Institut de Recherche en Informatique
et Systèmes Aléatoires

Leveraging side-channel signals for IoT malware classification and rootkit detection

Univ Rennes 1 • CNRS • IRISA

Supervisors: Pierre-Alain Fouque, Annelie Heuser, Olivier Zendra

Jury members: Lejla Batina, Davide Balzarotti, Yerom-David Bromberg, Damien Couroussé

PhD Defense - Duy-Phuc Pham - duyphuc.pham@trellix.com

January 13, 2023

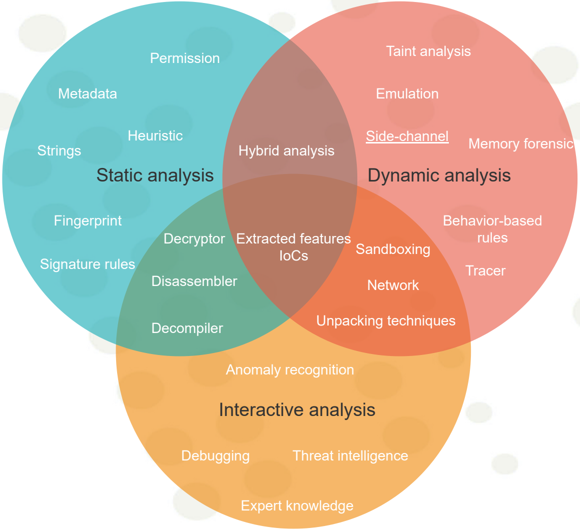
Outline

1. Introduction
   - Context
   - State of the art
2. AHMA: Obfuscated Malware Classification
3. ULTRA: Ultimate Rootkit Detection over the Air
4. Conclusion and Perspectives

- Trending of attacks on embedded devices.
- Difficulties for antivirus solutions on IoT devices: Resource constraints.
- Malware detection bypasses

Malware analysis techniques

- Malware detection
- Malware similarities
- Malware classification



Malware-evasion techniques

Static analysis

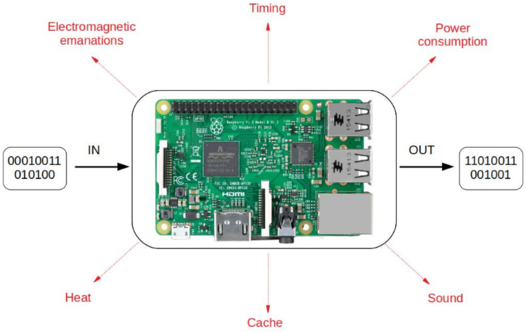
- Malware obfuscation
- Packers

Dynamic analysis

- Anti-debugging
- "Side-channel information"

Proposed solutions: dynamic analysis and side-channel

- Bare-metal device
- Side channel information
  - Power consumption
  - Electromagnetism (EM)
  - Cache, HPC (software)



- Anomaly detection using power consumption and EM.

RQ1

How can we build and setup an IoT malware classification and detection on embedded device using EM?

Contribution

Automated framework to automatically classify IoT malware by leveraging EM.

- Anomaly detection using power consumption and EM.
- Lack of research of side-channel detection for real-world malware.
- No variations regarding obfuscation and packers.

RQ2

If a malware analyst has a dataset of unlabeled binaries. Would it be possible to classify the dataset into labeled types, families, variants of malware or rootkits, obfuscation techniques used etc.?

Contribution

Real-world malicious and benign IoT dataset classification.

State of the art

- Anomaly detection using power consumption and EM.
- Lack of research of side-channel detection for real-world malware.
- No variation regarding obfuscation and packers.
- Utilize benchmark software to detect rootkit.

RQ3

Is it feasible to utilize EM for stealthy rootkit detection on embedded devices?

Contribution

Novel *baits* to detect rootkit in real-time.

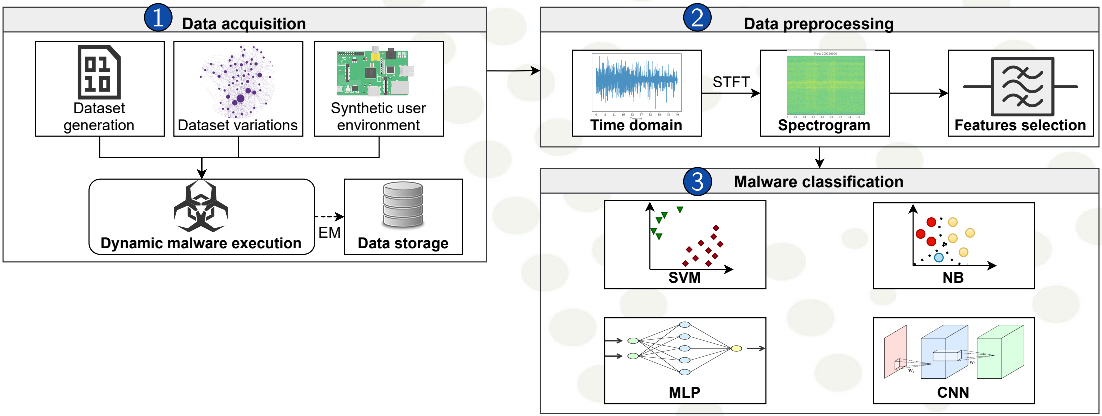
Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification

Joint work with Damien Marion, Mathieu Mastio and Annelie Heuser

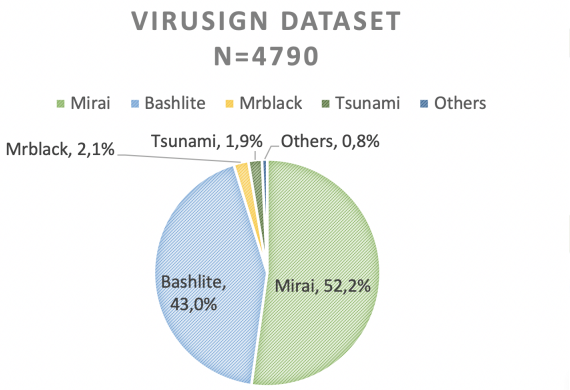
Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "Poster: Obfuscation Revealed-Using Electromagnetic Emanation to Identify and Classify Malware". In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2021, pp. 710–712.

Duy-Phuc Pham et al. "Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification". In: *Annual Computer Security Applications Conference (ACSAC)*. 2021.

