Demo: AntWall - A System for Mobile Adblocking and Privacy Exposure Prevention*

Anastasia Shuba
UC Irvine
ashuba@uci.edu

Athina Markopoulou
UC Irvine
athina@uci.edu

ABSTRACT

Mobile devices have become an essential part of our every-day lives but are also suffering from various privacy and security risks. The ease of app development has led to a plethora of apps that employ poor security practices and often expose personal identifiers to remote servers. Moreover, these privacy exposures often come from third-party libraries that are leveraged by multiple apps, leading to cross-app tracking of users. This tracking is typically used to serve personalized ads, which cost the user extra data and take up screen real estate. In this demo, we will showcase AntWall, a system for preventing exposures of personal information and blocking ads.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; • **Networks** → **Network monitoring**;

KEYWORDS

Privacy, Ad blocking, Network Monitoring

ACM Reference Format:
Anastasia Shuba and Athina Markopoulou. 2018. Demo: AntWall - A System for Mobile Adblocking and Privacy Exposure Prevention. In *Mobihoc '18: The Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3209582.3225207

1 SYSTEM OVERVIEW

System Architecture. In order to prevent mobile apps from fetching ads and/or sharing personally identifiable information (PII) over the network, one needs visibility into the network traffic generated by all apps. However, this is typically impossible without modifying the Android OS or rooting the phone, due to the way Android separates different apps in user-space. To address this challenge, we choose to operate at a different layer: we inspect all network traffic in and out of the device by leveraging the VPN APIs that have been made available since Android 4.0+, and are now also available on iOS. This allows us to block ads and prevent privacy exposures completely from user space. We use the AntMonitor Library [2], which runs a VPN service on the device but without re-routing traffic to a remote VPN server. The library exposes the necessary APIs to intercept traffic and has minimal impact on device resources. We refer the reader to [2] for details on the library. The AntWall architecture is shown in Fig. 1(a): AntWall receives outgoing IP packets via the PacketFilter interface provided by the AntMonitor Library and determines whether or not these packets contain PII and/or a request for an ad.

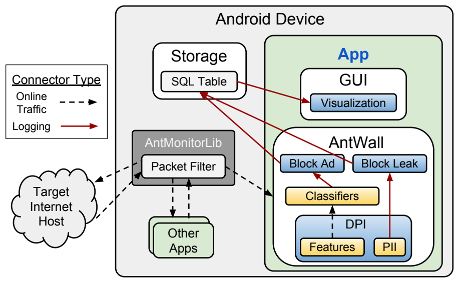
Detecting PII. In this work, we assume that the system knows the set of PII in advance. This is a reasonable assumption as most PII (such as email address and phone number) are available to any Android app through APIs, and AntWall provides a GUI for users to enter more PII of interest (Fig. 2(a)). For each outgoing packet, AntWall uses the Aho-Corasick algorithm (also provided by the AntMonitor Library) to find PII in one pass of the packet. Whenever a PII is detected, the user is notified. From there, the user can decide to allow the PII exposure to happen, replace the exposed PII with a random string of the same length (so as not to alter the payload size), or block the packet completely (Fig. 2(b)). Whichever action the user picks is remembered for future occurrences of the same PII and offending app combination, and can be changed at any time by the user in AntWall's settings (Fig. 2(c)). AntWall also provides a visual graph of connections made by other apps in real-time (Fig. 2(d)).

Blocking Ads. To block ads (Fig. 2(e)), AntWall blocks any outgoing packet containing a request for an ad. Since there are no ad blocking lists that have been tailored to mobile devices [1], we had to manually create additional rules to supplement the most popular ad blocking list - EasyList. We then used our modified EasyList to label outgoing packets from 50 most popular Android apps that display ads. The data collection was done using the AntMonitor Library, and the packets were saved in PCAPNG format, which is especially useful as it allows us to annotate each packet with a comment. In our case, we use the comment to label the packet with the following information: which app is responsible for generating the packet, what PII it contains, and whether or not it contains a request for an ad. We use these labels, along with the packet data, to train a C4.5 Decision Tree (DT) classifier that predicts whether or not a given packet is requesting an ad. For more details about feature extraction, feature selection, and training, we refer the reader to our full paper [3]. Our classifier is able to achieve F1 scores of over 95%, even when tested on applications that were not part of the training set.

