Fuzzing Android OMX

Mingjian Zhou and Chiachih Wu
C0RE Team

About Us

- Mingjian Zhou, 周明建
  - Security researcher @ 360 C0RE team
  - Focused on Android vulnerability research and exploit development
- Chiachih Wu, 吳家志 (@chiachih_wu)
  - Security researcher @ 360 C0RE team
  - Android/Linux system security research
  - C0RE team (c0reteam.org) founding member
- C0RE Team
  - A security-focused group started in mid-2015
  - With a recent focus on the Android/Linux platform, the team aims to discover zero-day vulnerabilities, develop proof-of-concept exploits, and explore possible defenses

Stagefright: Scary Code in the Heart of Android

Researching Android Multimedia Framework Security

ZIMPERIUM MOBILE DEFENSE

Joshua "jduck" Drake
August 5th 2015
Black Hat USA

Nexus Security Bulletin - October 2015

Published October 05, 2015 | Updated April 28, 2016

IN THIS DOCUMENT

Acknowledgements

We would like to thank these researchers for their contributions:

- Brennan Lautner: CVE-2015-3863
- Chiachih Wu and Xuxian Jiang of C0RE Team from Qihoo 360: CVE-2015-3868, CVE-2015-3869, CVE-2015-3862
- Yajin Zhou, Lei Wu, and Xuxian Jiang of C0RE Team from Qihoo 360: CVE-2015-3865

Nexus Security Bulletin - February 2016

Published February 01, 2016 | Updated March 7, 2016

IN THIS DOCUMENT

Acknowledgements

We would like to thank these researchers for their contributions:

- Android and Chrome Security Team: CVE-2016-0809, CVE-2016-0810
- Broadgate Team: CVE-2016-0801, CVE-2015-0802
- Chiachih Wu (@chiachih_wu), Mingjian Zhou (@Mingjian_Zhou), and Xuxian Jiang of C0RE Team, Qihoo 360: CVE-2016-0804

Fuzzing Android System Services by Binder Call to Escalate Privilege

Guang Gong
Security Reacher
Qihoo 360
Twitter &Weibo:@oldfresher

Black Hat USA 2015

Agenda

- Introduction
- Fuzzing Android OMX
- Confirmed Vulnerabilities
- Patterns of OMX Vulnerabilities

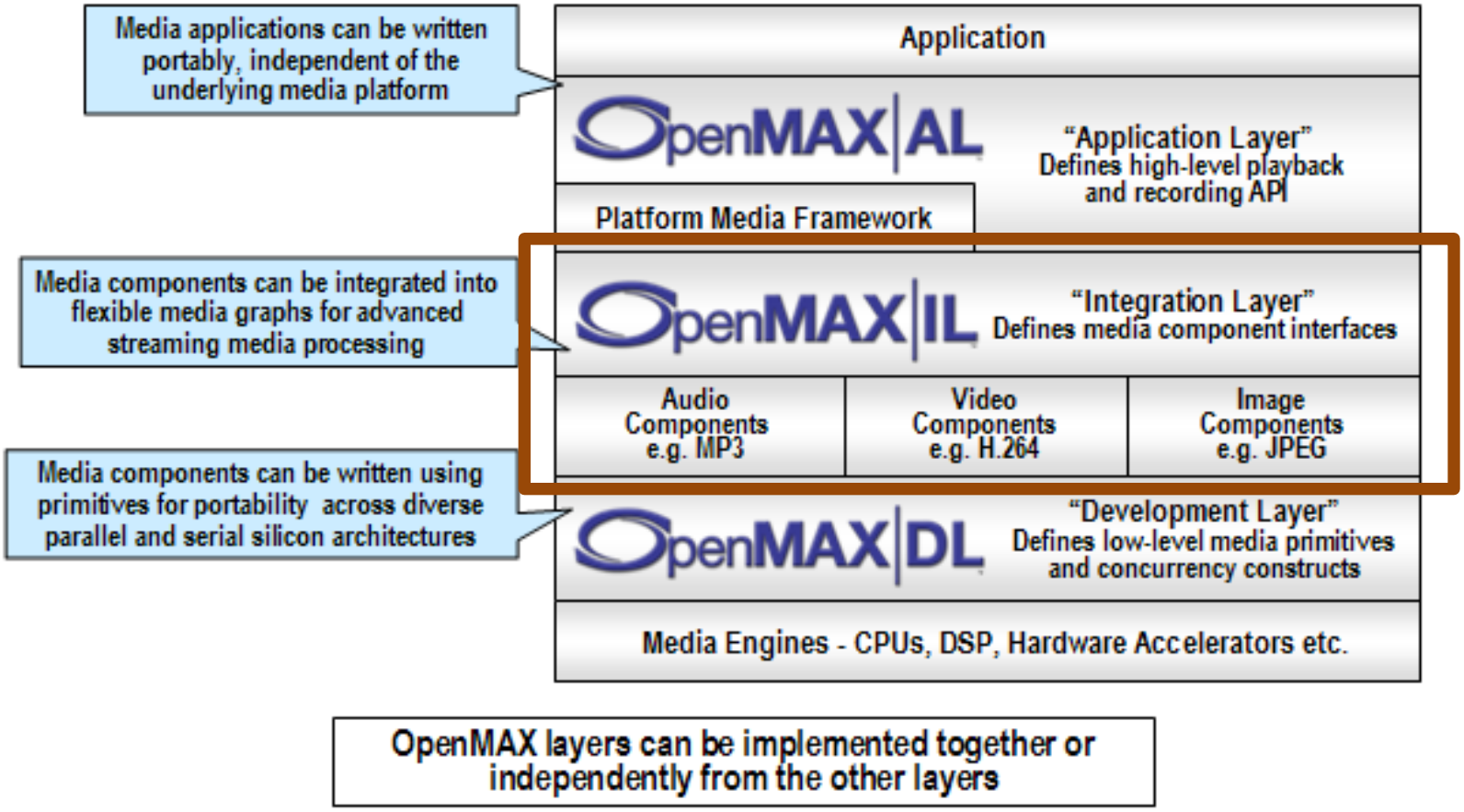
About OMX

INTRODUCTION

What is OMX (1/2)

