IoTReplay: Troubleshooting COTS IoT Devices with Record and Replay

Kaiming Fang and Guanhua Yan
Binghamton University

IoT security

1. wide spread of IoT devices
2. limited capabilities for security testing
3. existing attacks: Mirai against Dyn's DNS service

Challenges for troubleshooting IoT devices

1. blackboxness of IoT devices

2. real world use

3. IoT device interacts with physical environment

Contribution

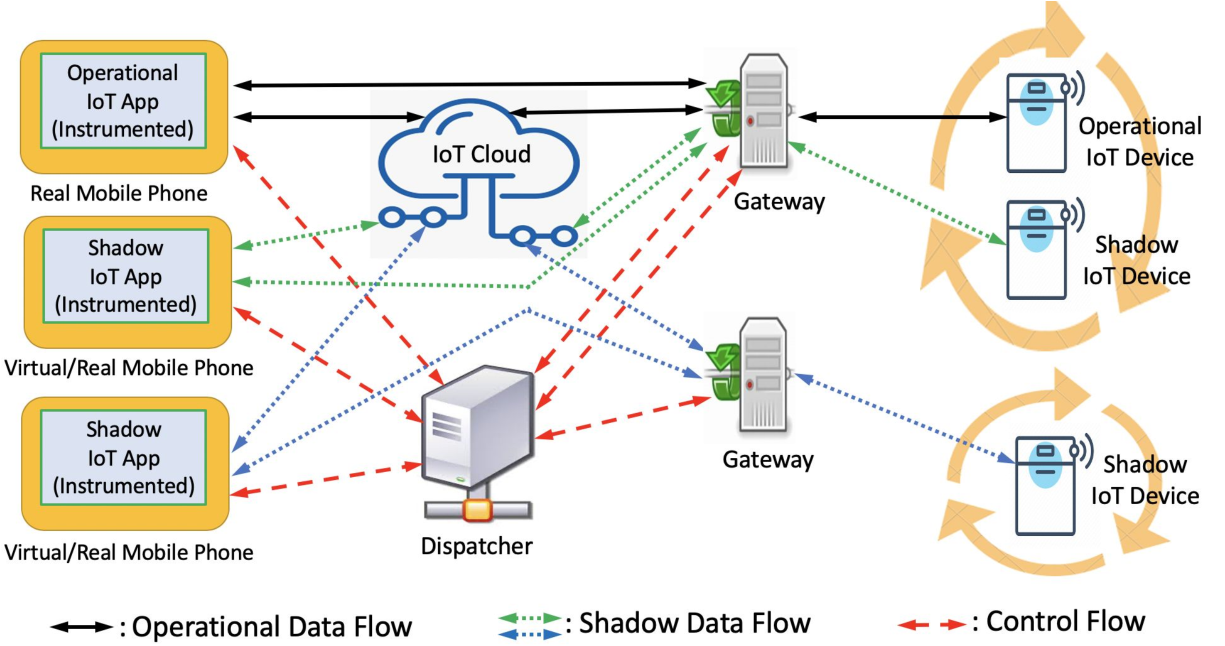
1. we identify contextual events for record & replay

