Detecting Privacy Infractions in Applications: A Framework and Methodology

Michael Smit
Department of Computing Science
University of Alberta
Edmonton, Alberta
Email: msmit@cs.ualberta.ca

Kelly Lyons
Faculty of Information Studies
University of Toronto
Toronto, Ontario
Email: kelly.lyons@utoronto.ca

Michael McAllister
Jacob Slonim
Faculty of Computer Science
Dalhousie University
Halifax, Nova Sotia
Email: mcallist,slonim@cs.dal.ca

Abstract

We describe a framework and methodology for managing the privacy policy of an enterprise, including creation (based on factors like legislation and consumer preferences), validation and verification, deployment and enforcement, and compliance testing for business processes and software.

To evaluate this approach, one module of our framework (compliance testing) is implemented for an existing prominent electronic commerce software application. Our unique approach monitors the personal information sent and received by the software application and converts it to a standardized representation. At defined points in the electronic commerce workflow, the transmissions are compared to a set of privacy rules (extracted from a privacy policy) to ascertain compliance. Non-compliant transmissions of personal information are labeled 'potential privacy infractions' and are reported. Though presently implemented for software testing, ultimately the methodology is intended to halt or alter a workflow to avoid privacy infractions.

1. Introduction

Privacy was once defined as the "right to be let alone" [1]. As new technology developed, this definition was extended to include the notion that individuals should have control over when and to whom they divulge personal information and what the recipient may do with the personal information upon receipt. Improved database management systems, pervasive computing applications and data mining algorithms enable the collection, aggregation, sharing and use of a growing amount of information, but can also offer the specification of individual privacy preferences and better privacy protection and compliance verification.

The collection, use, and dissemination of information is a crucial element of the knowledge economy. As applications shift from a desktop paradigm to a pervasive computing model, success and growth will require that businesses manage personal information in a manner that meets a variety of preferences, requirements, and incentives held by the businesses' stakeholders.

In addition to consumer requirements and legislative requirements, an enterprise will have privacy requirements based on the cost-benefit analysis of privacy protections, industry standards, its contracts with other enterprises, and the privacy policies of its competitors. From these requirements, an enterprise must determine its privacy policy, which is then deployed throughout the enterprise. The enterprise must ensure that its employees, business processes, software, and systems comply with the policy. As the influences on the enterprise change, or as the enterprise itself changes, so might its privacy policy; a revised policy must again be implemented and tested for compliance. Compliance can be a very complex process.

This paper describes an enterprise privacy policy management framework which determines an enterprise's privacy policy (based on the influence of factors from both outside and inside the enterprise), validate and verify this privacy policy, deploy and enforce it, and test employees, business processes, and software applications for compliance.

A proof-of-concept implementation of the latter component of the framework, the compliance testing module, is also presented. It tests an enterprise software application for privacy policy compliance without requiring modification of the original software application. An information flow report is created by modeling the flow of personal information through the access and exit points of the software application. Identifying the flow of information is useful on its own; however, we also compare the report to a set of privacy rules in order to detect which flows are potentially non-compliant with the privacy rules. Our implementation was evaluated on a commercial e-commerce software application in cooperation with the software vendor.

In the remainder of this paper, Section 2 describes the background and related work in privacy as it relates to electronic commerce, policy, and technology. Section 3 describes the various influences on an enterprise's internal privacy policy. Section 4 describes the overall framework for enterprise privacy policy management and the specific methodology for privacy policy compliance testing. Section 5 describes the implementation and evaluation of a privacy compliance testing module. The conclusions and suggestions for future work are provided in Section 6.

978-1-4244-5113-5/09/$25.00

2. Background and Related Work

The term privacy is a nuanced word. The definition of privacy used in this paper is drawn from the definition used by the Privacy Commissioner of Canada: "the right to control access to oneself and to personal information about oneself"[1]. For a business or an enterprise operating in today's e-economy, this means ensuring that consumers have control over how their personal information is handled by the enterprise. Business practices and technology exist to help organizations manage and protect the privacy of their customers (*e.g.*, [2], [3]).