- Open Media Acceleration, aka Open MAX, often shortened as "**OMX**"
- *WIKI:* a non-proprietary and royalty-free **cross-platform** set of **C-language programming interfaces** that provides abstractions for routines especially useful for **audio, video, and still images processing**.

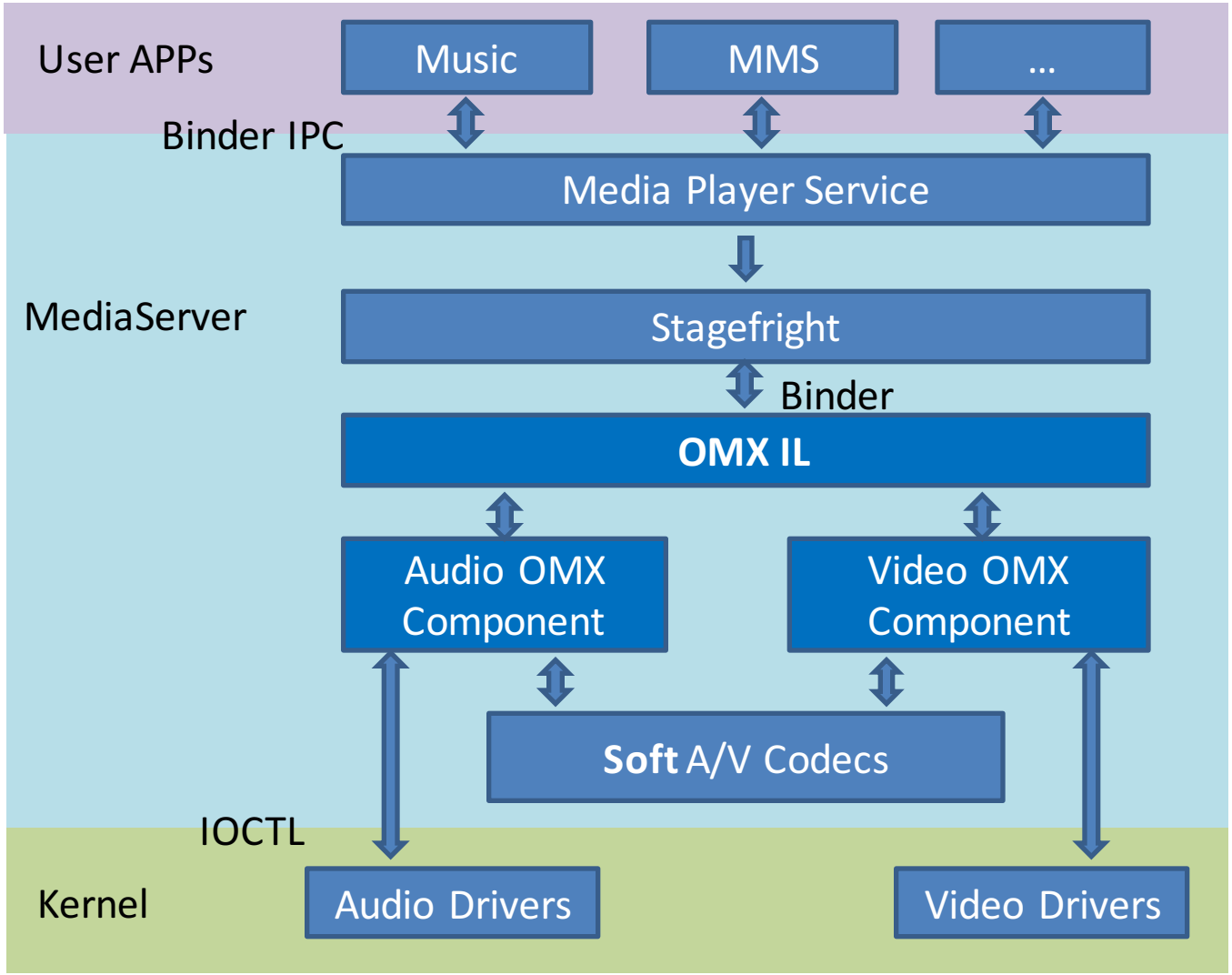
What is OMX (2/2)



OMX in Android (1/2)

- OMX Integration Layer (IL)
  - provides a standardized way for **Stagefright** to recognize and use custom hardware-based **multimedia codecs called components**.
- Vendors provide the **OMX plugin** which links custom codec components to Stagefright.
- Custom codecs **must** be implemented according to the OMX IL component standard.

