Privacy Engineering in Dynamic Settings

Inah Omoronyia
School of Computing Science
University of Glasgow, UK
inah.omoronyia@glasgow.ac.uk

***Abstract*—Modern distributed software platforms are linking smart objects such as smartphones, cars and health devices to the internet. A frequent challenge in the design of such platforms is determining the appropriate information disclosure protocol to use when one object interacts with another. For example, how can a software architect verify that when the platform constrains the sender to obtain consent from the subject before disclosure or notifying the subject after disclosure, then the privacy needs of the subject are addressed? To this end, this research presents an analysis framework for privacy engineering. We demonstrate how the framework's outputs can help software architects achieve privacy-by-design of software platforms for smart objects.**

I. Introduction

Taint analysis, a technique used for tracking the flow of sensitive information from data sources to sinks in program code [1][3], has shown to be effective in dealing with a range of privacy problems[2][15][8]. But the usage of applications categorised as benign after taint analysis may still lead to privacy violation when information is disclosed inappropriately. For example, while the program code for a Facebook application on a mobile device may be perfectly legitimate, users can still *forward* or *like* messages within Facebook network in a manner that violates the privacy of the subject.

Furthermore, traditional privacy engineering practices do not fit well with distributed object scenarios. The common practice involves three steps. First, the software presents the user with a privacy preference settings interface to express a privacy requirement. The software then determines the appropriate disclosure protocol to ensure the satisfaction of the privacy requirement. The final step is the enforcement or monitoring adherence to agreed disclosure protocols [13][6][9]. But described steps are only suitable for self-contained settings where privacy analysis is independently executed, and the impact of disclosure behaviour localised to avoid contagion on other users privacy requirements[4]. Whereas, the distribution of objects introduces new dynamics that makes these assumptions difficult to hold. For example, it is more difficult to understand the consequence of information disclosure on privacy. This is because it is easy for information once disclosed to reach unintended recipients, and users are frequently unclear if an information-flow path will ultimately lead to privacy violation [9][7][14].

One way to address this challenge is an approach that enables objects to determine alternative disclosure protocol(s) that maximises privacy requirement satisfaction. The approach should also at one end provide the capability for objects to relax or enhance the extent to which a privacy requirement is satisfied, and at the other end forfeit or update the privacy requirement depending on emergent disclosure behaviour of objects. The aim of this research is to show that by leveraging on knowledge abstraction models [11][10][12], such analytical approach is achievable over an information-flow path. This is by analysing traces from the statespace of possible disclosure protocols to determine the extent to which they each satisfy a privacy requirement along a path. Subsequently, traces that result in higher satisfaction levels becomes the preferred disclosure protocol(s). Our proposed framework is shown in Figure 1, and takes as input an information-flow path, a privacy requirement and the desired satisfaction level. The output is the appropriate disclosure protocol for transferring information from one object to another. Alternatively, where a satisfaction level is not achievable, different privacy adaptation measures are suggested.

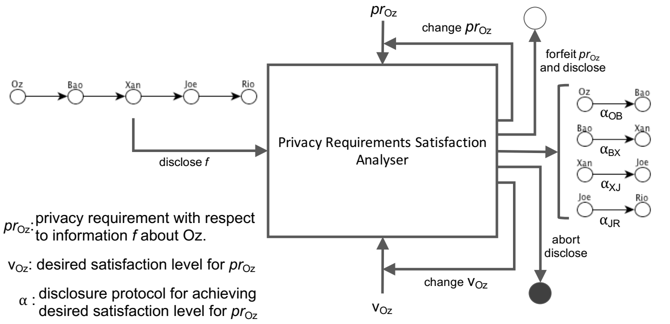
pr_{Oz}
change pr_{Oz}
forfeit pr_{Oz} and disclose
Oz
Bao
Xan
Joe
Rio
disclose f
Privacy Requirements Satisfaction Analyser
α_{OB}
α_{BX}
α_{XJ}
α_{JR}
abort disclose
change V_{Oz}
V_{Oz}
pr_{Oz}: privacy requirement with respect to information f about Oz.
V_{Oz}: desired satisfaction level for pr_{Oz}
α : disclosure protocol for achieving desired satisfaction level for pr_{Oz}

