Towards Energy-Aware Intrusion Detection Systems on Mobile Devices

M. Curti, A. Merlo, M. Migliardi, S. Schiappacasse

The 2013 International Conference on High Performance Computing & Simulation (HPCS 2013)

3rd of July, 2013

Agenda

- The idea of “Energy-aware” IDS
- Modelling the Energy Consumption on Mobile Devices
- Measuring the energy consumption of a Wi-Fi component
- Energetic Signatures of legal/illegal activities: some experiments
- Future developments & conclusion

Energy-Aware Intrusion Detection Systems

- Intrusion Detection has been widely adopted in wired and wireless networks:
  - Signature-based detection
  - Anomaly-based detection

- The idea: <u>Energy-Aware</u> Intrusion Detection on <u>smartphones</u>
  - Signature-based: “energy footprints” of malware
  - Anomaly-based: “non standard” energy consumption profile

Why smartphones, IDS and energy-awareness?

- Smartphones and tablets (S&T) are becoming the «new computing paradigm» → they store personal information
- The consolidation of the Android mobile OS (deployed on the 72% of smartphones in 3Q2012) is pushing the adoption of S&T also in professional scenario (US DoD, Chicago Hospital, Loyola University Medical Center, BYOD paradigm in private and public agencies) → a honeypot for malware developers!

Why smartphones, IDS and energy-awareness?

- S&T strongly depend on battery → emerging battery-drain attacks are aimed at mining the availability of devices.
- There is a cornucopia of standard approaches to malware detection ported on Android platforms based on malware "behavioral signatures"→ we aim to extend such detection with "energetic signatures" useful for both signature-based and anomaly-based malware detection.

The road to Energy-Aware IDS on Android…

… into five steps:

1. Modeling the energy consumption of mobile devices
2. Measuring energy consumption in Android
3. Building "energetic signatures" of legal/malicious behaviors
4. Populating a database of signatures
5. Testing controlled but realistic applicative scenarios
6. Implementing an Android IDS

Modeling the energy consumption

- We defined a general consumption model defining the total consumption C of the device as the *sum of consumptions of each hardware component*:

$$C = \sum_i (f_i + g_i) = B + P_s$$

where:

f_i = base consumption of the i-th component

g_i = activity-specific consumption related to the i-th component

B = $\sum_i f_i$ = base consumption of all components

P_S = $\sum_i g_i$ = consumption of the single activity

Modeling the energy consumption

- The general model is based on the idea of *isolating* the power contribution of different hardware components.
  - Measuring single contributions may require specific per-component modules.
  - Such modules may require modelling the characteristics of the components.
- We modeled the energy consumption of a Wi-Fi component in terms of sending/receive operations.
- We implemented a measurement module, accordingly.