OMX in Android (2/2)



OMX Codecs

- Android provides built-in software codecs for common media formats
- Vendors' codecs

Built-in Soft Codecs Example

```
OMX.google.aac.decoder
OMX.google.aac.encoder
OMX.google.amrnb.decoder
OMX.google.amrnb.encoder
OMX.google.amrwb.decoder
OMX.google.amrwb.encoder
OMX.google.flac.encoder
OMX.google.g711.alaw.decoder
OMX.google.g711.mlaw.decoder
OMX.google.gsm.decoder
```

Vendor Codecs Example

```
OMX.qcom.audio.encoder.aac
OMX.qcom.audio.encoder.amrnb
OMX.qcom.audio.encoder.evrc
OMX.qcom.audio.encoder.qcelp13
OMX.qcom.file.muxer
OMX.qcom.video.decoder.avc
OMX.qcom.video.decoder.avc.secure
OMX.qcom.video.decoder.divx
OMX.qcom.video.decoder.divx311
OMX.qcom.video.decoder.divx4
OMX.qcom.video.decoder.h263
OMX.qcom.video.decoder.hevc
```

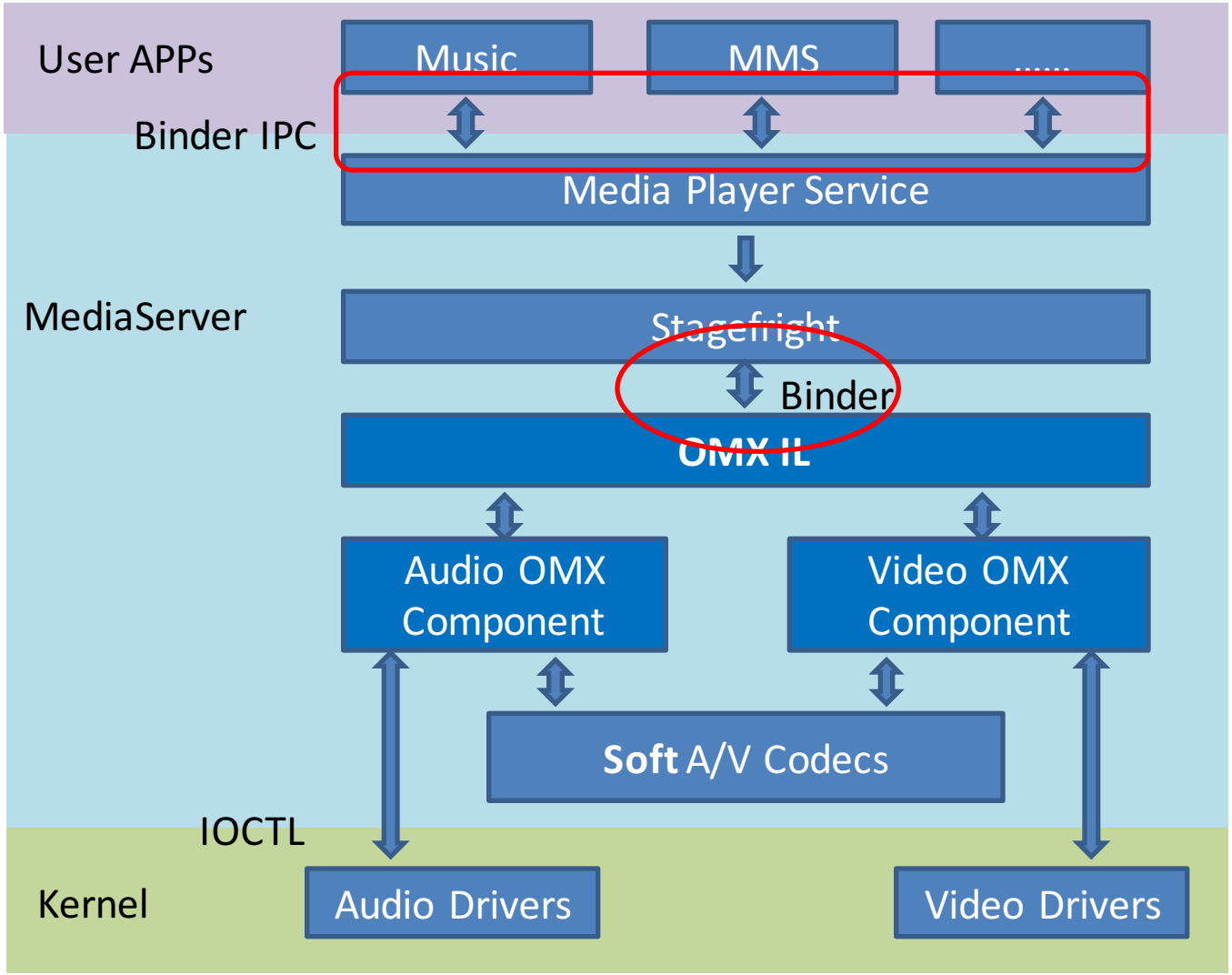
Why OMX?