In general, a policy is "a set of considerations designed to guide decisions on courses of action" [2]. These *enterprise policies* evolve over time and must be enabled and enforced by employees and incorporated within the processes of the business [4]. *Enterprise privacy policies* are business policies that address privacy and handling of personal information in the enterprise. An enterprise policy may differ from the privacy policy displayed to customers, though ideally only in scope and the level of detail offered and not in content. Enterprise policy management systems manage all aspects of an enterprise's privacy (including the establishment, communication, maintenance, and execution of business policies [2]) and provide structure to the task of creating, deploying, and enforcing enterprise policies.

A *privacy impact assessment* (PIA) is a formal process used to determine how an organization's business processes impact individuals' privacy and to provide ways to mitigate or avoid any adverse effects [5]. The general approach is similar for all organizations; one implementation used by the Treasury Board of Canada is to determine the flow of data through the organization, analyze these data flows in the context of the organization's established privacy policy, and document areas that present the greatest risk of violating the privacy policy and appropriate remedial action. PIAs, and similar *external privacy audits*, are offered as services by accounting and enterprise services firms. Conducting PIAs does not enable an enterprise to enforce compliance with the policies or monitor actual practices because they only examine high-level business processes and best-practices documents. A PIA treats a software application as a "black box" that performs as required by the business practices, rather than as a complicated entity which has many components that may or may not comply with privacy policies.

Security threat models are an approach to security made popular by Microsoft. In their documentation [6], [7], they present a structured approach to modeling targets and potential attacks which is intended to aid in designing and testing software systems. They do not specifically address privacy, nor does their approach call for any level of automation. However, some of their principles, such as data flows and trust boundaries, are similar to the concepts we identify when automatically detecting privacy compliance issues.

Concepts similar to security threat models have been employed by Hong *et al.* [8] when discussing privacy in ubiquitous computing. The authors discuss methods for assessing the privacy risk present in computer programs or advanced systems that track user's movements, location, actions, and the like. Their "privacy risk model" is a set of questionnaires designed to identify and manage the aspects of a ubiquitous computing system that present a risk of violating a user's privacy. The methodology proposed is not automated and does not examine as-implemented software. The aspects of the software examined are a subset of those described in security threat models.

1. http://www.privcom.gc.ca/speech/02_05_a_020516_e.asp

3. Privacy Policy Influences

An enterprise engaging in e-commerce has positive incentive to adhere to privacy protections advocated by its various stakeholders. We call these stakeholders *privacy policy influences* (PPIs). In this section, we briefly describe two major PPIs on enterprise privacy policy, legislation and consumer preferences. Beyond these two, contracts, industry standards, the influence of corporate executives, and any number of other stakeholders may be PPIs. The privacy protections advocate by PPIs comprise the input to our enterprise privacy policy management framework.

3.1. Legislation

The Canadian private sector is governed at the federal level by the *Personal Information Protection and Electronic Documents Act (PIPEDA)* [9], which applies to any private sector organization executing commercial transactions. PIPEDA codifies the Canadian Standards Association's model code [10], which include notification, consent, limited collection and use, access, and accuracy.

The United States has legislation protecting specific types of personal information or groups of people (*e.g.*, the *Children's Online Privacy Protection Act* protects the information of children under the age of thirteen [11]). In the European Union, the Data Protection Directive (DPD) [12], passed by the European Parliament in 1995, sets a standard of privacy for digital data processing in member countries. As of April 2006, all twenty-five member countries had passed national privacy laws compliant with the DPD.

In most countries or jurisdictions, regional laws exist that vary in similarity to federal or country-level laws. In Canada, for example, the provinces and territories have their own privacy or access to information legislation. In the United States, there are a number of state-level laws protecting the privacy of personal information collected and/or stored by corporations.

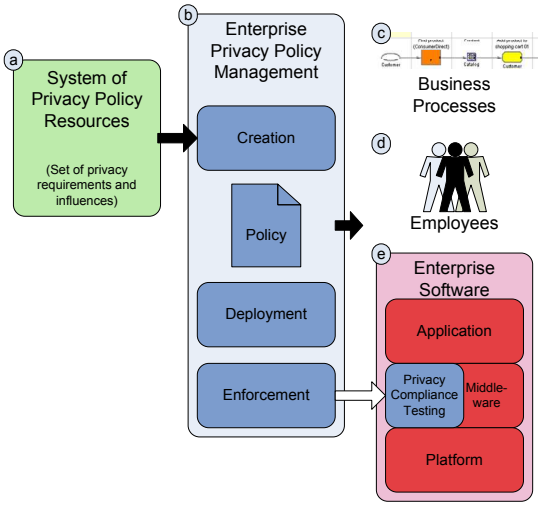

Figure 1. An overall view of enterprise privacy policy management.

