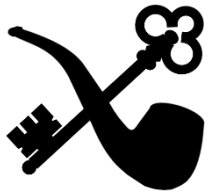
BIGMOMAL — Big Data Analytics for Mobile Malware Detection

SARAH WASSSERMANN, PEDRO CASAS

SherLock Dataset

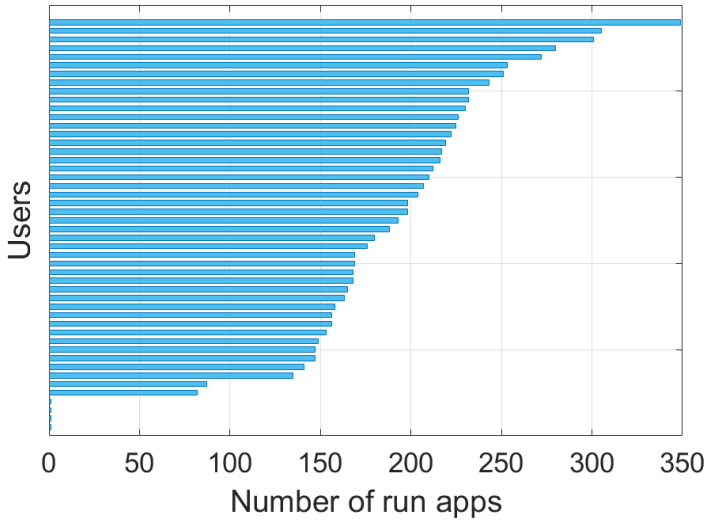
- Dataset published by the BGU Cyber Security Research Center
- Smartphone data collected from 50 Android users, from 01.2015 until 12.2017
- 2 agents
  1. **SherLock**: app to gather information about the smartphone
  2. **Moriarty**: app to simulate malicious behaviour on the device
- 600,000,000,000+ samples



SherLock vs Moriarty: A Smartphone Dataset for Cybersecurity Research
Yisroel Mirsky, Asaf Shabtai, Lior Rokach, Bracha Shapira, Yuval Elovici
in Proceedings of 9th ACM Workshop on Artificial Intelligence and Security (AISec)

Dataset Analysis (1)

- **In this work, we focus on the Q2 2016 dataset**
- 600,000,000+ records
- Features related to the network traffic generated by the apps, and corresponding to the footprint of the app on different components

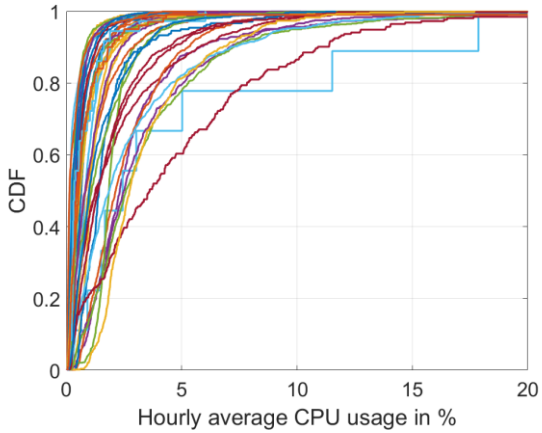


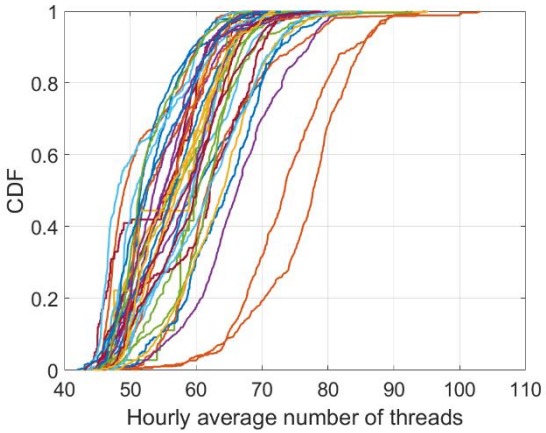
Rank	Application name
#1	Chrome
#2	Google App
#3	WhatsApp
#4	S Finder
#5	S Health
#6	Hangouts
#7	Samsung text-to-speech engine
#8	Geo News
#9	Peel Smart Remote
#10	Contacts

