Android SmartTVs Vulnerability Discovery via Log-Guided Fuzzing

Yousra Aafer, Wei You, Yi Sun, Yu Shi, Xiangyu Zhang, Heng Yin

Why is SmartTV Security Important? A *Few Reasons*

Smart TVs

Account for the largest market share of Home IoT devices

Expected to achieve a market value of 253 billion USD by 2023

Plethora of attack vectors:

- **Physical channels: e.g., sending crafted broadcast signals**
- **Malware: SmartTV users can download SmartTV-specific Apps**

Broad Spectrum of Attack Consequences: Cyber + Physical

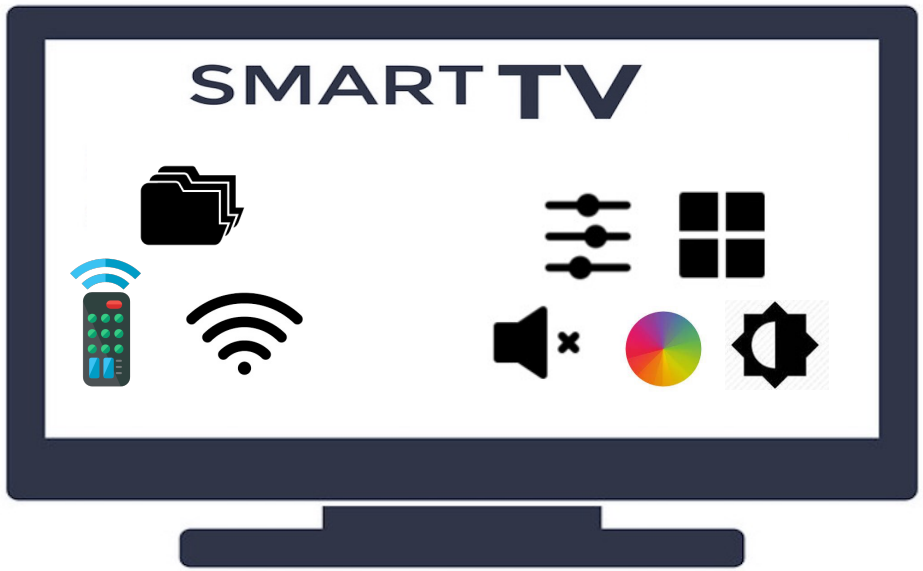
Goal

- Perform a systematic security evaluation of Android SmartTVs.
- Focus on customization aspects, performed to tailor the original OS for the SmartTV functionalities.

Background

Android SmartTVs run a heavily customized version of AOSP:

- Additional hardware, system components.
- Custom Functionalities are exposed to system and app developers through *dedicated APIs.*
  - *These APIs execute in the context of highly privileged processes.*

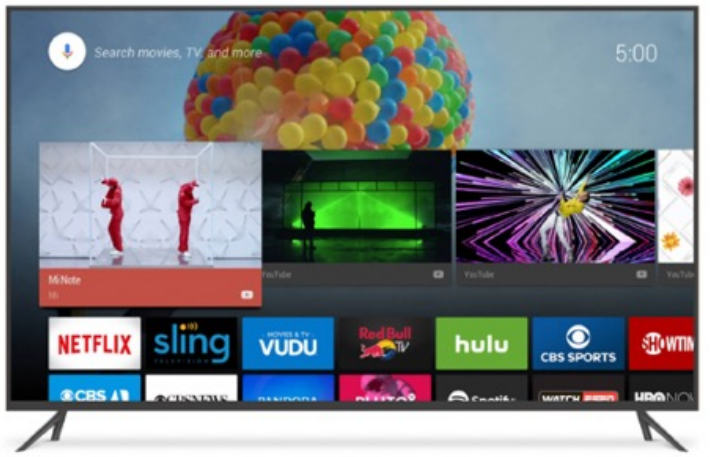


SmartTV APIs can open the door to various damages if not properly protected.

Motivating Example

- Xiaomi MiBox3 introduces a new native API **SystemControl. setPosition(x, y, w, h)**

SystemControl. setPosition(x, y, w, h)



Motivating Example

- Xiaomi MiBox3 introduces a new native API **SystemControl. setPosition(x, y, w, h)**
  - The API does not enforce any access control !
  - With the SmartTV ransomware on the rise, such APIs can be exploited to mount DoS attacks.

