AntMonitor: Network Traffic Monitoring and Real-Time Prevention of Privacy Leaks in Mobile Devices

Anastasia Shuba
CalIT2, EECS, CPCC
UC Irvine
ashuba@uci.edu

Anh Le
CalIT2, UC Irvine
anh.le@uci.edu

Minas Gjoka
CalIT2, UC Irvine
mgjoka@uci.edu

Janus Varmarken
IT Univ. of Copenhagen
janv@itu.dk

Simon Langhoff
IT Univ. of Copenhagen
siml@itu.dk

Athina Markopoulou
CalIT2, EECS, CPCC
UC Irvine
athina@uci.edu

ABSTRACT

Mobile devices play an essential role in the Internet today, and there is an increasing interest in using them as a vantage point for network measurement from the edge. At the same time, these devices store personal, sensitive information, and there is a growing number of applications that leak it. We propose AntMonitor – the first system of its kind that supports (i) collection of large-scale, semantic-rich network traffic in a way that respects users' privacy preferences and (ii) detection and prevention of leakage of private information in real time. The first property makes AntMonitor a powerful tool for network researchers who want to collect and analyze large-scale yet fine-grained mobile measurements. The second property can work as an incentive for using AntMonitor and contributing data for analysis. As a proof-of-concept, we have developed a prototype of AntMonitor, deployed it to monitor 9 users for 2 months, and collected and analyzed 20 GB of mobile data from 151 applications. Preliminary results show that fine-grained data collected from AntMonitor could enable application classification with higher accuracy than state-of-the-art approaches. In addition, we demonstrated that AntMonitor could help prevent several apps from leaking private information over unencrypted traffic, including phone numbers, emails, and device identifiers.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network monitoring*; D.4.6 [**Operating Systems**]: Security and Protection—*access controls*

Keywords

Mobile Network Monitoring; Android Security; Privacy Leakage Detection

1. INTRODUCTION

Mobile devices, such as smart phones and tablets, have become ubiquitous. With multiple wireless interfaces, including Wi-Fi and 3G/4G, these devices have persistent Internet connectivity throughout the day. As a result, the amount of traffic generated by these devices has grown rapidly in recent years and is expected to grow 10 times in the next 5 years [1]. Consequently, collecting and studying mobile network traffic has become a critical task in network infrastructure planning and Internet measurement research.

The growth of these mobile devices has been accompanied by an increasing number of personal information leakage [2, 3]. Examples of such information include personally identifiable information (PII) that can be used to uniquely identify an individual in a specific context (IMEI, email), data associated with the user (contacts, SMS messages), and demographic information (age, location).

We present a novel system, called AntMonitor, to address the needs of researchers for mobile traffic data and the needs of users for enhanced privacy, as outlined below.

Objective 1: Large Scale, Semantic-Rich Data Collection. First, AntMonitor is compatible with Android OS versions 4.0+, which makes it work with more than 94% of Android devices today [4]. Second, AntMonitor is carefully designed to scale and supports tens of thousands of users [5]. Third, AntMonitor collects packet traces in PCAP Next Generation format [6], which allows the system to collect arbitrary information alongside with the raw packets, such as the names of applications that are associated with packets. Such information is only available at the client side, and yet it plays a critical role in subsequent analyses by providing ground truth for application classification. Fourth, AntMonitor is designed to provide maximum user comfort: it runs seamlessly in the background, does not require a rooted phone, and most importantly, has modest CPU and battery usage while maintaining high network performance [5]. Last, AntMonitor entices users by providing privacy protection as discussed next.

Objective 2: Enhanced User Privacy. First, to address privacy concerns in data collection, such as those discussed in the Menlo report [7], AntMonitor is designed to provide users with complete control over what data they may want to contribute. In particular, they can choose specific applications, and either full packets or just packet headers to contribute. Second, AntMonitor is able to

This work has been supported by NSF Awards 1228995 and 1028394. Varmarken and Langhoff were visiting UCI when this work was conducted. App Icon in Fig. 2(a): ©UCI Networking Group

S3'15, September 11, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3701-4/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2801694.2801707.

search unencrypted packets for sensitive information. Moreover, it can prevent this information from leaking on-the-fly, by blocking the current communication or replacing the sensitive strings with randomly generated ones, depending on the user's decision.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the design and implementation of AntMonitor. Section 4 concludes the paper.

