Botconf 2024

IoT Malware and Rookit Detections Using Electromagnetic Insights: Unveiling the Unseen

Duy-Phuc Pham, Damien Marion, Annelie Heuser

April 26^{th}, 2024

UMR IRISA
Trellix

ANR-JCJC: Automated Hardware Malware Analysis (AHMA)

"Can side-channel play a role against Malware?"

Duy-Phuc Pham
Threat intelligence
Malware researcher
at Trellix ARC

Damien Marion
Assistant Professor at
University of Rennes, IRISA

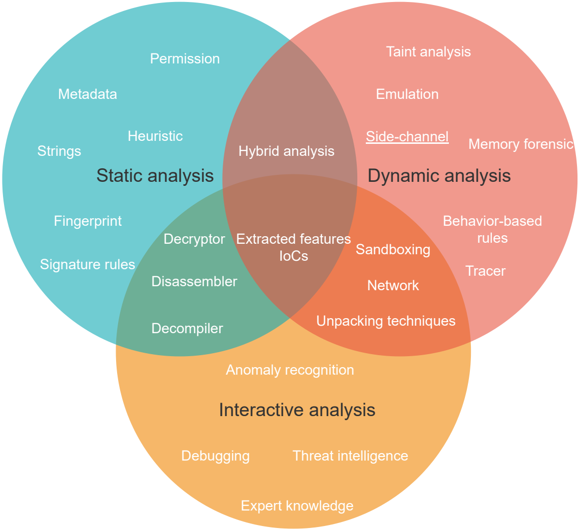
Outline

Why this topic?

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

Malware analysis techniques

- Malware detection
- Malware similarities
- Malware classification



Malware-evasion techniques

Static analysis

- Malware obfuscation
- Packers

Dynamic analysis

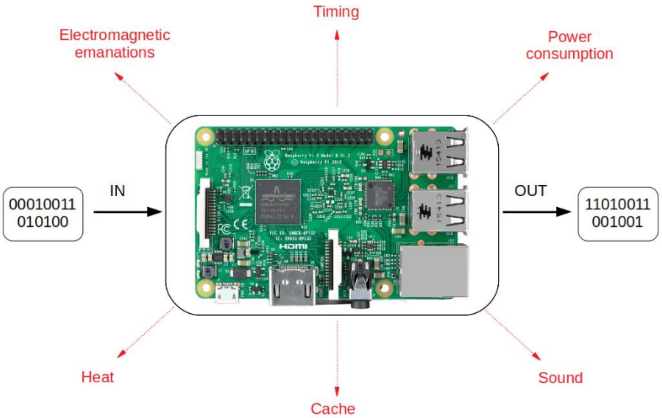
- Anti-debugging
- "Side-channel information"

Evasion of Detection Technologies (EDR/XDR)

- Evade behavioral detection
- Gaps in detection capabilities

(hackaday)

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



- Anomaly detection using power consumption and 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.

Contribution

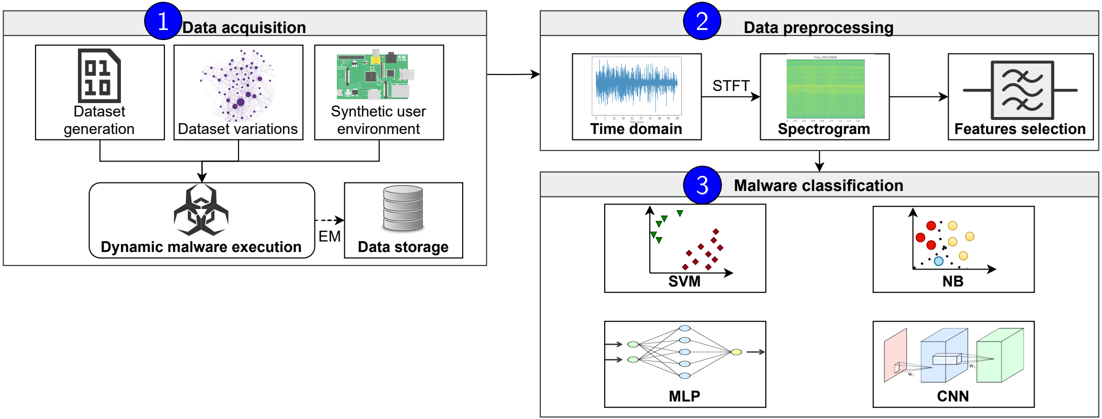
Real-world malicious and benign IoT dataset classification.

