When Side-channel Meets Malware

Duy-Phuc Pham[2], Damien Marion[1] and Annelie Heuser[1]

[2] *Trellix*
[1] *Univ Rennes, CNRS, Inria, IRISA Rennes, France*

Abstract

The Internet of Things (IoT) is a collection of interconnected devices, each becoming increasingly complicated and numerous. They frequently employ modified hardware and software without taking security risks into account, which makes them a target for cybercriminals, especially malware and rootkit crafter. In this extended abstract, we will present two strategies for exploiting electromagnetic side channels to address two issues: rootkit detection difficulties and malware categorization challenges in the presence of obfuscations. Both tactics center on IoT devices, target ARM (raspberry-Pi) and MIPS (CI.20) architectures, and use machine/deep learning techniques.
These results were published at,
• **ACSAC-2021:** "Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification" [1] (with an extended version presented at hardwear.io'22 USA),
• **RAID-2022:** "ULTRA: Ultimate Rootkit Detection over the Air"[2].

The talk will highlight all the results obtained from the ARN project "Automated Hardware Malware Analysis" (AHMA - Annelie's JCJC) and the ongoing next-steps.

Keywords

Malware classification, obfuscation, side-channel analysis, rootkit detection, SDR (software defined radio), machine learning/deep learning, Electromagnetic, IoT devices

1. Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification

We outline a cutting-edge method for determining the types of threats that are aimed at the device by leveraging side channel information. Even in the face of obfuscation tactics that may prohibit static or symbolic binary analysis, a malware analyst can use our approach to gain exact knowledge about the type and identity of malware. We gathered 100,000 measurement traces from an IoT device that was hacked using a variety of real-world malware types. A picture of our setup is available in Figure 1. The target device doesn't need to be changed in any way for our solution to work. As a result, it can be deployed without any overhead independently of the resources at hand. Our strategy also has the benefit of being difficult for malware authors to identify and avoid. In our tests, we achieved an accuracy of 99.82% in predicting three generic malware categories (and one benign class). Even more, our results

C&ESAR'23: Computer & Electronics Security Application Rendezvous, Nov. 21-22, 2023, Rennes, France
duyphuc.pham@trellix.com (D. Pham); damien.marion@irisa.fr (D. Marion); annelie.heuser.fr (A. Heuser)
https://phdphuc.github.io/web/ (D. Pham); https://damien-marion.github.io/ (D. Marion);
https://axnxlxe.github.io/ (A. Heuser)

show that we are able to classify altered malware samples with unseen obfuscation techniques during the training phase, and to determine what kind of obfuscations were applied to the binary, which makes our approach particularly useful for malware analysts.

Figure 1: Probe setup consists of a H-Field probe placed 45 degree above the system processor.

Setup description

- **Targets:** Raspberry Pi 2B (ARM processor), CI20 (MIPS processor),
- **Acquisition:** Picoscope 6407, H-Field Probe (Langer RF-R 0.3-3), connected to a H-Field Probe (Langer RF-R 0.3-3), where the EM signal is amplified using a Langer PA-303 +30dB (Fig. 1).
- **Samples, labels, number of traces:** all information available in tabular 3.

Resources

- code:
  → https://github.com/ahma-hub
- data:
  → https://zenodo.org/record/5414107
- talk at hardwear.io'22 USA of an extended version (with a additional target board CI20 embedded a MIPS processor):
  → https://m.youtube.com/watch?v=oCohqwfUpsQ&feature=youtu.be

State-of-the-art A summary of the state-of-the-art, regarding malware analysis throught side-channel is available in Tab. 1.

Article	SCM detection	Anomaly detection	SCM classification	Real-world SCM	Real-world analysis environment	Samples size	Variations	Benign dataset	Window size	Open source	Device under test
WattsUpDoc [3]	✗	-	-	✗	-	15	-	-	5s	-	Windows XP Embedded 664 MHz
IDEA [4]	-	✗	-	-	-	3	-	-	<40μs	-	AT328p 16MHz, Cortex A8
REMOTE [5]	-	✗	-	✗	-	3	-	-	<10ms	-	Single-core ARM 1Ghz
Wang *et al.* [6]	-	✗	-	-	-	1	-	-	10s	-	Raspberry Pi, Arduino, Siemens PLC
Khan *et al.* [7]	✗	-	-	-	-	3	-	-	<150μs	-	Cyclone II FPGA & NIOS II soft-processor
DeepPower [8]	✗	-	✗	✗	-	5	-	-	1s	-	MIPS/ARM OpenWRT
Chawla *et al.* [9]	✗	-	✗	✗	-	137	-	✗	10s	-	Android Intrinsyc Open-Q 820
Our paper	*(✗)**	-	✗	✗	✗	35	✗	✗	2.5s	✗	Multi-core, 900 Mhz ARM

Table 1
Comparison with related works on side-channel malware (SCM) analysis using EM or power consumption. *(*)*: Our paper aims at SCM classification, however we also achieve good results in SCM detection scenario.