Large organizations which do business in many jurisdictions have to deal with the issue of having their privacy practices governed by a variety of sometimes conflicting laws. Formulating privacy policies and updating workflows, re-educating employees, and upgrading software to comply with new and varied policies can be time-consuming, expensive, and prone to errors.

3.2. Public Opinion and Consumer Preferences

In addition to abiding by privacy laws, businesses must meet the privacy expectations of their customers or risk consequences such as civil lawsuits, restrictions on operations due to lawsuits, decreased stock value, cost of responding to complaints, recurring privacy audits, and negative publicity.

Public opinion polls show that individuals' opinions on privacy vary [13], a conclusion shared by Culnan and Armstrong [14]. An individual's preferences will be influenced by factors including recent news reports about privacy [15], age and geography [16], education [17], pre-existing levels of trust [16], and the stated practices of a given enterprise [14], [16]. An individual's views on privacy may change over time as the influencing factors evolve.

A 2006 survey reported that 71 percent of respondents had decided against registering or making a purchase online because those actions required them to provide information that they did not want to divulge. In the previous six months, 41 percent had provided inaccurate information to Web sites to avoid providing personal information which they did not wish to share [18]. The same survey showed that consumer trust varies depending on which organization he or she is dealing with (*e.g.*, 51 percent trust online shopping sites like Amazon; 10 percent trust social networking sites).

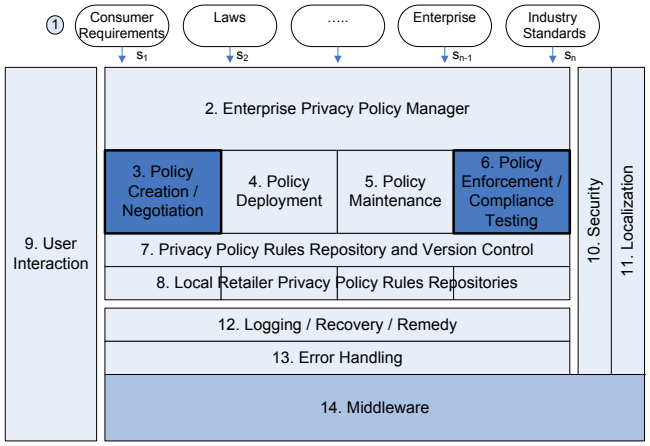

Figure 2. The enterprise privacy policy management framework.

4. Enterprise Privacy Policy Management Framework

The role of privacy policy management in an enterprise is to create a privacy policy based on the influence exerted by the enterprise's stakeholders (PPIs) and then deploy and enforce this policy on the employees, processes, and software applications of the enterprise. Figure 1 shows the role of policy management in an enterprise.

In this section, we present our framework for enterprise privacy policy management, shown in Figure 2. The Enterprise Privacy Policy Manager (2) is the coordinator of the privacy management software. It retrieves privacy policies from the PPIs (1) and invokes the policy creation module (3) to combine the separate policies into one central enterprise privacy policy. The policy is maintained by another module (5) which controls updates and revisions to the policy in coordination with the policy creation module.

The policy is deployed (4) to the enterprise. This includes dissemination of privacy policies to the employees of the enterprise through training, examining business processes to ensure each complies with the privacy policy, and modifying software applications that implement or support these business processes through updates to design, changes in configuration, upgrades, patches, or any of the above.

The compliance module (6) ensures that the deployment phase has adequately implemented changes based on the policy and that the environment has not changed without corresponding changes in the privacy policy. Employees may receive refresher courses or privacy certification exams. Business processes undergo regular privacy impact assessments to identify privacy infractions. Software applications are tested to ensure compliance with the privacy policy.

The policy repository (7) is used by the other modules to store and retrieve enterprise policies and is responsible for version control. This policy repository stores the policies of

the entire enterprise and of each subset of the enterprise. A local policy repository stores policies for any subset of the enterprise (8) which may have specific privacy requirements. The enterprise privacy policy manager and these modules (3 through 8 in Figure 2) are the contribution of the framework.

