Meddle: Practical Mobile Diagnosis Through Traffic Indirection

First
First Institution

Second
Second Institution

Third
Third Institution

Fourth
Fourth Institution

Fifth
Fifth Institution

Sixth
Sixth Institution

Abstract

We present, Meddle, a platform that relies on traffic indirection to diagnose mobile Internet traffic. Meddle is motivated by the absence of built-in support from ISPs and mobile OSes to freely monitor and control mobile Internet traffic; the restrictions imposed by mobile OSes and ISPs also make existing approaches impractical. Meddle overcomes these hurdles by relying on the native support for traffic indirection by mobile OSes. Specifically, Meddle proxies mobile Internet traffic through a software defined middleboxes configured for mobile traffic diagnosis. In this paper, we use Meddle to tests the limits of the network perspective of mobile Internet traffic offered by traffic indirection.

We use this perspective to characterize and control the behavior of mobile applications and provide a first look at ISP interference on mobile Internet traffic. [**TBD: We report on controlled experiments we conducted to analyze the network behavior of 100 most popular iOS and Android applications.**] [**TBD: We also report on how ISP interfere with mobile HTTP Internet traffic.**]

1 Introduction

Mobile systems consist of walled gardens inside gated communities, i.e., locked-down operating systems running on devices that interact over a closed and opaque mobile network. Despite a large collection of privacy, policy and performance issues in mobile networks [3, 6, **?**, 7], researchers are faced with few options to characterize and address them.

The key challenge is that mobile devices, their OSes, and ISPs provide no built-in service to monitor and control *all network traffic*. As a result, previous studies [**?**, 5, 1, 3, 12, **?**] are constrained by at least one of the following: mobile OSes, access technology, device manufacturer, installed applications, and user behavior. In this work, we are the first to present an approach that compromises none of these, an approach that can be used across carriers, mobile devices, apps, and access technologies.

Meddle redirects all Internet traffic through a software-defined middlebox for the purpose of analysis and interposition. Specifically, *Meddle* builds on the native support for VPN tunnels by mobile OSes to tunnel all the Internet traffic regardless of the access technology used by the device.

[**TBD: tripwire stuff**]

In this paper, we use *Meddle* to test the limits to which mobile Internet traffic can be diagnosed using traffic redirection. The main contributions of this paper are as follows:

1. Platform for practical mobile diagnosis. Single server solution empowers users to install and configure them on home-gateways. Researchers can deploy them for measurement studies.
2. Controlled experiments using off-the-shelf Android and iOS devices.
3. Controlled experiments to analyze ISP interference in US and France.

The remainder of this paper is structured as follows.

2 Motivation and Goals

Meddle is motivated by the impractical nature of existing solutions that can diagnose the behavior of mobile devices and what ISPs do with the traffic naturally generated by these devices. Existing solutions rely on a combination of OS modifications [3, 6, 7, 8, 13], static and dynamic analysis of mobile applications[2, 9], analysis of OS service logs[4], and traces collected by ISPs or at gateways [11]. [**TBD: add more references from mobisys and imc paper**]. These solutions are impractical because they either violate the terms and conditions of device usage, they are heavily dependent on the specific version of the OS and application, they cannot scale to support new OSes, or they are limited to a specific geographic region. This impractical nature thus creates a high-barrier to deployment with end-users.

Indeed, there exists a trade-off between a practical user-friendly solution and a solution that offers a fine-grained control over mobile devices. Unlike existing approaches, we accept the bait of coming up with a practical and user-friendly solution and testing the limits of its usefulness. Specifically, we relinquish OS-level controls to focus on the Internet traffic generated by mobile devices and try to use this perspective to diagnose mobile devices and the ISPs that serve these devices.