2. RELATED WORK

There is a large body of work on collecting and analyzing network traffic data. Depending on the vantage point of data collection, there are the following approaches: (i) applications installed on the device [8], (ii) traffic collection inside the network [9], (iii) custom operating systems or rooted phones [10, 11], and (iv) the Virtual Private Network (VPN) based approach, which AntMonitor takes. (i) provides fine-grained but small-scale traces from a limited set of users; (ii) suffers from coarse-grained traces; and (iii) inconveniences the user. Although the VPN approach (iv) alters the path of the packets and introduces additional processing per packet, it allows for interception of all network traffic, and thus can enable useful features, *e.g.*, privacy leakage prevention. Most importantly, the VPN approach works on almost all mobile devices today.

Privacy Leakage Detection. Detecting leakage of privacy sensitive data has also been extensively studied in the literature. TaintDroid [2] was one of the early tools built to identify privacy leaks in realtime using taint tracking, and it was used to identify a variety of privacy leaks on 30 popular Android apps. Similar work [3] automatically explores the GUI of Android apps and uses Taintdroid to detect privacy leaks. These approaches, however, are not suitable for large-scale deployment as they require a rooted phone. Another approach is to use static analysis of binary code [12]; yet, this method can be fooled by obfuscated code. Meddle [13] also adopts the VPN-based approach and supports detection of private information leakage; however, this detection is carried out at the server, when the information already leaked out of the device.

AntMonitor. The full system description and a demo of AntMonitor will appear in the upcoming SIGCOMM [5] and MobiCom [14] workshops, respectively. A video demonstrating the capabilities of AntMonitor can be found on our website [15].

3. SYSTEM OVERVIEW

The AntMonitor system consists of three components: a client-side Android application, called AntClient, and two server applications, called AntServer and LogServer for routing and collecting packets, respectively. Fig. 1 shows how the three work together. Each component is described in detail elsewhere [5]. Here, we provide the overview of the functionalities of AntMonitor.

Traffic Interception and Routing. AntClient establishes a VPN service on the device. This VPN service creates a virtual (layer-3) TUN interface that intercepts all outgoing traffic. Once a packet arrives at the TUN interface, AntMonitor sends it through a UDP socket to AntServer. AntServer routes the packets to the intended Internet host and delivers responses back to AntClient using another TUN interface and IP Masquerading (packet forwarding with Network Address Translation).