Fig. 1: Privacy analysis framework

II. Beliefs, Uncertainties and Privacy

Our knowledge abstraction technique is motivated by the possible worlds semantics as the logic of knowledge for describing alternative ways (modes) the world might have been like [5]. This is used to model the transformations in beliefs and uncertainties of users as information is disclosed from one object to another. Assume information f about $\texttt{Oz}$ flows along the path of objects $\texttt{Oz} \rightarrow \texttt{Bao} \rightarrow \texttt{Xan} \rightarrow \texttt{Joe} \rightarrow \texttt{Rio}$, where each object is labelled by the user it represents. Then the disclosure protocol used for each information-flow will determine the manner the beliefs and uncertainties that users have about f is transformed. For instance, if the disclosure protocol $\alpha_{\texttt{OB}}$ where after $\texttt{Oz}$ sends f to $\texttt{Bao}$, then $\texttt{Bao}$ reciprocates by acknowledging the receipt of f is used for $\texttt{Oz} \rightarrow \texttt{Bao}$. The consequence is Bao will belief that Oz knows that it knows f, represented as $B_{\texttt{Bao}}K_{\texttt{Oz}}K_{\texttt{Bao}}f$. Without such acknowledgement, Bao will be uncertain, represented as $B_{\texttt{Bao}}K_{\texttt{Oz}}K_{\texttt{Bao}}f|\neg B_{\texttt{Bao}}K_{\texttt{Oz}}K_{\texttt{Bao}}f$. Also, considering $\alpha_{\texttt{BX}}$ for Bao → Xan. If before Bao sends f to Xan, it is granted consent to do so by Oz with the consent subsequently acknowledged, and Bao notifying Oz after $f$is sent. Then, the belief proposition $B_{\texttt{Oz}}K_{\texttt{Bao}}K_{\texttt{Xan}}f$ will hold as a result of the notification, and uncertain otherwise. Same technique of varying disclosure protocols to derive a different belief/uncertainty transformation can also be considered for Xan → Joe and Joe → Rio.

Consequently, the combination of information request, consent, sent and/or notice defines the transactions in a disclosure protocol trace. This results in a statespace containing 112 traces, each generating a different user belief and uncertainty transformation after information is disclosed. At any point, the uncertainty level of an object u about f is the ratio of uncertainty to total elements in u's knowledge.

$$\texttt{UncertaintyLevel}(u,f) = \frac{|\texttt{UncertaintyElement} \in P_u|}{|P_u|}$$

Conversely, belief level is the ratio of belief to total elements:

$$\texttt{BeliefLevel}(u,f) = 1 - \texttt{UncertaintyLevel}(u,f)$$

When an object has high uncertainty level, then associated information is more private, whereas high belief level means less private information. A privacy objective may therefore involve regulating uncertainty and belief levels of objects.

Hence, by specifying privacy requirements as assertions using our proposed knowledge abstraction, it becomes possible to determine which disclosure protocol(s) will maximise its satisfaction. For example the requirement that when f is disclosed, then Oz should belief that Don knows f, and that Joe knows that it knows f, as well as Bao be uncertain that Xan knows f, is specified by:

$$\begin{aligned} pr_1 : \textbf{if}(\texttt{Oz}, \texttt{Oz}, \texttt{Joe})\ \textbf{then}\{&B_{\texttt{Oz}}K_{\texttt{Bao}}f, B_{\texttt{Oz}}K_{\texttt{Joe}}K_{\texttt{Oz}}f, \\ &B_{\texttt{Bao}}K_{\texttt{Xan}}f|\neg B_{\texttt{Bao}}K_{\texttt{Xan}}f\} \end{aligned}$$

The satisfaction of this type of expression is determined by checking whether assertion elements in the then segment of the privacy requirement are true or false in the knowledge of objects (i.e. Oz and Bao for pr_1). Our analysis logic is as follows: For a given disclosure protocol α_x, the privacy requirement pr is evaluated in the current step t_i in object u along the path h. The extent of satisfaction of pr by u is the ratio of elements in A that is also contained in P_u to the total elements in A. This is represented as:

$$t_i : \texttt{sat}(pr, \alpha_x, u) = \frac{|(a_i \in A \wedge a_i \in P_u)|}{|A|}$$