- Exposed via multiple attack vectors
- Media native codes are often vulnerable

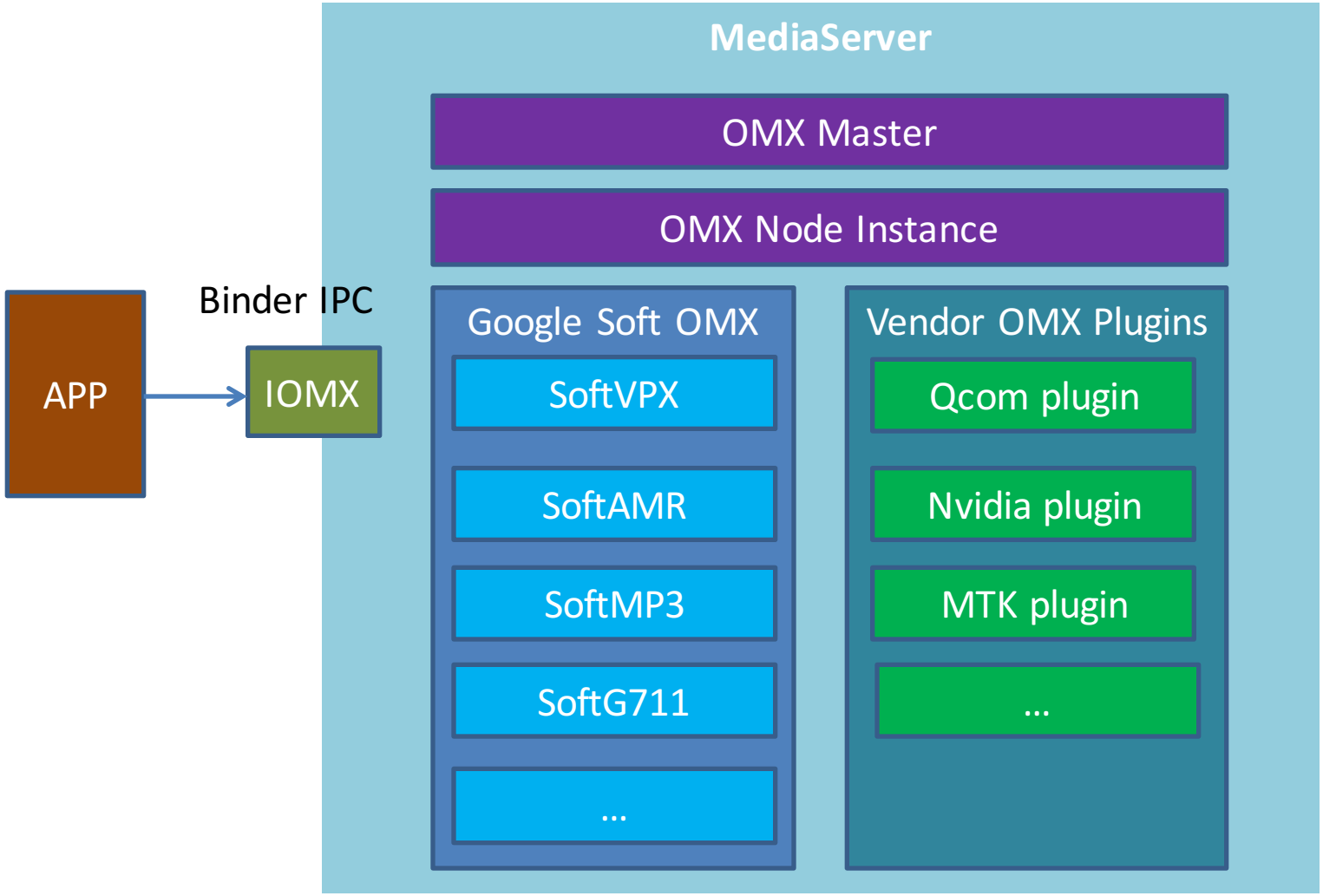
Attack Surface & Flow

FUZZING ANDROID OMX

The Attack Surface (1/2)



The Attack Surface (2/2)

