Privacy Revelations for Web and Mobile Apps

D. Wetherall*, D. Choffnes*, B. Greenstein†, S. Han*, P. Hornyack*, J. Jung†, S. Schechter‡, and X. Wang*
University of Washington, Intel Labs † and Microsoft Research‡*

Abstract

Users of Web and mobile apps must often decide whether to give the apps access to personal information without knowing what they will do with it. We argue that users could better manage their privacy and privacy standards would rise if the operating system simply revealed to users how their apps spread personal information. However, for this strategy to be effective, the research community must go well beyond today's low-level monitoring techniques to develop predictive, user-facing descriptions of information exposure that are grounded in measurement and analysis.

1 Introduction

Popular interest in privacy issues on the Web and mobile phones has grown along with news stories that make it clear that existing permissions and controls do not prevent personal information from being spread widely. The research community has responded in two ways. It has uncovered further exploits, especially clever ways of leaking private information via weak identifiers and side channels, e.g., information flows in Javascript [9] and leakage via search engines [10]. And it has come up with new designs that better restrict information exposure, e.g., privacy-preserving online advertising [8] and controls that fake inputs [1].

Our goal is to help people to manage better how their personal information is spread as they use networked applications, which is one aspect of privacy. To this end, our view is that there is a more basic approach than finding new exploits and designing new controls. It is simply to measure how personal information is spread as networked applications are used and to reveal these descriptions to and share them among users. We argue here that this approach is broadly applicable, valuable, and has been mostly overlooked by the research community.

We call the descriptions of how personal information is spread over the network *privacy revelations* (Section 2). An example privacy revelation is that a mobile app regularly sends location and identity information to advertisers. This information tells users the privacy that they will receive in practice if they use the application. As the example suggests, our emphasis is on high-level descriptions of the spread of information rather than low-level metrics that bound exposure in bits of entropy, or in terms of k-anonymity [12] or ϵ-differential privacy [3].

We see privacy revelations as desirable in their own right to help users manage their privacy and improve privacy standards over time. They are not merely convenient data to inform work on new privacy controls. To make this case, we begin with the lack of user awareness of actual information exposure (Section 3). We then explain how privacy revelations can have a positive impact by informing usage decisions and tackling the disincentives that make it difficult for users to express their privacy tradeoffs (Section 4).

The key research challenge we identify is how to generate user-relevant descriptions of information exposure automatically. There is a large gap between traditional low-level measurements (from tools such as information flow) and privacy revelations that convey actionable information to the average user. This gap cannot be papered over by a user interface veneer. Instead, operating systems need to do a better job of treating user-facing concepts as first-class abstractions that can be programmatically analyzed and measured.

To support our position, we draw on two privacy revelation systems that we are developing, though our arguments extend to other types of networked applications. The first system is a Web browser extension that analyzes pages and HTTP traffic and crowdsources measurements. It warns users of logins that will be done in the clear, and provides a login report showing sites and risks. The second system is based on the TaintDroid engine for Android [6]. It tells users how their smartphone apps expose sensor, device and personal information (e.g., GPS location, IMEI, and contacts) over the network based on crowdsourced information flow measurements.

2 Privacy Revelations

We propose the use of privacy revelations for operating systems in which application code has access to both personal information (that the user may not want leaked) and the network (a means with which to expose data). These systems include Web browsers, which support JavaScript-based apps, mobile phones, tablets, and PCs. Information that users may not want leaked may include their name, address, location, login credentials, contacts, emails, photos, and other files.

Since the apps are untrusted, the operating system (i.e., Web browser or mobile phone runtime) must enforce any privacy monitoring or controls. We consider only the spread of personal information off of the user's platform. It would be interesting to include the further spread of information between back-end servers. However, we leave this more general formulation to future work since we believe that much can be learned in the more actionable setting of user platforms.

Privacy revelations are automated descriptions of the spread of personal information based on measurement and analysis made by the operating system. To be useful, we require that they be: (i) based on user-relevant concepts and context; (ii) actionable, in the sense of giving feedback before information exposure; and (iii) shareable with other users, minus personal information.