2. we design a scalable architecture for IoTReplay

3. we implement IoTReplay with static and dynamic instrumentation

4. we evaluate the effectiveness of IoTReplay

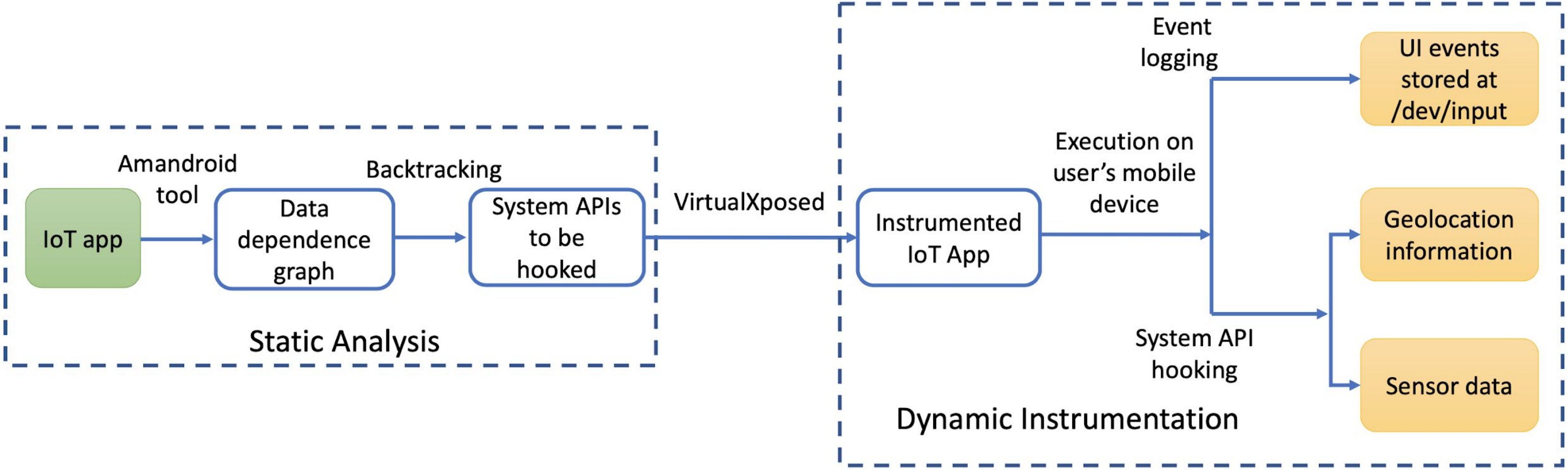
System architecture



Contextual events in record & replay

1. IoT App: UI events, geolocation, sensor data
2. IoT Device: human or physical
3. network traffic: partial

Record for IoT App



Replay for IoT App and devices

- Geolocation information and sensor data: xposed hooking
  - Geolocation APIs: `WifiManager.getScanResults`, `LocationManager.getGpsStatus` …
  - Sensor APIs: `onSensorChanged(SensorEvent se)`
- UI events: translation between different phones
  - `ABS_MT_TRACKING_ID -1 --> BTN_TOUCH`
- Exotic network traffic: replay the payloads

Experiment setup

COTS IoT devices:

- Google Nest camera
- D-Link smart plug
- TCL Roku TV
- Tycam smart camera

Android devices:

- Android virtual device
- Samsung Galaxy S5
- Nexus 5

Android version 7.1 AOSP

Workstation PC: i7-9700, 32GB

Execution trajectory and similarity

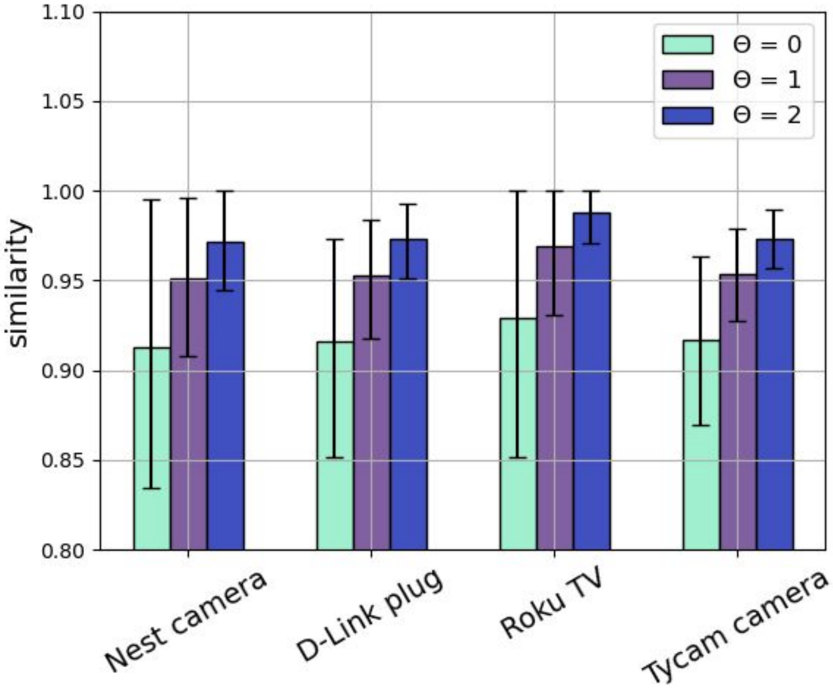
- trajectory: sequences of methods, that have been invoked for every message received by IoT App
  - example: `[‘a()Z’, ‘c(Lh/v/b$a;)V’, ‘a(Z)V’, ‘b()J’, ‘a(FF)V’]`
  - in smali format
- similarity: edit distance, normalized to (0, 1)