Proposed framework AHMA



- AVClass to classify malware labels



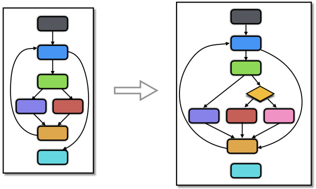
- AVClass to classify malware labels
- Code reviews and reverse engineering

DDoS	Ransomware	Rootkits
Mirai	GonnaCry	spy
Bashlite	(AES, Blowfish, DES)	MaK_It

Dataset: Variations

- AVClass to classify malware labels
- Code reviews and reverse engineering
- Obfuscations

- UPX, Tigress, O-LLVM
- Opaque predicates, bogus control flow, instructions substitution, control-flow flattening; packer and code virtualization



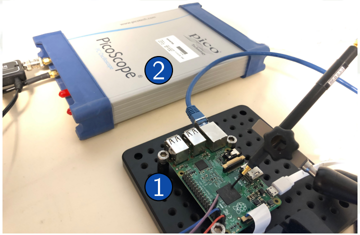
Dataset: Variations

- AVClass to classify malware labels
- Code reviews and reverse engineering
- Obfuscations
- Benign dataset
  - Random Linux activities
  - IoT activities
    - Video encoding
    - Camera captures
    - Music

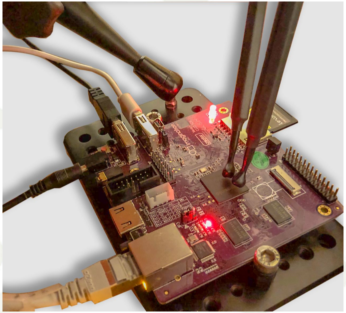
Specifications

- Multi-purpose embedded device.
- Prominent architecture: ARM and MIPS.

→ Raspberry Pi B+, Creator CI20



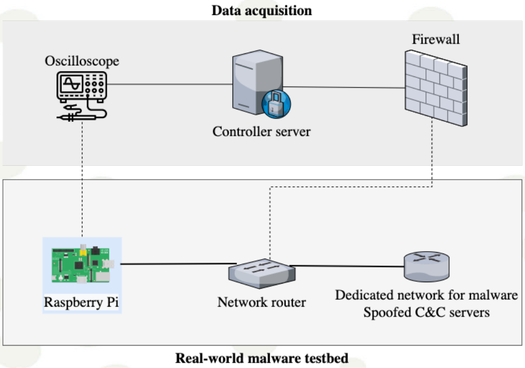
Raspberry Pi B+



Creator CI20

Target environment

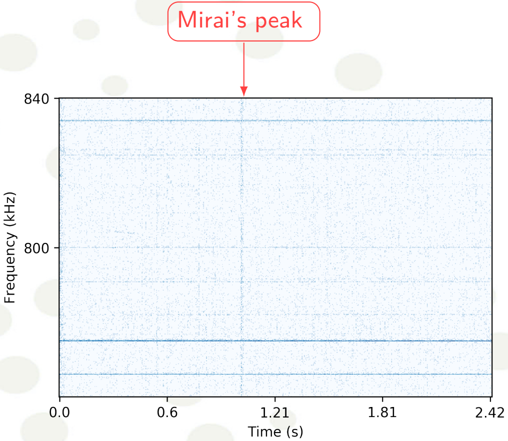
- Isolated controller server
- Embedded device inside synthetic environment
  - Randomized files (to trigger ransomware)
  - Keyboard emulation (to trigger keylogger)
  - Default services (no artifacts)
- Spoofed C&C server



Data and pre-processing

- **Raw traces:**
  106k(traces) $\times$ 2(MS/s) $\times$ 2.5(s) [1.2TB]
- **Time-frequency representation:**
  Short-time Fourier transform
  $spectro\{x(n)\}(m,\omega) = |\sum_{n=0}^{N} x(n)w(n-m)e^{-j\omega n}|^2$

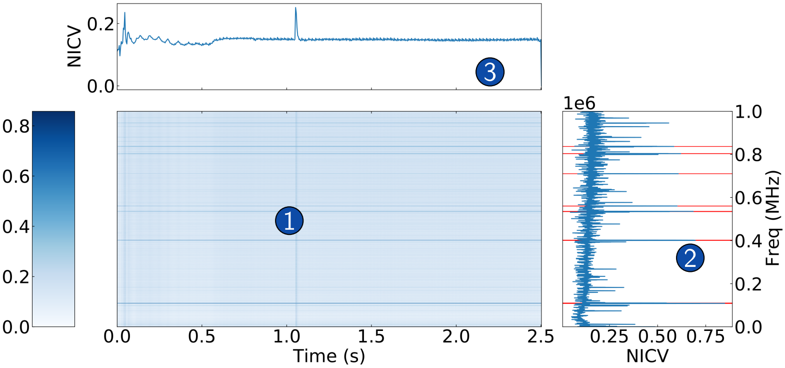
$$\begin{cases} \text{windows} & = 8192 \\ \text{overlap} & = 4096 \end{cases}$$



Features selection: Normalized Inter-Class Variance [Bha+14]

$$\mathrm{NICV}(X, Y) = \frac{\mathrm{Var}[\mathbb{E}[X|Y]]}{\mathrm{Var}[X]}$$

$$F_{\text{extract}} = \mathrm{argmax}_{\epsilon}\left(\left\{\max\left[\mathrm{NICV}(X, Y)_f^D\right]\right\}_{f<F}\right)$$



Machine Learning & Deep Learning models

Machine Learning

- Linear Discriminant Analysis (LDA) + Naive Bayes (NB)
- Linear Discriminant Analysis (LDA) + Support vector machine (SVM)

Deep Learning

- Multi-Layer Perceptron (MLP)
- Convolutional Neural Network (CNN)

Malware classification results

Scenarios	#	MLP	CNN	LDA+NB	LDA+SVM
Executables	31	73.56 [24]	**82.28** [24]	70.92 [28]	71.84 [20]
Type	4	99.75 [28]	**99.82** [28]	97.97 [24]	98.07 [24]
Family	6	98.57 [28]	**99.61** [28]	97.19 [28]	97.27 [28]
Novelty	5	88.41 [16]	**98.85** [24]	98.25 [28]	98.61 [28]
Virtualization	2	95.60 [20]	**95.83** [24]	91.29 [6]	91.25 [6]
Packer	2	93.39 [28]	**94.96** [20]	83.62 [16]	83.58 [16]
Obfuscation	7	73.79 [28]	**82.70** [24]	64.29 [10]	64.47 [10]

Table 1. Accuracy obtained with MLP, CNN, LDA + NB and LDA + SVM applied on several scenarios.

