinterface
{
public:
    DECLARE_META_INTERFACE(AudioPolicyService);

    3 → virtual status_t setEffectEnabled(int id, bool enabled) = 0;

    $ service call media.audio_policy 3 i32 23 i32 1
```

Context

Service CLI

Remote interface – not a data message

```
class IResourceManagerService : public Iinterface
{
public:
    DECLARE_META_INTERFACE(IResourceManagerService);

1 → virtual status_t method_1(int ...);
2 → virtual status_t method_2(int ...);
3 → virtual status_t method_3(int ...);
4 → virtual status_t method_4(int ...);

5 $ service call media.resource_manager 5
??? Parcel(Error: 0xfffffffffffffb6 "Not a data message")
```

Context

Service CLI

Remote interface – not a data message

```
class ISurfaceComposer: public IInterface {
public:
    DECLARE_META_INTERFACE(SurfaceComposer);

1 → virtual sp<ISurfaceComposerClient> createConnection() = 0;
2 → virtual sp<IGraphicBufferAlloc> createGraphicBufferAlloc() = 0;
...
13 → virtual int getActiveConfig(const sp<IBinder>& display) = 0;
    Result: Parcel(Error: 0xfffffffffffffb6 "Not a data message")

14 → virtual int getActiveConfig(const sp<IBinder>& display) = 0;
    Result: Parcel(fffffea '....')
```

Remote interface – permissions