Results: execution similarities

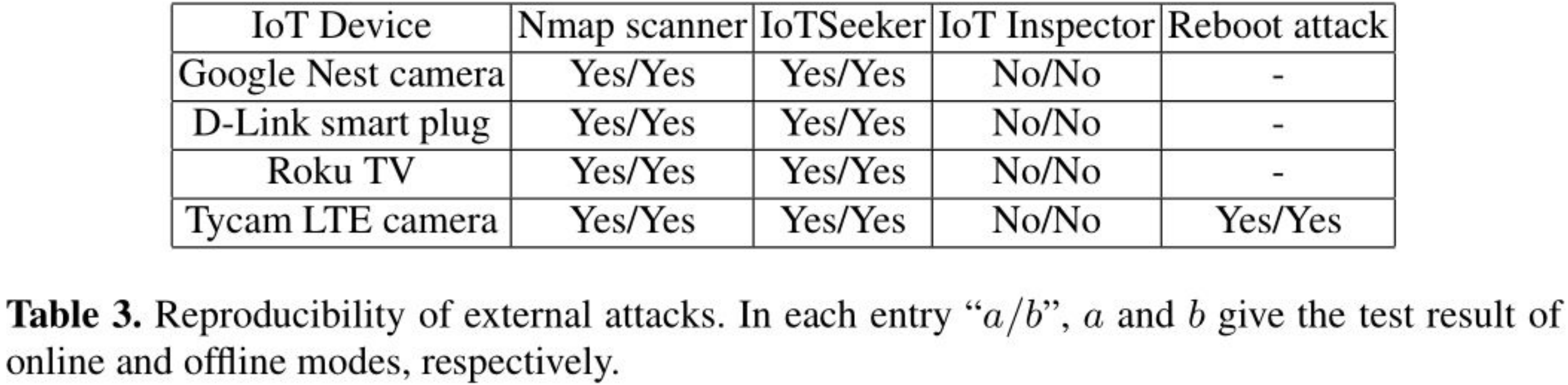
θ = 0: avg similarity > 0.90

θ = 1: avg similarity > 0.95

θ = 2: avg similarity > 0.97



Results: attack record and replay



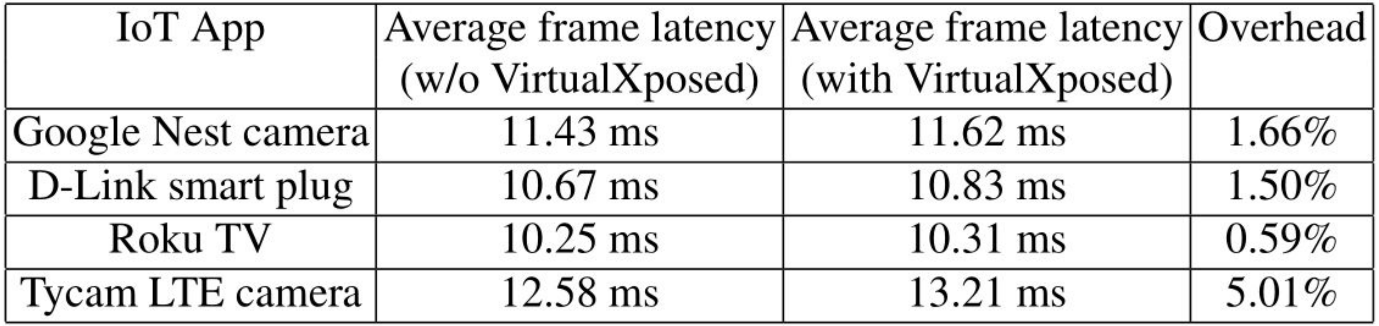
IoT Device	Nmap scanner	IoTSeeker	IoT Inspector	Reboot attack
Google Nest camera	Yes/Yes	Yes/Yes	No/No	-
D-Link smart plug	Yes/Yes	Yes/Yes	No/No	-
Roku TV	Yes/Yes	Yes/Yes	No/No	-
Tycam LTE camera	Yes/Yes	Yes/Yes	No/No	Yes/Yes

Table 3. Reproducibility of external attacks. In each entry "a/b", a and b give the test result of online and offline modes, respectively.

online: record & replay in real-time

offline: record & replay in different time

Results: performance