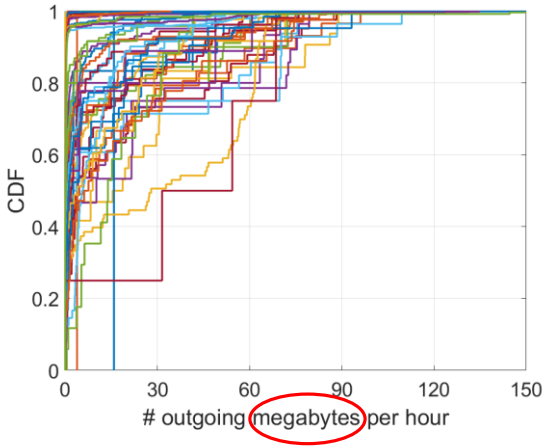
10 most popular apps

Dataset Analysis (2)

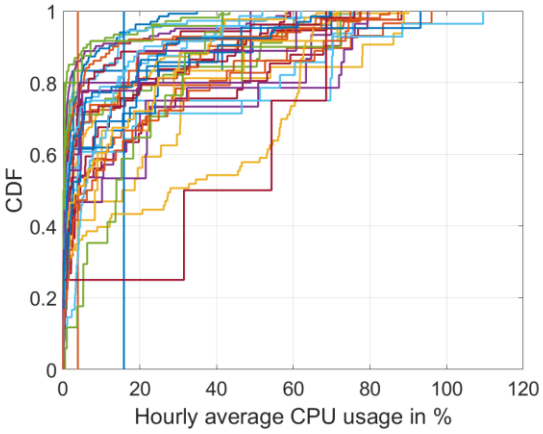
WhatsApp

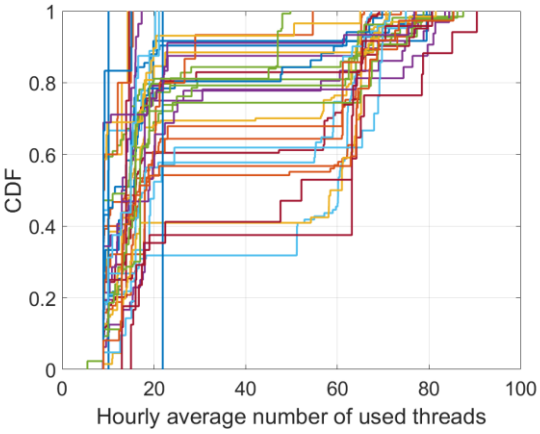


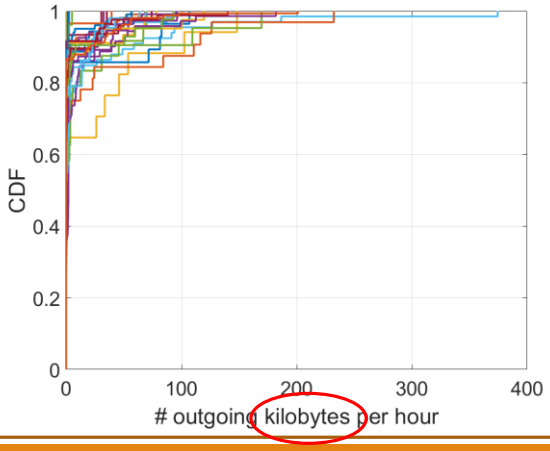




Moriarty







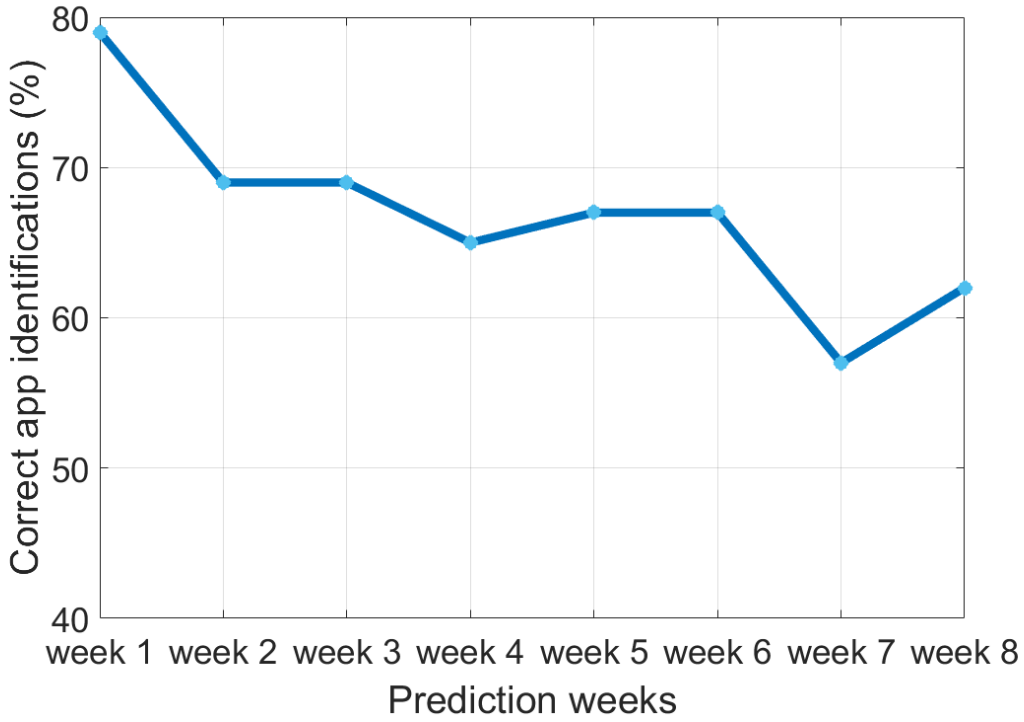
Application Identification (1)

- **Goal**: identify a running application on the user's smartphone