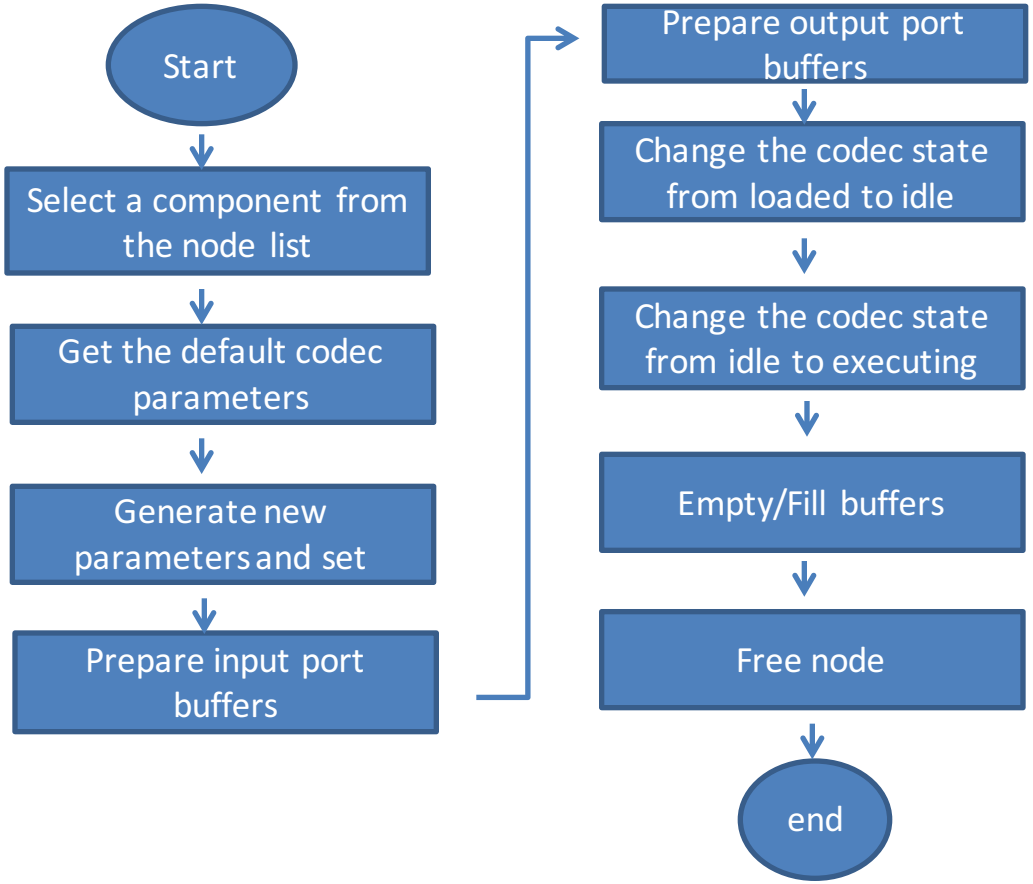


OMX Interfaces

- Defined in IOMX

API	Functions
listNodes	List names of all the codec component
allocateNode	Create a codec component
allocateBuffer	Allocate input/output buffers for codec
useBuffer	Provide a share buffer to the server
emptyBuffer	Request (or receive) an empty input buffer, fill it up with data and send it to the codec for processing
fillBuffer	Request (or receive) a filled output buffer, consume its contents and release it back to the codec
sendCommand	Send commands to codecs, such as changing state, port disable/enable
getParameter	Get codecs' parameters
setParameter	Set codecs' parameters

Fuzzing Flow



CONFIRMED VULNERABILITIES

Confirmed Vulnerabilities (1/3)

- By 2016/07/07, total **21** vulnerabilities are confirmed.
  - **16** vulnerabilities (15 high, 1 moderate) have been disclosed on Android Security Bulletins.
  - **Others** will be disclosed on **later** Android Security Bulletins.
- **Almost all** the codecs implemented by **Google** and **vendors(QualComm, Nvidia, MediaTek)** are vulnerable.

Confirmed Vulnerabilities (2/3)

NO.	CVE	Android ID	Codec
1	CVE-2016-2450	ANDROID-27569635	Google SoftVPX encoder
2	CVE-2016-2451	ANDROID-27597103	Google SoftVPX decoder
3	CVE-2016-2452	ANDROID-27662364	Google SoftAMR decoder
4	CVE-2016-2477	ANDROID-27251096	Qcom libOmxVdec
5	CVE-2016-2478	ANDROID-27475409	Qcom libOmxVdec
6	CVE-2016-2479	ANDROID-27532282	Qcom libOmxVdec
7	CVE-2016-2480	ANDROID-27532721	Qcom libOmxVdec
8	CVE-2016-2481	ANDROID-27532497	Qcom libOmxVenc
9	CVE-2016-2482	ANDROID-27661749	Qcom libOmxVdec
10	CVE-2016-2483	ANDROID-27662502	Qcom libOmxVenc

Confirmed Vulnerabilities (3/3)

NO.	CVE	Android ID	Codec
11	CVE-2016-2484	ANDROID-27793163	Google SoftG711 decoder
12	CVE-2016-2485	ANDROID-27793367	Google SoftGSM decoder
13	CVE-2016-2486	ANDROID-27793371	Google SoftMP3 decoder
14	CVE-2016-3747	ANDROID-27903498	Qcom libOmxVenc
15	CVE-2016-3746	ANDROID-27890802	Qcom libOmxVdec
16	CVE-2016-3765	ANDROID-28168413	Google SoftMPEG2 decoder
17	CVE-2016-3844	AndroidID-28299517	Not disclosed yet
18	CVE-2016-3835	AndroidID-28920116	Not disclosed yet
19	CVE-2016-3825	AndroidID-28816964	Not disclosed yet
20	CVE-2016-3824	AndroidID-28816827	Not disclosed yet
21	CVE-2016-3823	AndroidID-28815329	Not disclosed yet