Context and high-level descriptions help users to understand risk. For example, users are likely to be more concerned about apps that spread their personal information even when they are not logged in or actively using them. The context we consider for our systems is whether: information is exposed to the main site or to third parties such as advertisers; the user is logged in to a site or actively running the app; the exposure is a user-initiated action; encryption is used to prevent exposure to eavesdroppers; and the kind of personal information that is exposed. *None* of this context is captured by currently available operating system measurements.

Actionable revelations predict exposure ahead of time and occur in situ during usage, when they may inform individual usage decisions. In contrast, other efforts that have measured information exposure present after-the-fact studies [5, 9, 13]. We also consider pausing midway through a sequence to let the operating system obtain user permission (e.g., for opening the firewall) to fall short of a useful prediction of exposure.

Finally, the ability to share revelations lets users benefit from each other's experiences. It provides a basis for increasing the coverage of app testing by aggregating revelations, and for actionable revelations by using measurements from one user to tell another user what is likely to happen *before* an app is run.

3 Users Lack Privacy Awareness

Revealing how sites and mobile apps handle personal information is valuable when users do not have an adequate sense of how personal information is exposed. This is certainly the case today.

3.1 A Trail of Vulnerabilities

Seemingly each week brings new stories in the popular press of privacy vulnerabilities ranging from the dubious practices of large companies (e.g., Facebook privacy policies) to specific new attack tools that may access personal information (e.g., session hijacking with Firesheep). These stories suggest that typical users are not aware of many ways that their personal information may spread. A recent, prominent example for the Web is the FTC report on tracking [7]. It describes profiling that is now pervasive as a cause for consumer concern that industry has not adequately addressed. For mobile apps, the Wall Street Journal reported on a study of 100 apps for the iPhone and Android, finding that more than half of them sent phone identification (IMEI) without user awareness or consent, while others sent age, gender and more [13]. Lawsuits are now pending seeking redress for consumers.

3.2 Ineffective Privacy Policies

Motivated by credit reporting in the 1970s, the FTC codified generally-accepted principles for how information systems should process and protect personal information. A cornerstone of these principles applied to the Internet is "notice-and-choice." It requires that users be given notice of what information an app collects and why, and that they can choose whether to participate. This is the rationale for the privacy policies that have proliferated on the Web as the standard means for making users aware of how personal information is handled. However, these policies are widely considered to be a failure from a consumer point of view [11]. The FTC report speaks of "long, incomprehensible privacy policies that users typically do not read, let alone understand" [7].

3.3 Actual Exposure is Hidden

Even if they want to, users cannot reliably be aware of how their personal information is handled because its processing, storage and transmission is normally hidden from view. A privacy policy may be reassuring when read, but there is no guarantee that it is an accurate reflection of what happens in practice due to oversights, changes over time, and deception. Similarly, a mobile app may be configured to use encryption or expected to protect privacy, but there is no independent check for correctness.

Even independent certifications of acceptable behavior are no guarantee if they are not based on measurements of real operation. For example, TRUSTe administers a well-known privacy seal for websites. The seal attests that TRUSTe believes the site adheres to Fair Information Practices that include protection against unauthorized disclosure of personal data. However, since TRUSTe makes money by certifying sites, it has little incentive to strictly enforce and raise privacy standards. Indeed, there are incidents in which companies certified by TRUSTe have been found to violate privacy policies [4]. Our measurements of the top 100 US sites show that the presence of TRUSTe Web privacy seals does not mean personal data is encrypted when it is sent over the network; in many cases names and other identifying information are sent in the clear, often embedded in cookies.

For the iPhone, Apple vets apps on the App Store and uses agreements to prevent developers from unnecessarily passing information to third parties. However, the implicit endorsement of having an app available on the App Store does not prevent it from exposing personal information. The Wall Street Journal and other studies show that a significant number of apps do send phone identifiers, user location and other data to third parties without

user awareness or consent [13, 5].

4 Benefits of Privacy Revelations

Privacy revelations highlight issues, but a possible objection is that they will not proactively fix them because they do not enforce greater privacy. In this section we explain how revelations nonetheless help users manage their own privacy and raise standards over time.