SystemControl. setPosition(1000, 1000, 1000, 1000)

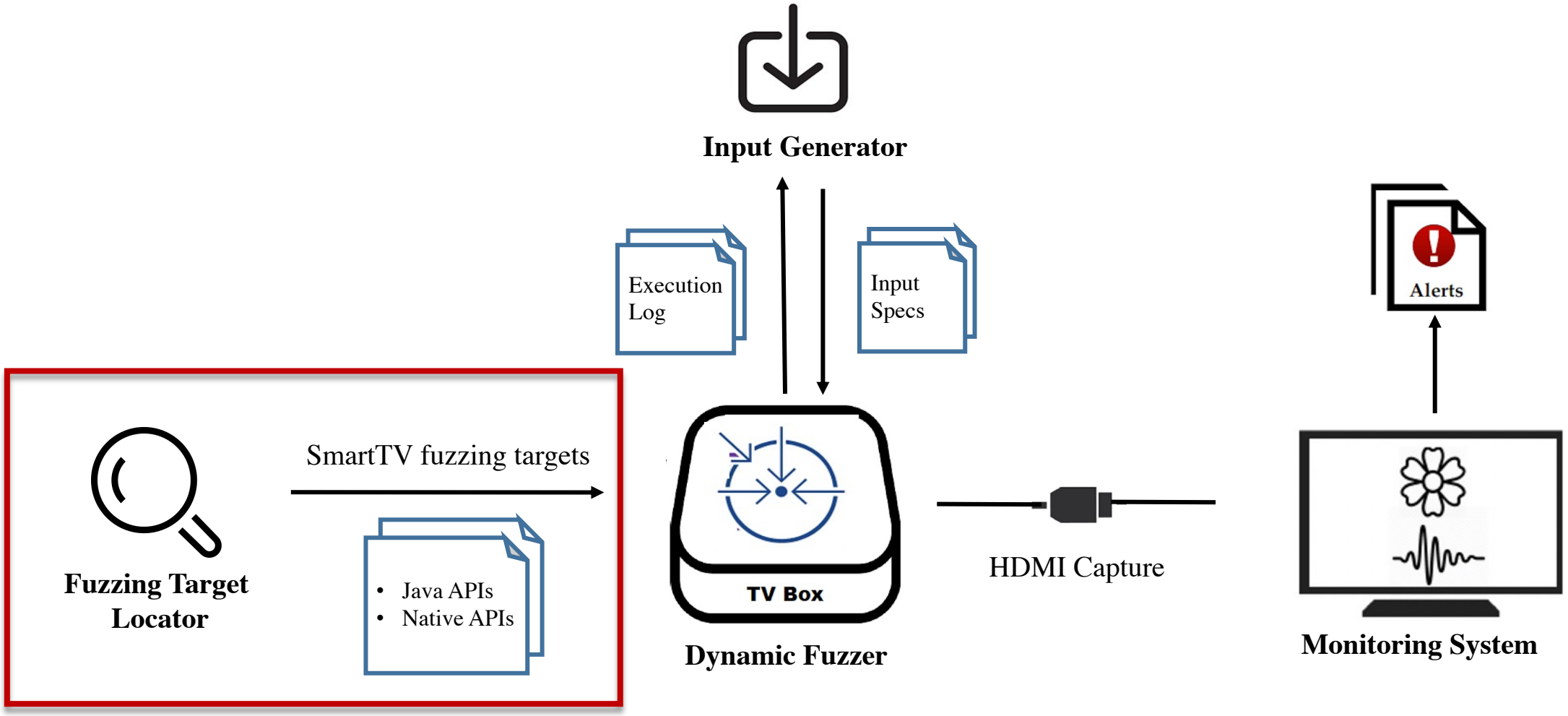
Detecting SmartTV Vulnerabilities

- We develop a specialized analysis framework to uncover hidden flaws, caused by unprotected APIs.