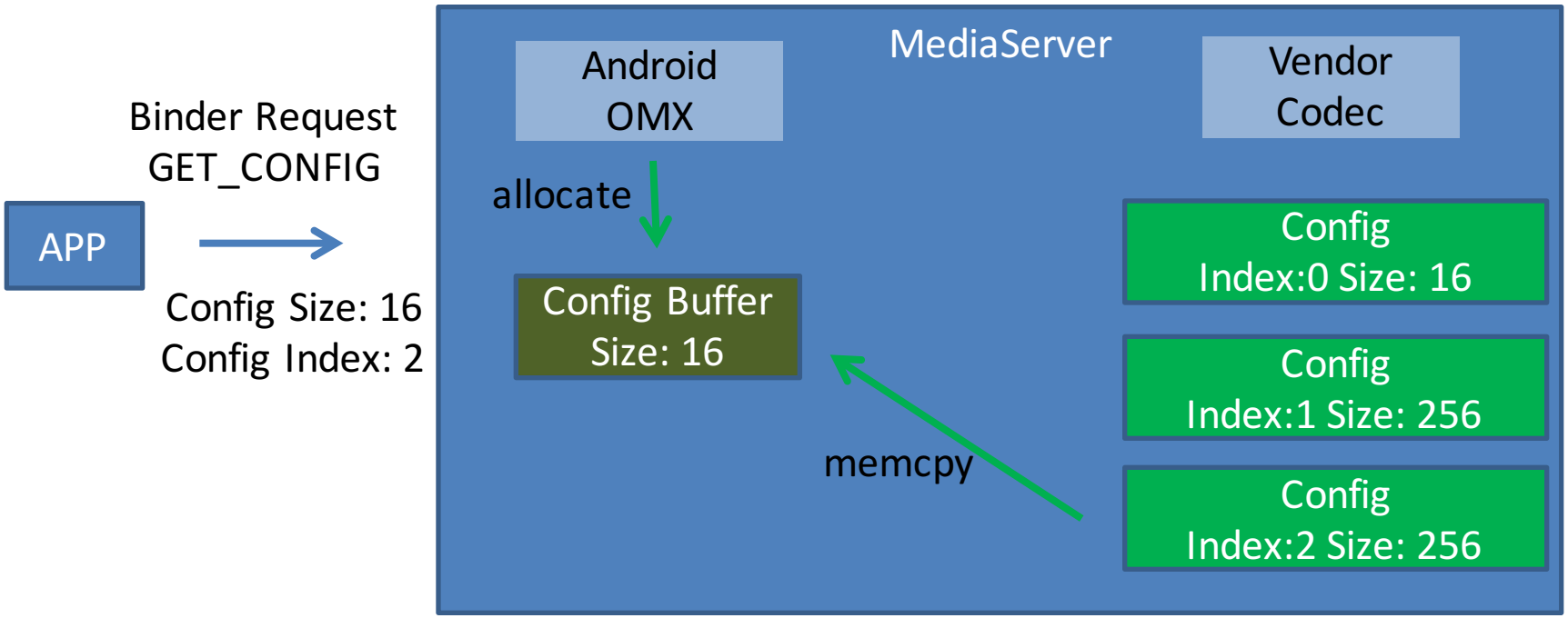
PATTERNS OF CONFIRMED VULNERABILITIES

Patterns of Confirmed Vulnerabilities

- Mismatch between Android OMX framework and vendor codecs' implementation
- Time of check to time of use
- Race condition
- Invalid input/output buffer length

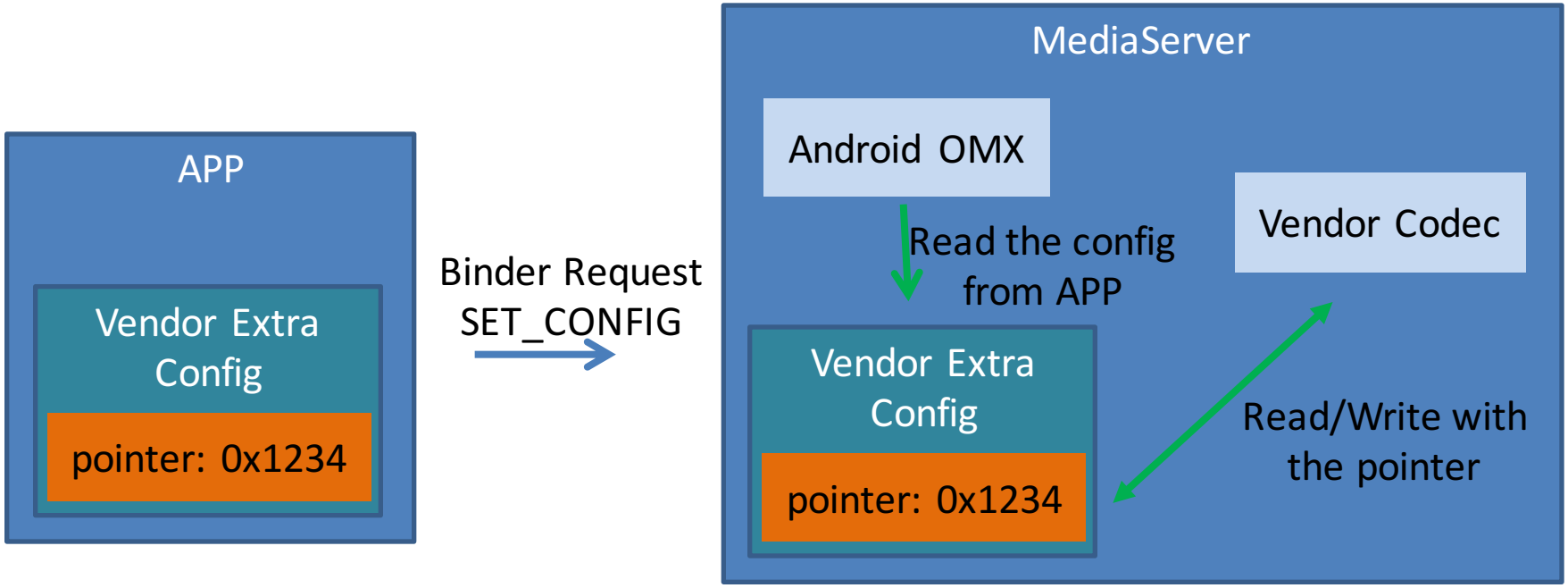
Mismatch between Android OMX and vendors' codec (1/2)

- CVE-2016-2480



Mismatch between Android OMX and vendors' codec (2/2)

