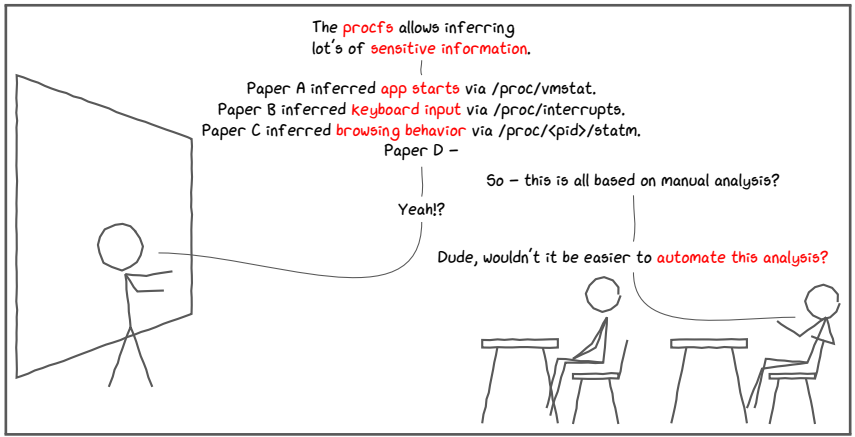
# ProcHarvester: Fully Automated Analysis of Procfs Side-Channel Leaks on Android

Raphael Spreitzer, Felix Kirchengast, Daniel Gruss, Stefan Mangard
IAIK, Graz University of Technology, Austria

AsiaCCS 2018, Incheon, Korea, 8th June 2018

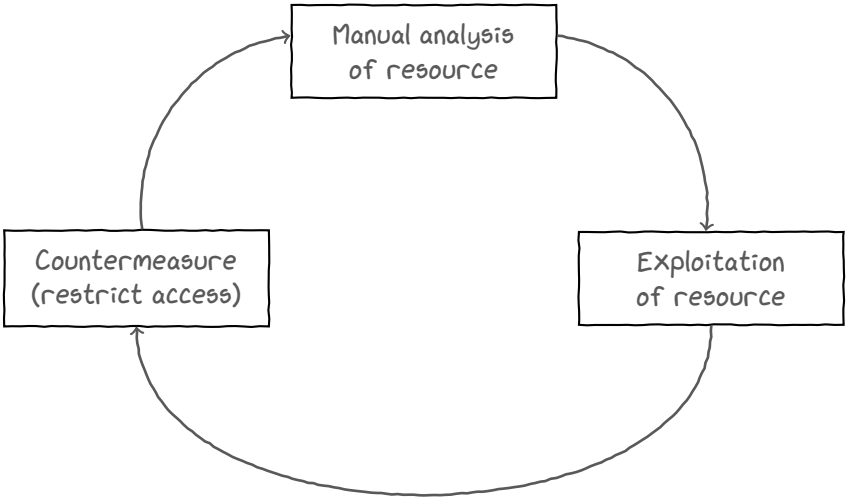
# Motivation and Contribution



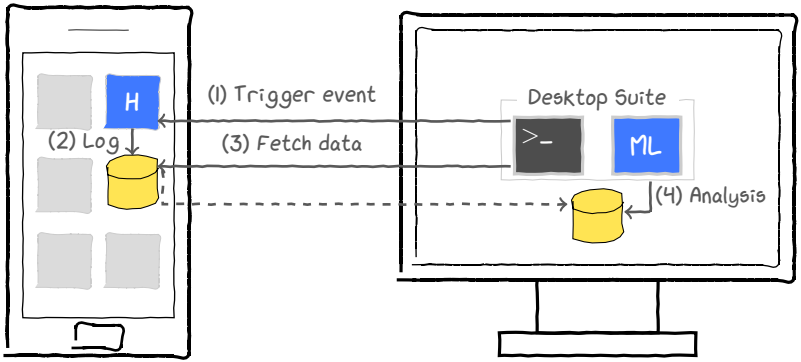
# The Linux procfs

- Virtual file system mounted under /proc/
- Per-application information
  - /proc/<pid>/...
  - /proc/uid_stat/<uid>/...
- Global information
  - /proc/interrupts
- Attacks and restrictions
  - Android 6: /proc/<pid>/ (partially) restricted
  - Android 7: procfs mounted with hidepid=2
  - Android 8: /proc/interrupts restricted