- **Why can't we directly adopt static analysis tools?**
  - Additions are implemented in C++ and / or Java
- **Why can't we directly adopt existing testing approaches?**
  - Assessing execution feedback is challenging

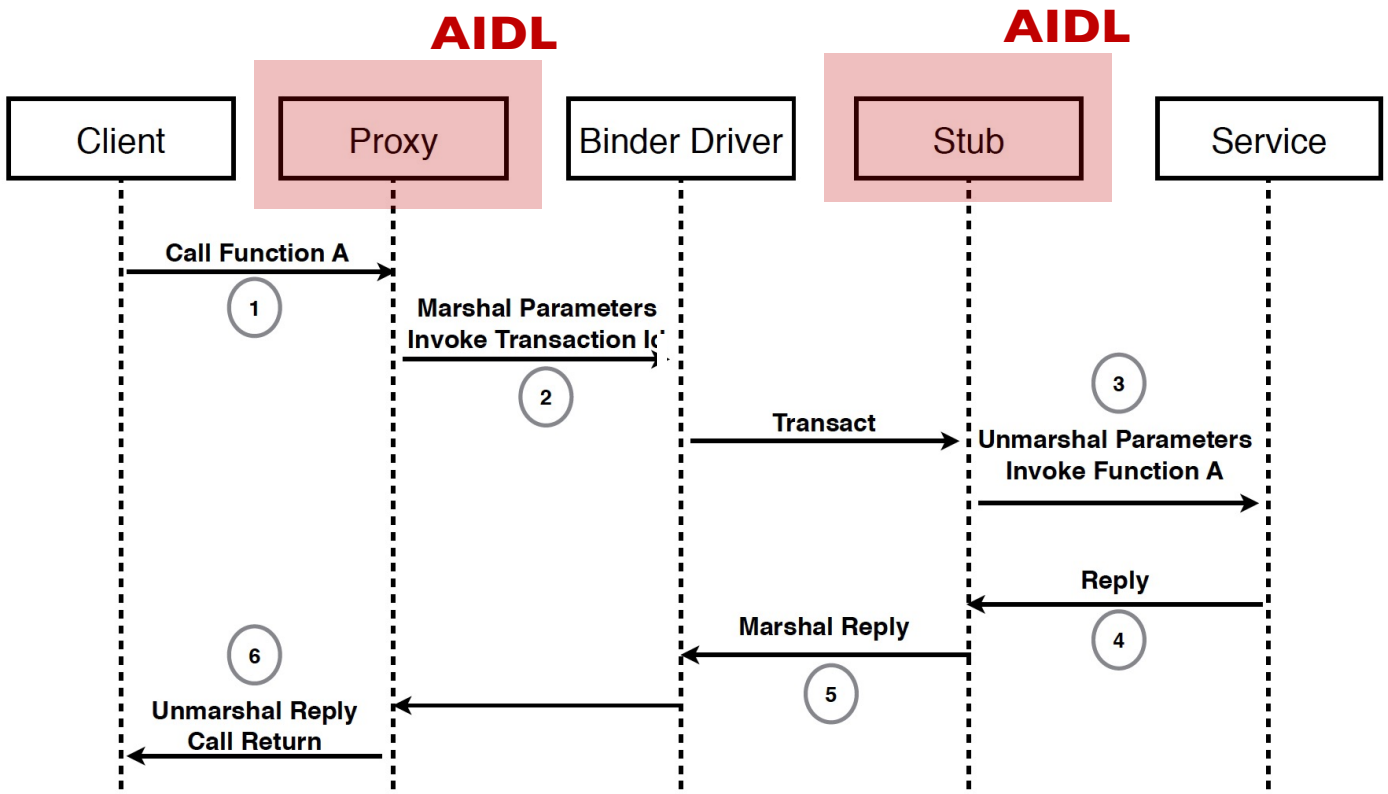
The Audio / Visual behavior is decoupled from the internal states → the system might be functioning correctly when the display / sound is messed up.