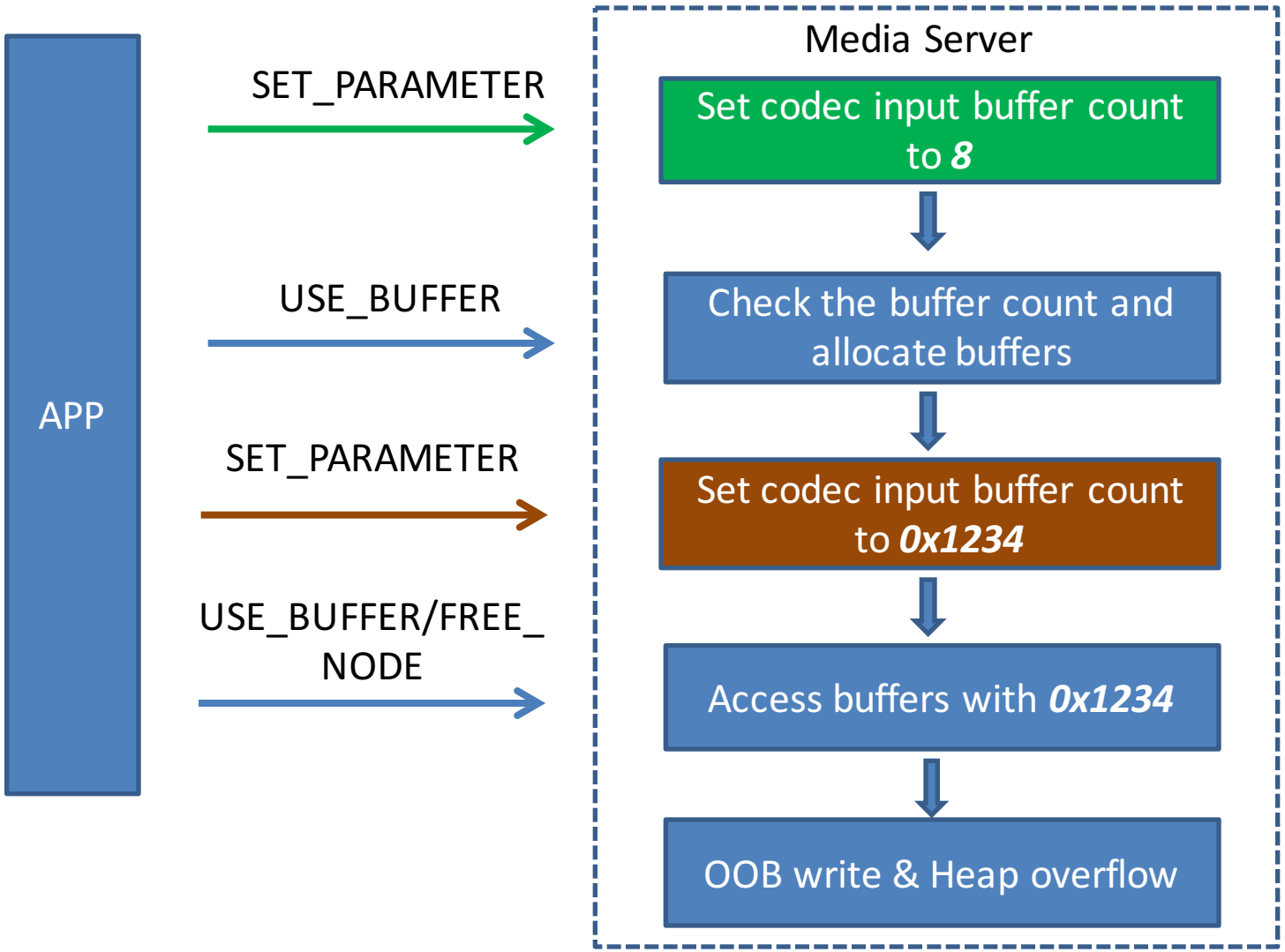
- CVE-2016-2477



Time of Check to Time of Use (1/2)

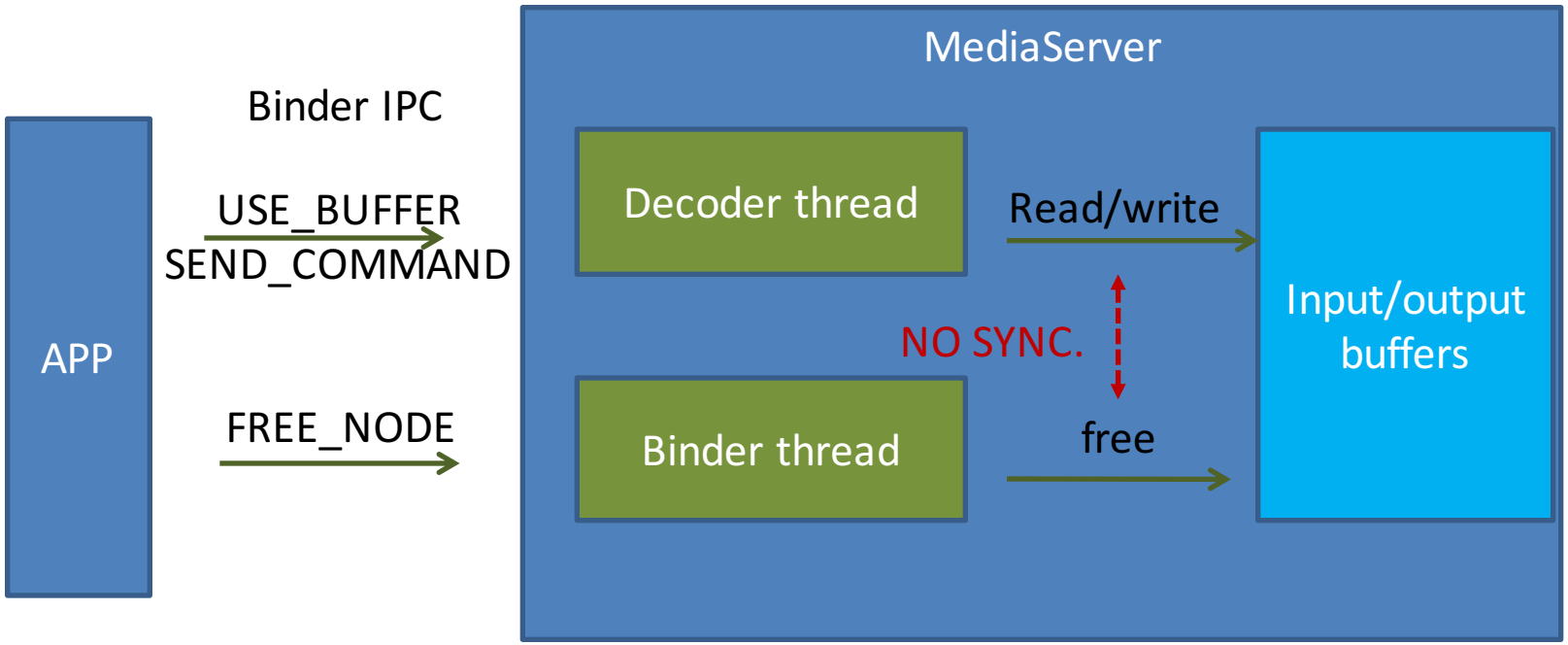
NO.	CVE	Android ID	Codec
1	CVE-2016-2479	ANDROID-27532282	Qcom libOmxVdec
2	CVE-2016-2481	ANDROID-27532497	Qcom libOmxVenc
3	CVE-2016-2482	ANDROID-27661749	Qcom libOmxVdec
4	CVE-2016-2483	ANDROID-27662502	Qcom libOmxVenc