# Cat and Mouse Game?



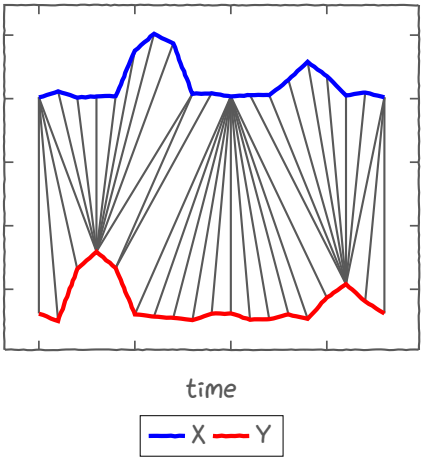
# ProcHarvester



# Analysis

## Dynamic time warping (DTW)

- Compare time series
  - $X = (x_1, \ldots, x_n)$
  - $Y = (y_1, \ldots, y_m)$
- No background information
- No human interaction
- Ignoring misaligned, stretched, or compressed traces



# Classification

## DTW-based approach (template attacks)

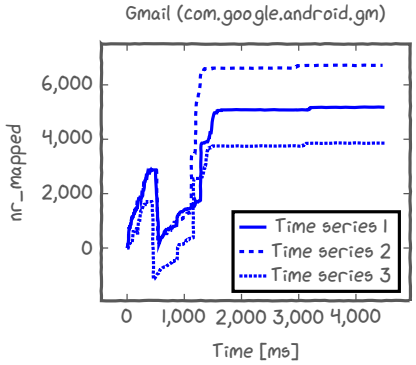
- Training data: $T = \{(e_i, X_i)\}$
- Test sample $s = (e_j, X)$: $i = \text{argmin } DTW(X, Y_i)$
- $\Rightarrow$ two time series result from the same event if they yield a low distance to each other

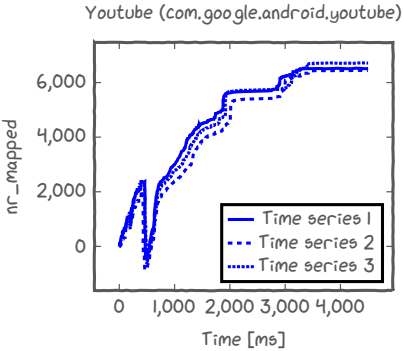
## K-fold cross validation

- Accuracy better than random guessing?
- $\Rightarrow$ information leak identified

# Case Study: App Inference

## Correlations between app start events and procfs information



# App Inference on Android 7

| procfs file | Property | Accuracy |
|---|---|---|
| /proc/vmstat | nr_mapped | 82.2% |
| /proc/net/sockstat | Sockets used | 74.1% |
| /proc/net/dev | wlan0: Receive bytes | 73.8% |
| /proc/vmstat | pgfault | 73.3% |
| /proc/interrupts | kgsl3do | 71.5% |
| /proc/vmstat | nr_anon_pages | 71.3% |
| /proc/interrupts | arch_timer | 70.1% |
| /proc/net/dev | wlan0: Transmit bytes | 68.4% |
| /proc/interrupts | MDSS* | 67.6% |
| /proc/sys/fs/inode-state | nr_inodes | 65.0% |
| /proc/interrupts | Rescheduling interrupts | 62.9% |
| /proc/vmstat | nr_dirty_threshold | 62.2% |
| /proc/vmstat | nr_shmem | 58.9% |
| /proc/meminfo | VmallocUsed | 55.9% |
| ⋮ | ⋮ | ⋮ |

* Also reported by Diao et al. [DLLZ16]

# App Inference on Android 8

| procfs file | Property | Accuracy |
|---|---|---|
| /proc/net/sockstat | sockets: used | 86.3% |
| /proc/net/xt_qtaguid/iface_stat_all | eth0: tx_packets | 77.2% |
| /proc/net/xt_quota/eth0 | eth0: interface quota | 76.9% |
| /proc/net/protocols | UNIX: sockets | 76.3% |
| /proc/net/xt_qtaguid/iface_stat_fmt | eth0: total_skb_tx_packets | 76.3% |
| /proc/meminfo | AnonPages | 76.3% |
| /proc/meminfo | Active(anon) | 75.9% |
| /proc/meminfo | MemFree | 70.9% |
| /proc/meminfo | Mapped | 62.5% |
| /proc/meminfo | Shmem | 55.0% |