4.1 Transparency Informs Decisions

Privacy concerns vary widely. Our intent is thus to let users make informed decisions that fit their own privacy concerns, rather than to mandate a one-size-fits-all kind of privacy for all users. In our work on WiFi privacy, we found showing users the information they expose to be a powerful way to help them to understand risks in practice [2]. Most users know at some level that WiFi suffers from security problems that may let other parties observe wireless traffic. However, this is an abstract threat. Telling users what information they are sending that can be read by other people (emails, passwords, sites visited) lets users reason in terms of concrete threats.

The transparency provided by privacy revelations directly supports informed usage decisions. After seeing revelations for one of their apps, one user may choose to proceed and give up a degree of privacy, e.g., to use a site or mobile app with a revenue model that is based on personalized ads. Other users may not, as there are often alternatives that provide roughly the same functionality but with other tradeoffs, including with respect to privacy. Both decisions are perfectly acceptable to us. What is not reasonable in our view is to force users who do value their privacy highly to use apps without an understanding of what personal information will be exposed. Similarly, it is not reasonable to preclude some app offerings, such as ones with personalized ads, to impose uniformly stronger privacy for all users.

4.2 Use with External Controls

While privacy revelations do not directly control information exposure, users can often exercise a useful degree of control outside of the app or operating system, if they have good reason. The ultimate control we have mentioned is that a user may decide not to use a site or mobile app. However, avoiding a site or app altogether is often impractical. Short of this, there are still measures that users can take. Consider the case of a site that sends passwords in the clear that a user wants to use regardless. The user can: ensure that the password is not shared with financial sites; prefer to use the site from a trusted network rather than an unsecured wireless network (e.g., coffee shops); and change the password occasionally. All these steps reduce risk. Since most users are lax about security, it is useful to know when there is good reason not to be lax.

For other kinds of personal information, users can exercise external control by providing false or one-time personal information as input to the site or app. Examples include fake telephone numbers and single-use credit card numbers. These measures are broadly applicable and already employed by users. Privacy revelations can help users to understand when they are most valuable, and it may even be the case that seeing how people use revelations will lead to natural candidates for new controls, e.g., automating fake inputs as a control [1]. In contrast, new controls that are devised to enforce greater privacy instead of provide transparency often have side-effects that interfere with widespread use, e.g., blocking third-party cookies is a control that can raise privacy but it breaks some sites.

4.3 Sharing Builds Reputations

Sharing privacy revelations across users tackles the security economics problem of guiding the market towards apps with the privacy controls that users want. At present, users learn of privacy considerations locally, by using apps and seeing what personal information they require, and by hearing the experiences of other local users. There is often little information about privacy and little incentive to report privacy behaviors for the benefit of others. Shared privacy revelations can be used to build reputation systems that help users to find alternative sites or apps that better meet their privacy tradeoffs. This guidance holds whether or not users decide to select or pay a premium for sites and apps that provide greater levels of privacy. Sites and apps with especially poor privacy tradeoffs will then be quickly spotted by users and serve as a disincentive to their developers. The effect is to raise privacy standards over time.

Given the benefits of reputation systems, some already exist. Web Of Trust (WOT) and WhatApp are examples of sites in which experts and users rate sites and apps on several dimensions so that better sites and apps can be easily found by users [14, 15]. However, both are vulnerable because they do not base ratings of trustworthiness or privacy on measured information exposure; both would benefit from privacy revelations.

5 Research Agenda

Despite its simplicity, the idea of revealing to users the information exposure of apps and sites as measured in practice poses challenges that require operating systems research. The key problem we highlight is the gulf between privacy as it can be measured today and concepts and descriptions that are suited to users. Traditional protection mechanisms such as access control and information flow operate at a low-level. To be meaningful, privacy revelations must operate at a high-level that is informed by user-facing descriptions. For example, a privacy revelation might be that a given app copies the

user's address book to the developer's server when the application is started, rather than that information sent to a certain IP address carries the taint of the contacts database. The semantic gap between the two levels involves at least the challenges below, which serve to push user-related concepts into the operating system.