Where A is the set of assertions in pr and P_u the knowledge of u. The satisfaction of pr along h is then a combination of extent to which each object in h satisfies pr. This is the mean sat values achieved along h, and represented by:

$$h : \texttt{pathsat}(pr) = \frac{\sum_{t_0}^{t_{|h|}} \texttt{sat}(pr, \alpha_x, u)}{|h|}$$

TABLE I: A comparison of satisfaction analysis outcome for pr_1 using different disclosure protocols

<table>
<tr><td colspan="6">$\alpha_{23} = (\texttt{Request}, \texttt{grantConsent}, \texttt{send}, \texttt{notice}_{[\texttt{s},\texttt{su}]}, \texttt{ackNotice}_{[\texttt{su},\texttt{s}]}, \texttt{notice}_{[\texttt{s},\texttt{r}]})$</td></tr>
<tr><td>s → r</td><td>sat : su</td><td>sat : s</td><td>sat : r</td><td>pathsat</td><td>assertion :pr_1</td></tr>
<tr><td>Oz → Bao</td><td>0</td><td>0</td><td>NA</td><td>0.00</td><td>$B_{\texttt{Oz}}K_{\texttt{Bao}}f$</td></tr>
<tr><td>Bao → Xan</td><td>0</td><td>1</td><td>NA</td><td>0.25</td><td>$B_{\texttt{Oz}}K_{\texttt{Bao}}f$
$B_{\texttt{Bao}}K_{\texttt{Xan}}f|\neg B_{\texttt{Bao}}K_{\texttt{Xan}}f$</td></tr>
<tr><td>Xan → Joe</td><td>1</td><td>NA</td><td>NA</td><td>0.40</td><td>$B_{\texttt{Oz}}K_{\texttt{Joe}}K_{\texttt{Oz}}f$</td></tr>
<tr><td>Joe → Rio</td><td>1</td><td>NA</td><td>NA</td><td>0.50</td><td>$B_{\texttt{Oz}}K_{\texttt{Joe}}K_{\texttt{Oz}}f$</td></tr>
<tr><td colspan="6">$\alpha_{46} = (\texttt{Request}, \texttt{send}, \texttt{ackSend}, \texttt{notice}_{[\texttt{s},\texttt{su}]}, \texttt{notice}_{[\texttt{s},\texttt{r}]}, \texttt{ackNotice}_{[\texttt{r},\texttt{s}]})$</td></tr>
<tr><td>s → r</td><td>sat : su</td><td>sat : s</td><td>sat : r</td><td>pathsat</td><td>assertion :pr_1</td></tr>
<tr><td>Oz → Bao</td><td>1</td><td>1</td><td>NA</td><td>1</td><td>$B_{\texttt{Oz}}K_{\texttt{Bao}}f$</td></tr>
<tr><td>Bao → Xan</td><td>1</td><td>0</td><td>NA</td><td>0.75</td><td>$B_{\texttt{Oz}}K_{\texttt{Bao}}f$
$B_{\texttt{Bao}}K_{\texttt{Xan}}f|\neg B_{\texttt{Bao}}K_{\texttt{Xan}}f$</td></tr>
<tr><td>Xan → Joe</td><td>1</td><td>NA</td><td>NA</td><td>0.80</td><td>$B_{\texttt{Oz}}K_{\texttt{Joe}}K_{\texttt{Oz}}f$</td></tr>
<tr><td>Joe → Rio</td><td>1</td><td>NA</td><td>NA</td><td>0.83</td><td>$B_{\texttt{Oz}}K_{\texttt{Joe}}K_{\texttt{Oz}}f$</td></tr>
</table>