2. ULTRA: Ultimate Rootkit Detection over the Air

We suggest the ULTRA framework, which operates outside of the “box” (literal device) and requires no resources from the target device, , as visible on Figure 2, to identify rootkits effectively and efficiently. A software-defined radio is used by ULTRA to measure electromagnetic emission, preprocess signals, and then detect and categorize rootkit activities. ULTRA baits the rootkit to elicit action. We focus on two IoT devices with ARM and MIPS architectures as use cases. During the offline learning phase, the suggested method produced encouraging results with high accuracy for detecting both known and unknown rootkits. The classification of rootkit families and distinctive variants, obfuscated rootkits, probe dislocation, benign noise (kernel) activities, and comparison with software-based solutions are all part of our experimental investigation.

Setup description

- **Targets:** Raspberry Pi 2B (ARM processor), CI20 (MIPS processor),
- **Acquisition:** SDR (software define radio, hackRF), H-Field Probe (Langer RF-R 0.3-3), connected to a H-Field Probe (Langer RF-R 0.3-3), where the EM signal is amplified using a Langer PA-303 +30dB (Fig. 2).

Resources

- code:
  → https://gitlab.com/ultra-RK/ultra
- data:
  → https://zenodo.org/record/5902451

State-of-th-art A summary of the state-of-the-art, regarding rootkit detection by side-channel is available in Tab. 2.

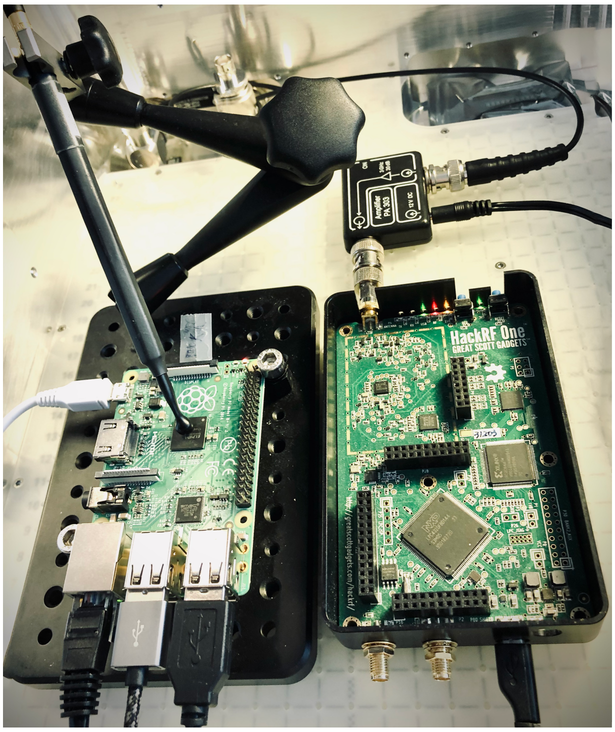

Figure 2: ULTRA framework data acquisition consists of a H-field probe, an amplifier, an HackRF and the target raspberry Pi.

Table 2
Comparison with related works on kernel-level or user-level rootkit (user RK) detection using different side-channel analysis techniques: HPC, DMA, Power consumption (Power) and EM.

	Article	WnP	Classification	Baits	ML	DL	Sample size	Open source	Benign set	User RK	Detection latency	Targeted device(s)/Architecture
HPC	Numchecker [10]	-	-	✗	-	-	8	-	-	-	262.3ms	32-bit Linux PC
HPC	[11]	-	-	-	✗	-	5	-	-	-	45s	Windows 7 Intel (VMWare)
HPC	LKRDet[12]	-	-	✗	✗	-	4	✗	-	-	2.91s	ARM Cortex-A53 (TEE)
DMA	Copilot [13]	-	-	-	-	-	12	-	-	-	30s	PCI-compatible Intel PC Linux
DMA	Gibraltar [14]	-	-	-	-	-	23	-	✗	-	20s	PCI-compatible Intel PC Linux
Power	[15]	-	-	-	✗	✗	5	-	-	✗	>5m	PC Windows 10 & Ubuntu 14
Power	[16]	-	-	-	✗	-	5	-	-	-	>1m	Dell OptiPlex 755 Windows 7
EM	ULTRA	✗	✗	✗	✗	✗	9	✗	✗	✗	1.3s	ARM Raspberry Pi & MIPS Ci20

3. Ongoing Next-steps

Currently, we are focusing on the reproducibility of our results. First, we are in contact with researchers that are building the same setup. Second, we built student projects to make the ULTRA framework more portable using a Jetson Nano board that embeds a GPU. Finally, we are collaborating to improve the classification step.