Our Approach: Fuzz-testing



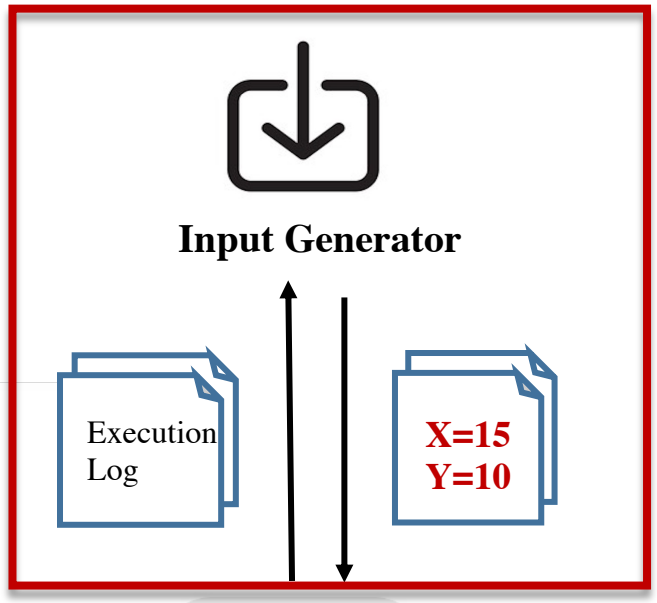
Fuzzing Target locator

- We recover native API interfaces at the low-level Binder IPC through binary analysis.
- Recovering Native APIs Interfaces: Binder transaction ids, arguments types and order.



Our Approach: Fuzz-testing

```
BatteryChangedJob: Running battery changed worker
ImagePlayerService: max x scale up or y scale up is 16
DiskIntentProviderImpl: Successfully read intent from disk
MediaPlayer: not updating
```



- Challenges to address:

1. Recognizing target logs
2. Recognizing input validations

Deep Learning for Message Classification

```
Intent buildRequestPermissionsIntent(String[] permissions) {
  if (ArrayUtils.isEmpty(permissions)) {
    Log.d("permission cannot be null or empty");
    return;
```

Input Validation

Input Validation

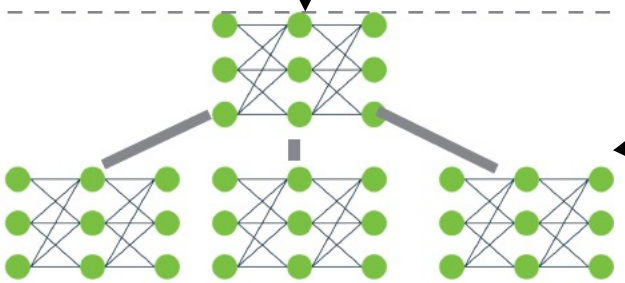
```
void requestBugReportWithDescription(String shareTitle,….){
  if (shareTitle.length() > 50) {
    String err = "shareTitle should be less than " +
                 50 + " characters";
    throw new IllegalArgumentException(err);}
  Slog.d(TAG, "Bugreport notification title" + shareTitle);
```

