



Group-4

Literature study-2

Team members

Jaya Sahithi Aravapalli
Amulya Aregunta
Azhar Chara



Paper 1

SoK: Security Evaluation of
Home-Based IoT
Deployments



Introduction

Internet of Things (IoT) devices are increasingly found in everyday homes, providing useful functionality for devices such as TVs, smart speakers, and video doorbells.



Problems

The heterogeneity of home-based IoT devices contributes to these insecurities because although core functionalities are alike, specific features based on the device type can be vastly different.



Problems

some vendors leave service backdoors in their devices that are later discovered and exploited by botnets.

Even unsophisticated criminal groups are taking advantage of the rampant insecurities to run distributed denial-of-service (DDoS) attacks.





Alexa, unlock the front door.



Methodology

Authors proposed a modeling methodology to study home-based IoT devices and evaluate their security posture based on component analysis.

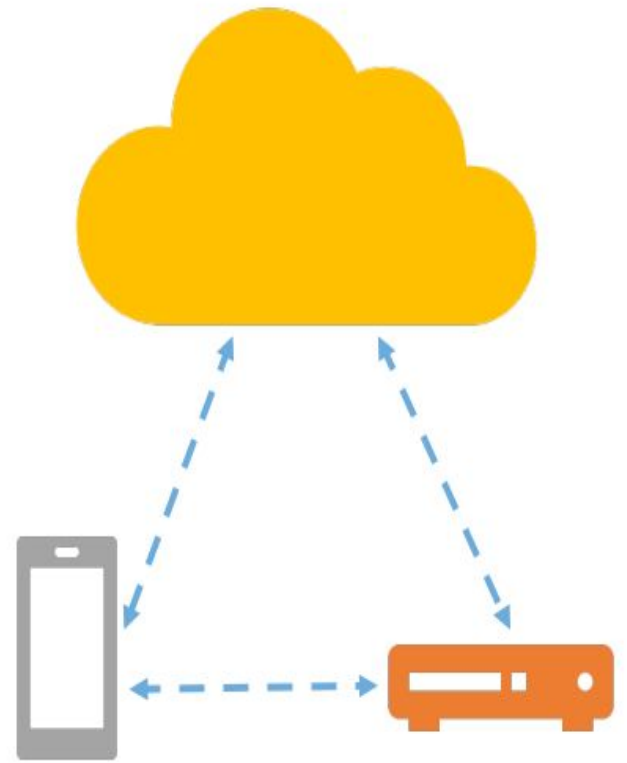
IoT Components

Device

Mobile
App

Cloud
Endpoints

Network





Methodology(Cont.)

Additionally, authors evaluated 45 home-based IoT devices that are available on the market today and provided an overview of their security properties across the IoT components.



Experiment Setup

Network setup has three main components, the IoT subnet, custom Linux gateway, and an assessment machine.

Lab Setup





Device

The evaluation uses

Nessus scanner to assess the device and cloud endpoints;

Kryptowire, MobSF, and Qark to assess the mobile applications;

Nessus Monitor, ntopng, ssllsplit, and Wireshark to assess the communication protocols.

Results

Device	Device Services Appendix A Table VI		Mobile Application Appendix A Table IV			Cloud Endpoints Appendix A Table V		Communication Appendix A Table VII	
	Running Services	Security Issues	Over-privileged	Sensitive Data	Crypto Issues	SSL Issues	Service Issues	MITM	Encryption
Amazon Echo	1	0	✓	✓		✓			●
Amazon Fire TV	1	0	✓	✓			✓		○
Apple HomePod	4	0	—	—	—	✓			●
Apple TV (4th Gen)	3	0	—	—	—	✓			●
August Doorbell	1	0	✓	✓		✓	✓	✓	○
Belkin Netcam	1	1		✓		✓	✓	✓	○
Belkin WeMo Crockpot	0	0		✓		✓	✓	✓	○
Belkin WeMo Link	1	1		✓			✓	✓	○
Belkin WeMo Motion	1	1		✓		—	—	✓	○





Device Grade

80.95% (B)

Mobile Grade

69.23% (D)

Network Grade

89.29% (B)

Cloud Grade

57.61% (F)

Device



Harmon Kardon Invoke



Conclusion

This work systematized the existing literature for home based IoT devices through an abstract model that allowed them to derive insights.

They made results available and the evaluation dataset on portal and invite researchers to contribute and reproduce their work.



Paper 2

SoK: The Challenges, Pitfalls, and Perils of Using Hardware Performance Counters for Security



Introduction

- Hardware Performance Counters (HPCs) have been available in processors for more than a decade. These counters can be used to monitor and measure events that occur at the CPU level.
- when it comes to the use of HPC's for security applications, measurement imprecisions or incorrect assumptions regarding the measured values can undermine the offered protection.