A user interaction layer (9) is provided to manage interactions with the user (generally an employee of the enterprise). The security layer (10) manages authentication and access control. The localization (11) layer provides a means to work in other languages and to translate privacy rules from and to other languages, as well as to map rules from one language or locality to existing rules in other languages or localities. The logging/recovery/remedy (12) and error handling (13) layers log activity within the framework and take remedial action when errors occur. This entire framework operates just above the middleware layer of an enterprise as an extension to existing middleware (14).

In Sections 4.1-4.2, we describe the role of the *Policy Creation / Negotiation* and the *Policy Enforcement / Compliance Testing* modules (boxes 3 and 6 in Figure 2). In Section 5, we describe our implementation of the *Policy Enforcement / Compliance Testing* module.

4.1. Influences and policy creation

Each PPI contributes specific privacy policies or sets of policy rules that describe the privacy-related information handling practices they advocate, either explicitly or implicitly. The challenge is to identify and represent these policies, and determine how they impact the overall privacy policy.

In our implementation, each of the policy rules is expressed in a machine-readable form as well as a plain-text description of what practices should be followed. Each policy rule is assigned a weight representing the importance of the rule itself as well as the importance of the PPI from which it originated, and an originator (*e.g.*, an originator in the legislation PPI might be the name of a privacy law).

To combine the individual contributing sets of policy rules into a managed enterprise privacy policy, E, we begin with n PPIs, $r_1, r_2, \cdots, r_n$, and a set of privacy rules for each PPI: $S_1, S_2, \cdots, S_n$. Consider the union of the sets of privacy rules, $P = S_1 \cup S_2 \cup \cdots \cup S_n$, the set of all privacy policy rules coming from the n PPIs. For each policy rule $p \in P$, we have a vector $(v(p) = \{v_{1p}, v_{2p}, \cdots, v_{np}\})$ where v_{ip} is the weight that PPI r_i has for rule p. The weight is 0 if the PPI does not contain that policy rule. Note that each privacy rule p has a weight and each PPI r_i also has a weight. The values of this vector are determined according to Equation 1. If a policy rule exists in the ith set, the ith value (v_i) is the value of the 'weight' property of that policy rule multiplied by the weight of PPI i; otherwise, the value is 0.

$$v_i = \begin{cases} weight(p) * weight(r_i) & \text{if } p \in S_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We define the *overall weight* of a policy rule p using the L^1-norm of the vector v associated with the rule p, $||v||_1$. This measure combines the weights assigned by different PPIs to determine the relative importance of enforcing the policy rule[2]. The L^1-norm of v is equivalent to the sum of each of the n elements of v ($\sum_{k=1}^{n} v_k$).

After determining the weight of each policy rule in P, the enterprise sets a threshold such that a policy rule p exists in the enterprise privacy policy E if its weight exceeds the minimum threshold. The weights and the threshold must be chosen to ensure that the policy rules from essential PPIs (*e.g.*, legal and contractual obligations) are included.

Policy rules may conflict with one another. These policy conflicts should be detected automatically and resolved either automatically or manually (*e.g.*, through the renegotiation of a contract).

This process must take place for each subset of the enterprise (*e.g.*, a division geographically by country or state, or a division based on business units such as sales, management, or service divisions). An additional set of policies will need to be established to govern the sharing of information between subsets of the enterprise. The resulting collection of policies is the enterprise's privacy policy.

4.2. Privacy Policy Compliance in Software

There is a need to test that new or existing software applications comply with an enterprise privacy policy. The source code for applications may not always be available (such as when the software is purchased from another company), so our approach considers the software application to be a set of black-box sub-components, each having different access and egress points for information input and output. This view is consistent with a service-oriented architecture, where an application consists of loosely-coupled separate components. It also works with other architectures such as client-server.

An enterprise privacy policy governs the processing of personal information. Testing software for compliance with the policy involves tracing the movement of information through the software application and to or from outside entities, a determination of whether or not this information is personal and the degree of sensitivity, and asserting that this behaviour complies with established rules.

In this section we describe our framework for testing compliance of software applications, using the example of an e-commerce application such as an on-line bookstore or clothing store. The sample enterprise is a retailer which collects and uses personal information about its customers.