Malware classification results

Scenarios	#	MLP	CNN	LDA+NB	LDA+SVM
Executables	31	73.56 [24]	**82.28** [24]	70.92 [28]	71.84 [20]
Type	4	99.75 [28]	**99.82** [28]	97.97 [24]	98.07 [24]
Family	6	98.57 [28]	**99.61** [28]	97.19 [28]	97.27 [28]
Novelty	5	88.41 [16]	**98.85** [24]	98.25 [28]	98.61 [28]
Virtualization	2	95.60 [20]	**95.83** [24]	91.29 [6]	91.25 [6]
Packer	2	93.39 [28]	**94.96** [20]	83.62 [16]	83.58 [16]
Obfuscation	7	73.79 [28]	**82.70** [24]	64.29 [10]	64.47 [10]

Table 1. Accuracy obtained with MLP, CNN, LDA + NB and LDA + SVM applied on several scenarios.

Malware classification results

Scenarios	#	MLP	CNN	LDA+NB	LDA+SVM
Executables	31	73.56 [24]	**82.28** [24]	70.92 [28]	71.84 [20]
Type	4	99.75 [28]	**99.82** [28]	97.97 [24]	98.07 [24]
Family	6	98.57 [28]	**99.61** [28]	97.19 [28]	97.27 [28]
Novelty	5	88.41 [16]	**98.85** [24]	98.25 [28]	98.61 [28]
Virtualization	2	95.60 [20]	**95.83** [24]	91.29 [6]	91.25 [6]
Packer	2	93.39 [28]	**94.96** [20]	83.62 [16]	83.58 [16]
Obfuscation	7	73.79 [28]	**82.70** [24]	64.29 [10]	64.47 [10]

Table 1. Accuracy obtained with MLP, CNN, LDA + NB and LDA + SVM applied on several scenarios.

Takeaways

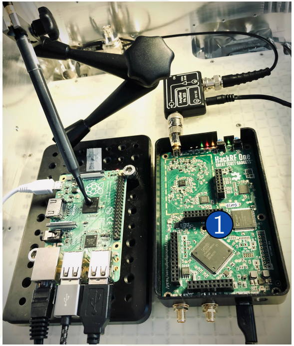
- Classify various malware samples in multiple in-the-wild scenarios.
- Obfuscation technique can be classified.
- Evaluation of both DL/ML.
- Evaluated Artifacts:
  - Code: https://github.com/ahma-hub/analysis/wiki
  - Data: https://zenodo.org/record/5414107

Disadvantages

- Oscilloscopes: difficulties in practical usage and expensive
- It only works with *active* malware not *passive*: **stealthy rootkits**.
- Difficulties for file-less and self-deleting malware detection.

SDR Advantages

- Flexible and adaptable
- Suitable for streaming mode
- Affordable and portable



ULTRA: Ultimate Rootkit Detection over the Air

Joint work with Damien Marion and Annelie Heuser

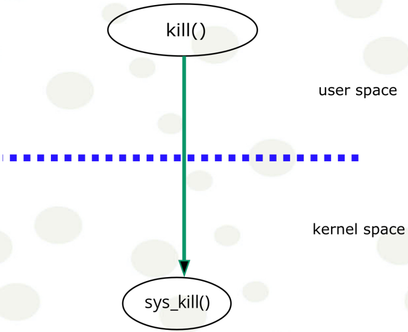
Media: Hackaday

Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "ULTRA: Ultimate Rootkit Detection over the Air". In: *25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2022.

Rootkit *modus operandi*

Classification

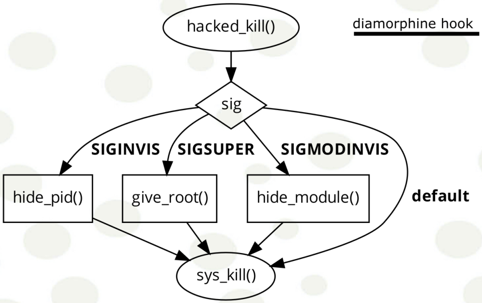
- User-level
- Kernel-level
- Boot and hypervisor level
- Hardware and firmware level



kill() syscall flow

Classification

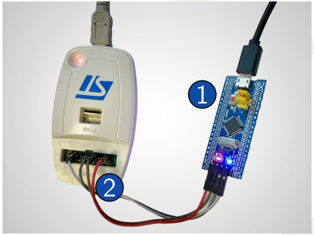
- User-level
- Kernel-level
- Boot and hypervisor level
- Hardware and firmware level



Diamorphine rootkit syscall hooking

Bait definition

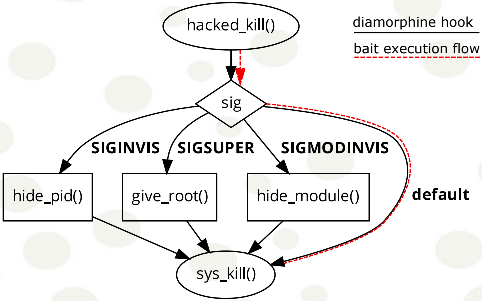
A *bait* β, which is a software or hardware stimulus on a device δ, has the following requirements:
(i) The bait can trigger partial or full behavior of rootkits without knowing *modus operandi* of the rootkit in advance;
(ii) It has a variable duration time of execution activities that can be remotely controlled;
(iii) It cannot be distinguished from common benign behavior (e.g., it relies on unprivileged execution).



An example of hardware bait

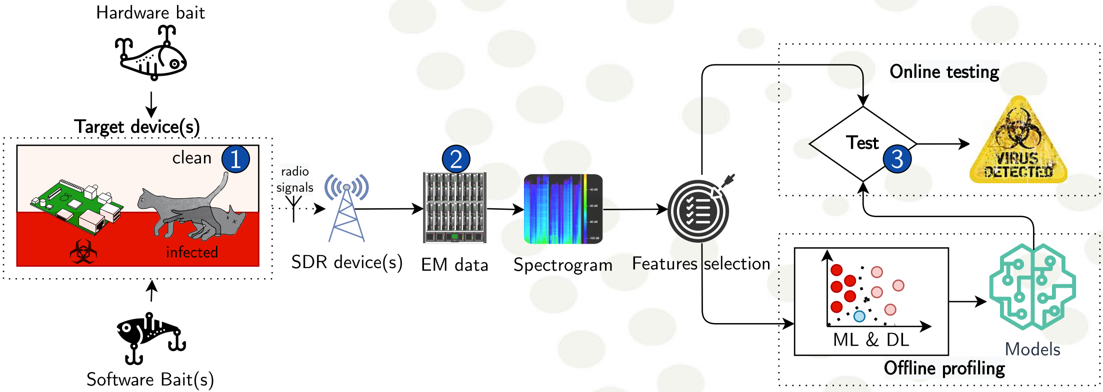
Classification

- User-level
- Kernel-level
- Boot and hypervisor level
- Hardware and firmware level