Time of Check to Time of Use (2/2)



Race Condition

- CVE-2016-3747

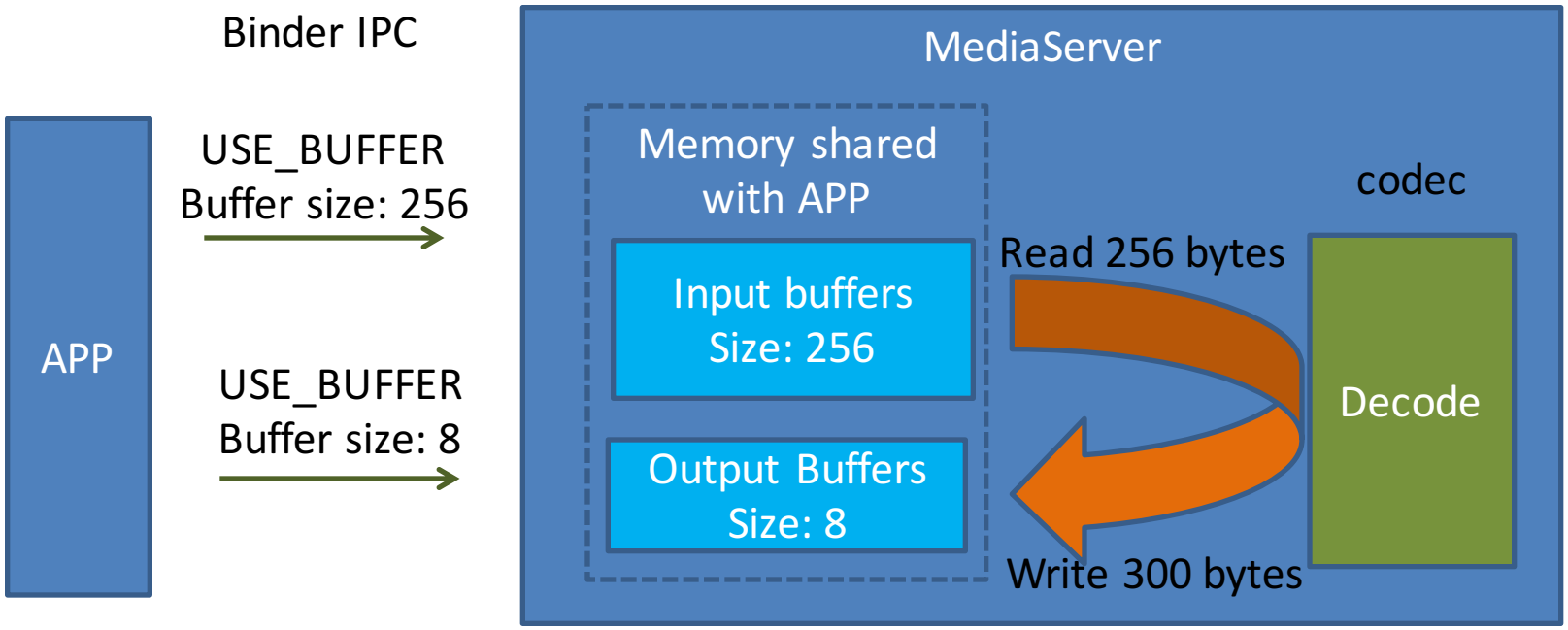


Invalid Input/Output Buffer Length

- Codecs don’t check the buffer length

NO.	CVE	Android ID	Codec
1	CVE-2016-2450	ANDROID-27569635	Google SoftVPX encoder
2	CVE-2016-2451	ANDROID-27597103	Google SoftVPX decoder
3	CVE-2016-2452	ANDROID-27662364	Google SoftAMR decoder
4	CVE-2016-2484	ANDROID-27793163	Google SoftG711 decoder
5	CVE-2016-2485	ANDROID-27793367	Google SoftGSM decoder
6	CVE-2016-2486	ANDROID-27793371	Google SoftMP3 decoder

Invalid Input/output Buffer Length



Conclusion

- Android OMX is vulnerable
  - OMX interfaces and OMX codecs are implemented by Google and vendors separately.
  - Media processing is complex.
- Fuzzing combined with code auditing is helpful for such modules.
  - Many codecs & parameters

Any Questions ?

- If you prefer to ask offline, contact us:
  - Mingjian Zhou
    - Twitter/Weibo: @Mingjian_Zhou
    - Mail: cn.zhou.mingjian@gmail.com
  - Chiachih Wu
    - Twitter: @chiachih_wu

APPENDIX

References

- Android
  - https://source.android.com/devices/media/
  - https://developer.android.com/reference/android/media/MediaCodec.html
- OMX
  - https://www.khronos.org/openmax/