- Prediction based on 45 **SherLock features**
- **Decision tree** as prediction model
- for each user: 20% of data for training, 80% for testing

→ Identification accuracy: **99.9%**

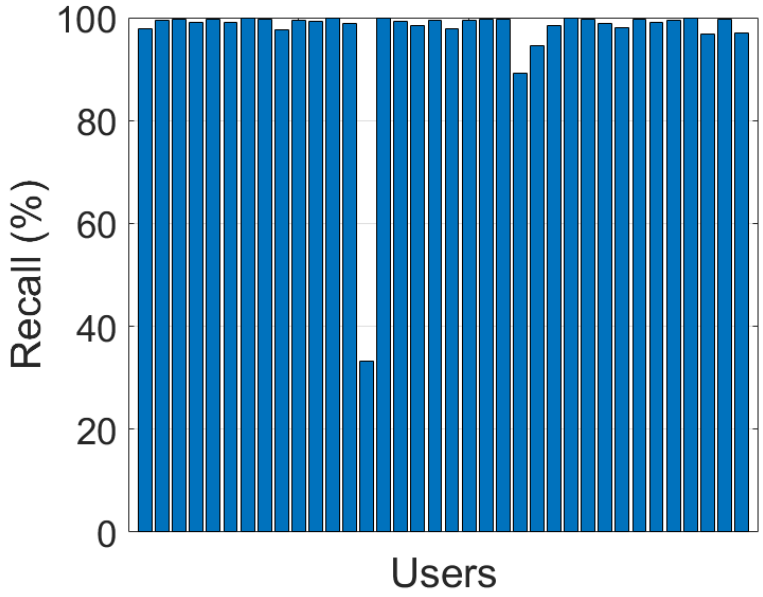
Application Identification (2)

- Prediction on a **weekly basis**
  - What about training during the first week and testing in the following ones?