- 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.

Contribution

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

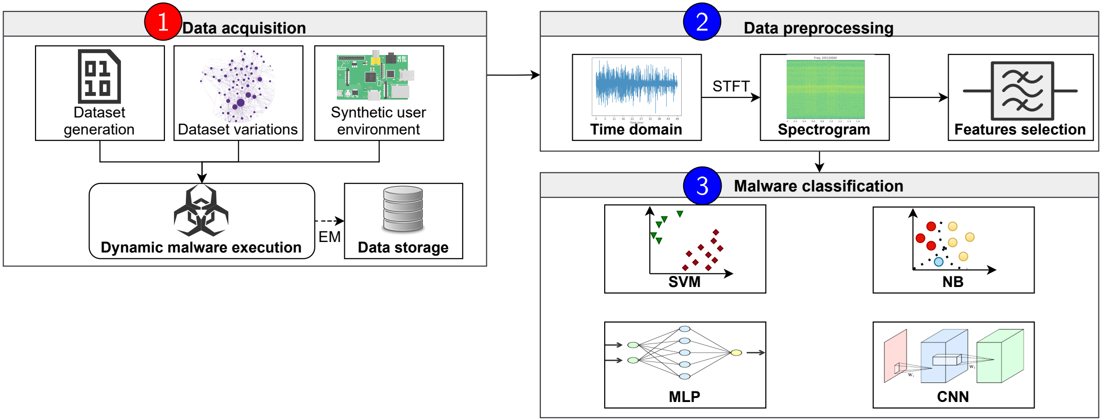
Frameworks overview



Open-source (code and dataset)

https://github.com/ahma-hub/analysis/wiki

https://gitlab.com/ultra-RK/ultra



Dataset

1 AHMA focuses on malwares' variations classification, while ULTRA focuses on rootkits detection.

Samples

- **benign [5]:** `random34, playaudio, recordcamera, takepicture, encodevideo`
- **mirai [8]**
- **gonnacry [12]**
- **rootkits [2]**
- **bashlite [8]**

Obfuscations and variations

obfuscations: _addopaque, _virtualize, _flatten, _cfflatten, _bcf, _sub, _upx.

encryption algorithm for gonnacry (by default blowfish): _aes, _des

- 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

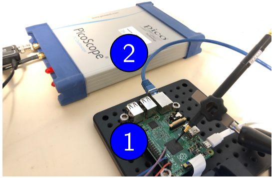
⋆ plus an obfuscated version.

Same target devices

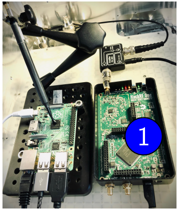
Specifications

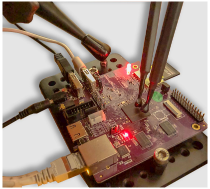
- Multi-purpose embedded device.
- Prominent architecture: ARM and MIPS.

→ Raspberry Pi B+, Creator CI20



Raspberry Pi B+





Creator CI20

Acquisition divergences: Picoscope Vs SDR

Picoscope

- cheap oscilloscope, but not very cheap
- standard in side-channel
- **Setup:** EM monitoring during 2.5s at 2MHz sampling rate.
  - 200k traces [2TB]

SDR Advantages

- Flexible and adaptable
- Suitable for streaming mode
- Affordable and portable
- **Setup:** EM monitoring during 0.5s using HackRF SDR with 2MHz window
  - Centered in 1222MHz for Raspberry Pi B+ and 792MHz for the Creator CI20.
  - 800k traces [6.6TB]

AHMA

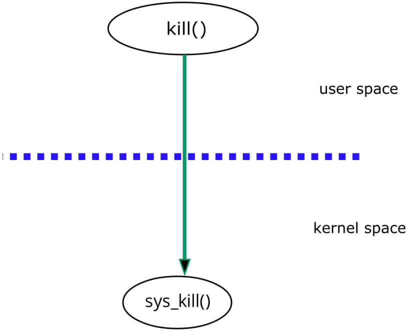
- recording start at the malware installation using a software trigger,

