Privacy-by-Design for the Security Practitioner

By Richard Chow (richard.chow@intel.com)

Privacy-by-Design is a strategy where privacy engineering is embedded throughout the entire product life-cycle rather than tacked onto the end. Privacy-by-Design has taken the privacy community by storm in the last few years, but the basic concept has been familiar to security practitioners for decades, where the prime commandment is designing in security at the beginning.

At the same time, engineers with significant experience in privacy are relatively scarce and many companies end up relying on security engineers to build privacy controls into their products and services. This makes some amount of sense, as the basic privacy concept of securing personal data is very familiar to security folk. However, there are some aspects of privacy that may be less familiar, and one of the goals of this paper is to outline and provide some of the basics on these aspects. In essence, my aim is to help the security engineer be conversant in privacy.

Security vs. Privacy

We start with an abstract comparison of privacy and security. With security, there is always an adversary or attacker, almost by definition. One common goal is not to release information to the attacker. For instance, if a user is sending another party a bit of information or storing a bit of information on a device, the attacker is supposed to not be able to guess with more than 50% accuracy whether the bit is 1 or 0. In other words, assuming appropriate security controls, the attacker is frozen out and in a rigorous sense is unable to obtain any knowledge about the user's data.

With privacy, we also assume the existence of a privacy attacker, but often the attacker has certain user data with (hopefully) full awareness and consent by the user. A hacker who steals the user's data is a privacy attacker, but if the user's data is collected by, say, Google, then (for the purposes of this paper) Google would also be considered a privacy attacker. Google may not be trying to subvert the user's privacy, but for analysis purposes one needs to understand what Google can infer about the user.

Because the user may be willingly releasing some data to the attacker, there are several difficulties. First, how does one describe the data collected to the user? For security, the *null* dataset is easy to understand and model. When the dataset is not null, describing the data collected becomes an issue, and the consent also becomes a problem. Just to illustrate the difficulties, when installing an Android application, how many users consent with an actual understanding of all types of data collected by the app?

Second, what can the privacy attacker infer about the released data? These inferences are often unclear to the user. For example, many users participating in studies involving collection of location data are unaware that the collected data can be used to infer their identity by deducing home addresses. Another example is that collection of data like browser type, plug-ins, and time-zone is becoming more common, but does the user understand that this data can identify the user's device?

The inference problem is compounded by the auxiliary knowledge of the attacker, i.e. what other data the attacker knows, about the user or in general. It is often hard for the user to understand the extent of this knowledge. Part of the concern with Google linking data from all its sites is the resulting auxiliary knowledge store. A more concrete example: on a social networking site if someone reports his location to be "at school," then one doesn't know much about where the person is. But if one knows the person lives in San Francisco, then perhaps one can narrow his whereabouts to a not-so-long list of schools. If one knows his address and age, then one could make a pretty good guess as to his location.

The inference problem, what it is possible to deduce with what certainty, is central to privacy. Hence, that is why, while the *lingua franca* of security is cryptography, the *lingua franca* of privacy is statistics, machine learning, and data mining. The growth of businesses such as online advertising and the rise of Internet-of-Things depend on increasingly accurate inferences about users, forcing constant calibrations to the inference problem.

Personally Identifiable Information (PII)

The data collected from a user may be traceable or identified with a particular person, and this is the idea behind *personally identifiable information* or PII. We find that project teams overly focus on what they consider to be the PII data. The idea is attractive on the surface – the team sifts through the data they collect to see which data is PII and then they apply existing policies for PII on this data. The problem is that PII is not so well-defined and furthermore depends on the totality of data collected. Whether a particular piece of data is considered PII may depend on what else is collected.

The APEC (1) definition of PII is "any information about an identified or identifiable individual." Usual examples of data that identifies are name, address, telephone numbers, birth date and place, photos, biometrics, IDs, login names, IP address, but of course any such list is incomplete. There are other definitions out there, but they share the difficulty in how to interpret "identifiable," how much auxiliary data is necessary and in what context. For example, the interpretation of what data is PII is generally broader in Europe compared to the US.

One must keep in mind that the concept of PII, like what constitutes a reasonable data description and reasonable consent, is not so much a technical construct, but one based in policy and law. Hence, our approach is to put aside the concept of PII and consider *all* the data that is collected. The reason is that we do not pretend to understand the legal subtleties of PII, but more importantly the concept of PII has no intrinsic technical meaning. Narayanan and Shmatikov (2) make the point well, and we summarize some of their reasoning below.

Even the most seemingly inconsequential pieces of data can be pieced together to identify a user or device. Device fingerprinting technologies are a prime example. One famous example (3) of re-identification is that four movie ratings were enough to identify most individuals in the anonymized Netflix Prize dataset.