```
flounder:/ $ service call bluetooth_manager 7
```

```
Result: Parcel(
```

```
0x00000000: ffffffff 0000006e 0065004e 00640065 '....n...N.e.e.d.'
0x00000010: 00420020 0055004c 00540045 004f004f ' .B.L.U.E.T.O.O.'
0x00000020: 00480054 00410020 004d0044 004e0049 'T.H. .A.D.M.I.N.'
0x00000030: 00700020 00720065 0069006d 00730073 ' .p.e.r.m.i.s.s.'
0x00000040: 006f0069 003a006e 004e0020 00690065 'i.o.n.:. .N.e.i.'
0x00000050: 00680074 00720065 00750020 00650073 't.h.e.r. .u.s.e.'
0x00000060: 00200072 00300032 00300030 006e0020 'r. .2.0.0.0. .n.'
0x00000070: 0072006f 00630020 00720075 00650072 'o.r. .c.u.r.r.e.'
0x00000080: 0074006e 00700020 006f0072 00650063 'n.t. .p.r.o.c.e.'
0x00000090: 00730073 00680020 00730061 00610020 's.s. .h.a.s. .p.'
0x000000a0: 00720065 0069006d 00730073 006f0069 'e.r.m.i.s.s.i.o.'
0x000000b0: 002e006e 004c0042 00450055 004f0054 'n.B.L.U.E.T.O.O.'
0x000000c0: 0054004f 005f0048 00440041 0049004d 'T.H._.A.D.M.I.N.')
```

Agenda

Context

Fuzzing tool implementation

Results and vulnerabilities

Fuzzing tool implementation

Main idea

Fuzz the
methods
of each
system
service

- service call command
- transaction code for each method
- fuzzed parameters

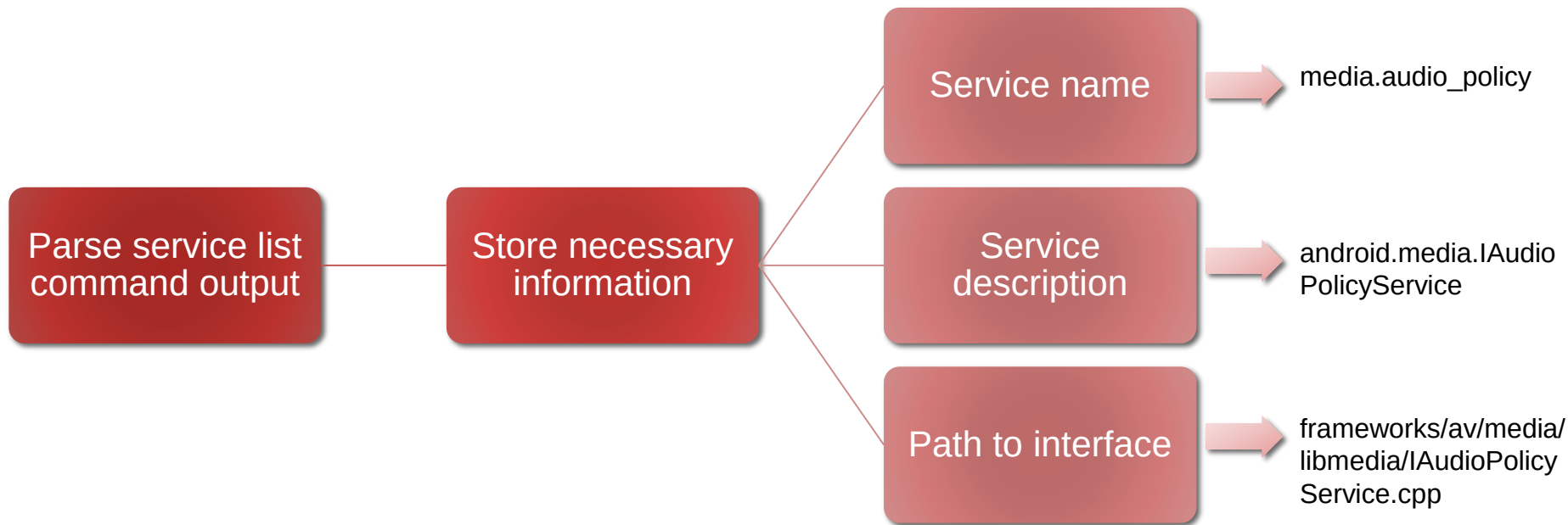
2 testing
scenarios

- 0 permissions
- all permissions

Fuzzing tool

Module 1

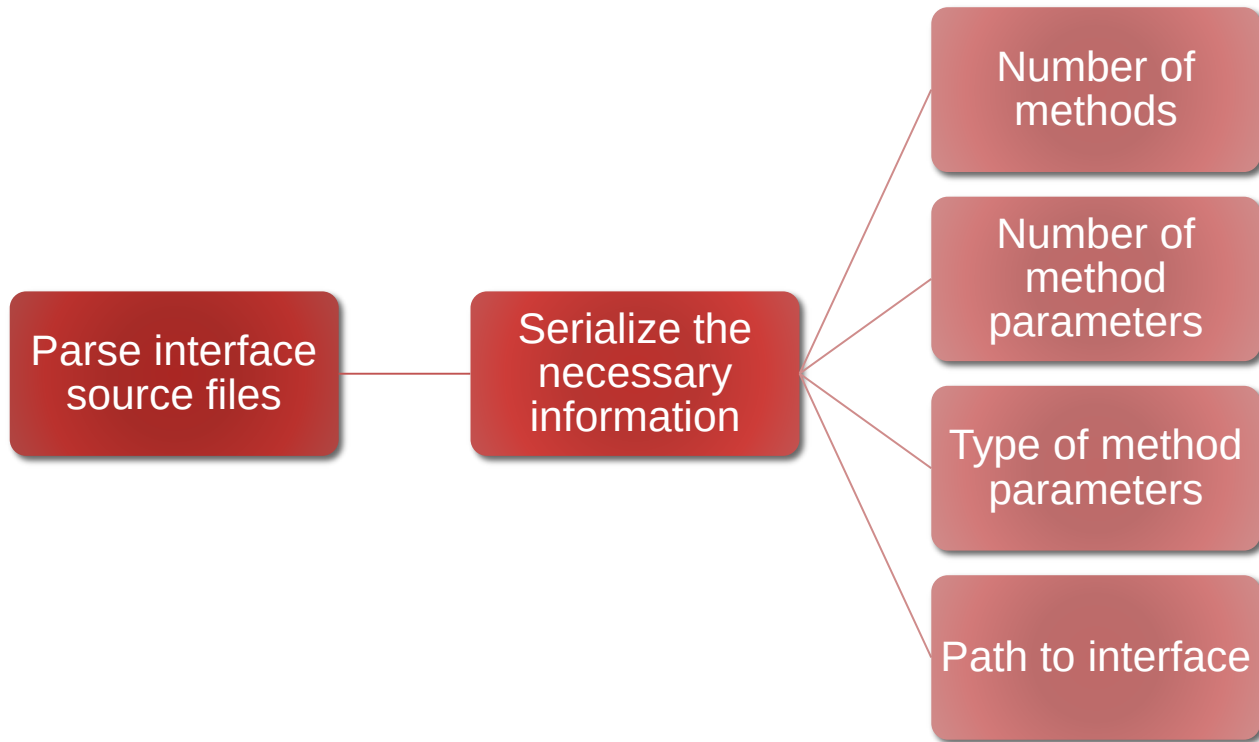
Get information regarding the available system services