Acknowledgments

The work was supported by the French Agence Nationale de la Recherche (ANR) under reference ANR-18-CE39-0001 (AHMA). We thank our colleague Ronan Lashermes and Olivier Zendra who provided hardware and side-channel insights and greatly assisted this work.

References

[1] D. Pham, D. Marion, M. Mastio, A. Heuser, Obfuscation revealed: Leveraging electromagnetic signals for obfuscated malware classification, in: ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021, ACM, 2021, pp. 706–719. URL: https://doi.org/10.1145/3485832.3485894. doi:10.1145/3485832.3485894.

[2] D. Pham, D. Marion, A. Heuser, ULTRA: ultimate rootkit detection over the air, in: 25th International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2022, Limassol, Cyprus, October 26-28, 2022, ACM, 2022, pp. 232–251. URL: https://doi.org/10.1145/3545948.3545962. doi:10.1145/3545948.3545962.

[3] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, W. Xu, K. Fu, Wattsupdoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices, in: 2013 USENIX Workshop on Health Information Technologies (HealthTech 13), USENIX Association, Washington, D.C., 2013. URL: https://www.usenix.org/conference/healthtech13/workshop-program/presentation/clark.

[4] H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, R. L. Callan, A. Yeredor, M. Prvulovic, A. Zajic, Idea: Intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems, IEEE Transactions on Dependable and Secure Computing (2019) 1–1.

[5] N. Sehatbakhsh, A. Nazari, M. Alam, F. Werner, Y. Zhu, A. Zajic, M. Prvulovic, Remote: Robust external malware detection framework by using electromagnetic signals, IEEE Transactions on Computers 69 (2020) 312–326.

[6] X. Wang, Q. Zhou, J. Harer, G. Brown, S. Qiu, Z. Dou, J. Wang, A. Hinton, C. A. Gonzalez, P. Chin, Deep learning-based classification and anomaly detection of side-channel signals, in: Cyber Sensing 2018, volume 10630, International Society for Optics and Photonics, 2018, p. 1063006.

[7] H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, M. Prvulovic, A. G. Zajic, Malware detection in embedded systems using neural network model for electromagnetic side-channel signals, J. Hardware and Systems Security 3 (2019) 305–318. URL: https://doi.org/10.1007/s41635-019-00074-w. doi:10.1007/s41635-019-00074-w.

[8] F. Ding, H. Li, F. Luo, H. Hu, L. Cheng, H. Xiao, R. Ge, Deeppower: Non-intrusive and deep learning-based detection of iot malware using power side channels, in: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, 2020, pp. 33–46.

[9] N. Chawla, H. Kumar, S. Mukhopadhyay, Machine learning in wavelet domain for electromagnetic emission based malware analysis, IEEE Transactions on Information Forensics and Security 16 (2021) 3426–3441. doi:10.1109/TIFS.2021.3080510.

[10] X. Wang, R. Karri, Numchecker: Detecting kernel control-flow modifying rootkits by using hardware performance counters, in: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2013, pp. 1–7.

[11] B. Singh, D. Evtyushkin, J. Elwell, R. Riley, I. Cervesato, On the detection of kernel-level rootkits using hardware performance counters, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 483–493.

[12] X. Jiang, M. Lora, S. Chattopadhyay, Efficient and trusted detection of rootkit in iot devices via offline profiling and online monitoring, in: Proceedings of the 2020 on Great Lakes Symposium on VLSI, 2020, pp. 433–438.

[13] N. L. Petroni Jr, T. Fraser, J. Molina, W. A. Arbaugh, Copilot-a coprocessor-based kernel runtime integrity monitor., in: USENIX security symposium, San Diego, USA, 2004, pp. 179–194.

[14] A. Baliga, V. Ganapathy, L. Iftode, Detecting kernel-level rootkits using data structure invariants, IEEE Transactions on Dependable and Secure Computing 8 (2011) 670–684. doi:10.1109/TDSC.2010.38.