Proposed bait running with **Diamorphine** on infected device

Proposed framework ULTRA (Open source)



Dataset: Real-world rootkit and device activities

- Benign activities
  - User-space: Linux utilities, etc.
  - Kernel-space: Kernel drivers, firewalls, etc.
- Rootkit dataset

	Hide files	Network	Keylogger	RAT	LPE	Mode
diamorphine⋆	✓				✓	Kernel
m0ham3d⋆	✓	✓			✓	Kernel
adore-ng	✓	✓		✓		Kernel
spy			✓			Kernel
maK_it			✓			Kernel
beurk	✓	✓		✓		User
vlany	✓	✓		✓		User

⋆ plus an obfuscated version.

Data acquisition and processing

Pre-processing

- EM monitoring during **0.5 seconds** using HackRF SDR with **2MHz** window,
  - Centered in 1222MHz for Raspberry Pi B+ and 792MHz for the Creator CI20.
- Time frequency representation: short-time Fourier transform

$$\begin{cases} \text{windows} & = 8192 \\ \text{overlap} & = 4096 \end{cases}$$

Open-data (traces and models)

https://zenodo.org/record/5902451

Deep Learning

- Multi-Layer Perceptron (MLP)

Machine learning

- Kernel PCA (KPCA) + Naive Bayes (NB)
- Kernel PCA (KPCA) + Support vector machine (SVM)

Hill climbing algorithm

Iterated “forward selection” of the sorted extracted bandwidth (using NICV) for optimal bandwidth selection.

Average processing (optional)

The testing traces can be average to increase the detection rate.

Results: Novelty detection using getdents

Testing on (columns) / Profiling on (rows)

MLP on δ_{ci20}

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.4	97.9	98.9	100	97.6
diamorphine	51.2	100	91.2	100	63.9
beurk	95.5	88.4	99.3	74.0	98.4
adore	100	100	93.3	100	99.0
vlany	51.2	50.7	55.7	50.7	98.8

KPCA + NB on δ_{ci20}

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	100	100	100	100	70.5
diamorphine	100	100	100	100	100
beurk	100	99.5	100	96.7	100
adore	51.1	84.5	51.5	100	51.9
vlany	100	100	100	100	100

KPCA + SVM on δ_{ci20}

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	100	100	100	100	100
diamorphine	51.1	100	98.1	100	63.6
beurk	100	99.5	100	97.0	100
adore	75.6	99.5	100	100	64.1
vlany	100	100	100	100	100

MLP on $\delta_{rasp.}$

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.6	98.0	61.0	50.3	59.7
diamorphine	71.7	95.1	61.9	86.5	56.2
beurk	80.5	100	96.9	100	56.3
adore	50.3	89.6	57.4	100	52.5
vlany	71.1	86.7	66.7	73.8	85.5

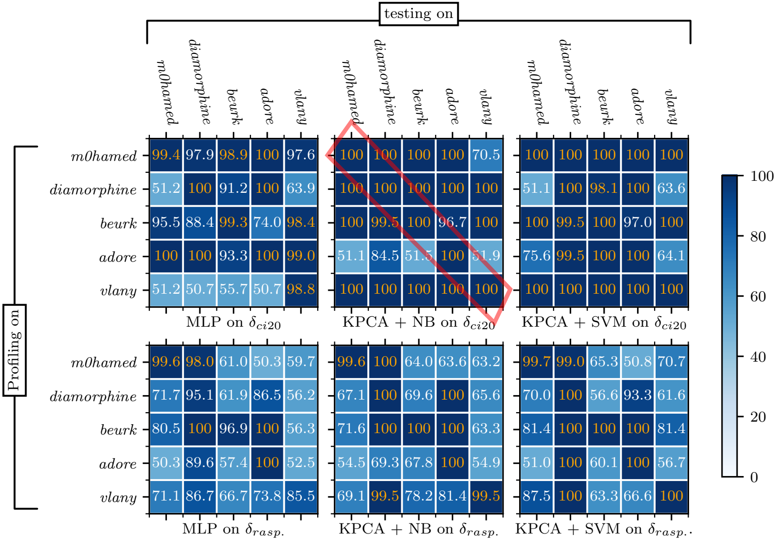
KPCA + NB on $\delta_{rasp.}$

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.6	100	64.0	63.6	63.2
diamorphine	67.1	100	69.6	100	65.6
beurk	71.6	100	100	100	63.3
adore	54.5	69.3	67.8	100	54.9
vlany	69.1	99.5	78.2	81.4	99.5

KPCA + SVM on $\delta_{rasp.}$.

Profiling on \ testing on	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.7	99.0	65.3	50.8	70.7
diamorphine	70.0	100	56.6	93.3	61.6
beurk	81.4	100	100	100	81.4
adore	51.0	100	60.1	100	56.7
vlany	87.5	100	63.3	66.6	100

Results: Novelty detection using getdents



Results: Novelty detection using getdents

Testing on (columns) / Profiling on (rows)

MLP on δ_{ci20}

	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.4	97.9	98.9	100	97.6
diamorphine	51.2	100	91.2	100	63.9
beurk	95.5	88.4	99.3	74.0	98.4
adore	100	100	93.3	100	99.0
vlany	51.2	50.7	55.7	50.7	98.8

KPCA + NB on δ_{ci20}

	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	100	100	100	100	70.5
diamorphine	100	100	100	100	100
beurk	100	99.5	100	96.7	100
adore	51.1	84.5	51.5	100	51.9
vlany	100	100	100	100	100

KPCA + SVM on δ_{ci20}

	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	100	100	100	100	100
diamorphine	51.1	100	98.1	100	63.6
beurk	100	99.5	100	97.0	100
adore	75.6	99.5	100	100	64.1
vlany	100	100	100	100	100

MLP on $\delta_{rasp.}$

	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.6	98.0	61.0	50.3	59.7
diamorphine	71.7	95.1	61.9	86.5	56.2
beurk	80.5	100	96.9	100	56.3
adore	50.3	89.6	57.4	100	52.5
vlany	71.1	86.7	66.7	73.8	85.5

KPCA + NB on $\delta_{rasp.}$

	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.6	100	64.0	63.6	63.2
diamorphine	67.1	100	69.6	100	65.6
beurk	71.6	100	100	100	63.3
adore	54.5	69.3	67.8	100	54.9
vlany	69.1	99.5	78.2	81.4	99.5

KPCA + SVM on $\delta_{rasp.}$.