IoT App	Average frame latency (w/o VirtualXposed)	Average frame latency (with VirtualXposed)	Overhead
Google Nest camera	11.43 ms	11.62 ms	1.66%
D-Link smart plug	10.67 ms	10.83 ms	1.50%
Roku TV	10.25 ms	10.31 ms	0.59%
Tycam LTE camera	12.58 ms	13.21 ms	5.01%

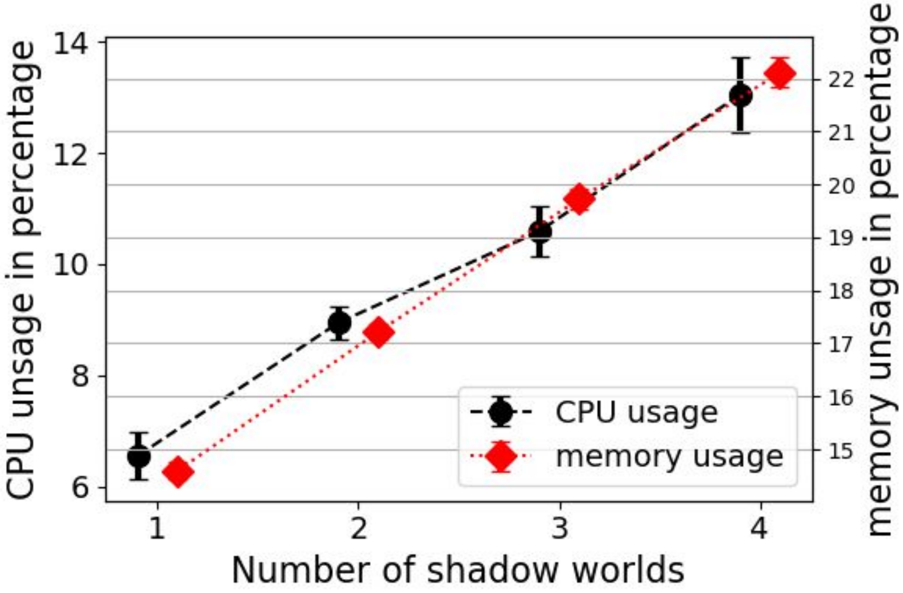
Table 4. Average frame generation time

performance overhead is negligible

Results: Scalability

One-to-multiple record & replay:

- one operational world
- multiple shadow world



Conclusion

1. we presents the design, and implementation details of IoTReplay

2. we perform extensive experiments using four types of COTS IoT devices

3. IoTReplay results in good similarities and incurs negligible performance

Thanks for watching