Input Validation

Input Validation

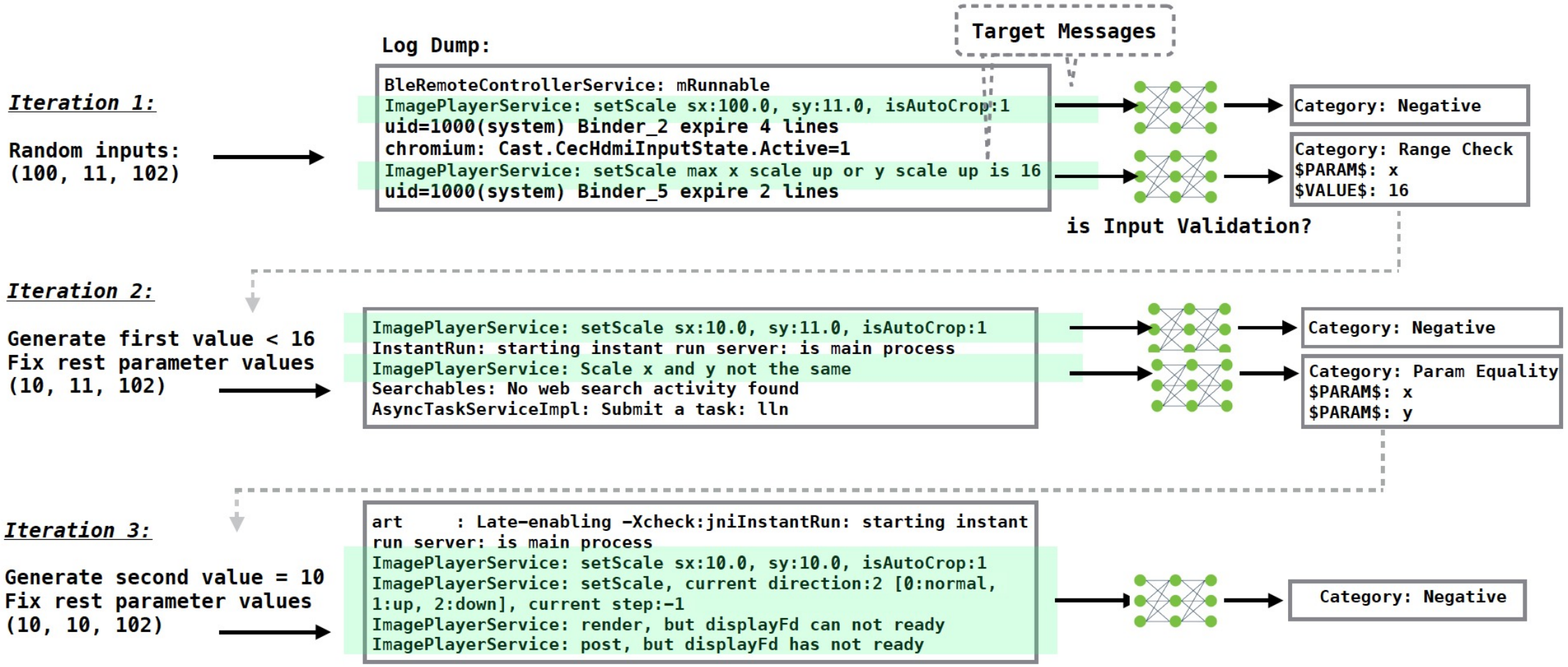
Non-Input Validation

●●●

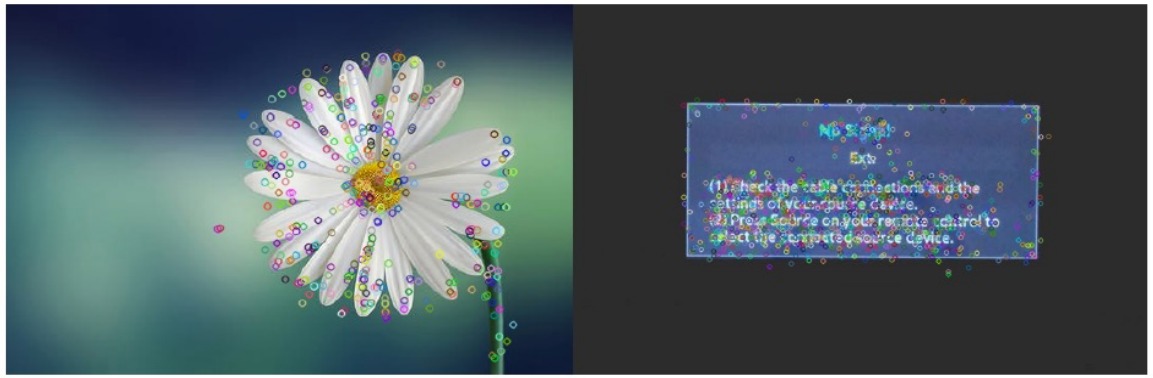


Log-Guided Fuzzing

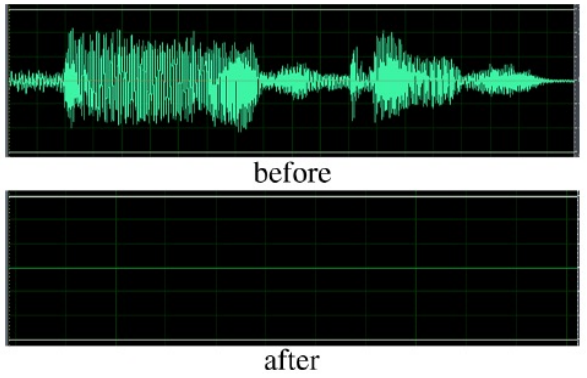
Example: fuzzing ABC(int, int, float)