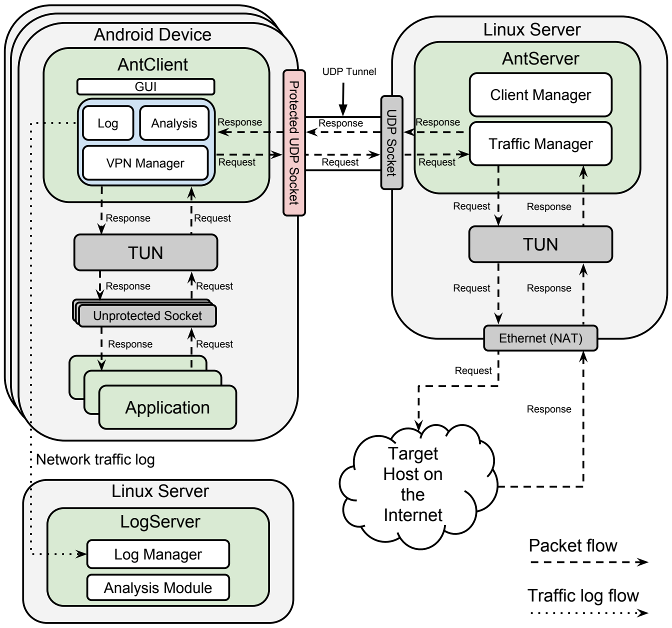


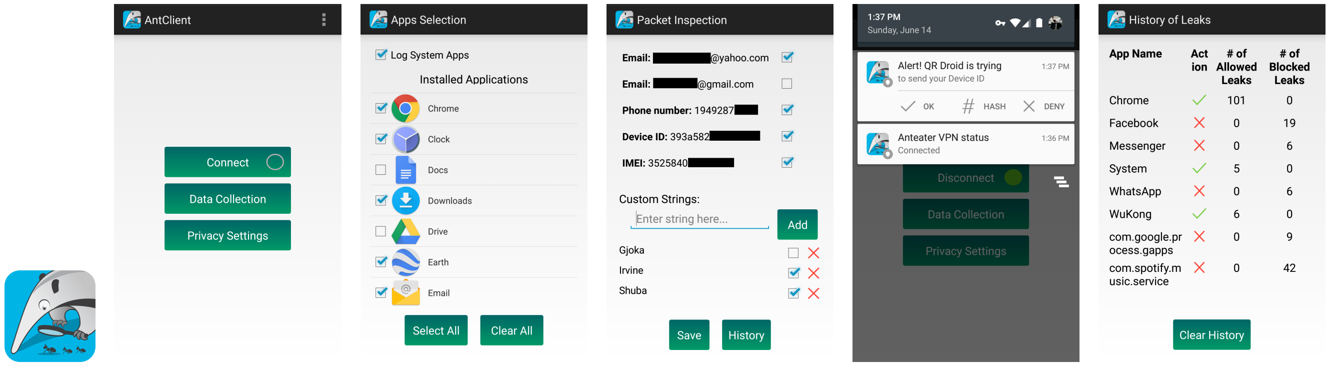
Figure 1: AntMonitor System Overview

Data Collection. AntClient saves packets in log files and uploads them to LogServer at a later time, *e.g.*, when the device is charging and has Wi-Fi or when explicitly requested by the user. LogServer extracts features from the log files and inserts them into a database to support various types of analysis. LogServer receives crowd-sourced data from a large number of devices, which enables global large-scale analysis. In our pilot deployment to volunteering students at UC Irvine, we collected and analyzed 20 GB of data from 151 applications, and were able to classify network flows to a specific app with F1-score of 70.1% using a Linear SVM [5]. To put this result in context, Meddle [13] reports a 64.1% precision score in classifying flows for the 92 most popular Android applications by using the Host and User-Agent payload features.

Enhanced Privacy Control. When designing AntMonitor, we made an explicit decision to decouple the routing and logging functionalities: they are provided by two separate servers. This separation is to provide transparency, fine-grained data collection, and enhanced privacy control. Our design choices for privacy are as follows: First, AntServer only routes traffic and must not log any traffic. This is in line with privacy protection provided by some of the most popular VPN services [16]. Second, the LogServer, which logs and analyzes traffic, must only have access to the information explicitly allowed by the user. In other words, the user must be able to choose which applications to log, as shown in Fig. 2(c).

Privacy Leakage Prevention. AntClient allows users to configure which private information they want to prevent from leaking, as shown in Fig. 2(d). The information can be of two types: (i) sensitive information that is readily available to applications on the phone, such as, IMEI, email, and phone number, or (ii) custom strings that the user wants to protect. Examples of custom strings include a user's home address, ethnicity, gender, age, *etc.* This type of information is typically not stored on the phone; however, a user may input and send them to a friend in a previous communication, and the user wants to make sure that no other apps can sniff (*e.g.*, keyboard apps) and send this information elsewhere.

If the user selects one or more strings to protect (as shown in Fig. 2(d)), then AntClient inspects every outgoing packet for any of the protected strings, before sending it out. The search is currently implemented with the widely used Aho-Corasick algorithm [17]. If a string is found within the packet, AntClient notifies the user, as shown in Fig.2(e). The user is then able to either allow the packet to continue on its way, replace the sensitive string, or block it. As deep packet inspection is costly, we have implemented this part of AntClient in native C so as not to significantly impact CPU usage and battery life. It has been shown that the Aho-Corasick algorithm is able to reach gigabits per second throughput [18], which is suf-



☑ Log System Apps

☑ Chrome
☑ Clock
□ Docs
☑ Downloads
□ Drive
☑ Earth
☑ Email

Email: [redacted]@yahoo.com ☑
Email: [redacted]@gmail.com □
Phone number: 1949287[redacted] ☑
Device ID: 393a582[redacted] ☑
IMEI: 3525840[redacted] ☑

Custom Strings: Enter string here...

Gjoka □ ✕
Irvine ☑ ✕
Shuba ☑ ✕

1:37 PM
Sunday, June 14

Alert! QR Droid is trying to send your Device ID 1:37 PM
✓ OK # HASH ✕ DENY

Anteater VPN status 1:36 PM
Connected

App Name	Action	# of Allowed Leaks	# of Blocked Leaks
Chrome	✓	101	0
Facebook	✕	0	19
Messenger	✕	0	6
System	✓	5	0
WhatsApp	✕	0	6
WuKong	✓	6	0
com.google.process.gapps	✕	0	9
com.spotify.music.service	✕	0	42

(a) Icon (b) Home Screen (c) Selecting Apps for Data Collection (d) Selecting Strings for Inspection (e) Privacy Leak Notification (f) History of Leaks

Figure 2: Screenshots of AntClient. A video demo can be found on the project website [15].

ficient for mobile devices whose wireless networks typically reach several megabits per second.