To re-iterate, the personal nature of the data may only be apparent after considering all the data collected and with a holistic view of the system. Even with a holistic view, there is a wide spectrum in how easily the data identifies an individual, ranging from, for example, names and addresses to an identification relying on auxiliary data held in third-party proprietary databases.

Now, practically speaking, most organizations have certain policies for PII, for example, how to safeguard and handle PII data. How do we apply these policies, given the lack of a rigorous technical definition for PII? One course of action is to examine the collected data in aggregate and adopt a risk management-based approach to data classification, involving how easily the data identifies an individual and how sensitive the data is. Similar risk implies similar measures for safeguarding and handling the data. For the security practitioner, the risk posed by PII data is naturally handled through traditional security measures and the data minimization described in the next section.

Basic Questions

We have so far presented some background and a certain mindset. But how does this translate into actual privacy engineering?

Ann Cavoukian developed Privacy-by-Design (4)in the 90’s, when the usual approach towards privacy was regulation. Privacy-by-Design instead advocates an approach where privacy arises organically from an organization and uses principles from the OECD (5) which in turn derives from the Fair Information Practices (6). These days the regulatory approach has shifted towards encouraging Privacy-by-Design (e.g., recommended by the FTC (7) and EU Data Protection Regulation (8)), and Privacy-by-Design has become the de facto standard for privacy engineering.

Privacy-by-Design has 7 Foundational Principles:

1. Proactive not Reactive; Preventative not Remedial
2. Privacy as the Default
3. Privacy Embedded into Design
4. Full Functionality – Positive-Sum, not Zero-Sum
5. End-to-End Security – Lifecycle Protection
6. Visibility and Transparency
7. Respect for User Privacy

One can easily find information on these principles (for example, (9)) and studying these principles would certainly be time well-spent. However, for the purposes of the security practitioner, one can perhaps simplify these principles down to three basic questions to keep in mind:

(1) Is the data "secure"? That is, are the expected data outflows the actual data outflows? This is actually a question about the security architecture, which privacy relies upon. This article is directed towards security practitioners, so I won't say more about this question.

(2) Have we minimized the data collected? The less data collected, the more the risk is reduced. Generally, one should only collect data that conforms to a particular purpose that we've communicated to the user and we should keep the data only as long as we need it for the specified purpose.

(3) Does the user understand? Does the user have a good understanding of the kind of data that he is releasing, who he is releasing it to, and what it will be used for? Does the user have a good understanding of the inferences possible with the data?

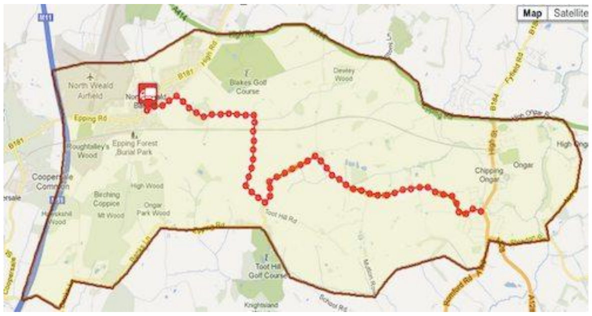
We expand on (2) and (3) below.

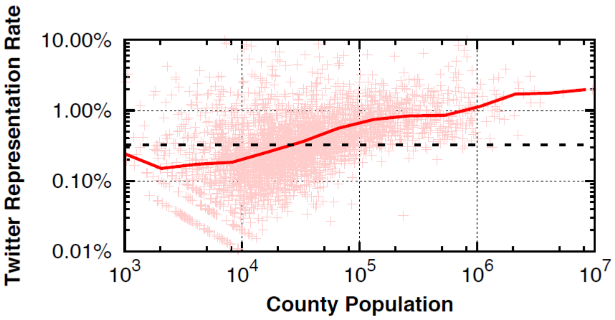
Have we minimized the data collected?

User or device identifiers provide several examples of the need for data minimization. The danger with identifiers is that they are the glue that enables sticking together bits of personal data. The Apple advertiser ID is problematical for privacy because it provides a global identifier that can be associated with longitudinal data across apps. Silos are not good in many contexts, but they are good for privacy.

The first question is, does one really need an identifier? In one project we consulted for, data was collected from devices for trouble-shooting and diagnostic purposes. Data needed to be associated with an identifier so that reports from the same device could be correlated. But the correlation needed only to be local, e.g., reports that are a week apart did not need to be correlated but reports that are an hour apart did. This suggested a data minimization strategy of periodically changing the identifier, say every few days. In this way, we got sufficient data for our purposes, but also enhanced the user's privacy by breaking the connection with the identifier every few days.