2. Conceptually, *weight(force) = (relative importance of PPI) + (relative penalty for violating requirements) - (relative cost of implementing requirements)*. The term 'relative' indicates that the weight matters only relative to other policy influences. Hence, the actual values in the formula should be normalized against the values of the other PPIs.

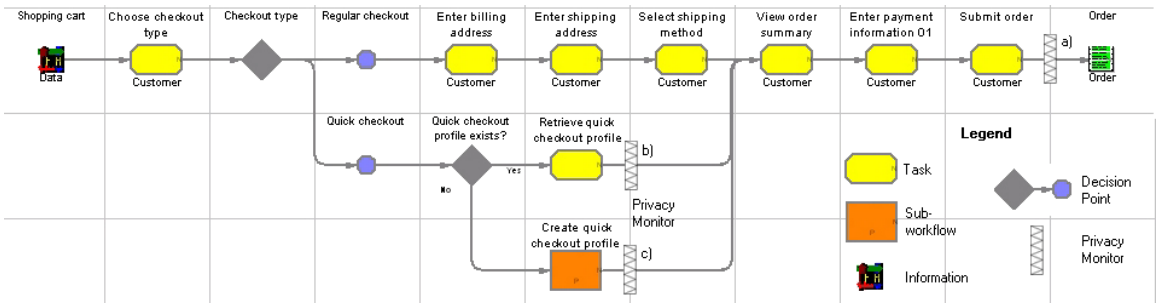

Figure 3. Workflow for creating an order on an e-commerce website, showing the privacy monitor filters.

The movement of information through a software application is dictated by a set of *workflows*. Georgakopoulos *et al.* describe a workflow as "a collection of tasks organized to accomplish some business process... [it] defines the order of task invocation or condition(s) under which tasks must be invoked, task synchronization, and information flow (dataflow)" [19]. These workflows are enabled by the flow of information from one task to the next (an *information flow*). An information flow is a set of data elements that are sent from a *source* to a *destination*. We assume the software application is divided into subsets called *components* that share information. We consider the component within the application that is sending or receiving the information as the source or destination. For example, a user submitting a login form on a website is an information flow; the source is the user and the destination is the application "authentication" component. The "checkout" component passing credit card information to the "payment processing" component is another example.

In our e-commerce example, the workflow for placing an order is shown in Figure 3. The customer can either do a regular checkout or a quick checkout. For the former, they will be prompted for billing and delivery information. For the latter, this information will be loaded from a stored profile (if one does not exist, the customer is asked to create one). In either case, an order summary will be displayed and the customer will be asked for payment information. The order is then submitted and stored in the retailer's database. Each of the arrows indicates that information (or control of the information) is passing from one task to another. Privacy compliance testing focuses on workflows to, from, and within the software application.

Figure 3 shows *privacy monitors* at various points throughout the workflow (labeled a,b,c) that are used to record information flows between sources and destinations. These allow detailed tracing of the information flow in the application; for performance reasons, privacy monitors are established only at manually-identified critical points. Privacy monitor a examines the flow of information submitted by the e-commerce customer to create the order. Privacy monitor b examines the information contained in the quick checkout profile which is retrieved from a database. Privacy monitor c examines the information submitted by the customer to create a quick checkout profile.

Each privacy monitor compares the transmission of the information (including the destination, source, and the presence of personal information) to the enterprise privacy policies. If a potential violation is detected, the privacy monitor may take one or more actions depending on how the privacy compliance testing is being employed. Actions include creating a report for a software tester, halting the transaction, warning the customer, asking for additional consent, or notifying the administrator. Privacy monitors can be applied during the deployment, testing, and/or production phases of software applications. In this paper, we describe an implementation used during development and testing.

5. Proof-of-Concept Implementation

This is an overview of the implementation of the privacy compliance module; a full description of the implementation is available [20]. Our proof-of-concept implementation is designed to work with any J2EE application. The compliance module was tested on a sample store supplied with the widely-used electronic commerce software package, IBM's WebSphere Commerce™.

The implementation of the policy compliance testing module was broken into a number of loosely-coupled, components that communicate using XML documents. Each component can be replaced by another implementation of the same component. Collectively, the components correctly implement our specification of a privacy monitor. In the next sections, we present the functionality of each of the four components: capture information flows, understand information flows, rule evaluation, and report on compliance.