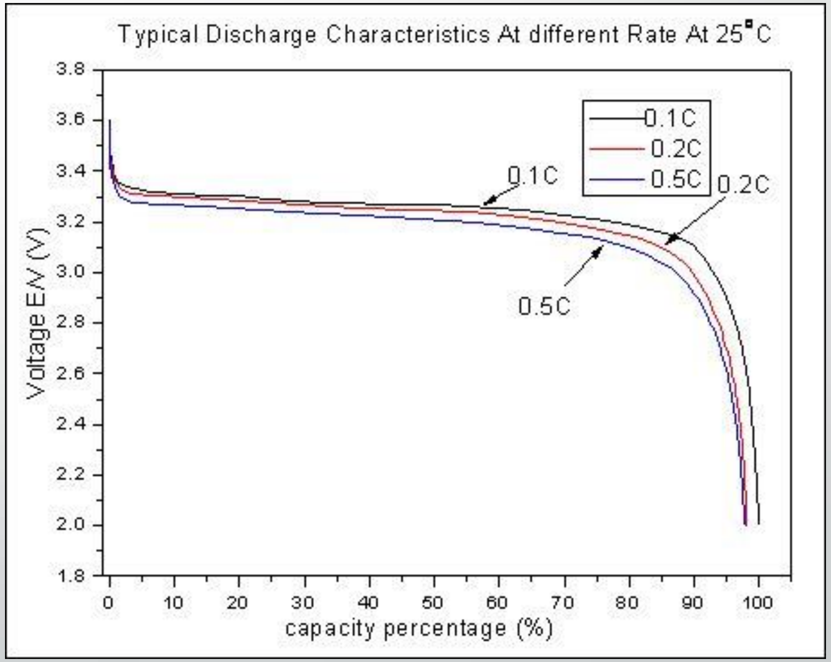
Measuring energy consumption in Android

- Why implementing an ad-hoc kernel module? Because currently available Android measurement tools are insufficient:
  - PowerTutor → coarse consumption measurement;
  - AppScope → strictly tied with the device model (static reference), available only for a single device (Google Nexus One), closed source.

- Thus, we have implemented an Android kernel module able to measure the *instantaneous* Power Consumption as **W=V*I** of sending/receiving single packets.

Building an Android Wi-Fi consumption module

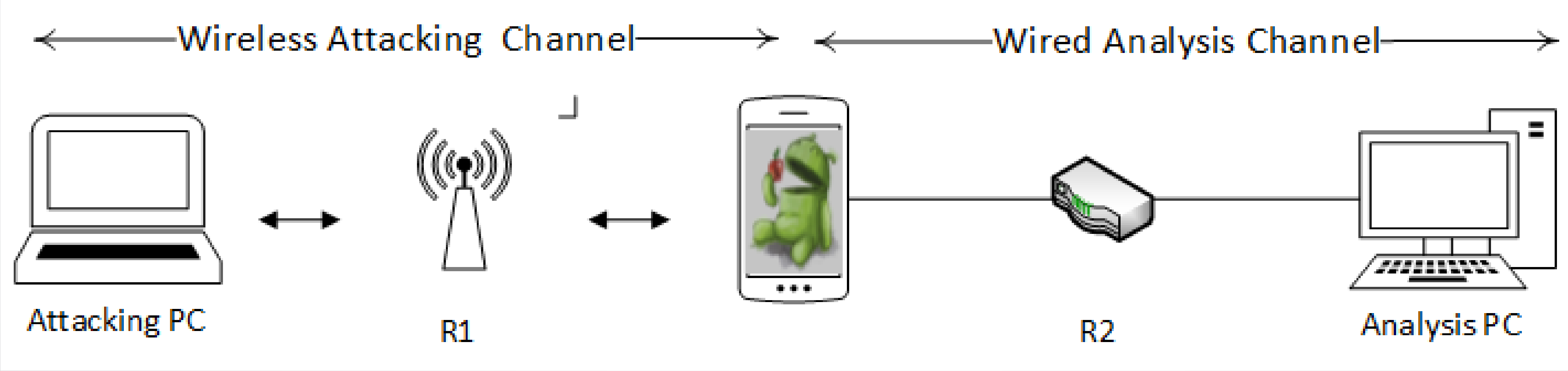
- Issues to solve:
  - Mobile batteries provide variable voltage → battery level should be kept into account during measurement
  - All Android devices provides the mere global consumption → there is no way to discriminate single contributions.
  - Few Android smartphones provide both the instantaneous value of I and V.