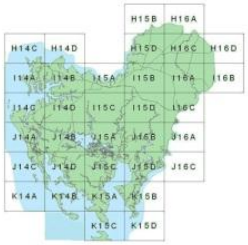
For another example of data minimization, we consider location, a primary ingredient in mobile apps and also one of the most privacy-sensitive. Suppose a web service provides the user's weather in his location. In order to get this data, the web service sends users' latitude-longitude pair, but does not send any user IDs, so that the third-party provider cannot distinguish a query corresponding to User A from a query corresponding to User B. This appears to be safe, but, after thinking a bit, does a provider ever see a series of queries such as the figure to the right? This would correspond to a geographically isolated user and would be a problem as the user's identity and location history would be revealed.





Such situations would seem unlikely as long as the number of users of the service is relatively large. But in fact this leakage of data to the web service is indeed a concern, even with a large set of users. Consider the figure to the left (from “Understanding the Demographics of Twitter Users” by Mislove et al.). Even with 3.2 million Twitter users, the bottom leftmost diagonal points in the plot show that many counties have only a single Twitter user, and hence are quite geographically isolated. This is another example of the interplay of statistics and privacy – often privacy depends on the nature of data distributions. Hiding in a crowd works OK if your data is similar to many others, not so well if your data stands out.

So, what to do? We suggest applying data minimization prior to sharing with the third-parties, which do not actually need the latitude-longitude pair for applications like the local weather. One approach would be to super-impose a coarse grid onto the map (assuming weather is fairly constant inside each square) and send a user’s grid square instead of actual location.



Data retention is not usually considered an aspect of data minimization, but the same philosophy applies. If practical, do not keep data longer than needed.

In some cases, however, there are compelling reasons to keep data around, and techniques to de-personalize the data such as anonymization and aggregation are commonly used. For instance, Google (10) and Yahoo! (11) use anonymization techniques with their search data, although the precise algorithms are not disclosed. We recommend caution in the area of de-personalizing data though. Simply removing identifiers is often insufficient. The example of the Netflix Prize Contest was described earlier. Other examples are the AOL search data (12) and the Latanya Sweeney analysis (13) of medical data.

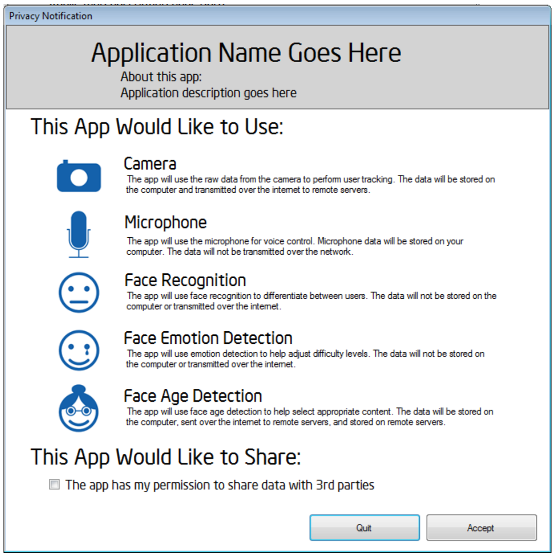
These attacks on data assumed to be de-personalized are an example of how unexpected inferences are possible using machine learning. A major challenge to data minimization is that such inferences will only increase as the size and pervasiveness of user data grows.

Does the user understand?

In the traditional notice-and-consent privacy model the user consents in some fashion to the data being collected, with the assumption that the user understands the specific data collected and the purpose(s) for which the data will be used. In practice, real understanding is difficult to achieve and the consent mechanism seems as much for the lawyers and policy-makers as it is for users. Hence, system designers should do as much as possible in the “minimizing data” area and reduce the need for the more difficult task of ensuring true ‘informed consent’ by the user.

As a simple illustration of the difficulties in user understanding, consider the Perceptual Computing SDK (PCSDK) (14), a bundle of drivers and middleware algorithms for using cameras and microphones that will be installed on the end user's device. Third-party application developers write applications that run on top of the PCSDK. These applications must notify users what and when sensors will be in use.

Application developers for the PCSDK are required to implement a dialog window as shown in the figure on the right. Icons representing each type of sensor usage are meant to enhance user understanding of what kinds of data the application collects.


Privacy Notification

Application Name Goes Here

About this app:
Application description goes here

This App Would Like to Use:

Camera
The app will use the raw data from the camera to perform user tracking. The data will be stored on the computer and transmitted over the internet to remote servers.