[15] P. Luckett, J. T. McDonald, W. B. Glisson, R. Benton, J. Dawson, B. A. Doyle, Identifying stealth malware using cpu power consumption and learning algorithms, Journal of Computer Security 26 (2018) 589–613.

[16] R. Bridges, J. H. Jiménez, J. Nichols, K. Goseva-Popstojanova, S. Prowell, Towards malware detection via cpu power consumption: Data collection design and analytics, in: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 1680–1684.

Binaries names	#	Types tags	Family tags	Virtualization tags	Packer tags	Obfuscation tags	Executable tags	Novelty (family) tags
random34	6000	benign	benign				random34	benign
mirai.arm7	6000	ddos	mirai	orig	not_packed		mirai	mirai [*]
mirai_addopaque	3000	ddos	mirai			addopaque	mirai_addopaque	mirai [*]
mirai_virtualize	3000	ddos	mirai	virtualized		virtualize	mirai_virtualize	mirai [+]
mirai_flatten	3000	ddos	mirai			flatten	mirai_flatten	mirai [+]
mirai-bcf	3000	ddos	mirai			bcf	mirai-bcf	mirai [*]
mirai-cfflatten	3000	ddos	mirai			cfflatten	mirai-cfflatten	mirai [+]
mirai-sub	3000	ddos	mirai			sub	mirai-sub	mirai [+]
upx-mirai	3000	ddos	mirai		packed	upx	mirai-upx	mirai [*]
gonnacry	6000	ransomware	gonnacry	orig	not_packed		gonnacry	gonnacry [*]
upx-gonnacry	3000	ransomware	gonnacry		packed	upx	gonnacry-upx	gonnacry [*]
aes-upx-gonacry	3000	ransomware	gonnacry		packed	upx	gonnacry-aes-upx	gonnacry [+]
aes-gonacry	3000	ransomware	gonnacry		not_packed		gonnacry-aes	gonnacry [+]
des-gonnacry	3000	ransomware	gonnacry		not_packed		gonnacry-des	
des-upx-gonnacry	3000	ransomware	gonnacry		packed		gonnacry-des-upx	
gonnacry_Virtualize2	3000	ransomware	gonnacry	virtualized		virtualize	gonnacry_virtualize2	gonnacry [*]
gonnacry_flatten	3000	ransomware	gonnacry			flatten	gonnacry_flatten	gonnacry [*]
gonnacry_bcf	3000	ransomware	gonnacry			bcf	gonnacry_bcf	gonnacry [*]
gonnacry_sub	3000	ransomware	gonnacry			sub	gonnacry_sub	gonnacry [*]
gonnacry_cfflatten	3000	ransomware	gonnacry			cfflatten	gonnacry_cfflatten	gonnacry [+]
gonnacry_addopaque	3000	ransomware	gonnacry			addopaque	gonnacry_addopaque	gonnacry [*]
maK_it4.19.57-v7+.ko	3000	rootkit	maK_it				rootkit_maK_it	rootkit [*]
kisni-4.19.57-v7+.ko	3000	rootkit	kisni				rootkit_kisni	rootkit [+]
bashlite	3000	ddos	bashlite	orig	not_packed		bashlite	bashlite [*]
bashlite_bcf	3000	ddos	bashlite			bcf	bashlite_bcf	bashlite [*]
bashlite_flatten	3000	ddos	bashlite			flatten	bashlite_flatten	bashlite [+]
bashlite_upx	3000	ddos	bashlite		packed	upx	bashlite_upx	bashlite [*]
bashlite_addopaque	3000	ddos	bashlite			addopaque	bashlite_addopaque	bashlite [*]
bashlite_cfflatten	3000	ddos	bashlite			cfflatten	bashlite_cfflatten	bashlite [*]
bashlite_sub	3000	ddos	bashlite			sub	bashlite_sub	bashlite [*]
bashlite_virtualize	3000	ddos	bashlite	virtualized		virtualize	bashlite_virtualize	bashlite [+]
playaudio	1000	benign	benign				playaudio	benign
recordcamera	1000	benign	benign				recordcamera	benign
takepicture	1000	benign	benign				takepicture	benign
encodevideo	1000	benign	benign				encodevideo	benign

Table 3: Malware tag map. The first column lists all malware and benign samples, followed by the number of recorded traces. Then each column refers to a scenario and gives for each sample the group it belongs to if it has been used. [*] (resp., [+]) means the sample has been used only during the training phase (resp. the testing phase), by default samples are used during both phases (80% for training, 20% for testing).