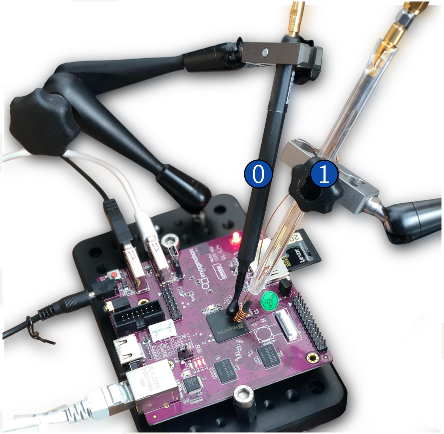
	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.7	99.0	65.3	50.8	70.7
diamorphine	70.0	100	56.6	93.3	61.6
beurk	81.4	100	100	100	81.4
adore	51.0	100	60.1	100	56.7
vlany	87.5	100	63.3	66.6	100

Results: Different types and locations of probes

Probe type

Scenario	MLP			KPCA + NB			KPCA + SVM		
	BA $_{[\epsilon_{\text{opt}}]}$	TPR	TNR	BA $_{[\epsilon_{\text{opt}}]}$	TPR	TNR	BA $_{[\epsilon_{\text{opt}}]}$	TPR	TNR
$\{0,0\} \to \{0,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{2,1\}$	$60.6_{[2]}$	21.4	99.9	$50.0_{[2]}$	0.0	100	$50.0_{[2]}$	0.0	100
$\{1,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[3]}$	100	100	$100_{[2]}$	100	100
$\{2,1\} \to \{2,1\}$	$100_{[1]}$	100	100	$100_{[4]}$	100	100	$100_{[4]}$	100	100

- **More scenarios available:** sample classification, keyloggers detection with software and hardware baits, influence of benign kernel activities, effect of background noise, influence of obfuscation.



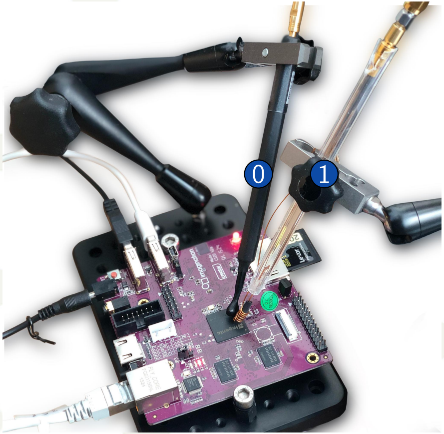
ULTRA with a cheap probe
beurk vs. *open* bait

Results: Different types and locations of probes

Probe location

Scenario	MLP BA $_{[\epsilon_{\text{opt}}]}$	MLP TPR	MLP TNR	KPCA + NB BA $_{[\epsilon_{\text{opt}}]}$	KPCA + NB TPR	KPCA + NB TNR	KPCA + SVM BA $_{[\epsilon_{\text{opt}}]}$	KPCA + SVM TPR	KPCA + SVM TNR
$\{0,0\} \to \{0,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{2,1\}$	$60.6_{[2]}$	21.4	99.9	$50.0_{[2]}$	0.0	100	$50.0_{[2]}$	0.0	100
$\{1,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[3]}$	100	100	$100_{[2]}$	100	100
$\{2,1\} \to \{2,1\}$	$100_{[1]}$	100	100	$100_{[4]}$	100	100	$100_{[4]}$	100	100

- **More scenarios available:** sample classification, keyloggers detection with software and hardware baits, influence of benign kernel activities, effect of background noise, influence of obfuscation.

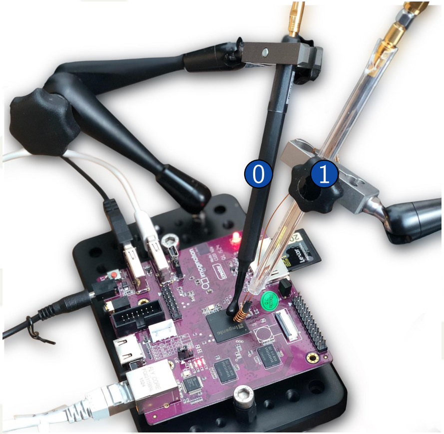


ULTRA with a cheap probe
beurk vs. *open* bait

Results: Different types and locations of probes

	MLP			KPCA + NB			KPCA + SVM		
Scenario	**BA** $[\epsilon_{opt}]$	**TPR**	**TNR**	**BA** $[\epsilon_{opt}]$	**TPR**	**TNR**	**BA** $[\epsilon_{opt}]$	**TPR**	**TNR**
$\{0,0\} \to \{0,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[2]}$	100	100	$100_{[2]}$	100	100
$\{0,0\} \to \{2,1\}$	$60.6_{[2]}$	21.4	99.9	$50.0_{[2]}$	0.0	100	$50.0_{[2]}$	0.0	100
$\{1,0\} \to \{1,0\}$	$100_{[2]}$	100	100	$100_{[3]}$	100	100	$100_{[2]}$	100	100
$\{2,1\} \to \{2,1\}$	$100_{[1]}$	100	100	$100_{[4]}$	100	100	$100_{[4]}$	100	100

- **More scenarios available:** sample classification, keyloggers detection with software and hardware baits, influence of benign kernel activities, effect of background noise, influence of obfuscation.



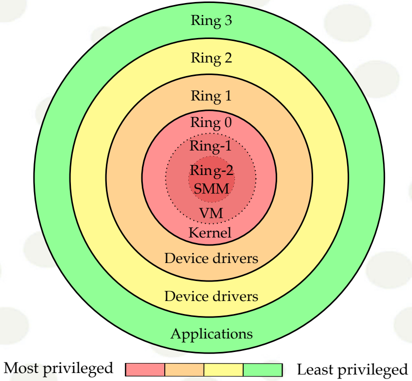
ULTRA with a cheap probe
beurk vs. *open* bait

Conclusion

- ULTRA framework: Wave-and-Play solution.
- Investigation of various experiments and real-world scenarios.
- Promising solution (detection accuracy up to 100%) and handy: tested with multiple probes and probe relocation with affordable SDR.