Fuzzing tool

Module 2

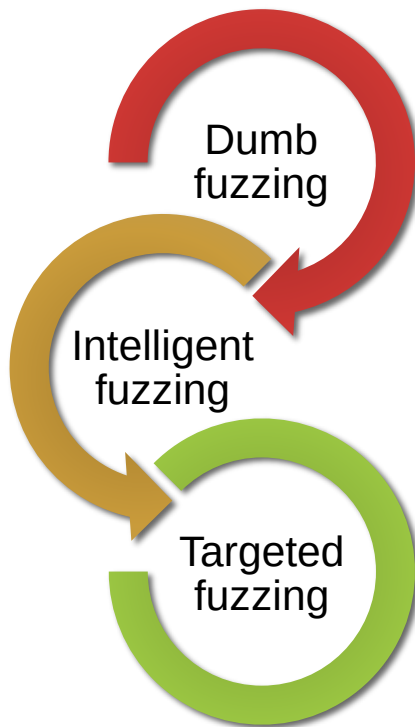
Get information regarding the methods of each service



Fuzzing tool

Actual fuzzing process

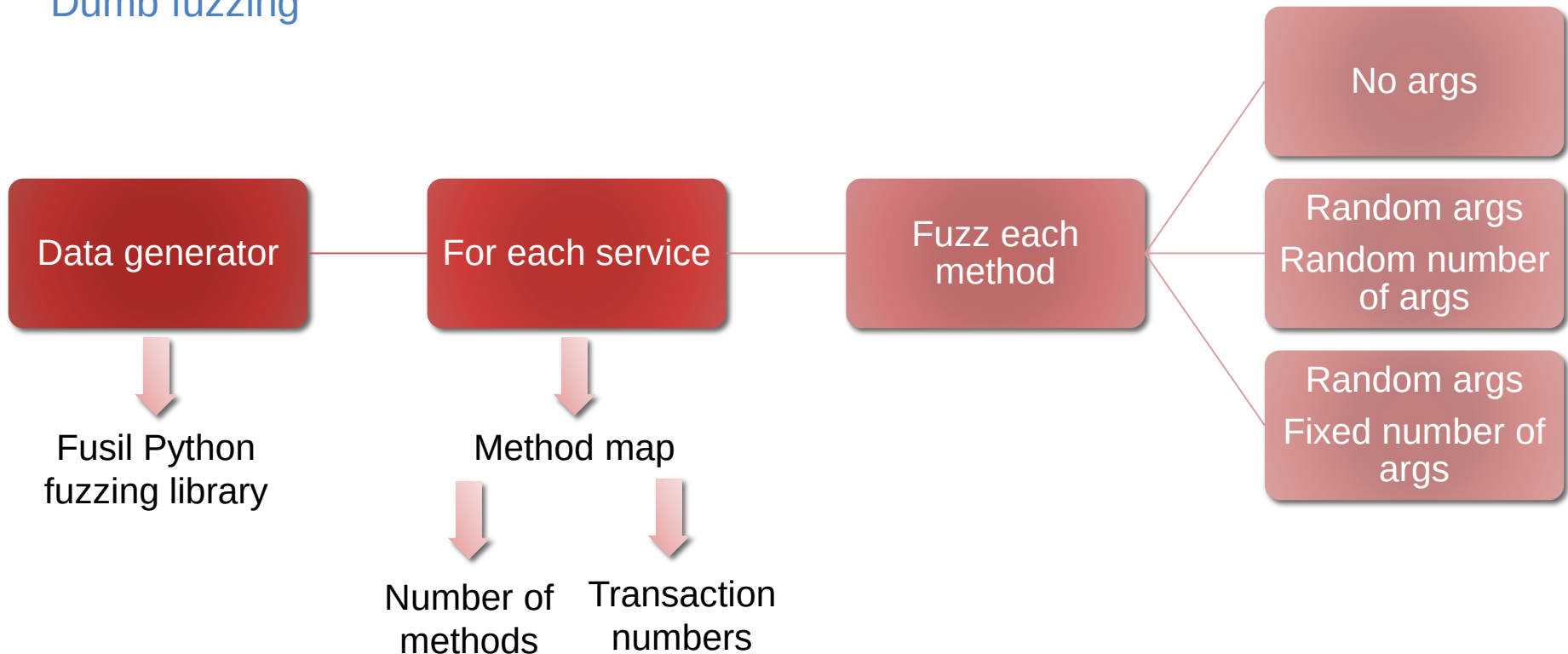
Module 3



Fuzzing tool

Dumb fuzzing

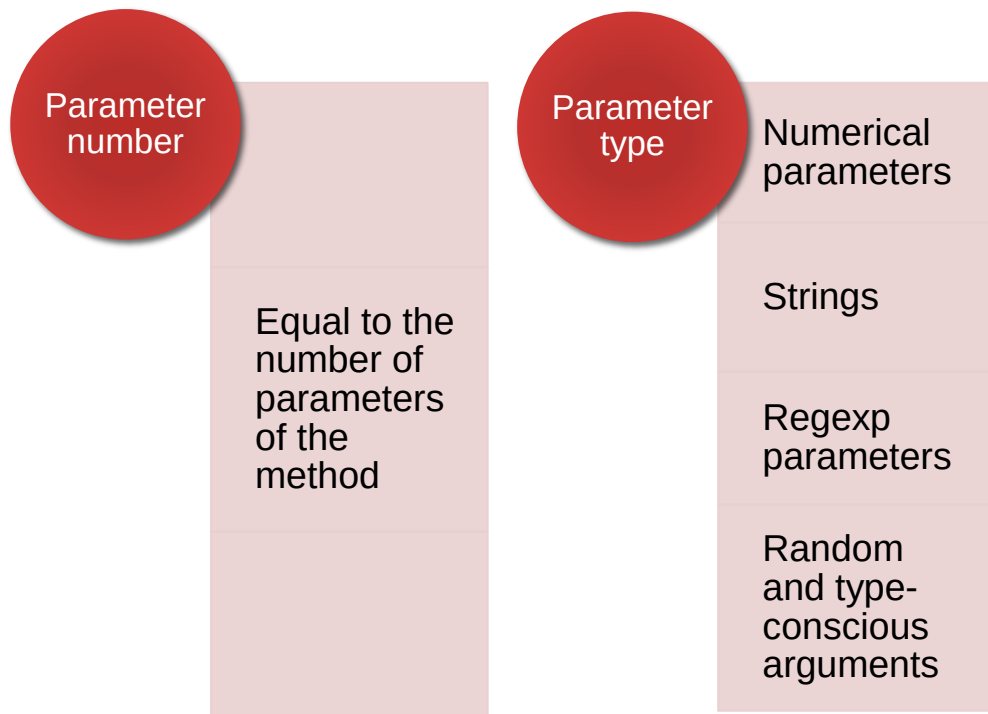
Module 3



Fuzzing tool

Intelligent vs dumb fuzzing

Module 3



Fuzzing tool

Targeted fuzzing

Module 3

- Create custom generation models for each particular interface
- Priority is on **native** interfaces, but **Java** based interfaces are interesting as well
- Allows triggering sections of code that would be otherwise inaccessible

Fuzzing tool

Module 4

Logging process