ULTRA

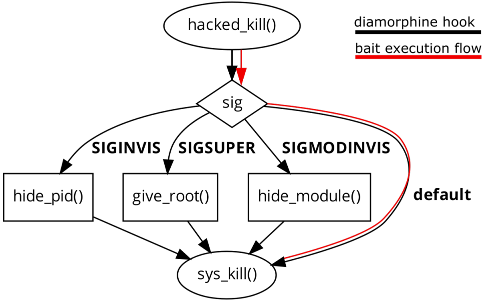
- record **baits** activities

Bait definition

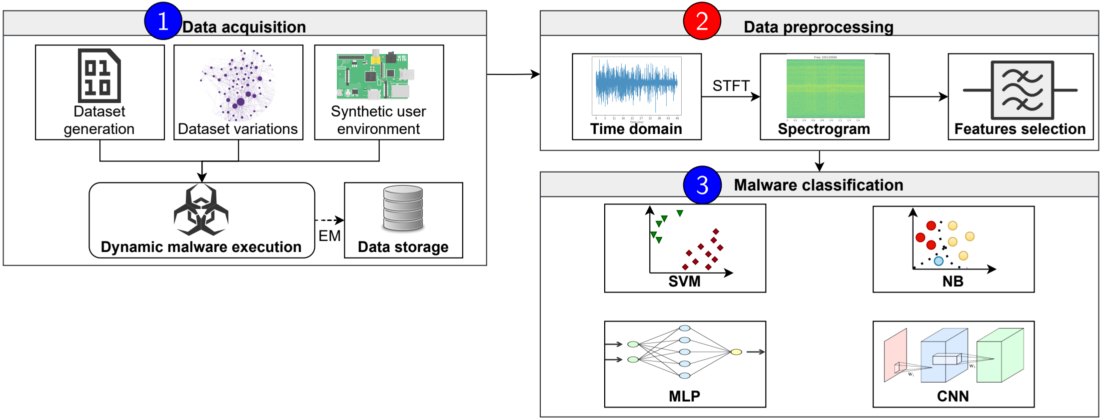
A *bait* β, which is a software or hardware stimulus on a device δ, has the following requirements:
(i) The bait can trigger partial or full behavior of rootkits without knowing *modus operandi* of the rootkit in advance;
(ii) It has a variable duration time of execution activities that can be remotely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



kill() syscall flow

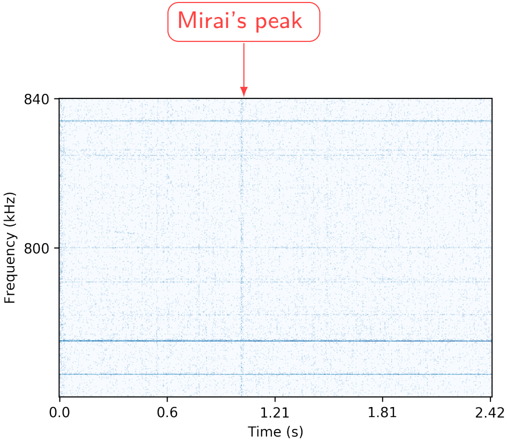


Diamorphine rootkit syscall hooking


1
Data acquisition
Dataset generation
Dataset variations
Synthetic user environment
Dynamic malware execution
EM
Data storage
2
Data preprocessing
Time domain
STFT
Spectrogram
Features selection
3
Malware classification
SVM
NB
MLP
CNN

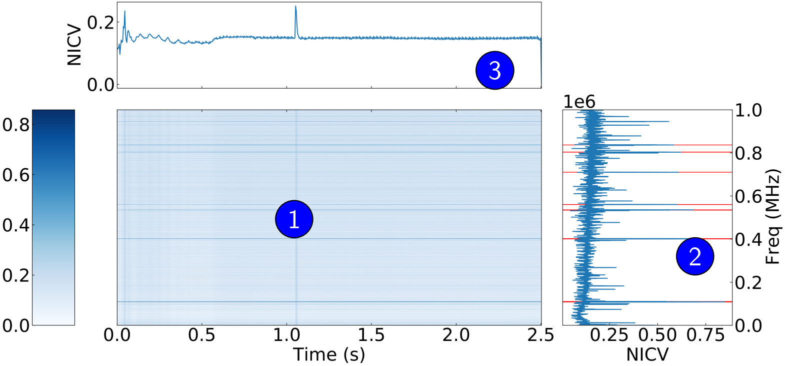
- **Raw traces:**
  106k(traces) $\times$ 2(MS/s) $\times$ 2.5(s) [1.2TB]
- **Time-frequency representation:**
  Short-time Fourier transform
  $spectro\{x(n)\}(m,\omega) = |\sum_{n=0}^{N} x(n)w(n-m)e^{-j\omega n}|^2$