Short-term

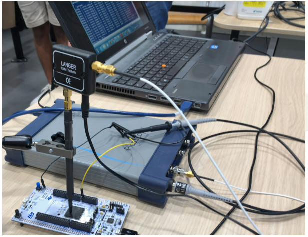
- Larger dataset and upcoming threats (*eg.* hypervisor, eBPF rootkits)
- IoT malware and rootkits from APT campaigns (*eg.* APT28, UNC3524/APT29)



CPU privilege levels

Long-term

- A standalone solution that uses electromagnetic waves to detect malware and similar threats for other platforms (PLC, Linux servers, etc.)
- Portable solution with GPU (*e.g.* Nvidia Jetson Nano)



Open questions and Perspectives (3/3)

Long-term

- Evasion techniques
  → Dynamic bare-metal malware analysis pitfalls
  → Electromagnetic noise (*eg.* NoiseSDR [CF22])
- Model explainability

Giovanni Camurati and Aurelien Francillon. "Noise-SDR: Arbitrary modulation of electromagnetic noise from unprivileged software and its impact on emission security". In: *IEEE Symposium on Security and Privacy*. San Francisco, CA: IEEE Computer Society, 2022.

Contributions

- Duy-Phuc Pham, Duc-Ly Vu, and Fabio Massacci. "Mac-A-Mal: macOS malware analysis framework resistant to anti evasion techniques". In: *Journal of Computer Virology and Hacking Techniques* 15.4 (2019), pp. 249–257
- Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "Poster: Obfuscation Revealed-Using Electromagnetic Emanation to Identify and Classify Malware". In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2021, pp. 710–712
- Duy-Phuc Pham et al. "Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification". In: *Annual Computer Security Applications Conference (ACSAC)*. 2021
- Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "ULTRA: Ultimate Rootkit Detection over the Air". In: *25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2022

Thank you!

Electromagnetism discussion

duyphuc.pham@trellix.com @phd_phuc

[BCH08] Rory Bray, Daniel Cid, and Andrew Hay. *OSSEC host-based intrusion detection guide*. Syngress, 2008.

[BGI11] Arati Baliga, Vinod Ganapathy, and Liviu Iftode. "Detecting Kernel-Level Rootkits Using Data Structure Invariants". In: *IEEE Transactions on Dependable and Secure Computing* 8.5 (2011), pp. 670–684. DOI: 10.1109/TDSC.2010.38.

[BH12] Michael Boelen and John Horne. "The rootkit hunter project". In: *Online. http://rkhunter.sourceforge.net* (2012). Accessed on 2021-06-23.

[Bha+14] Shivam Bhasin et al. "NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage". In: *International Symposium on Electromagnetic Compatibility (EMC '14 / Tokyo)*. eprint version: https://eprint.iacr.org/2013/717.pdf. IEEE, 2014.

[Bri+18] Robert Bridges et al. "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.

[CF22] Giovanni Camurati and Aurelien Francillon. "Noise-SDR: Arbitrary modulation of electromagnetic noise from unprivileged software and its impact on emission security". In: *IEEE Symposium on Security and Privacy*. San Francisco, CA: IEEE Computer Society, 2022.

[CKM21] Nikhil Chawla, Harshit Kumar, and Saibal Mukhopadhyay. "Machine Learning in Wavelet Domain for Electromagnetic Emission Based Malware Analysis". In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 3426–3441. DOI: 10.1109/TIFS.2021.3080510.

[Cla+13] Shane S. Clark et al. "WattsUpDoc: Power Side Channels to Nonintrusively Discover Untargeted Malware on Embedded Medical Devices". In: *2013 USENIX Workshop on Health Information Technologies (HealthTech 13)*. Washington, D.C.: USENIX Association, Aug. 2013.

[Din+20] Fei Ding et al. "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.

[Gur+15] Mordechai Guri et al. "JoKER: Trusted detection of kernel rootkits in android devices via JTAG interface". In: *2015 IEEE Trustcom/BigDataSE/ISPA*. Vol. 1. IEEE. 2015, pp. 65–73.

[HP18] Seunghun Han and JH Park. "Shadow-box v2: The practical and omnipotent sandbox for arm". In: *2018, slideshow at Blackhat Asia* (2018).

[JLC20] Xingbin Jiang, Michele Lora, and Sudipta 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.

[Kha+19] H. A. Khan et al. "IDEA: Intrusion Detection through Electromagnetic-Signal Analysis for Critical Embedded and Cyber-Physical Systems". In: *IEEE Transactions on Dependable and Secure Computing* (2019), pp. 1–1.

[Kha+19] Haider A. Khan et al. "Malware Detection in Embedded Systems Using Neural Network Model for Electromagnetic Side-Channel Signals". In: *J. Hardware and Systems Security* 3.4 (2019), pp. 305–318. DOI: 10.1007/s41635-019-00074-w. URL: https://doi.org/10.1007/s41635-019-00074-w.

[LGO04] John Levine, Julian Grizzard, and Henry Owen. "A methodology to detect and characterize kernel level rootkit exploits involving redirection of the system call table". In: *Second IEEE International Information Assurance Workshop, 2004. Proceedings.* IEEE. 2004, pp. 107–125.

[Luc+18] Patrick Luckett et al. "Identifying stealth malware using CPU power consumption and learning algorithms". In: *Journal of Computer Security* 26.5 (2018), pp. 589–613.

[MSJ01] Nelson Murilo and Klaus Steding-Jessen. "Métodos para detecção local de rootkits e módulos de kernel maliciosos em sistemas UNIX". In: *Anais do III Simpósio sobre Segurança em Informática (SSI'2001)*. 2001, pp. 133–139.

[Pha+21] Duy-Phuc Pham et al. "Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification". In: *Annual Computer Security Applications Conference (ACSAC)*. 2021.

[PJ+04] Nick L Petroni Jr et al. "Copilot-a Coprocessor-based Kernel Runtime Integrity Monitor.". In: *USENIX security symposium*. San Diego, USA. 2004, pp. 179–194.

[PMH21] Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "Poster: Obfuscation Revealed-Using Electromagnetic Emanation to Identify and Classify Malware". In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2021, pp. 710–712.

[PMH22] Duy-Phuc Pham, Damien Marion, and Annelie Heuser. "ULTRA: Ultimate Rootkit Detection over the Air". In: *25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. 2022.

