Classifying Smartphone Screen ON/OFF State Based On WiFi Probe Patterns

Shuja Jamil
Science and Technology Unit
Umm Al-Qura University
Makkah, Saudi Arabia
sjamil@gistic.org
Univ. Grenoble Alpes, LIG
F-38000 Grenoble, France
Shuja.Jamil@imag.fr

Anas Basalamah
Computer Engineering
Department & Science and
Technology Unit
Umm Al-Qura University
Makkah, Saudi Arabia
ambasalamah@uqu.edu.sa

Sohaib Khan
Science and Technology Unit
Umm Al-Qura University
Makkah, Saudi Arabia
saashfaq@uqu.edu.sa

Ahmed Lbath
Univ. Grenoble Alpes, LIG
F-38000 Grenoble, France
Ahmed.Lbath@imag.fr

UbiComp/ISWC '16 Adjunct, September 12-16, 2016, Heidelberg, Germany
ACM 978-1-4503-4462-3/16/09.
http://dx.doi.org/10.1145/2968219.2971377

Abstract
WiFi enabled smartphones regularly send probe request messages to actively discover nearby access points. Previously, these probes have been exploited to understand users' mobility patterns and were also identified to be a privacy threat. To highlight the increased threats to user's privacy, we aim to identify the user behavior by characterizing the changes in probe patterns, which occur as an effect of user's smartphone usage. In this demo, we developed a browser based interactive front-end client, with a Node JS backend server, to interact with the sniffing hardware and to display realtime packets visualization with smartphone screen state predictions. We were able to detect the screen ON and OFF states by analyzing the probe patterns using a Decision-Tree classifier with accuracy ranging from 93% to 100% on specific smartphone models.

Author Keywords
WiFi; Probe Request; Sniffing; Privacy; Classification; Smartphone Usage.

ACM Classification Keywords
C.2.1 [Network Architecture and Design]: Wireless communication; K.4.1 [Public Policy Issues]: Privacy

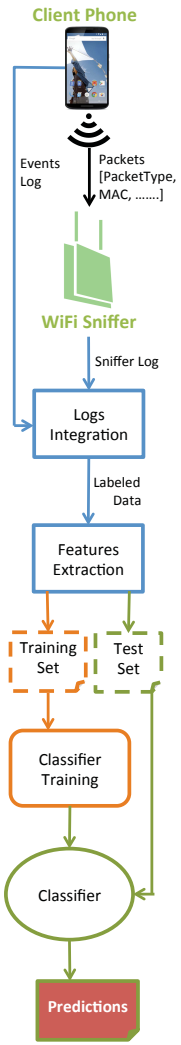

Figure 1: System Architecture.

Introduction

The high demand of cloud associated services require mobile devices to look for a nearby WiFi Access Point (AP) and remain connected to a WiFi network for data offloading. The commonly utilized active lookup method to discover nearby WiFi APs require the mobile devices to periodically broadcast Probe Request messages. This method is commonly adopted due to its faster discovery speed, and possibility to find hidden networks.

WiFi probes have been identified as a privacy concern, mainly because they include a MAC address, which is a unique device identifier, sent in clear text. This unique identifier can be used to locate and track the movement of a mobile device with the help of passive adversary listening for the WiFi traffic over the air. For instance, in [5, 3], authors were able to detect the presence of nearby smartphones, extract their trajectories, infer social structure and the socioeconomic status of individuals in the crowd by just sniffing WiFi probes in the ambient environment. Industry implementations [2, 7] use probes to understand population and traffic in specific regions, as well as dwelling times in malls and other indoor spaces. One recent work attempts to quantify WiFi probe requests' threat to privacy by conducting an experimental study of popular smartphones in different settings [4].

The objective of this work is to identify the user behavior by characterizing the Probe Request patterns originating from the user's smartphone. In this demo, we build an interactive system to passively capture and analyze the probe patterns, in order to highlight the increased threat to user's privacy caused by the changing probe patterns. Specifically, we extract unique probing patterns from the captured log and use them to classify the screen ON and OFF states of the user's smartphone, with more than 93% accuracy.

System Overview

Initially, we conducted an explorative study to understand the probing behavior of various smartphones. Based on the findings of the study, we developed a smartphone screen state classification system. Following subsections describe the details of the system illustrated in Figure 1.

Experimental Setup

WiFi traffic is easily captured using commodity hardware, such as, Personal Computers (PC) [1] or specialized hardware [6]. We use a PC built-in WiFi adapter to capture the packets originating from the client smartphone. Moreover, we deploy an event logging application on a number of client smartphones to understand their probing behavior in different states. The event log keeps chronological record of various events occurring on the phone, which helps to identify specific events affecting the probing pattern.

Data Gathering

We first set the initial state of the client smartphone as shown in Table 1. Then we start the sniffing hardware to capture ambient packets originating from the nearby smartphones. Next, we periodically turn ON and OFF the client smartphone screen for specific duration. Each screen ON and OFF state is recorded on the smartphone along with the timestamp using the event logging application. The sniffer constantly captures probe packets from the smartphone throughout the experiment duration.

Data Exploration

We observed that probe messages appear more frequently when the smartphone is not connected to a WiFi network, and reduce significantly or disappear as soon as the device is connected to a network. During disconnection phase, we observed consistent discriminatory probing patterns in the screen ON and OFF cycles, across multiple smartphone

Setting	Value
WiFi	Disconnected
GPS	OFF
Charging	Disconnected
Screen	OFF

Table 1: Client smartphone initial settings.