Monitoring System

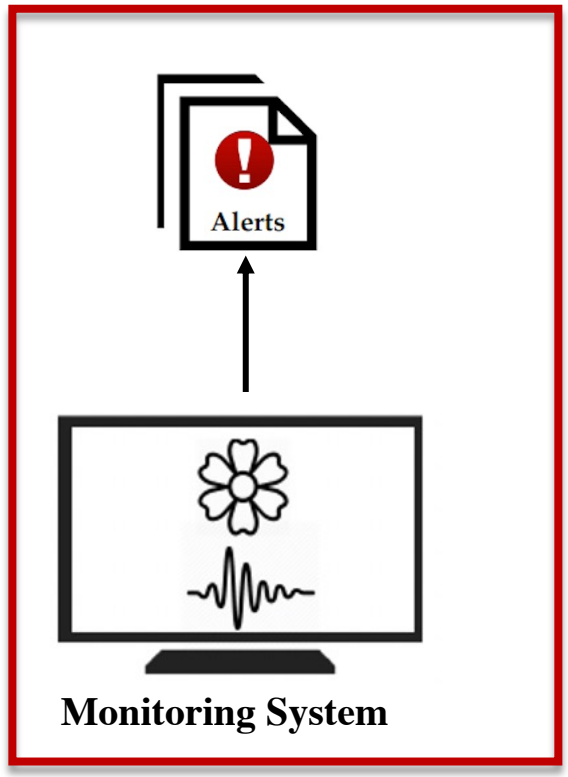


(a) Display before and after invoking `DisplayManager.enableInterface`



(b) Audio comparison

HDMI Capture



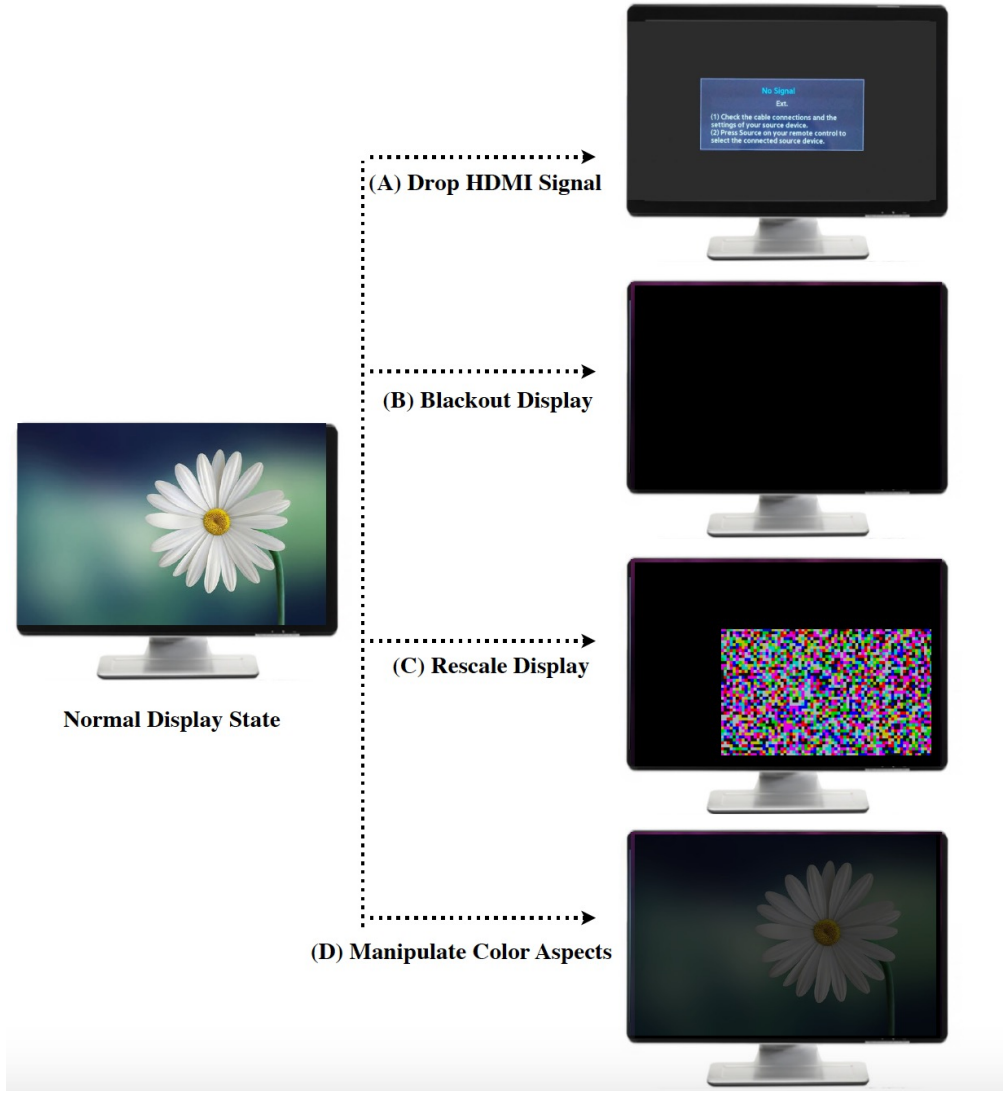
Evaluation

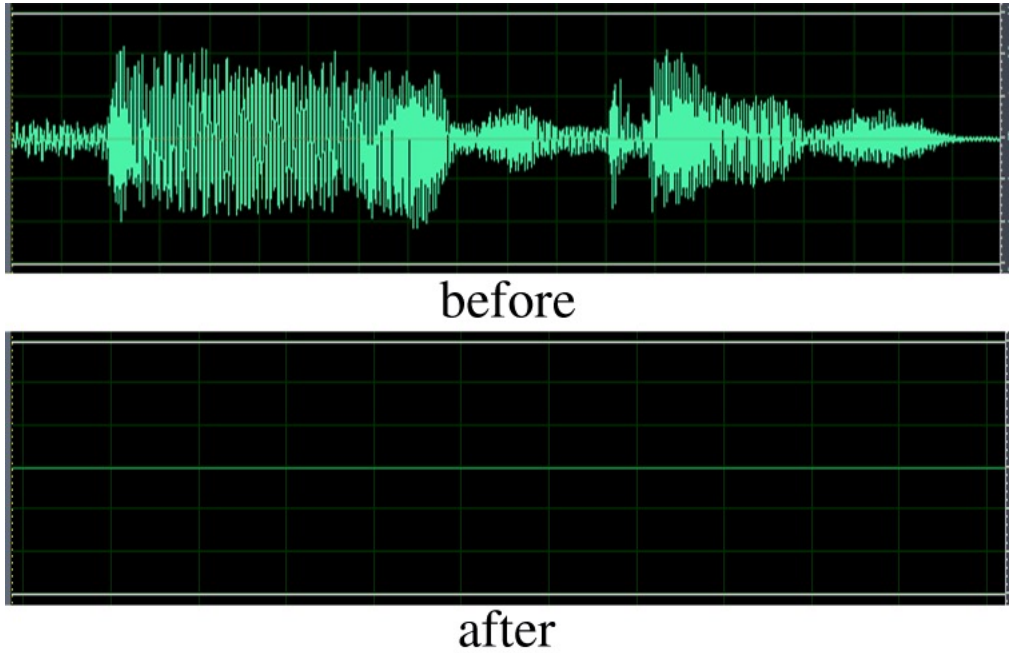
Cyber threats and Memory Corruptions

Description	Victim Devices (s)	Log-Guided Input Inference	Log-Guided Feedback Inference	External Feedback	Exposing Time	
					Random	Guided
Corrupt boot environment variables	H96 Pro	✓	✓	✓	Timed out	0.11h
Overwrite System Directories	Nvidia Shield	✓	✓	✓	Timed out	4.71h
Delete Files in internal memory	Nvidia Shield	✓	✓	✓	Timed out	2.14h
inject mouse coordinates	V88, Max	✗	✗	✓	0.03h	0.04h
inject mouse coordinates	V88, Max	✗	✗	✓	0.03h	0.03h
Change persistent system properties	Q+	✓	✓	✗	Timed out	0.14h
read highly-sensitive data	Q+	✓	✓	✗	Timed out	0.14h
overwrite certain system files	Q+	✓	✓	✗	Timed out	0.19h
read highly-sensitive data	Q+	✓	✓	✗	Timed out	0.15h
create hidden files under /sdcard/	GT King	✓	✓	✗	Time out	0.05h
reboot device into recovery mode	MIBOX4	✗	✓	✓	0.03h	0.03h

Evaluation

Physical Vulnerabilities





Thank you!

Q & A

Contact:
yaafer@waterloo.ca
youwei@ruc.edu.cn