```
03-17 13:43:17.310 F/service_call:DUMB:fuzzer(29448): createDisplay[4] - param: 2 -  
seed: 0  
03-17 13:43:17.461 F/service_call:DUMB:fuzzer(29453): createDisplay[4] - param: 2 -  
seed: 4736359305080745519  
03-17 13:43:17.533 F/service_call:DUMB:fuzzer(29456): createDisplay[4] - param: 2 -  
seed: 3491175988003079  
03-17 13:43:17.229 F/libc( 9876): Fatal signal 11 (SIGSEGV), code 1, fault addr 0x0 in  
tid 9890 (Binder_2)  
03-17 13:43:18.137 F/service_call:DUMB:fuzzer(29476): destroyDisplay[5] - param: 1 -  
seed: 0  
03-17 13:43:18.281 F/service_call:DUMB:fuzzer(29481): destroyDisplay[5] - param: 1 -  
seed: 3218211437215368928  
03-17 13:43:18.430 F/service_call:DUMB:fuzzer(29486): destroyDisplay[5] - param: 1 -  
seed: 5058432304378629718
```

Fuzzing tool

Logging process

Module 4

Problems

- Some of the arguments are not printable
- Generate as many random values as the number of parameters

Ideal case

- Save the random generator state before each test case is executed (not feasible)

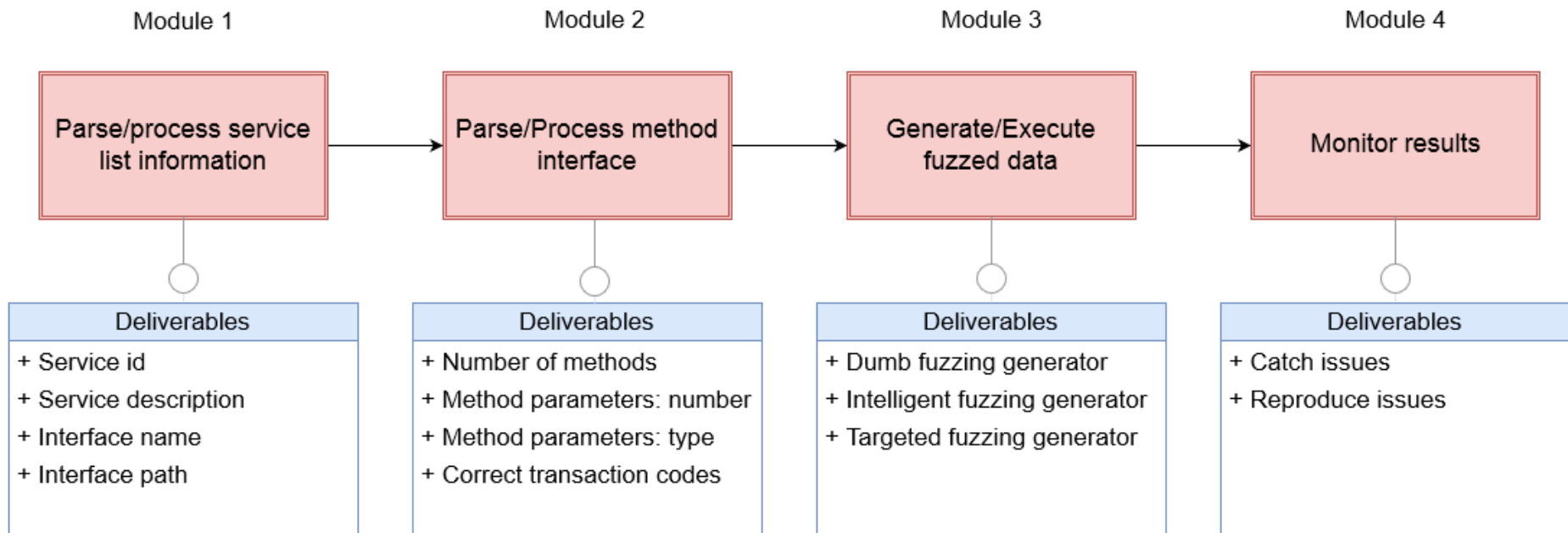
Solution

- For a given test case, generate a single seed