Building an Android Wi-Fi consumption module

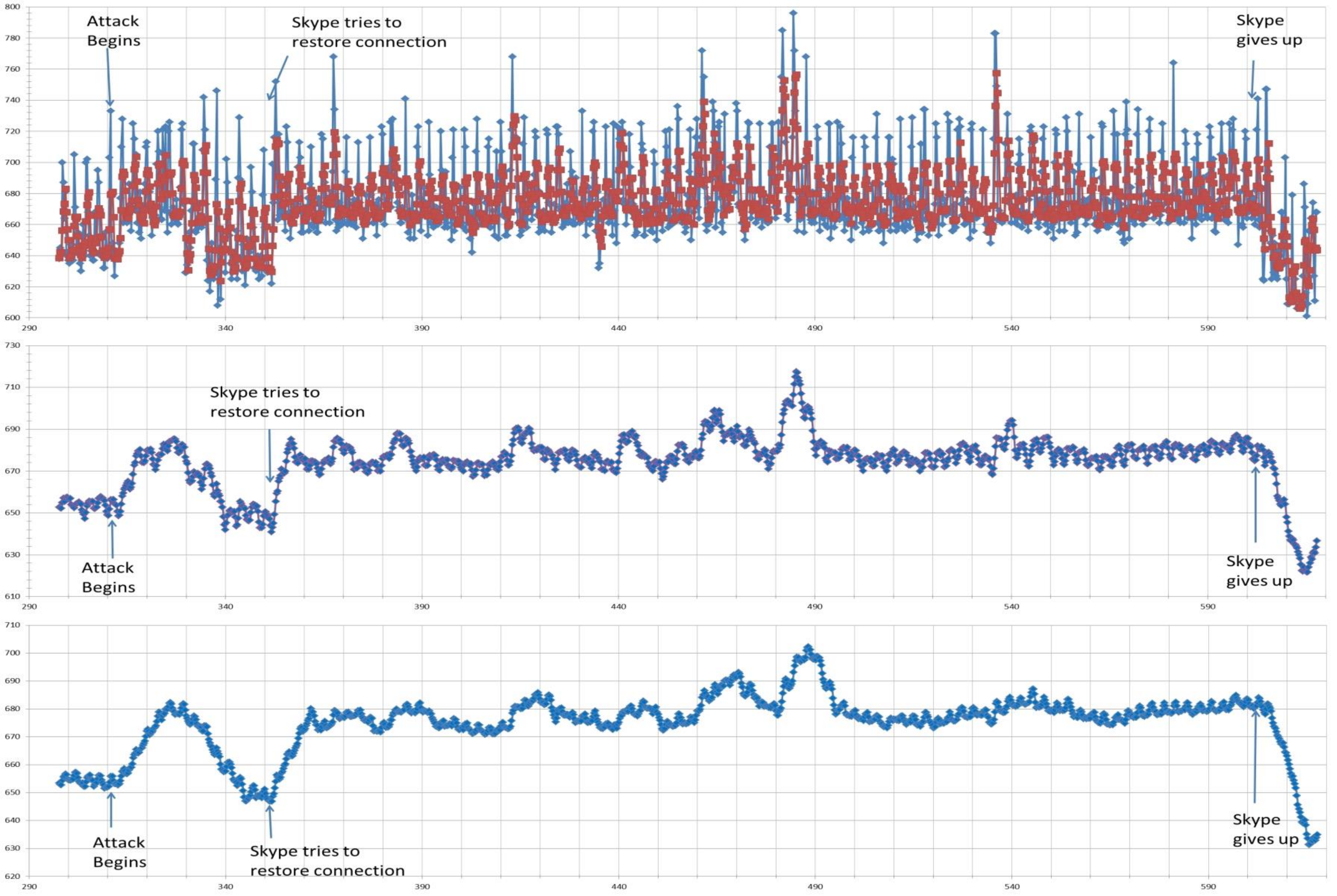
- Some consumption measurements are provided by the battery driver.
- Our module needs to calculate both *voltage* and *current*. However, the coulomb counter (CC) is provided by the battery driver in less than 20% of devices.
- The *ab_8500fg* (used in Sony Xperia and some early Samsung) implements a CC regularly measuring current (250ms); however, it is not natively accessible → we customized the *ab_8500fg* driver in order to get access to the corresponding kernel primitives.
- The module also hooks kernel functions providing the instant transmission rate and the number of byte exchanged in a period.