Two further problems are how to share privacy revelations without compromising user privacy, and how to present them without overwhelming users. However, anonymization in various forms has received substantial attention from the research community, and usability issues are beyond the scope of this paper.

5.1 Application Traces not Network Traces

Descriptions of privacy in practice depend on the type of measurements that are available. Much Internet measurement research uses packet traces as a source of low-level data. However, as we quickly realized when building our systems, those traces are of limited value for understanding how apps expose personal information over the network.

Network traces do not reveal the contents of encrypted transfers (e.g., HTTPS) that are an important part of the spread of personal information. Nor do they record application interactions. Hence it is not possible to tell if a HTTP request and response were directly caused by a click or screen press, or silently triggered by a script or code that is hidden from the user. Similarly, it is not possible to tell if exposed personal information was recently entered (for sharing with a server) or sourced elsewhere (from storage or embedded in code).

For all these reasons, privacy revelations must be based on application-level traces (which include network activity). The challenge is to come up with generic ways of tracing high-level application behavior that can be supported by operating systems, rather than leaving untrusted apps to monitor themselves in different ways. We speculate that the relative ease of obtaining network traces and difficulty of obtaining application traces is one reason why there are measurement studies but a dearth of work akin to privacy revelations.

5.2 Measuring User-Facing Context

To be useful, revelations need to characterize privacy in terms that users can understand. However, operating system level traces presently lack formal notions of context that users take for granted. We give several examples.

Logins. It is easy for users to see whether they are logged into a website, and natural to expect that personal information will be exposed to a larger extent when logged in. However, the notion of logins does not exist in measurements of Web browser communications. It is not present in network protocols such as HTTP. The closest indicator here is a session cookie, which is defined by individual sites and cannot reliably be detected and understood. It is not present in page content. Logins may be effected with different mechanisms (e.g., form POST or AJAX), and even login buttons cannot be reliably detected because they may have various names or be an image. There is significant variation across sites as well as complications like single-sign-on (e.g., Facebook credentials used with another site).

Sites. Users are also cognizant of the site with which they are interacting, making it a natural part of a privacy revelation. However, there is no clear notion of site that can be measured for our purposes. Web pages are comprised of information drawn from many servers in potentially many different domains. At a high-level, users relate sites with companies but business relationships confuse the definition of site, as when a large company such as Google has multiple separate Web services. At a low-level, protection mechanisms such as the widely used Same Origin Policy (SOP) work in terms of domain names, which are a lower-level construct than users normally associate with sites.

Foreground vs. Background. Some network activity may occur in direct response to user interaction with an app or page. Other activity may happen in the background, e.g., due to services. These two cases are different to users, but difficult to separate with available measurements. For Web browsers, APIs give much information on user interface events and network events, but not the linkage between them. There is uncertainty even for mobile apps where foreground and background activity are clearly distinguished (e.g., under Android). For example, we observe apps that trigger a short burst of network activity when the user places them in the background. This is booked as background activity, but most users would categorize it as closing out foreground activity. It has different privacy implications than an app that continues to send the location after the user has finished with it.

For these and other contexts, the short-term challenge is to use heuristics to infer them from the available low-level measurements. Often this involves a substantial amount of work, e.g., our tool requires knowledge of a large number of patterns to recognize logins on Web pages with low false negative and low false positive rates. Applied machine learning can help. So can third party services, such as a service that developers can leverage to map between pages or apps and sites.

The long-term challenge is for browser, mobile phone and other operating systems to incorporate user-facing context directly so that it can be measured and enforced reliably. For example, browsers could support a standard login element that is used to identify logins clearly.

When present, it would enable further functionality such as consistently signaling logins to users, and protecting logins from eavesdropping.

5.3 High-Level Information Flow

The exposure of personal information across the network is more naturally expressed in terms of information flow than access control. With access control paradigms, e.g., as used in Android, exposure is only weakly restricted. For example, a sound meter app might require access to the microphone to measure ambient noise levels and the network to display ads. Yet these permissions do not prevent it from shipping sound bites across the network, something that most users would find undesirable.

