COS 597D Project - Mobile vs Traditional Web Tracking (FourthPartyMobile)

Marcela Melara, Christian Eubank and Diego Perez Botero
{melara, cge, diegop}@princeton.edu

ABSTRACT
INSERT ABSTRACT HERE??

Categories and Subject Descriptors
H.3.5 [**Information Systems**]: Information Storage and Retrieval—*Web-based services*; K.4.1 [**Computing Milieux**]: Computers and Society—*Privacy*

General Terms
Documentation, Measurement, Security

Keywords
FourthParty, Web crawling, cookies, privacy policy, ...

1. INTRODUCTION
We wish to automate the detection of third-party tracking mechanisms while browsing the web on a mobile device. To this end, we will adopt the FourthParty[1] project's approach and instrument a popular open-source mobile browser (i.e. Firefox) to be used as an enhanced web crawler. This enables us to log realistic end-user interactions (e.g. execution of embedded scripts) as opposed to just downloading each web page's static content, which is what traditional web crawlers do.

The mobile web crawler is not our main objective for this project, but rather the tool that we will use to collect valuable information in order to conduct our comparison between the Mobile and Traditional third-party tracking ecosystems and their practices.

2. BACKGROUND AND MOTIVATION

3. RELATED WORK
MARCELA

4. IMPLEMENTATION
DIEGO

4.1 Challenges
Mobile application development poses a variety of challenges that will need to be addressed for a mobile web crawler to be materialized:

- Mobile devices have limited amounts of RAM, so applications should not rely on large data structures stored in main memory.

- Security permissions in mobile devices are strict, which means that writing data into persistent memory is not always an option.

- Processing power in mobile devices is limited, so computationally intensive procedures, such as parsing a web page, should be delegated to an external entity.

- Mobile network bandwidth is a limited resource, so large data transfers should be avoided.

- Battery life must be preserved as much as possible by a mobile application if it is being aimed towards the general public.

4.2 Mobile Web Crawler's Architecture
FourthPartyMobile's architecture (see Figure 1) delegates most of the computation and storage to a supporting server, limiting the mobile deviceâĂŹs responsibilities to fetching one website at a time and generating a log of its latest interactions (e.g. cookies, javascript, embedded HTTP objects). The crawling plugin running on the mobile device sends the interaction log corresponding to the website being visited in the form of SQL statements to the crawling backend running on a server. This way, the amount of state kept in the mobile deviceâĂŹs main memory is minimal and the crawl database, which can be several Megabytes in size, is generated by the supporting server's side.

4.3 Prototype
We took advantage of the fact that the FourthParty[2] project is open-source. After analyzing its codebase, we ported its core functionality over to support Android-based mobile devices, such as smartphones and tablets. FourthPartyMobile is implemented in Java and Javascript, leveraging both the Android SDK and the Mozilla Add-On SDK. Persistent storage is fully compliant with FourthParty's SQLite database schema. Thus, we provide a standardized representation for traditional and mobile crawls, which facilitates data analysis. Our Crawling Backend is written in java with a SQLite JDBC library that supports Mac OS, Linux and Windows, so it should be fully multi-platform. It also supports concurrency, so multiple crawls can be recorded simultaneously.
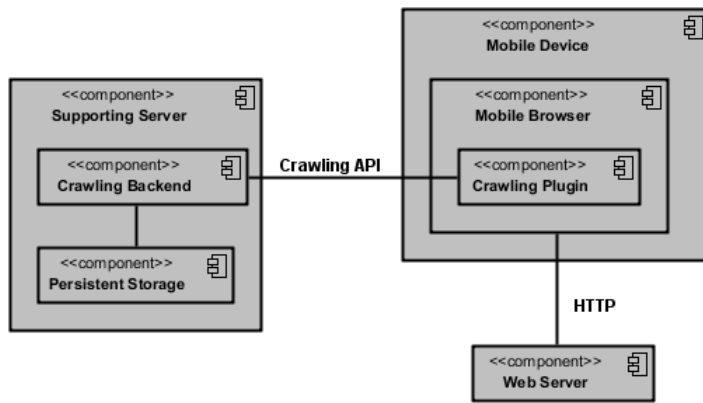
---

[1] http://www.fourthparty.info

[2] http://www.fourthparty.info

Figure 1: Prototype's Runtime Interactions.

5. METHODOLOGY
DIEGO

6. DATA ANALYSIS
6.1 Main Players
CHRIS

Three types of players: advertisers, content providers, and third-party content providers (embedded in sites)

6.2 Cookie/Javascript Pervasiveness
CHRIS

By website category (e.g. porn, news, etc) and by domain (e.g. com, net, etc)

6.3 Desktop vs Mobile Tracking
ALL OF US

6.4 Physical vs Emulated Devices
DIEGO

6.5 Privacy Policy Case Study
After running our web crawls, we designed a case study in which we look at the privacy policies and related data collected from our crawls of three categories of websites that we found were amongst the most popular sites today: social networks, news sites, and e-commerce sites. While pornography sites are also amongst the most popular sites online, we do not include this fourth category in our case study for reasons of decency.

Based on the list of the Alexa Top 100 US visited websites, we chose three websites amongst the top 25 most visited sites, one representing each of our three categories. Our case study examines the privacy policies of LinkedIn[3] (social network), CNN[4] (news) and Amazon[5] (e-commerce).

The case study has five stages, each stage looking at the privacy policies in more detail:

1. Compare the contents of the privacy policies displayed when visiting each of the sites on our three platforms (desktop, tablet, and smartphone).

2. Compare the length of the privacy policies of each website.

3. Examine the topics covered, i.e., the sections included in the policies.

4. Inspect some of the language used in select sections of the policies.

5. Compare the presented cookie policies with the collected web crawl data.

7. CONCLUSIONS AND FUTURE WORK
8. REFERENCES

---

[3]http://www.linkedin.com
[4]http://www.cnn.com
[5]http://www.amazon.com