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



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

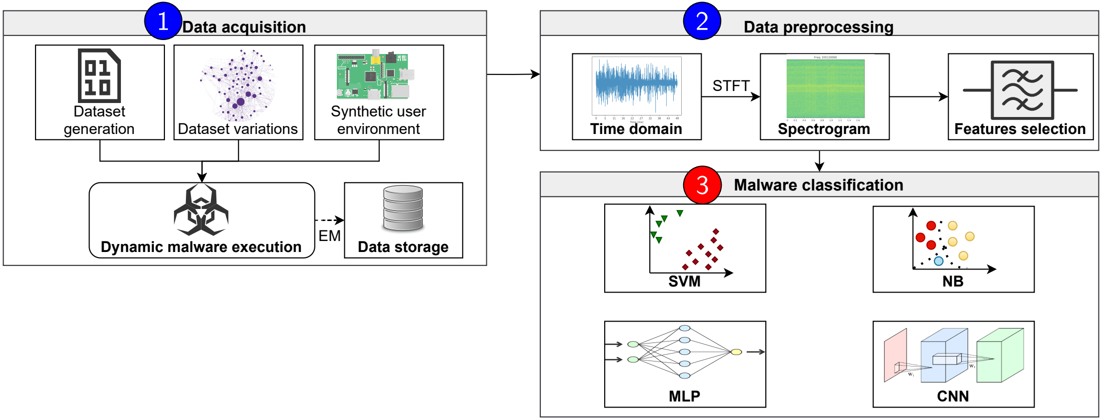
$$F_{\text{extract}} = \operatorname{argmax}_{\epsilon}\left(\left\{\max\left[\mathrm{NICV}(X, Y)_f^D\right]\right\}_{f<F}\right)$$



Iterative Vs hill-climbing for selection

Based on NICV's results

- AHMA: try from one to F bandwidth without rejections,
- ULTRA: hill-climbing process, with rejection process to select optimal bandwidth subspace.


1
Data acquisition
Dataset generation
Dataset variations
Synthetic user environment
Dynamic malware execution
EM
Data storage
2
Data preprocessing
Time domain
STFT
Spectrogram
Features selection
3
Malware classification
SVM
NB
MLP
CNN

Machine Learning & Deep Learning models

Machine Learning

- (AHMA)Linear Discriminant Analysis (LDA) / (ULTRA)Kernel PCA
  + (AHMA + ULTRA) Naive Bayes (NB)
- (AHMA)Linear Discriminant Analysis (LDA) / (ULTRA)Kernel PCA
  + (AHMA + ULTRA) Support vector machine (SVM)

Deep Learning

- (AHMA + ULTRA) Multi-Layer Perceptron (MLP)
- (AHMA) Convolutional Neural Network (CNN)

	#	MLP	CNN	MLP	CNN
Scenarios		**Rasp.**		**ci20**	
Family	6	98.00 [14]	**99.36** [11]	96.96 [11]	**96.99** [11]

Table 1. Accuracy obtained with MLP, CNN applied on several scenarios for the ci20 and raspberry.

Labels	*benign,*	*mirai,*	*gonnacry,*	*maK_it spy*	*bashlite*
Samples	random34, encodevideo, takepicture, recordcamera, playaudio	mirai, mirai_cfflatten, mirai_virtualize, mirai_flatten, mirai_bcf, mirai_addopaque, mirai_sub, mirai_upx	gonnacry, gonnacry_upx, gonnacry_aes_upx, gonnacry_aes, gonnacry_virtualize, gonnacry_flatten, gonnacry_bcf, gonnacry_sub, gonnacry_cfflatten, gonnacry_addopaque, gonnacry_des, gonnacry_des_upx	maK_it, spy	bashlite, bashlite_bcf, bashlite_flatten, bashlite_upx, bashlite_addopaque, bashlite_cfflatten, bashlite_sub, bashlite_virtualize

	#	MLP	CNN	MLP	CNN
Scenarios		**Rasp.**		**ci20**	
Novelty (family)	5	**96.65** [11]	89.04 [4]	**99.92** [25]	99.91 [11]

Table 1. Accuracy obtained with MLP, CNN applied on several scenarios for the ci20 and raspberry.