[PVM19] Duy-Phuc Pham, Duc-Ly Vu, and Fabio Massacci. "Mac-A-Mal: macOS malware analysis framework resistant to anti evasion techniques". In: *Journal of Computer Virology and Hacking Techniques* 15.4 (2019), pp. 249–257.

[Seh+20] N. Sehatbakhsh et al. "REMOTE: Robust External Malware Detection Framework by Using Electromagnetic Signals". In: *IEEE Transactions on Computers* 69.3 (2020), pp. 312–326.

[Sin+17] Baljit Singh et al. "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*. ASIA CCS '17. Accessed on 2019-03-26. New York, NY, USA: ACM, 2017, pp. 483–493. ISBN: 978-1-4503-4944-4. DOI: 10.1145/3052973.3052999.

[Wan+09] Zhi Wang et al. "Countering kernel rootkits with lightweight hook protection". In: *Proceedings of the 16th ACM conference on Computer and communications security*. 2009, pp. 545–554.

[Wan+18] Xiao Wang et al. "Deep learning-based classification and anomaly detection of side-channel signals". In: *Cyber Sensing 2018*. Vol. 10630. International Society for Optics and Photonics. 2018, p. 1063006.

[WK13] Xueyang Wang and Ramesh 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.

[Zab18] Adam Zabrocki. “Linux Kernel Runtime Guard (LKRG) under the Hood”. In: *CONFidence Conference*. 2018.

Balanced Accuracy

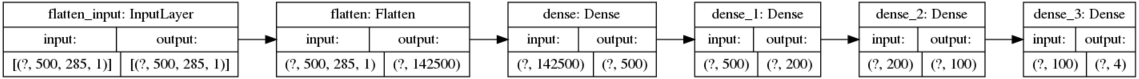
Consider a dataset that contains 99 negative samples and 1 positive sample. Classifying all values as negative yields a 0.99 accuracy score.
Balanced Accuracy is not affected by this issue. It normalizes true positive and true negative predictions by the number of positive and negative samples, respectively, and divides their sum by two:

$$\mathbf{BA} = \frac{TPR + TNR}{2} \tag{1}$$

- Picoscope 6000
- Keysight Infiniium
- HackRF SDR

Deep Learning models

- Multi-Layer Perceptron (MLP)
- Convolutional Neural Network (CNN)



Deep Learning models (MLP)

Table: Proposed MLP architecture of ULTRA framework

Layer	Size	Filter	Activation
Flatten	spectrogram_size	_	leaky relu
Dense	500	_	leaky relu
Dense	200	_	leaky relu
Dense	100	_	leaky relu
Dense	N	_	softmax (multi-class) or sigmoid (two-class)

Deep Learning models (CNN)

Layer	**Size**	**Filter**	**Activation**
Convolution	64	7×7	relu
Max Pooling	64	2×2	_
Convolution	128	3×3	relu
Convolution	128	3×3	relu
Max Pooling	128	2×2	_
Convolution	256	3×3	relu
Convolution	256	3×3	relu
Max Pooling	256	2×2	_
Dense	128	_	relu
Dense	64	_	relu
Dense	nb_labels	_	softmax

State of the art (1)

Article	Year	Techniques
WattsUpDoc: Power SC to Nonintrusively Discover Untargeted MW on Embedded Medical Devices	2013	- Detection of 12 MW variants - Power & MLP & 3NN &RF
Detecting crypto-ransomware in IoT networks based on energy consumption footprint	2017	- MW detection of Ransomware - PowerTutor & KNN
Deep learning-based classification and anomaly detection of side-channel signals	2018	- Anomaly detection of botnet - Power & MLP & LSTM
HLMD: a signature-based approach to HW-level behavioral MW detection and classification	2019	- MW classification of 14 variants - HPC & singular values

State of the art (2)

Article	Year	Techniques
EDDIE: EM-based detection of deviations in program execution	2017	- Code Inj. detection - EM & STFT & KS
MW detection in embedded systems using NN model for EM SC signals	2019	- MW detection of DDoS, Ransomware, CF Hijack - EM & MLP

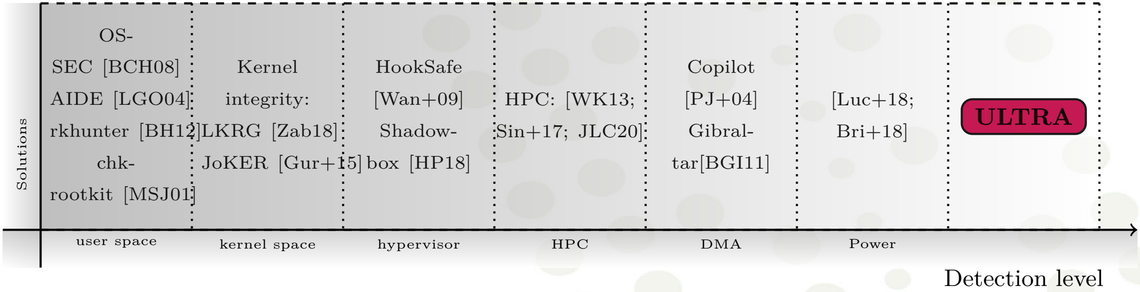
$\rightarrow$ Real world malware.

State of the art (3)

Table: Comparison with related works on side-channel malware (SCM) analysis using EM or power consumption.

Article	SCM detection	Anomaly detection	SCM classification	Real-world SCM	Real-world analysis environment	Samples size	Varia-tions	Benign dataset	Window size	Open data, source code	Device under test
WattsUpDoc [Cla+13]	✓	-	-	✓	-	15	-	-	5s	-	Windows XP Embedded 664 MHz
IDEA [Kha+19]	-	✓	-	-	-	3	-	-	$<40\mu s$	-	AT328p 16MHz, Cortex A8
REMOTE [Seh+20]	-	✓	-	✓	-	3	-	-	<10ms	-	Single-core ARM 1Ghz
Wang *et al.* [Wan+18]	-	✓	-	-	-	1	-	-	10s	-	Raspberry Pi, Arduino, Siemens PLC
Khan *et al.* [Kha+19]	✓	-	-	-	-	3	-	-	$<150\mu s$	-	Cyclone II FPGA & NIOS II soft-processor
DeepPower [Din+20]	✓	-	✓	✓	-	5	-	-	1s	-	MIPS/ARM OpenWRT
Chawla *et al.* [CKM21]	✓	-	✓	✓	-	137	-	✓	10s	-	Android Intrinsyc Open-Q 820
Chapter ??	(✓)*	-	✓	✓	✓	35	✓	✓	2.5s	✓	Multi-core, 900 Mhz ARM

State of the art (4)

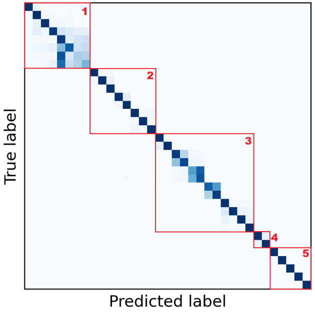


Taxonomy of rootkit detection approaches and positioning our approach in the state of the art and open source tools.