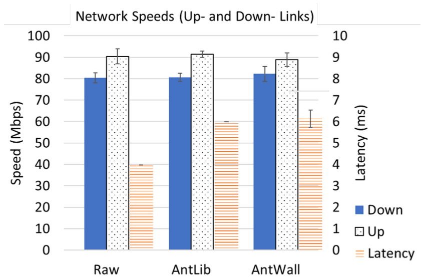
Performance Evaluation. To evaluate the performance of AntWall we ran the *Ookla Speedtest* and sampled the overall system CPU and memory usage using the top command. We evaluted 3 scenarios: (i) raw device performance, (ii) the device with the AntMonitor Library intercepting packets, and (iii) the full AntWall system, where packets were intercepted and inspected for PII and ad requests. Each test case was repeated 5 times on the Pixel device, running Android 8.1, and containing only basic system apps, the *Speedtest* app, and various

*This work has been supported by NSF Award 1649372, a DTL Grant 2016, and CPCC at UCI. A. Shuba has been partially supported by an ARCS Fellowship.

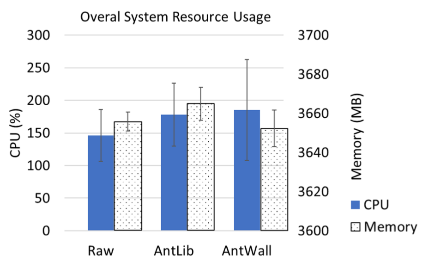
Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3225207



(a) System architecture

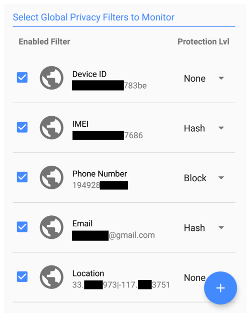


(b) Up- and downlink speeds, and latency (ping) as reported by *Speedtest*



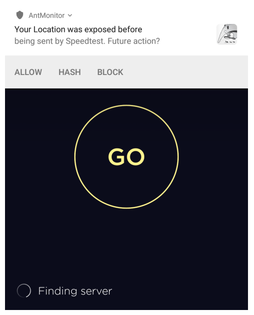
(c) Overal system CPU (out of 400% on a quad-core) and memory usage as reported by the *top* command

Figure 1: System overview: architecture and performance


Select Global Privacy Filters to Monitor

Enabled Filter	Protection Lvl
Device ID [redacted]783be	None
IMEI [redacted]7686	Hash
Phone Number 194928[redacted]	Block
Email [redacted]@gmail.com	Hash
Location 33.[redacted]973|-117.[redacted]3751	None

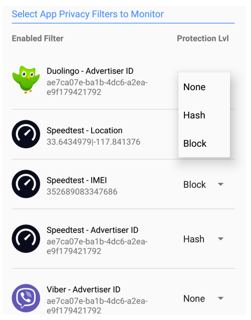
(a) Partial list of PII to protect


AntMonitor

Your Location was exposed before being sent by Speedtest. Future action?