Microphone
The app will use the microphone for voice control. Microphone data will be stored on your computer. The data will not be transmitted over the network.

Face Recognition
The app will use face recognition to differentiate between users. The data will not be stored on the computer or transmitted over the internet.

Face Emotion Detection
The app will use emotion detection to help adjust difficulty levels. The data will not be stored on the computer or transmitted over the internet.

Face Age Detection
The app will use face age detection to help select appropriate content. The data will be stored on the computer, sent over the internet to remote servers, and stored on remote servers.

This App Would Like to Share:

☐ The app has my permission to share data with 3rd parties

We note the *unchecked* box for sharing data with third parties at the bottom of the dialog. There is an important principle in Privacy-by-Design: private-by-default. This is analogous to *fail-safe* for security systems. The idea is that even if the user doesn't understand or doesn't act, the user maintains his or her privacy. Of course, it is an interesting question as to what private-by-default means for a product like Facebook, where most users do not change their default privacy settings.

The dialog window is given to the user at install time or at the time the application first runs, and from the main part of the dialog the user is meant to understand which data is collected by the application and why it is collected. The installation procedure is similar to the procedure for an Android application. But like an Android application, even though the letter and spirit of privacy laws are met, it's unclear how effective this install-time mechanism actually is; see, for example, Felt et al: *Android permissions: user attention, comprehension, and behavior* (15).

Furthermore, even leaving aside the issue of whether the user is actually paying attention, in some situations the method of giving notice or obtaining consent is unclear. For example, in the new world of the Internet-of-Things, the user has no clear interface to pervasive sensors recording his data for analysis. For both these newer settings and the more traditional notice-and-consent model, the level of user understanding suggests that there is opportunity for innovation in this area.

Summary

Privacy-by-Design has become the de facto standard for privacy engineering. Security practitioners are venturing (or are being drafted) into privacy in greater numbers, and the principles and philosophy of Privacy-by-Design will be familiar for the most part. We have concentrated on two areas that may be less familiar: data minimization and enhancing user transparency or understanding. Both are areas that are rapidly increasing in importance. Data minimization emphasizes machine learning, and its relevance is surging as the size and pervasiveness of user data grows. Managing user understanding meanwhile emphasizes the human-computer interface. Besides the need for general improvement in this area, the rise of mobile computing and the Internet-of-Things forces different modes of interaction.

References

1. APEC Privacy Framework. [Online] http://www.apec.org/Groups/Committee-on-Trade-and-Investment/~/media/Files/Groups/ECSG/05_ecsg_privacyframewk.ashx.

2. **Narayanan, Arvind and Shmatikov, Vitaly.** Myths and Fallacies of 'Personally Identifiable Information'. [Online] http://cacm.acm.org/magazines/2010/6/92489-myths-and-fallacies-of-personally-identifiable-information/fulltext.

3. *Robust De-anonymization of Large Sparse Datasets.* **Narayanan, Arvind and Shmatikov, Vitaly.** s.l. : IEEE S&P, 2008.

4. Privacy-by-Design. [Online] http://www.privacybydesign.ca/.

5. OECD. [Online] http://www.oecd.org/internet/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflowsofpersonaldata.htm.

6. Fair Information Practices. [Online] http://itlaw.wikia.com/wiki/Fair_Information_Practice_Principles.

7. FTC. [Online] http://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-report-protecting-consumer-privacy-era-rapid-change-recommendations/120326privacyreport.pdf.

8. EU Data Protection Regulation. [Online] http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf.

9. **Cavoukian, Ann.** Privacy by Design, 7 Foundational Principles. [Online] http://www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf.

10. Google anonymization. [Online] https://support.google.com/accounts/answer/162743?hl=en.

11. Yahoo! anonymization. [Online] http://info.yahoo.com/privacy/us/yahoo/datastorage/details.html.

12. **Jr., MICHAEL BARBARO and TOM ZELLER.** A Face Is Exposed for AOL Searcher No. 4417749. [Online] August 9, 2006. http://www.nytimes.com/2006/08/09/technology/09aol.html.

13. **Sweeney, Latanya.** Computational Disclosure Control, A Primer on Data Privacy Protection . [Online] http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/sweeney-thesis-draft.pdf.

14. Perceptual Computing SDK. [Online] http://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk.

15. *Android Permissions: User Attention, Comprehension and Behavior.* **Adrienne Porter Felt, Serge Egelman , Ariel Haney , Erika Chin , David Wagner , Ariel Haney , Erika Chin , David Wagner.** 2012. SOUPS.

The opinions expressed in this paper are the author's own and do not reflect the view of Intel Corporation.