```
seed_list = []  
init_seed = randint (0, max_value)  
seed_list.append (init_seed)  
seed[i] = F (init_seed, i)  
seed_list.append (seed[i])
```

Fuzzing tool

Summary



Agenda

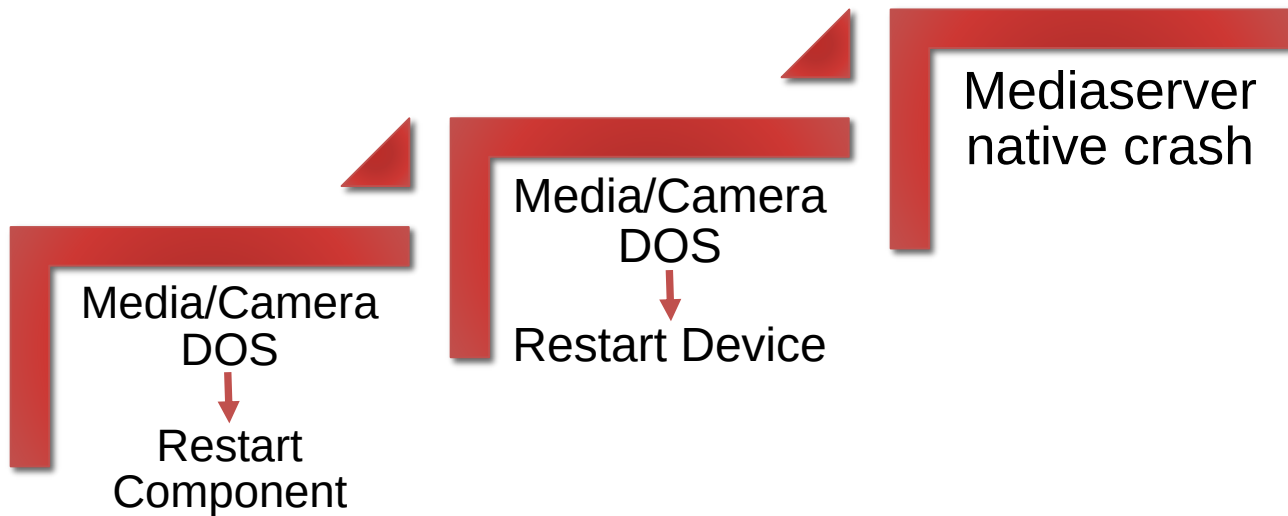
Context

Fuzzing tool implementation

Results and vulnerabilities

Vulnerability example

- Vulnerability in libaudiopolicyservice.so – triggered when calling portConfig() method in IAudioPolicyService interface
- Impact:



Vulnerability example

Logcat snapshot