5.1. Capture Information Flows

For each of the communication mediums used by the e-commerce application, we capture and record the information and its source and destination as it is transmitted

between the e-commerce application and other entities (*e.g.*, users of the application, external third-party applications).

We use filters as described by the J2EE specification to monitor the information received and sent by the e-commerce application server (the filters are based on the Sun Intercepting Filter pattern[3]). This provides access to the unencrypted HTTP requests submitted by the e-commerce customer, and the response from the application. A filter can be added to a J2EE application without modifying the source code and without changing the function of the application.

We extract the data elements from different forms of incoming information such as POST, GET, cookie data, or the URL. *Data elements* are name-value pairs where the name is a data descriptor and the value is the data value. Outgoing information includes HTML pages, cookies, and parameters in the URL. Data elements already exist in the cookie and URL as name-value pairs. We extract data elements from the HTML page by examining the JSP template used to generate the HTML page. The template consists of static content and instructions to insert dynamic values; the value is the text inserted into the HTML page and the name is the variable name in the template. Captured data elements, and their sources and destinations, are sent to the next component via an XML document.

5.2. Understand Information Flows

The data descriptor portion of the data element captured in the information flow is assigned a specific name by the application. We "understand" its meaning by mapping it to a pre-defined label. For example, the application-specific data descriptors 'user_real_name', 'firstName', '$u' could map to the pre-defined label 'user.name'. The pre-defined label has a meaning, it is known to be personal information, and it has a level of sensitivity as determined by the enterprise privacy policy. Its meaning can also be understood by 'grouping' the data descriptor with other descriptors. For example, 'age' and 'gender' might be grouped and assigned the label 'user.demographics'. Grouped descriptors have similar properties and identical levels of sensitivity and can be treated the same way. Non-personal information not relevant to the privacy compliance testing can be added to a single group and subsequently ignored.

Inferring meaning and semantics from this type of data is non-trivial. Our approach uses a file which maps known application-specific data descriptors to appropriately understood abstracted groups. Simplified regular expressions are used to make this mapping file more powerful. Each abstracted data label has a `sensitivity` attribute that gives the level of sensitivity of that piece of information on a numeric scale. An additional value for the level of sensitivity is `default`, which indicates that a default sensitivity level was assigned. The default sensitivity level may be configured differently for each application depending on the sensitivity of information generally handled.

Constructing the mapping file is semi-automated. If a data descriptor cannot be located in the existing mapping file, heuristics are used to create a mapping to an existing abstract data category. First, data descriptors are broken into words based on camelCase[4] and by replacing underscores and other separators with spaces. The mapping file is checked to see if the newly separated words appear. If not, the words are checked against the Princeton Wordnet[5], a lexical reference system that organizes words into *synonym sets* based on similar lexical concepts. The mapping file is searched to determine the existence of mappings for other words in the synonym set. For example, a Wordnet lookup of `postal_code` returns: 'ZIP code', 'ZIP', and 'postcode'. Where a word exists in more than one synonym set with mappings, the mapping with the greatest level of sensitivity is chosen. Finally, a set of regular expressions is used to identify potential mappings based on the nature of the data value (for example, 800-555-1234 is a phone number).

When the automated approach identifies a mapping, it is added to the mapping file. This allows for manual correction of incorrect mappings. This process works best when the application being tested follows programming style guides.

5.3. Rule Evaluation

This component identifies flows of information that do not comply with the rules as policy infractions. Each infraction is assigned importance based on the weight of the rule.

A privacy compliance rule is a set of compliance tuples joined by the Boolean operators AND and OR (each of equal precedence). Tuples may be nested using parentheses. A compliance tuple is of the form *(variable, operator, value[s])*. Valid variables are based on information generated in the previous components. Valid operators depend on the variable type and follow the standard conventions for operators in computer programming languages. The '!=' operator means that the variable may not take on any value in the given set; the '==' operator means that the variable must take on one of the values in the given set. Only a single compliance tuple is necessary to make a complete rule. A set of exemplar rules is shown in Table 1.