Experimental Results

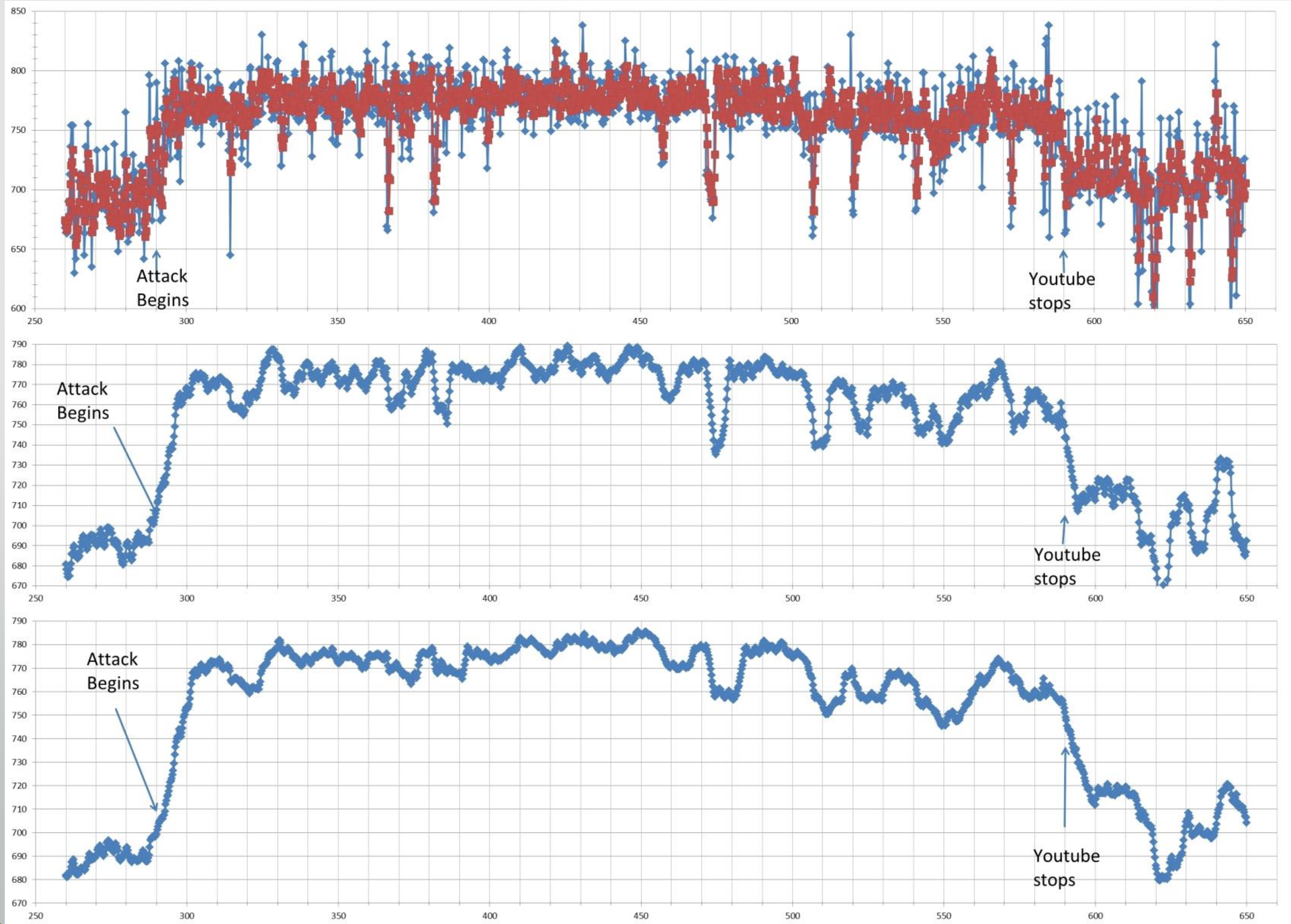


- We tested the our modified battery driver on both a *Sony Xperia U* and a Galaxy S Advance.
- We tested an "early" Energy-Aware IDS in the form of an *Android app*.
- We defined proper script to keep consumptions of other components as much negligible as possible.
- We calculated Wi-Fi energetic signatures of:
  - Legal activities: Skype, YouTube, Shazam;
  - Attacks: Ping Flood, Repeated HTTP GET Requests.

Skype + Ping Flood attack



YouTube + HTTPGET attack



Conclusions & Future Works

- Our experiments show that
  - Energy-based analysis of attacks for mobile devices is feasible and promising.
  - there is no simple superposition of effects → finer grained analysis must be performed. Energy contributions from other components are needed.
- Future directions will focus on:
  - Modeling other components and building measurement modules;
  - Calculating energy signatures from other legal activities/attacks;
  - Defining energy-based *strategies* for the intrusion detection…

… and, of course, keep developing the Energy-Aware IDS application for Android!

THANK YOU !!!