Labels	*benign,*	*mirai,*	*gonnacry,*	*rootkit,*	*bashlite*
Samples learning	`random34`, `encodevideo`, `takepicture`, `recordcamera`, `playaudio`	`mirai`, `mirai_cfflatten`, `mirai_bcf`, `mirai_upx`	`gonnacry`, `gonnacry_upx`, `gonnacry_virtualize`, `gonnacry_flatten`, `gonnacry_bcf`, `gonnacry_sub`, `gonnacry_addopaque`,	`maK_it`,	`bashlite`, `bashlite_bcf`, `bashlite_upx`, `bashlite_addopaque`, `bashlite_cfflatten`, `bashlite_sub`
Samples testing	`random34`, `encodevideo`, `takepicture`, `recordcamera`, `playaudio`	`mirai_virtualize`, `mirai_flatten`, `mirai_addopaque`, `mirai_sub`	`gonnacry_aes_upx`, `gonnacry_aes`, `gonnacry_cfflatten`	`spy (kisni)`	`bashlite_flatten`, `bashlite_virtualize`

	#	MLP	CNN	MLP	CNN
Scenarios		**Rasp.**		**ci20**	
Obfuscation	7	71.1 [11]	**78.9** [15]	42.8 [5]	**44.7** [5]

Table 1. Accuracy obtained with MLP, CNN applied on several scenarios for the ci20 and raspberry.

Labels	*addopaque,*	*virtulalize,*	*flatten*
Samples	`mirai_addopaque,`	`mirai_virtualize,`	`mirai_flatten,`
	`gonnacry_addopaque,`	`gonnacry_virtualize,`	`gonnacry_flatten,`
	`bashlite_addopaque`	`bashlite_virtualize`	`bashlite_flatten`

Labels	*cflatten,*	*upx,*	*sub,*	*bcf*
Samples	`mirai_cflatten,`	`mirai_upx,`	`mirai_sub,`	`mirai_bcf,`
	`gonnacry_cflatten,`	`gonnacry_upx,`	`gonnacry_sub,`	`gonnacry_bcf,`
	`bashlite_cflatten`	`bashlite_upx`	`bashlite_sub`	`bashlite_bcf`

	#	MLP	CNN	MLP	CNN
Scenarios		**Rasp.**		**ci20**	
Executables	35	70.7 [5]	**77.1** [10]	29.9 [9]	**34.3** [6]

Table 1. Accuracy obtained with MLP, CNN applied on several scenarios for the ci20 and raspberry.

Labels: *random34, mirai, mirai_addopaque, mirai_virtualize, mirai_flatten, mirai_bcf, mirai_cfflatten, mirai_sub, mirai_upx, gonnacry, gonnacry_upx, gonnacry_aes_upx, gonnacry_aes, gonnacry_des, gonnacry_des_upx, gonnacry_virtualize, gonnacry_flatten, gonnacry_bcf, gonnacry_sub, gonnacry_cfflatten, gonnacry_addopaque, maK_it, spy (kisni), bashlite, bashlite_bcf, bashlite_flatten, bashlite_upx, bashlite_addopaque, bashlite_cfflatten, bashlite_sub, bashlite_virtualize, playaudio, recordcamera, takepicture, encodevideo*

Rootkit detection: Novelty detection using getdents

Rows: profiling on; columns: testing on.