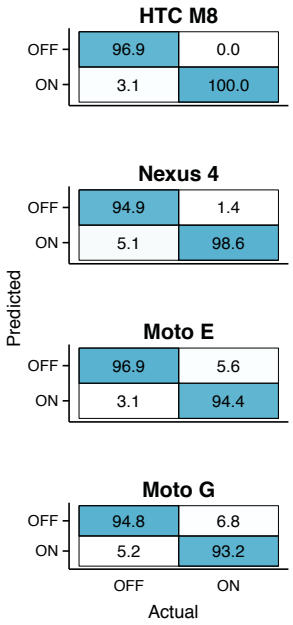

Figure 2: System Accuracy (%).

models. Generally, the frequency of probes reduces significantly when the screen is turned OFF.

Screen State Classification

To develop a smartphone screen state classifier, we first obtain the ground truth by integrating the sniffed probes log with the events log recorded on the client phone. This step provides us with a labeled data set, that is, probe variations marked with events happening on the smartphone. Next, we compute features on the labeled data set by aggregating the n seconds data, and then apply moving averages with a window of size m, where n and m are smartphone specific parameters computed based on the duration of delay between probe bursts. To account for the variations between probe burst delays across different smartphone models, we also compute the time-since-last-peak value for every data point in the data set. Next, the first 70% samples of the computed features set are used to train a Decision-Tree classifier, which gives a model for predictions. The remaining 30% of the data is then provided to the trained model to predict the screen state and evaluated against the event labels. The accuracy results for the selected smartphone models are shown in Figure 2. Finally, the trained classifiers are used to predict the screen states in realtime.

Scope of Work

There exist a large variety of smartphone models, where a user is free to choose any Operating System (OS) version for his device. During the explorative study, we observed significant variations in the probing behavior based on the vendor and OS version differences. We restrict the scope of this work by selecting a limited number of smartphone models and OS versions, as shown in Table 2. Moreover, we only focus on a scenario where smartphones are not connected to a WiFi network and can easily expose themselves to an adversary who is scanning for ambient packets.

Figure 3: Demo User Interface.

Demo Description

In this work, we show the passive sniffing of the probe packets from a nearby smartphone and predict whether the smartphone is being used or not. Our demo comprises of a sniffing hardware and an interactive software application.

Demo UI

Figure 3 shows the user interface of the browser based interactive tool which comprises of three panels. 1) *Right Panel*: At the top, the target client phone is selected which we want to listen to for the probes. Next, the start and stop buttons are used to instruct the sniffing hardware for probe collection. At the bottom, the captured packets log appears as soon as the probes start to emerge from the selected smartphone. In the middle, a smartphone is being displayed which simulates the screen OFF and ON states in realtime. 2) *Bottom Panel*: This panel visualizes the detected probe counts in a live timeseries graph. This live graph is useful to observe the frequency of instantaneous

Samrtphone	OS Version
HTC M8	5.0.1
Nexus 4	5.1.1
Moto E	5.0.2
Moto G	4.4.4

Table 2: Selected smartphone models and Android OS versions.

probes and the inter probe burst delays, which vary across different smartphone models. 3) *Center Panel*: The center panel is used to show the results of offline analysis, where the overall performance of the system is displayed, once the sniffing is stopped.

Implementation

We developed a NodeJS server with which the front-end application interacts to fulfill the desired functionality of the system with the help of R-script implementations. 1) *Sniffer Controls*: When the sniffing is started, the ambient packets are captured by asynchronously executing the TShark tool which interacts with the hardware for sniffing packets. 2) *Logging and Visualization*: As soon as the packets are captured by the TShark tool, the capture log is reported to the NodeJS server, which then pushes it back to the front-end to display as a log and on the live graph. 3) *Realtime Prediction*: Periodically, the front-end client requests the NodeJS server to process the last *N* seconds of data and predict the current screen state. For prediction we use a Decision-Tree classifier, which was pre-trained for a variety of phones. 4) *Offline Analysis*: When the sniffer is stopped, the complete log is processed using the pre-trained Decision Tree classifier to identify the screen state changes during the whole run.

Conclusion

In this work, we demonstrate the smartphone screen state prediction using passive and non-intrusive WiFi sniffing to capture ambient packets emerging from the nearby smartphone. We are able to successfully classify the sniffed packets information in realtime and offline with classification accuracy ranging from 93% up to 100% on selected smartphone models. This result shows another level of privacy threat, where a passive adversary can easily track the user's engagement with the smartphone.

Acknowledgements

This work was supported by the KACST National Science, Technology and Innovation Plan under grant #11-INF2061-10, and the Science and Technology Unit at Umm Al-Qura University.

REFERENCES

1. WLAN (IEEE 802.11). 2016. Capture Setup using WireShark. `https://wiki.wireshark.org/CaptureSetup/WLAN`. (23 July 2016).
2. Averos. 2016. Loris TMS Solutions. `http://www.averos.com`. (23 July 2016).
3. Marco V Barbera, Alessandro Epasto, Alessandro Mei, Vasile C Perta, and Julinda Stefa. 2013. Signals from the crowd: uncovering social relationships through smartphone probes. In *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 265–276.
4. Julien Freudiger. 2015. How Talkative is Your Mobile Device?: An Experimental Study of Wi-Fi Probe Requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '15)*. ACM, New York, NY, USA, Article 8, 6 pages.
5. A. B. M. Musa and Jakob Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors.. In *SenSys*, M. Rasit Eskicioglu, Andrew Campbell, and Koen Langendoen (Eds.). ACM, 281–294.
6. WIFI PINEAPPLE. 2016. MARK V STANDARD. `https://hakshop.myshopify.com/products/wifi-pineapple`. (23 July 2016).
7. Blip Systems. 2016. Helping Transportation Sectors and Retail. `http://blipsystems.com/`. (23 July 2016).