What did they do?



- (i) Study the best practices for obtaining accurate measurement of events using performance counters,
- (ii) Understand the challenges and pitfalls of using HPCs in various settings.
- (iii) Explore ways to obtain consistent and accurate measurements across different settings and architectures.
- (iv) Extended the seminal work of Weaver and McKee from almost 10 years ago on non-determinism in HPCs, and applied our findings to 56 papers across various application domains.

- They found that the use of HPCs for security purposes is in stark contrast to other areas, as evidenced by the increasing number of techniques that rely on HPC measurements for offensive and defensive research.
- Architectural events comprise events that remain consistent across different processor architectures, *e.g.*, instructions, branches, cycles.
- Non-architectural events consist of events that are specific to the micro-architecture of a given processor, *e.g.*, cache accesses, branch prediction, and TLB accesses. Unlike architectural events, non-architectural events vary among processor architectures

TABLE I: Hardware Events


Architectural Events	Description	Non-architectural Events	Description
1. Ins	Instruction retired	9. Uops_Retired	All micro-operations that retired
2. Clk	Unhalted core cycles	10. Mem_Load_Uops_Retired	Retired load uops
3. Br	Branch instructions	11. Mem_Store_Uops_Retired	Retired store uops
4. Arith_Ins	Arithmetic instructions	12. Br_Miss_Pred_Retired	Mispredicted branches that retired
5. Call	Near call instructions	13. Ret_Miss	Mispredicted return instructions
6. Call_D	Direct near call instructions	14. Call_D_Miss	Mispredicted direct call instructions
7. Call_ID	Indirect near call instructions	15. Br_Far	Far branches retired
8. Ret	Near return instructions	16. Br_Inst_Exec	Branch instructions executed
		17. ITLB_Miss	Misses in ITLB
		18. DTLB_Store_Miss	Store uops with DTLB miss
		19. DTLB_Load_Miss	Load uops with DTLB miss
		20. LLC_Miss	Longest latency cache miss

BACKGROUND AND CHALLENGES



- 1) Configuring the counters: only a limited number of counters can be used simultaneously.
- 2) *Reading Counter Values*: Performance counters can be read by either sampling or polling.
 - Polling: The counters can be read at any instant. Counters are read using the MSRs. For that purpose, Intel uses specific instructions (*rdmsr*, *wrmsr*) to read from and write to MSRs, respectively.
 - Event-based sampling: HPCs also support sampling of various metrics based on the occurrence of events. This feature is enabled in most CPUs through a specific interrupt, called *Performance Monitoring Interrupt (PMI)*,

Challenges and Pitfalls



- External sources: The runtime environment may vary across runs.
- Non-determinism: Weaver provided an overview of the impact of non-deterministic events (e.g., hardware interrupts like periodic timers) in HPC settings.
- Overcounting: Performance counters may overcount certain events on some processors.
- Variations in tool implementations: Many tools have been developed to help obtaining performance counter measurements.



PROPER INSTANTIATION AND USAGE

- Context switch monitoring
- Interrupt handling
- Process filtering upon PMI
- Minimizing the impact of non-deterministic events



Trends and Concerns

Given the major issues of non-determinism and overcounting in HPCs, they decided to look more closely at the 56 papers to assess

i) whether the authors acknowledged the impact of these issues in their analysis

ii) whether they addressed the challenges that arise due to non-determinism,

Case study : On weak security Foundations



Two of the most prominent areas of security research that have used performance counters include exploit protection and malware detection.

A. Using HPCs for ROP detection

The CS-PMI approach correctly raises an alert 79 times, whereas the PMI approach does so 77 times. Hence both approaches have ample opportunities to detect the prescribed signature for a ROP attack.

B. Using HPCs for Malware Detection



Result of Case Study

Our experiments show that differences in the way the data is recorded and the approaches taken to do so, not only affect the accuracy of the resulting techniques, but also impact the reproducibility of the research findings. For that reason, we do not recommend the use of PMI.

Recommendations



- First, (for both reviewers and authors alike) empirical results based on performance counters should not be compared with those from other profiling tools .
- Modern processors are inherently complex, so it is important that authors verify HPC-based findings on different CPU architectures.
- For profiling a specific application, it is imperative that per-process filtering of events is applied by saving and restoring the counter values at context switches.
- For critical applications such as security defenses, the issues related to non-determinism and overcounting cannot be overlooked.



Conclusion

Overall, the fact that the number of events being added to processors is growing faster than the number of counters, is unsettling. It has come to the point where machine learning is required to manage the abundance of information.

Thank you