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

MLP 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 + NB 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

KPCA + SVM on δ_{ci20}

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

MLP 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 + NB 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

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

Profiling on (rows) / testing on (columns)

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 $\delta_{rasp.}$

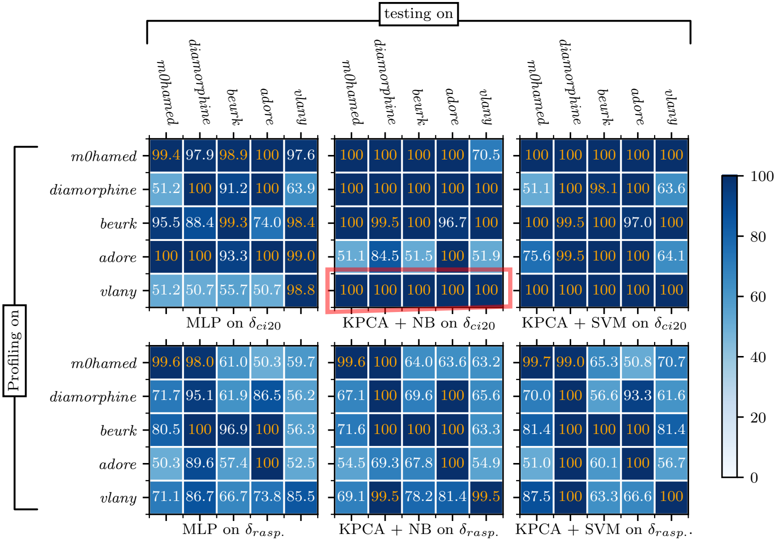
	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.6	98.0	61.0	50.3	59.7
diamorphine	71.7	95.1	61.9	86.5	56.2
beurk	80.5	100	96.9	100	56.3
adore	50.3	89.6	57.4	100	52.5
vlany	71.1	86.7	66.7	73.8	85.5

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 $\delta_{rasp.}$.

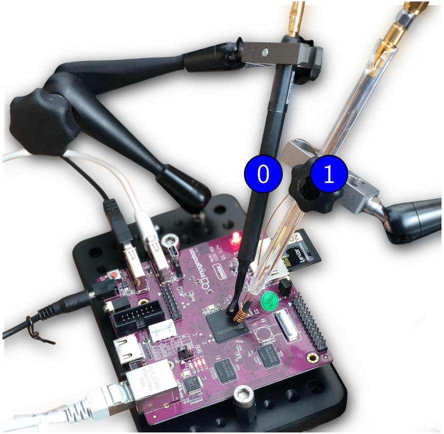
	m0hamed	diamorphine	beurk	adore	vlany
m0hamed	99.7	99.0	65.3	50.8	70.7
diamorphine	70.0	100	56.6	93.3	61.6
beurk	81.4	100	100	100	81.4
adore	51.0	100	60.1	100	56.7
vlany	87.5	100	63.3	66.6	100


testing on
Profiling 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
MLP on δ_{ci20}
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 + NB on δ_{ci20}
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
KPCA + SVM on δ_{ci20}
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
MLP on $\delta_{rasp.}$
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 + NB on $\delta_{rasp.}$
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
KPCA + SVM on $\delta_{rasp.}$.
100
80
60
40
20
0

Probe type

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

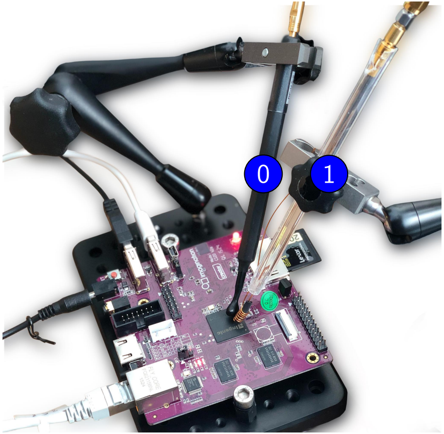


ULTRA with a cheap probe
beurk vs. *open* bait

Probe location

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

DEMO

Conclusion

Achievement

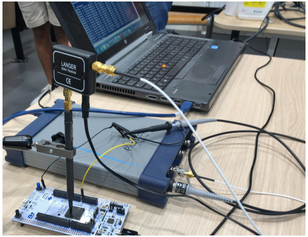
- (fully open-source) AHMA framework: classification in presence of obfuscation
- (fully open-source) ULTRA framework: Wave-and-Play solution.
- Investigation of various experiments and real-world scenarios.
- Promising solution (detection accuracy up to 100%) and handy: tested with multiple probes and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

Media Coverage

- The Hacker News, Schneier on Security, 01 net...
- ...But be Careful to the fake news, we **never** used raspberry to detect malware on your computer!