III. Preliminary Results

Based on our analysis logic, we implemented PSat - a tool for reasoning about the satisfaction of privacy requirements (www.dcs.gla.ac.uk/ inah/prisoft/). The outcome of analysing pr_1 in PSat using two different disclosure protocols α_{23} and α_{46} is as shown in Table I. One way to consider the viability of disclosure protocols is to compare their relative sat values. Hence, a disclosure protocol is more viable for a given information-flow when it yields a higher sat value for a privacy requirement. For example, consider the information-flow Oz → Bao, the disclosure protocol α_{23} is less viable. This is because it does not yield any satisfaction value for pr_1 in the knowledge of Oz or Bao. Conversely, α_{46} is more viable since it yields the maximum satisfaction value for the same information-flow. For Bao → Xan, while α_{23} yields no satisfaction of pr_1 in *Bao*'s knowledge, the maximum value is generated in Xan's knowledge. The reverse is the case for α_{46}. Finally, both protocols are equally viable for Xan → Joe and Joe → Rio. Thus, given the higher pathsat value for α_{46} it is the preferred disclosure protocol for pr_1 compared to α_{23}. Such insights can be used by software architects to help make appropriate design decisions, especially in the privacy engineering of platforms for smart objects.

IV. Conclusion

One key finding from our preliminary results is that when distributed software platforms are instrumented to simply apply generic disclosure protocols for every information-flow along a path, then the satisfaction of privacy requirements is not necessarily maximised. While this is arguably a common privacy engineering practice, a rigorous evaluation of the disclosure protocols that fits every information-flow is necessary for today's new generation of smart objects. Such disclosure protocol preferences can also be influenced by factors such as the privacy requirements satisfaction level that should be achieved, the degree of freedom it offers users during interaction and the cost of its execution.

V. Acknowledgments

This research is supported by EPSRC-ISF 2016 Institutional support grant award EP/P51133X/1

References

[1] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *ACM SIGPLAN Notices*, 49(6):259–269, 2014.

[2] V. Avdiienko, K. Kuznetsov, A. Gorla, A. Zeller, S. Arzt, S. Rasthofer, and E. Bodden. Mining apps for abnormal usage of sensitive data. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 426–436. IEEE, 2015.

[3] L. N. Q. Do, K. Ali, B. Livshits, E. Bodden, J. Smith, E. Murphy-Hill, and I. Fraunhofer. Just-in-time static analysis. *month*, 2016.

[4] E. Ferrara and Z. Yang. Measuring emotional contagion in social media. *PloS one*, 10(11):e0142390, 2015.

[5] J. Hintikka. *Knowledge and belief: An introduction to the logic of the two notions*, volume 181. Cornell University Press Ithaca, 1962.

[6] J.-M. Horcas, M. Pinto, L. Fuentes, W. Mallouli, and E. M. de Oca. An approach for deploying and monitoring dynamic security policies. *Computers & Security*, 58:20 – 38, 2016.

[7] A. Joshi, T. Finin, L. Kagal, J. Parker, and A. Patwardhan. Security policies and trust in ubiquitous computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881):3769–3780, 2008.

[8] L. Li, T. F. Bissyandé, D. Octeau, and J. Klein. Reflection-aware static analysis of android apps. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ASE 2016, pages 756–761, New York, NY, USA, 2016. ACM.

[9] R. Neisse, G. Steri, D. Geneiatakis, and I. N. Fovino. A privacy enforcing framework for android applications. *Computers & Security*, 2016.

[10] I. Omoronyia. The case for privacy awareness requirements. *Int. J. Secur. Softw. Eng.*, 7(2):19–36, Apr. 2016.

[11] I. Omoronyia. Reasoning with imprecise privacy preferences. In *ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, FSE 2016, New York, NY, USA, 2016. ACM.

[12] I. Omoronyia, L. Cavallaro, M. Salehie, L. Pasquale, and B. Nuseibeh. Engineering adaptive privacy: On the role of privacy awareness requirements. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 632–641, Piscataway, NJ, USA, 2013. IEEE Press.

[13] F. B. Schneider. Enforceable security policies. *ACM Trans. Inf. Syst. Secur.*, 3(1):30–50, Feb. 2000.

[14] J. Singh, T. F.-M. Pasquier, and J. Bacon. Securing tags to control information flows within the internet of things. In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[15] J. Yang, T. Hance, T. H. Austin, A. Solar-Lezama, C. Flanagan, and S. Chong. Precise, dynamic information flow for database-backed applications. In *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 631–647. ACM, 2016.