Each of the compliance tuples resolves to true or false; the value of the resulting boolean expression determines the final result of the rule (true or false). A rule with a final result of 'true' is said to have passed; otherwise, assertion of the rule is said to have failed and the appropriate entry ('warning', 'error', or 'unknown') is made.

3. http://java.sun.com/blueprints/corej2eepatterns/Patterns/InterceptingFilter.html

4. camelCase is a variable name convention that concatenates words together with the first letter of each word in upper-case *e.g.* thisIsAnExample.

5. http://wordnet.princeton.edu/

Original	Sensitive information must be encrypted during transmission
Translated	(Sensitivity,<, 2) OR (Encrypted, ==, 1)
Original	Information in the cookies or the URL must be encrypted during transmission
Translated	(Sensitivity,<, 1) OR (Encrypted, ==, 1) OR ((Source, !=, [Cookie, URL]) AND (Destination, !=, [Cookie, URL]))
Original	The Social Insurance Number or Social Security Number must not be collected
Translated	(Abstracted Data Label, !=, SIN)
Original	No personal information may be collected.
Translated	(Sensitivity,<, 1)
Original	Information may not be sent to any entity other than a delivery agent.
Translated	(Sensitivity,<, 1) OR (Destination, ==, [Delivery, User, Cookie, URL])
Original	Information may not be sent to any third parties.
Translated	(Sensitivity,<, 1) OR (Destination, ==, [User, Cookie, URL])

Table 1. Exemplar rules translated from the legislative rules and a sample privacy policy.

Rule	Sensitive information must be encrypted during transmission
Test	Modify the user registration page to submit using HTTP instead of HTTPs
Rule	Information in the cookies or the URL must be encrypted during transmission
Test	Include sensitive information as parameters in the URL of the request
Rule	The Social Insurance Number or Social Security Number must not be collected
Test	Modify the user registration page to request a social insurance number
Rule	Passwords should not be sent to the user
Test	Modify the user registration page to confirm a user's password by displaying it
Rule	An individual should be the sole source of personal information about himself or herself
Test	Personal information that the user did not submit is sent to the user

Table 2. The rules enforced by this implementation and the tests used to verify violations can be detected.

5.4. Report on Compliance

Rather than dynamically attempting to fix infractions, our proof-of-concept implementation generates a report on the compliance of the software application. The outcome of the information flows captured and the rules-based analysis in the Rule Evaluation step (Section 5.3) are stored in our XML-based format. Readily customizable reports are generated from the XML to a form that varies depending on who requested the report (*e.g.*, programmers assigned to resolve compliance errors will require more detail than a manager required to review testing progress). The output includes information such as which module of the software application may be non-compliant and why, which data elements were transmitted by non-compliant information flows, the rule that was violated, how the rule was violated, and the original source of the rule.

5.5. Evaluation

The online retail store under test was modified to introduce a number of previously fixed customer-reported bugs involving inappropriate disclosure of personal information (specifically in flows between the user and the e-commerce application). The modifications made are listed in Table 2. Using the rules generated from the enterprise privacy policy, the privacy policy compliance testing module was able to detect these potential privacy infractions and report them to the administrator.

In this simple test, the implementation detected 100% of the non-compliant information flows. A false detection of non-compliance was generated when the form requested personal information (social insurance number) but the user entered non-personal information (specifically, the string "will not disclose"). The abstraction component classifies data elements based on what the application requested, not what the user provided. These results are promising, but this set of rules is not large enough to obtain an accurate false-positive/false-negative rate.

6. Conclusions and Future Work

This paper presented an enterprise privacy policy management framework, including modules responsible for creating, deploying, enforcing, and testing for compliance with an enterprise privacy policy. The framework is readily extensible; it was designed to be incrementally developed as the details of the privacy challenge emerge and as the challenge evolves over time. This framework co-exists with existing software applications; it is designed as a layer between the middleware and the software applications (an extension of middleware that does not require the modification of existing software applications). To demonstrate the feasibility of our proposed framework, we defined and implemented two of the modules, privacy policy creation and privacy policy compliance testing. A proof-of-concept implementation of privacy compliance testing was developed and tested in a commercial electronic commerce software application.

The current implementation is readily extensible to capture additional information flows, to use additional mappings when abstracting data, to compare additional rules, and to

generate different reporting views. The component-based architecture with defined Java interfaces dictating the behavior of components allows for existing components to be extended or replaced with alternate implementations, without the need to modify the entire implementation.