Future Work. Currently, AntClient can examine just unencrypted traffic and match only simple strings. We are working on extending inspection to encrypted traffic by leveraging the SSL Bumping technique [19]. Furthermore, we plan to add regular expression matching to the inspection by using Deterministic finite automata (DFA) or its extensions [20]. Regular expressions will allow us to potentially detect malware and dynamic sensitive information, such as user location. In the latter case, we can also explore HTTP requests to find tokens that signify a privacy leak. For instance, in our pilot deployment we encountered several packets that passed the user location as a key-value pair in the URL string, with "location" being the key and coordinates being the value.

4. CONCLUSION

In this work, we present AntMonitor – a system for collecting large-scale, yet fine-grained network measurements from mobile devices, and for detecting and preventing leakage of private information in real time. Our pilot deployment of AntMonitor shows that it can greatly assist research activities, such as network measurements and traffic classification. Furthermore, AntMonitor can prevent apps from leaking sensitive information over unencrypted traffic. Our core contribution lies in making our system accessible to the majority of Android users and in providing users with various privacy protection options that serve as an enticement for using AntMonitor.

5. REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014-2019. http://goo.gl/Zu8f2r.

[2] W. Enck, P. Gilbert, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 2014.

[3] Vaibhav Rastogi, Yan Chen, and William Enck. Appsplayground: automatic security analysis of smartphone applications. In *Proc. of the 3rd ACM conference on Data and application security and privacy (CODASPY)*, 2013.

[4] Android Versions. developer.android.com/about/dashboards.

[5] Anh Le, Janus Varmarken, Simon Langhoff, Anastasia Shuba, Minas Gjoka, and Athina Markopoulou. AntMonitor: A System for Monitoring from Mobile Devices. In *(to appear) Proc. of ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big Data*, 2015.

[6] PCAPNG File Format. http://goo.gl/y89d9U.

[7] Erin Kenneally and David Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[8] J. Sommers and P. Barford. Cell vs. WiFi: On the Performance of Metro Area Mobile Connections. In *Proc. of IMC*, 2012.

[9] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying Diverse Usage Behaviors of Smartphone Apps. In *Proc. of IMC*, 2011.

[10] PhoneLab, University at Buffalo. https://www.phone-lab.org/.

[11] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A First Look at Traffic on Smartphones. In *IMC*, 2010.

[12] Clint Gibler, Jonathan Crussell, Jeremy Erickson, and Hao Chen. Androidleaks: Automatically detecting potential privacy leaks in android applications on a large scale. In *Proc. of the International Conf. on Trust and Trustworthy Computing*, 2012.

[13] A. Rao, A. M. Kakhki, A. Razaghpanah, A. Tang, S. Wang, J. Sherry, P. Gill, A. Krishnamurthy, A. Legout, A. Mislove, and D. Choffnes. Using the Middle to Meddle with Mobile. Technical report, Northeastern University, Dec. 2013.

[14] Anastasia Shuba, Anh Le, Minas Gjoka, Janus Varmarken, Simon Langhoff, and Athina Markopoulou. Demo: AntMonitor - A System for Mobile Traffic Monitoring and Real-Time Prevention of Privacy Leaks. In *(to appear) Proc. of the 21st International Conference on Mobile Computing and Networking*, 2015.

[15] AntMonitor: Project Webpage and Demo. http://antmonitor.calit2.uci.edu/.

[16] Private Internet Access Privacy Policy. http://goo.gl/Yt8jNx.

[17] Multifast. http://multifast.sourceforge.net/.

[18] Nathan Tuck, Timothy Sherwood, Brad Calder, and George Varghese. Deterministic memory-efficient string matching algorithms for intrusion detection. In *Proc. of INFOCOM*, 2004.

[19] Squid Proxy. Squid-in-the-middle SSL Bump.

[20] Sailesh Kumar, Sarang Dharmapurikar, Fang Yu, Patrick Crowley, and Jonathan Turner. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. *ACM SIGCOMM Computer Communication Review*, 36(4):339–350, 2006.