The network perspective is promising because mobile devices are increasingly becoming the primary gateway to access Internet based services. This vision is backed by the recent International Telecommunication Union (ITU) report: *"in developing countries, mobile-broadband services cost considerably less than fixed-broadband services"* [10]. The increasing Internet usage from mobile devices makes the traffic generated by these devices comparable to the traffic generated by PCs [4]. This perspective becomes even more important because a large number of free applications use Internet based advertisements to make-up for their costs [7, 11].

Intuitively, the practical way to obtain this network perspective is to redirect the mobile Internet traffic through a middlebox configured for traffic diagnosis. By offloading traffic diagnosis to a middlebox, the offered solution has the potential to be agnostic to the OS, device manufacturer, application installed, and access technology used by the device.

3 Goals

[**TBD: For each goal, What does it mean followed by why is it important?**]

1. *Deployable.* Easy to install/use/configure. It must not require warrant voiding of the device. This is important to support a large user-base.
2. *On-demand.* The user must be able to enable and disable service easily. It must be easy to fallback to the original state. This is important to ensure that users can easily opt-out and are not blocked if there are some problems with the system.
3. *Agnostic to OS, ISP, access technology, and applications..* The research work that comes out must not be limited to specific OS, ISP, or device manufacturer, or set of applications.
4. *Scalable.* Can support many users. To ensure that research results have statistical significance.
5. *Always On.* Capable of monitoring and controlling all the Internet traffic once the user enables the system. This is important to ensure passive and pervasive operations that do not demand periodic inputs from end-users.
6. Flexible - supports many services
7. Reliable - always works
8. Safe - does not adversely affect traffic (interaction between modules)
9. Secure - handles user data with care

Figure 1: Figure showing traffic redirection and flow of traffic through VPN and HTTP proxy

4 Platform Description

We take an approach of traffic redirection. Proxying internet traffic meets above goals. Issues with network perspective of traffic

4.1 Architecture

Architecture as shown in Figure 1.

Explain two Proxies, one VPN and other HTTP.

Explain how each goal is met and why we need two proxies.

The role of the VPNs Proxy. VPNs allow tunnel everything (all IPv4 traffic). All IPv4 traffic can be monitored and controlled on the middlebox.

The role of HTTP Proxy. Check ISP interference. What How tripwires are implemented.

Configuration requirements. All iOS devices (version 3.0 and above) support *VPN On-Demand*, which forces traffic for a specified set of domains to use VPN tunnels. To ensure all possible destinations match this list, we exploit the fact that iOS uses suffix matching to determine which connections should be tunneled; accordingly, we specified the domain list as the set of alphanumeric characters (a-z, 0-9, one character per domain). Android version 4.2 and above supports an *Always On VPN* connection that provides the same functionality; for Android version 4.0 and above there is an app API that allows apps to manage VPN tunnels. We support both options.

4.2 Existing Deployment

Where is this currently deployed?

What is the objective of this deployment?

Who are the ones signed up?

What was the incentive given?

4.3 Overheads

1. **Increase in Network Latency.**

[**AR:TBD:**] x axis contains Android iOS — Android iOS — Android iOS — Android iOS
Location 1 — Location 2 — Location 1 — Location 2
Wi-fi — Cellular With error bars for min and max.

Figure 2: Latency Overheads. Tests performed by Adrian and Sam

Figure 2. Cite results from latency to home gateways based on DSL results in PAM and IMC.

2. **Increase in Power Consumption.** Results from battery tests performed by Dave for Android battery monitoring tests performed by Ashwin for Android and iPhone.

3. **Increase in Traffic Volume.**

4.4 Limitations

1. **At most one tunnel.** [**TBD: Not true for http proxy**] Currently iOS and Android support exactly one VPN connection at a time. This allows *Meddle* to measure traffic over either WiFi or cellular interfaces, but not both at once. The vast majority of traffic uses only one of these interfaces, and that interface uses the VPN

2. **Proxy location.** When traffic traverses the *Meddle* box, destinations will see the *Meddle* box address, not the device IP, as the source. This might impact services that customize (or block access to) content according to IP address (e.g., in case of localization). A solution to this problem is to use a *Meddle* instance with an appropriate IP address

3. **ISP support.** Some ISPs block VPN traffic, which prevents access to our current *Meddle* implementation. We note that few ISPs block VPN traffic, and there is an incentive not to block VPN traffic to support enterprise clients.