Malware Detection (1)

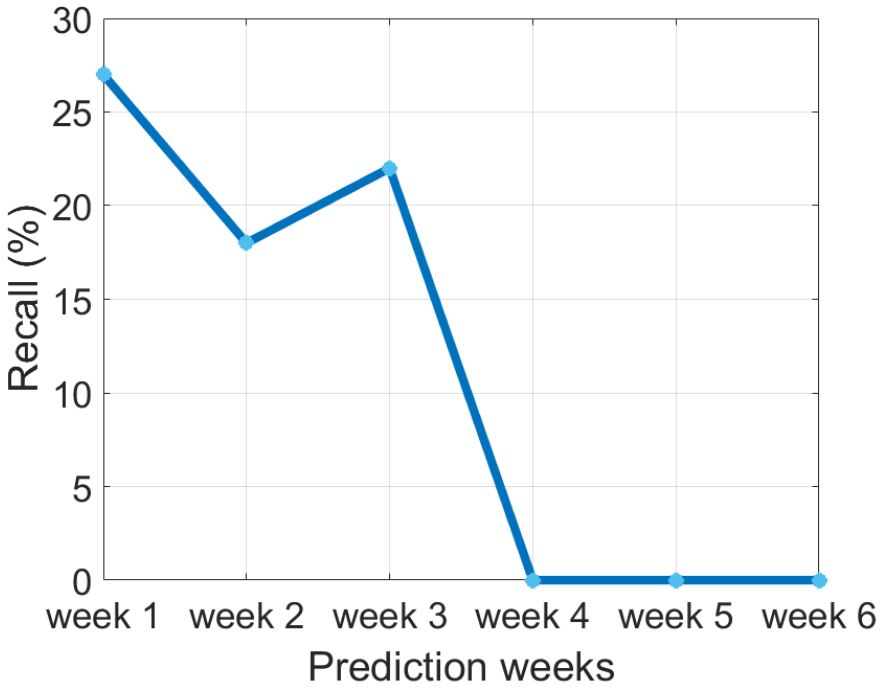
- **Goal**: detect whether the running app is Moriarty or not
- Prediction based on 45 **SherLock features**
- **Decision tree** as ML model
- For each user: 40% of data for training, 60% for testing



	predicted malicious	predicted benign
real malicious	55,210	604
real benign	5578	30,405,691

Malware Detection (2)

- Moriarty detection on a **weekly basis**
  - as for the application detection, results are not very promising



Concept drift!

Malware Detection (3)

- **Better together?** Detecting Moriarty through different users
  - Data of 3 users for training, data of 3 others for testing
- Results are disappointing
  - ➢ This underlines the diversity among the users
  - ➢ Building a malware-detection tool from multiple users is thus very challenging

	predicted malicious	predicted benign
real malicious	3472	35,401
real benign	16,784	79,775,923

Feature Selection

Features	Application identification	Malware detection
#1	virtual memory size	shared dirty pages used by Dalvik heap
#2	number of threads	number of minor page faults
#3	CPU utilisation	time process scheduled in user mode
#4	process priority level (foreground, background, service, sleeping, etc.)	proportional set size for Dalvik heap
#5	ordering within a particular priority category	time process-children scheduled in user mode
#6	process life time	virtual memory size
#7	time process scheduled in user mode	time process scheduled in kernel mode
#8	time process-children scheduled in user mode	process ID
#9	number of minor page faults	process life time
#10	number of private dirty pages used by everything else than the native heap	number of private dirty pages used by the native heap

Highly similar performance when using only top 5/10 features!

Conclusion and Future Work

- We tackled **2 prediction tasks** in the domain of **smartphone security**
- **Highly promising results** for both
  - But some scenarios remain very challenging, most probably due to concept drift
- Next step: stream-based machine-learning approaches


Q
&
A