However, existing information flow tools are of limited use for privacy revelations because they focus on the low-level concept of the flow of taint, not the higher-level ways in which personal information is transformed as it flows across the network. As an example, a significant number of apps literally send the mobile phone identifier (IMEI) over the network to identify the user, e.g., to record their highest score in a game. This is problematic for privacy because the IMEI uniquely identifies a phone across apps and so can be used to profile users over time. Instead, we have seen apps send a value derived from the literal IMEI, e.g., hashing it to a less sensitive identifier. This is beneficial in terms of privacy. However, standard information flow cannot distinguish these cases. The general point is that there is a difference to the user between the literal spread of personal information and the flow of taint that does not reveal the personal information (e.g., credit card number versus last four digits).

Information flow also cannot distinguish when important transformations are applied. For example, whether data is encrypted is a significant privacy factor. The challenge here is to develop techniques that report flow in terms of high-level transformations without trusting the site or app. We speculate that progress here may need different techniques for the risks of poorly written or configured apps versus malicious programs that attempt to conceal transformations using side channels.

5.4 Predictions versus History

To be useful, privacy revelations must tell users what will happen when they decide to use a site or app, rather than what has already occurred. For example, a tool that warns users that an app will send all their contacts to a spam operator is more helpful than one that tells users that this just happened. This places a premium on the ability to predict exposure rather than to summarize historical measurements. In some cases, prediction may be done by static analysis of information flow, e.g., looking for login processing on pages. In the more general case, sharing of privacy revelations is useful because one user's experience may benefit other users. This might be done by offline testing or crowdsourcing over a pool of users, with more sharing being beneficial because different users exercise apps in different ways, which increases coverage. The challenge is how to combine measurements taken from different users, and how to assess whether the app or site test coverage is sufficient for accurate predictions.

6 Conclusion

Privacy revelations are measurements of how personal information is spread by networked applications. They are intended to be shared and presented to users while they interact with applications. This idea is simple and has been largely unexplored by the research community, yet we argue that it is valuable to help users manage their privacy and to improve privacy standards over time. However, to be effective, privacy must be measured by the operating system and in terms of user-facing concepts. This requires new and better application analysis methods than the research community has today.

References

[1] A. Beresford, A. Rice, N. Skehin, and R. Sohan. MockDroid: trading privacy for application functionality on smartphones. In *HotMobile 2011*, 2011.

[2] S. Consolvo et al. The Wi-Fi Privacy Ticker: Improving Awareness & Control of Personal Information Exposure on Wi-Fi. In *Ubicomp 2010*, 2010.

[3] C. Dwork. Differential Privacy. *Automata, languages and programming*, pages 1–12, 2006.

[4] B. Edelman. Adverse Selection in Online "Trust" Certifications. In *ICEI 2009*, pages 205–212, 2009.

[5] M. Egele et al. PiOS: Detecting Privacy Leaks in iOS Applications. In *NDSS 2011*, 2011.

[6] W. Enck et al. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *OSDI 2010*, 2010.

[7] Federal Trade Commission. Protecting Consumer Privacy in an Era of Rapid Change. http://www.ftc.gov/, 2010.

[8] S. Guha, B. Cheng, and P. Francis. Privad: Practical Privacy in Online Advertising. In *NSDI 2011*, 2011.

[9] D. Jang, R. Jhala, S. Lerner, and H. Shacham. An Empirical Study of Privacy-Violating Information Flows in JavaScript Web Applications. In *CCS 2010*, 2010.

[10] J. John et al. Searching the Searchers using SearchAudit. In *USENIX Security 2010*, 2010.

[11] A. McDonald and L. Cranor. The Cost of Reading Privacy Policies. In *Technology Policy Research Conf.*, 2008.

[12] L. Sweeney. k-anonymity: a model for protecting privacy. *Intl. Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[13] S. Thrum and Y. Kane. Your Apps are Watching You. Wall Street Journal, http://online.wsj.com/, 2010.

[14] Web of Trust. http://www.mywot.com/, 2011.

[15] WhatApp. http://whatapp.org/, 2011.