What next?

- Is it possible to reverse the classification?
- Larger dataset and upcoming threats (*eg.* hypervisor, eBPF rootkits)
- 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)



- 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. "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.
  - Extended version presented at hardware.IO USA (San Francisco) 2022.
- 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.
- Duy-Phuc Pham thesis: Leveraging side-channel signals for IoT malware classification and rootkit detection

Thank you!

duyphuc.pham@trellix.com @phd_phuc

Institut de Recherche en Informatique et Systèmes Aléatoires

[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.

[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.

References

[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.

[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.

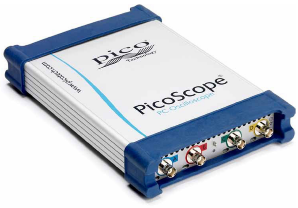
[WK13] Xueyang Wang and Ramesh Karri. "Numchecker: Detecting kernel control-flow modifying rootkits by using hardware performance counters". In: *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE. 2013, pp. 1–7.

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

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)

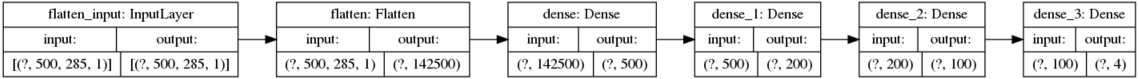


Table: Proposed MLP architecture of ULTRA framework

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

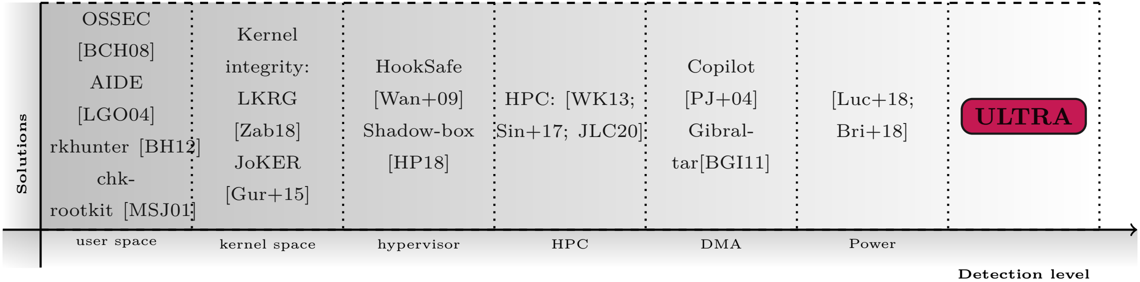
State of the art (2)

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.

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

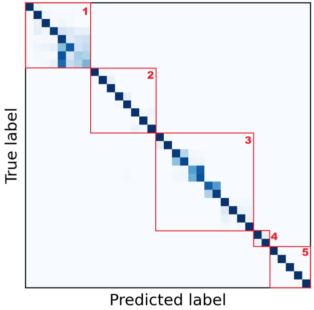
Article	SCM detection	Anomaly detection	SCM classification	Real-world SCM	Real-world analysis environment	Samples size	Variations	Benign dataset	Window size	Open data, source code	Device under test
WattsUpDoc [Cla+13]	✓	-	-	✓	-	15	-	-	5s	-	Windows XP Embedded 664 MHz
IDEA [Kha+19]	-	✓	-	-	-	3	-	-	$<40\mu s$	-	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



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	Classification	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
	[Sin+17]	-	-	-	✓	-	5	-	-	-	45s	VMWare Windows 7 Intel
	[JLC20]	-	-	✓	✓	-	4	-	-	-	2.91s	ARM Cortex-A53
DMA	Copilot [PJ+04]	-	-	-	-	-	12	-	-	-	30s	PCI-compatible Intel PC Linux
	Gibraltar [BGI11]	-	-	-	-	-	23	-	✓	-	20s	PCI-compatible Intel PC Linux
Power	[Luc+18]	-	-	-	✓	✓	5	-	-	✓	>5m	PC Windows 10 & Ubuntu 14
	[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 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.

- 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

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.*

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‡

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	$75.0_{[40]}$ $^{75.4}/_{75.0}$	$50.6_{[10]}$ $^{51.5}/_{55.6}$	$59.2_{[9]}$ $^{59.4}/_{59.2}$