# Attack Scenario

## Infer app starts from unprivileged app

- Allows more targeted attacks
- Bypass GET_TASKS, REAL_GET_TASKS

## Analysis phase

- Gather procfs resources for apps of interest
- Establish fingerprint database (templates)

## Attack phase

- Malicious app monitors procfs
- Infer app starts

# Evaluation

## Android 7

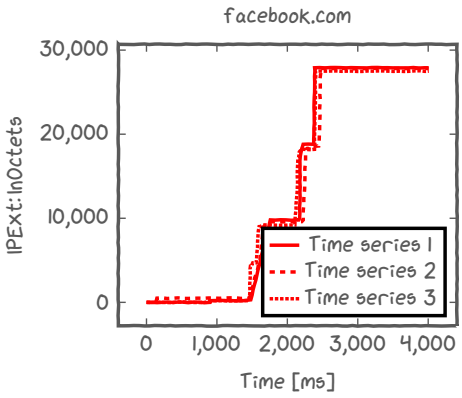
| Attacks | # Apps | Accuracy |
|---|---|---|
| App cold starts | 100 | 96% |
| App resumes | 20 | 86% |
| Mixed (cold starts and app resumes) | 20 | 90% |
| Manual cold starts (by human being) | 20 | 98% |

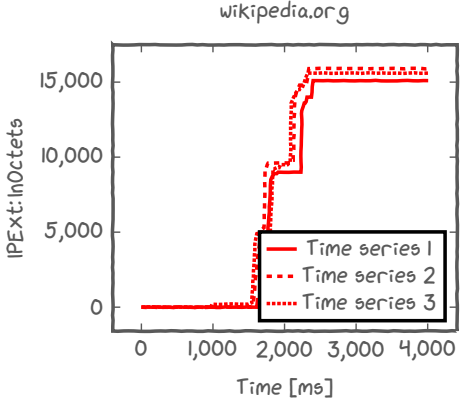
## Android 8

- 20 apps: inference accuracy of 87%

# Case Study: Website Inference

## Correlations between website launches and procfs information



# Discussion

## Limitation: false negatives

- No leaks identified $\rightarrow$ secure?
- More specialized features

## Countermeasures

- Restrict access to procfs resources
- ProcHarvester could be used to eliminate side channels in upcoming Android versions (before they are released)

# Take-Home Message

## Procfs side channels are still a threat

- Several new side channels on Android 7 and Android 8
- Some leaks "moved"
  - /proc/vmstat (Android 7) → /proc/meminfo (Android 8)

## ProcHarvester

- Framework to scan the procfs automatically
- Available at https://github.com/IAIK/ProcHarvester

# ProcHarvester: Fully Automated Analysis of Procfs Side-Channel Leaks on Android

Raphael Spreitzer, Felix Kirchengast, Daniel Gruss, Stefan Mangard
IAIK, Graz University of Technology, Austria

AsiaCCS 2018, Incheon, Korea, 8th June 2018

# Disclaimer

The xkcd comic, in particular the stick figures, and the plots have been drawn based on StackExchange [stal2] and the xkcd comic "Teaching Physics" [xkcll].

# Bibliography

[DLLZ16] Wenrui Diao, Xiangyu Liu, Zhou Li, and Kehuan Zhang.
No Pardon for the Interruption: New Inference Attacks on Android Through Interrupt Timing Analysis.
In IEEE Symposium on Security and Privacy — S&P 2016, pages 414–432, 2016.

[stal2] StackExchange: Create xkcd style diagram in TeX.
https://tex.stackexchange.com/questions/74878/create-xkcd-style-diagram-in-tex/74881#74881, 2012.
Accessed: May 31, 2018.

[xkc11] xkcd Comic: Teaching Physics.
https://xkcd.com/895/, 2011.
Accessed: May 31, 2018.