4. **IPv6.** *Meddle* cannot be currently used on networks using IPv6 because IPv6 is not fully supported by mobile devices. Indeed, we observe that though iOS and Android support IPv6 they currently do not support IPv6 traffic through VPN tunnels

4.5 Costs

1. **Deployment and Running Costs**

2. **Trust Provider**

4.6 Incentive for End-user Deployment

1. **Deploy on Home Gateway.** This is why we need the single machine constraint.

2. **Packet Filtering.** Custom ad blocks. Protect against data leaks.

3. **Security from untrusted Wi-Fi APs.**

4. **Modular Architecture for Offloading Activities.**

5 Evaluation

6 Conclusion

References

[1] Chen, X., Jin, R., Suh, K., Wang, B., and Wei, W. Network Performance of Smart Mobile Handhelds in a University Campus WiFi Network. In *Proc. of the Internet Measurement Conference (IMC)* (2012), pp. 315–328.

[2] Egele, M., Kruegel, C., Kirda, E., and Vigna, G. PiOS: Detecting Privacy Leaks in iOS Applications. In *Proceedings of the Network and Distributed System Security Symposium* (2011).

[3] Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., and Sheth, A. N. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *Proc. of the USENIX Operating Systems Design and Implementation (OSDI)* (2010), pp. 1–6.

[4] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., and Estrin, D. Diversity in Smartphone Usage. In *Proceedings of the International conference on Mobile systems, applications, and services (Mobisys)* (2010), pp. 179–194.

[5] Gerber, A., Pang, J., Spatscheck, O., and Venkataraman, S. Speed Testing without Speed Tests: Estimating Achievable Download Speed from Passive Measurements. In *Proc. of the Internet Measurement Conference (IMC)* (2010), pp. 424–430.

[6] Hornyack, P., Han, S., Jung, J., Schechter, S., and Wetherall, D. "These Arent the Droids Youre Looking For": Retrofitting Android to Protect Data from Imperious Applications. In *Proc. of CCS* (2011), pp. 639–652.

[7] Pathak, A., Hu, Y. C., and Zhang, M. Where is the energy spent inside my app? Fine Grained Energy Accounting on Smartphones with Eprof. In *Proc. of Eurosys* (2012).

[8] Qian, F., Wang, Z., Gerber, A., Mao, Z., Sen, S., and Spatscheck, O. Profiling Resource Usage for Mobile Applications: A Cross-layer Approach. In *Proc. of MobiSys* (2011).

[9] Ravindranath, L., Padhye, J., Agarwal, S., Mahajan, R., Obermiller, I., and Shayandeh, S. AppInsight: Mobile App Performance Monitoring in the Wild. *Proc. of the USENIX Operating Systems Design and Implementation (OSDI)* (2012).

[10] Sanou, B. ICT Facts and Figures. Tech. rep., International Telecommunications Union, 2013.

[11] Vallina-Rodriguez, N., Shah, J., Finamore, A., Grunenberger, Y., Papagiannaki, K., Haddadi, H., and Crowcroft, J. Breaking for Commercials: Characterizing Mobile Advertising. In *Proc. of the Internet Measurement Conference (IMC)* (2012), pp. 343–356.

[12] WANG, Z., QIAN, Z., XU, Q., MAO, Z., AND ZHANG, M. An Untold Story of Middleboxes in Cellular Networks. In *Proc. of the ACM SIGCOMM Conference* (2011), pp. 374–385.

[13] WEI, X., GOMEZ, L., NEAMTIU, I., AND FALOUTSOS, M. ProfileDroid: Multi-layer Profiling of Android Applications. In *Proc. of MobiCom* (2012).