```
03-25 12:05:28.774 W/AudioSystem( 580): AudioFlinger server died!
03-25 12:05:28.774 W/AudioSystem( 580): AudioPolicyService server died!
03-25 12:05:28.774 W/SoundTrigger( 580): Sound trigger service died!
03-25 12:05:28.774 I/ServiceManager( 171): service 'media.resource_manager' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.audio_flinger' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.player' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.camera' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.audio_policy' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.sound_trigger_hw' died
03-25 12:05:28.775 I/ServiceManager( 171): service 'media.radio' died
03-25 12:05:28.775 W/AudioSystem( 1148): AudioFlinger server died!
03-25 12:05:28.775 W/AudioSystem( 728): AudioPolicyService server died!
03-25 12:05:28.775 E/AudioService( 580): Media server died.
```

Vulnerability example

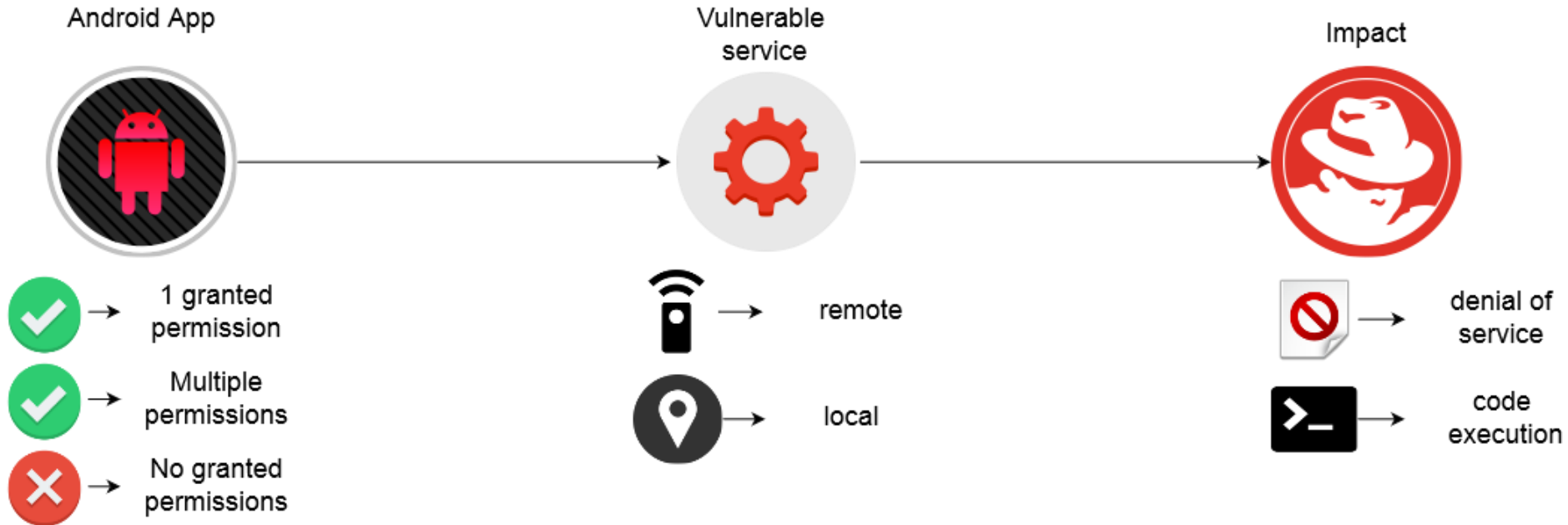
Mediaserver native crash

```
pid: 224, tid: 909, name: Binder_3 >>> /system/bin/mediaserver <<<signal 11  
(SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x18
```

backtrace:

```
#00 pc 00006608 /system/lib/libaudiopolicyservice.so  
#01 pc 0008aa11 /system/lib/libmedia.so  
(android::BnAudioPolicyService::onTransact(unsigned int, android::Parcel const&,  
android::Parcel*, unsigned int)+2172)  
#02 pc 00019999 /system/lib/libbinder.so (android::BBinder::transact(unsigned  
int, android::Parcel const&, android::Parcel*, unsigned int)+60)  
#03 pc 0001ecf9 /system/lib/libbinder.so  
(android::IPCThreadState::executeCommand(int)+560)  
#05 pc 0001eeb5 /system/lib/libbinder.so  
(android::IPCThreadState::joinThreadPool(bool)+48)  
#07 pc 0001006d /system/lib/libutils.so (android::Thread::_threadLoop(void*)+112)
```

Attack scenario



blanda.alexandru@gmail.com
@bigdinrock
github.com/fuzzing



Q&A