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

	Article	WnP	Classifi-cation	Baits	ML	DL	Sample size	Open source	Benign	User RK	Window size	Device under test
HPC	Numchecker [WK13]	-	-	✓	-	-	8	-	-	-	262.3 ms	32-bit Ubuntu PC
HPC	[Sin+17]	-	-	-	✓	-	5	-	-	-	45s	VMWare Windows 7 Intel
HPC	[JLC20]	-	-	✓	✓	-	4	-	-	-	2.91s	ARM Cortex-A53
DMA	Copilot [PJ+04]	-	-	-	-	-	12	-	-	-	30s	PCI-compatible Intel PC Linux
DMA	Gibraltar [BGI11]	-	-	-	-	-	23	-	✓	-	20s	PCI-compatible Intel PC Linux
Power	[Luc+18]	-	-	-	✓	✓	5	-	-	✓	>5m	PC Windows 10 & Ubuntu 14
Power	[Bri+18]	-	-	-	✓	-	5	-	-	-	>1m	Dell OptiPlex 755 Windows 7
EM	**ULTRA**	✓	✓	✓	✓	✓	9	✓	✓	✓	1.3s	ARM Raspberry Pi & MIPS Ci20

Confusion matrix of a CNN classification into 35 binaries



Confusion matrix of a CNN classification into 35 binaries from left to right (with and without obfuscation).
(1) *bashlite_cfflatten, bashlite_upx, bashlite_bcf, bashlite, bashlite_addopaque, bashlite_sub, bashlite_flatten, bashlite_virtualize*;
(2) *mirai_sub, mirai_bcf, mirai_cfflatten, mirai, mirai_upx, mirai_addopaque, mirai_flatten, mirai_virtualize*;
(3) *gonnacry_des, gonnacry_des_upx, gonnacry, gonnacry_aes, gonnacry_aes_upx, gonnacry_upx, gonnacry_flatten, gonnacry_virtualize, gonnacry_addopaque, gonnacry_bcf, gonnacry_sub, gonnacry_cfflatten*;
(4) *spy, maK_It*;
(5) *benign:* encode video, play audio, take picture, record camera, random.

Dataset: Benign

- Binaries from fresh Linux installation
- Random activities

Activities	Executables			
Linux Utilities	mknod	vdir	more	find
	zgrep	ls	cat	findmnt
	zmore	as	ed	rm
	touch	dmesg	sleep	cd
	less	grep	objdump	
Network	wget	hostname	ss	ip
Compression	gunzip	bunzip2	bzip2	tar
	uncom-press			
Data backup	dd			
Scripting	python			
Photo & Video	raspistill	raspivid		
Video Encoding	MP4Box			
Audio player	mpg321			

ULTRA's targeted devices specification

Table: ULTRA's targeted devices specification, architectures (Arch.), and their targeted frequency leakage (Fc) and CPU in MHz.

Device δ	Arch.	CPU	RAM	OS	Fc
Raspberry Pi B+	ARM32	700	512MB	Linux 4.1.7	1222
Creator CI20	MIPS32	1200	1GB	Linux 3.18.3	792

ULTRA's bill of materials

Table: ULTRA's bill of materials

Equipment	Rate/Unit	Count	Amount (Euro)
HackRF One SDR	309	1	309
Adapter SMA Male BNC Female RG316	5	1	5
Amplifier Langer PA-303 BNC	375	1	375
*Probe Langer RF-U 5-2**	250	1	250
Total			939

* *This can be omitted in the case of using a hand-crafted probe.*

Comparison

Table: Performance evaluation of rootkit (RK) and their obfuscated variants$^{(*)}$ detection results, and execution latency. List of indicators: (✓) RK detected; (-) Not detected; (†) Malicious behavior trigger required; (⚠) Kernel panicked; Executed on (‡) CPU ; (§) GPU.

RK	AV solutions			
	rkhunter	*chkrootkit*	*LKRG*	*ULTRA*
diamorphine	✓	-	✓†	✓
diamorphine$^{(*)}$	-	-	✓†	✓
m0ham3d	✓	-	✓†	✓
m0ham3d$^{(*)}$	-	-	✓†	✓
adore-ng	-	-	✓† ⚠	✓
spy	-	-	-	✓
maK_it	-	-	-	✓
beurk	-	-	-	✓
vlany	-	-	-	✓
Latency *(sec)*	1326.6‡	44.3‡	2.6‡	1.3§-1.5‡

Rootkit Classification

Table: Classification by family and by activity obtained with MLP, LDA + NB and LDA + SVM. The column "#" gives the number of classes per scenario.

			MLP		LDA + NB		LDA + SVM	
Scenario		#	AC $_{[\epsilon_{opt}]}$	$^{PR}/_{RC}$	AC $_{[\epsilon_{opt}]}$	$^{PR}/_{RC}$	AC $_{[\epsilon_{opt}]}$	$^{PR}/_{RC}$
δ_{ci20}	family	19	$91.3_{[65]}$	$^{83.0}/_{83.0}$	$76.0_{[10]}$	$^{65.6}/_{65.4}$	$85.6_{[8]}$	$^{76.1}/_{76.3}$
δ_{ci20}	activity	46	$82.5_{[45]}$	$^{83.0}/_{82.5}$	$62.5_{[10]}$	$^{63.2}/_{62.4}$	$76.0_{[10]}$	$^{75.8}/_{76.0}$
$\delta_{rasp.}$	family	19	$82.1_{[50]}$	$^{79.1}/_{76.5}$	$54.7_{[10]}$	$^{53.9}/_{55.3}$	$66.2_{[10]}$	$^{66.9}/_{60.1}$
$\delta_{rasp.}$	activity	46	$75.0_{[40]}$	$^{75.4}/_{75.0}$	$50.6_{[10]}$	$^{51.5}/_{55.6}$	$59.2_{[9]}$	$^{59.4}/_{59.2}$