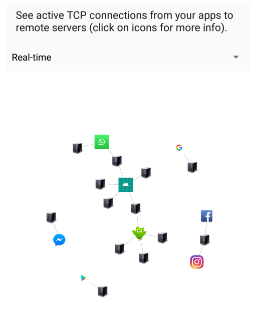
ALLOW HASH BLOCK

(b) A real-time notification of an exposure

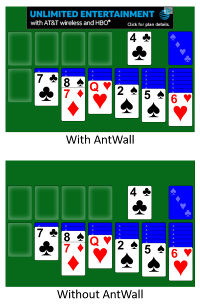

Select App Privacy Filters to Monitor

Enabled Filter	Protection Lvl
Duolingo - Advertiser ID ae7ca07e-ba1b-4dc6-a2ea-e9f179421792	None / Hash / Block
Speedtest - Location 33.6434979|-117.841376	
Speedtest - IMEI 352689083347686	Block
Speedtest - Advertiser ID ae7ca07e-ba1b-4dc6-a2ea-e9f179421792	Hash
Viber - Advertiser ID ae7ca07e-ba1b-4dc6-a2ea-e9f179421792	None

(c) Adjusting settings for PII/app combos of past leaks


See active TCP connections from your apps to remote servers (click on icons for more info).

(d) Real-Time Visualization: which apps connect to which servers



(e) Ad blocking in action

Figure 2: Screenshots of AntWall in action: (a)-(d) blocking PII leaks and (e) blocking ads, both in real-time.

versions of the AntWall app. The experiments were performed in our lab during late night hours to minimize network interference. As we show in Fig. 1(b), both the AntMonitor Library and AntWall have negligible impact on the network throughput. However, intercepting packets adds ~2ms to latency, as indicated by *Speedtest* ping results. We believe this to be of minimal impact on user experience. Moreover, as shown in Fig. 1(c), most of the resource consumption comes from the AntMonitor Library itself and the impact is minimal.

2 DEMONSTRATION PLAN

Description. Our demo will show AntWall's capability to detect exposures of PII and requests for ads across multiple apps without significantly impacting CPU usage and network throughput. First, we will use several apps with AntWall disabled. For example, we will read an article inside *Chrome*, play *Solitaire*, and run *Speedtest*. All of these apps will display ads and *Speedtest* will also display the network speed of the raw device. Next, we will turn on AntWall and repeat the same activities. This time, no ads will be shown, and we will also see (through *Speedtest*) that AntWall does not significantly impact the network throughput. Furthermore, AntWall will allow us to block/replace/allow any privacy exposures made by apps such as *Speedtest*. We will also showcase a test app and test server that we made to prove that AntWall indeed can replace and block PII. Finally, we will navigate to the visualization screen of AntWall (as in Fig. 2(d)) and we will see how many servers are being contacted in the background as a result of only a few minutes of interaction with the phone. During the demo we will have CPU overlay enabled from Android Developer Options for real-time CPU usage statistics.

Video. A preliminary video of the demo is available at http://athinagroup.eng.uci.edu/antwall-mobihoc-2018/.

Setup Requirements are minimal. Our demo requires one Android phone (or several phones to demo to multiple people at once), which we will provide on our own. We need Internet connectivity for the phone, preferably Wi-Fi. A poster stand and a TV screen would be preferred, but not required, for our demo. A small table with a close-by power outlet will suffice (we will only use power for the TV screen, if provided). The setup will only take several minutes.

3 CONCLUSION

In this demo we presented AntWall – a system for blocking exposures of personal information and requests for ads. AntWall runs in the background and does not require rooting the device. It has minimal impact on network speeds and low resource usage. Although AntWall currently relies on manually labeled datasets to build classifiers for ad blocking, in the future we hope to automate this process by changing the Android OS to track whether the network requests come from the app itself or from a third-party library.

REFERENCES

[1] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. Weippl. 2017. Block me if you can: A large-scale study of tracker-blocking tools. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*. IEEE, 319–333.
[2] A. Shuba, A. Le, E. Alimpertis, M. Gjoka, and A. Markopoulou. 2016. AntMonitor: System and Applications. *arXiv preprint arXiv:1611.04268* (2016).
[3] A. Shuba, Z. Shafiq, and A. Markopoulou. 2018. NoMoAds: Effective and Efficient Cross-App Mobile Ad-Blocking. *Proceedings on Privacy Enhancing Technologies* (2018). To Appear.