Future work includes further implementing our framework, including more development on implemented modules, implementing other modules, or integrating existing software into our framework. We would like to explicitly address the issue of risk management in privacy compliance testing. Also, automatically generating machine-readable privacy policies from plain-text sources such as legislation would be a useful addition to policy creation. Finally, we would like to research ways to automatically take action based on detected privacy infractions.

We have demonstrated that we can test a software application for compliance with a defined set of privacy policy rules. Though not a complete solution, this initial result demonstrates that privacy management in technology is not an impossible task. With appropriate legislation, demand from individuals, and support from businesses, technology can help address the issue of privacy.

Acknowledgements

The authors sincerely thank Jen Hawkins, Darshanand Khusial, and the IBM Toronto Lab WebSphere Commerce team for their help in formulating and refining our approach. This research was funded by the IBM Centers for Advanced Studies (Toronto), with additional funding from NSERC and Precarn.

References

[1] L. Brandeis and S. Warren, "The right to privacy," *Harvard Law Review*, vol. IV, no. 5, pp. 193–220, 1890.

[2] M. J. Maullo and S. B. Calo, "Policy management: An architecture and approach," in *Proceedings of the IEEE First International Workshop on Systems Management*, Los Angeles, CA, April 1993.

[3] IBM Corporation, "IBM Tivoli Privacy Manager for e-business," last visited May 2008, http://www-306.ibm.com/software/tivoli/products/privacy-mgr-e-bus/.

[4] R. J. Ziegler, "Business policies and decision making," in *Meredith Publishing Company*, New York, N.Y., 1966.

[5] Treasury Board of Canada Secretariat, "Privacy impact assessment guidelines: A framework to manage privacy risks," 2002, http://www.tbs-sct.gc.ca/pubs_pol/ciopubs/pia-pefr/paipg-pefrld_e.asp.

[6] M. Howard and D. LeBlanc, *Writing Secure Code*. Microsoft Press, 2002.

[7] F. Swiderski and W. Snyder, *Threat Modeling*. Microsoft Press, 2004.

[8] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay, "Privacy risk models for designing privacy-sensitive ubiquitous computing systems," in *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques*. New York, NY: ACM Press, August 2004.

[9] "*Personal Information Protection and Electronic Documents Act* (Canada)," Second Session, Thirty-sixth Parliament, 48-49 Elizabeth II, 1999-2000. Assented to April 2000.

[10] Canadian Standards Association, "Model code for the protection of personal information," March 1996.

[11] "*Children's Online Privacy Protection Act* (United States of America)," 15 U.S.C. § 6501; P.L. 105-277, 1998. Approved 1998.

[12] European Parliament, "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," October 1995.

[13] Harris Interactive, "Most people are "privacy pragmatists" who, while concerned about privacy, will sometimes trade it off for other benefits," 2003, http://www.harrisinteractive.com/harris_poll/index.asp?PID=365.

[14] M. J. Culnan and P. K. Armstrong, "Information privacy concerns, procedural fairness, and impersonal trust: An empirical investigation," *Organization Science*, vol. 10, no. 1, pp. 104–115, 1999.

[15] B. Givens, "Identity theft: How it happens, its impact on victims, and legislative solutions," Privacy Rights Clearinghouse, Written Testimony for U.S. Senate Judiciary Subcommittee on Technology, Terrorism, and Government Information, last updated July 2000, http://www.privacyrights.org/ar/id_theft.htm.

[16] Roy Morgan Research, "Community attitudes to privacy," Office of the Federal Privacy Commissioner, Australia, 2001, http://www.privacy.gov.au/publications/rcommunity.html.

[17] R. C. Angell, "Preferences for moral norms in three problem areas," *The American Journal of Sociology*, vol. 67, no. 6, pp. 650–660, May 1962.

[18] TRUSTe, "Consumers have false sense of security about online privacy actions inconsistent with attitudes," 2006.

[19] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and Parallel Databases*, vol. 3, no. 2, pp. 119–153, 1995.

[20] M. Smit, *Privacy in e-Commerce Software*. VDM Verlag Dr. Mueller e.K., 2007.