W3C Workshop on the long term Future of P3P

Workshop Position Paper: Shortcomings of P3P for Privacy Authorization - Lessons learned when Using P3P-based Privacy Manager 1.1

By Paul Ashley, S. Hada, Günter Karjoth, M. Schunter

Introduction

Privacy is the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others[1]. When an individual gives their private data (PII[2]) to an enterprise, the enterprise should consider itself the custodian of the data, and let the individual as the data owner decide how it should be used. There are many types of private data, for example medical records, home address, home phone number, email address, web site usage patterns, and shopping patterns. All of this PII should be treated with absolute care by the enterprise that collects it.

As a first step towards managing privacy effectively organizations publish privacy promises as text or P3P[3]. The text policy can be read by an individual and usually contains legal language. The P3P statements can be used by a P3P client (e.g. the Internet Explorer 6 web browser) to notify the user automatically whether the privacy policy of the enterprise matches that configured by the user. The idea of the text and P3P policy is that the individual has an unambiguous statement of how the enterprise handles PII data.

Alongside the privacy policy should be a set of user preference options for use of the individuals PII. So when an enterprise accepts PII data from an individual, it should have BOTH clear confirmation of acceptance of the privacy policy, as well as a recording of the individual's own preferences for use of the data. These preference options which include opt-in and opt-out choices should give the user full control over what purposes the data is used. For example the privacy policy may state:

> *When an individual consents, the individual's home address is provided to our trusted partners for use in updating the individual with new product releases from our partners.*

This will then be accompanied by an opt-in or opt-out choice to allow the individual to make the choice whether they want their address to be provided to the enterprise's partners. For example, the user may be provided with a selection on a web page:

---

[1] Alan Westin

[2] Information is considered PII or Personally Identifiable Information if it can be linked to a person. Information that has been de-identified or anonymized would not be considered PII – unless there are ways to linking it back to the person through re-identification or inference.

[3] The Platform for Privacy Preferences (P3P), W3C Recommendation, 16 April 2002, http://www.w3.org/p3p

☐ *Mark the box if we can send your home address to our trusted partners.*

The processes of providing the user with the privacy policy, and collecting their acceptance of the policy and their preferences for use of PII, is commonly called **Notice** and **Consent**. An enterprise should NOT collect PII without having both notice and consent implemented.

Many enterprises are currently underway creating the processes or have completed the processes for providing notice and consent. A quick look around the web for instance will find most major company web sites with a privacy policy (including in P3P) and facilities for collecting policy acceptance and preferences.

However, many enterprises are finding that this is just the tip of the privacy iceberg. Just because enterprises have advertised their privacy promises, and collected user consent and preferences, it doesn't mean they are providing good privacy protection to that data. Enterprises are finding that they do not have the privacy technology to **Enforce** the promises throughout the enterprise, and **Audit** the accesses to the PII data. This has resulted in privacy violations being a common occurrence today, even from well meaning companies.

What are the choices for implementing Privacy Policy Enforcement and Auditing?

Unfortunately up until quite recently there have not been any software tools available for privacy policy enforcement and auditing. Enterprises have had two choices really:

1. Do nothing and pray that they don't violate too many regulations and they don't annoy too many of their customers.
2. Try to implement their privacy policy across their application environment. This usually means coding privacy policy into applications. This causes a number of problems:
    a. The cost of coding privacy policy into applications and the maintenance of this quickly becomes prohibitive.
    b. The time to change to a new policy is far too large. Each of the applications has to be modified every time a policy change is required. This cannot be done quickly.
    c. An enterprise is never sure that there is a consistent implementation of the privacy policy within all applications. What if some applications are still running with an old policy?
    d. An enterprise has no transparency to the policy in place. How does the CPO or her policy team know that an application has implemented the correct policy? Does she read the source code?

What is Privacy Manager?

To address the problems faced by organizations in enforcing their privacy policy, Tivoli development has been underway for the last few years for a new product called IBM Tivoli Privacy Manager for e-business (Privacy Manager). This product was released in 2002.

The value of Privacy Manager is in separating the privacy policy from applications. Privacy Manager allows the CPO or her staff to enter the privacy policy at a high level, and through the use of Privacy Manager monitors, have this enforced across the application environment.

Some of the main features of Privacy Manager are:

1. The ability to track different versions of privacy policy. This is very important so the enterprise can keep an historic log of when the privacy policy was changed.
2. Can store consent of the individual to the privacy policy when PII data is collected. Without this consent Privacy Manager can be configured not to release the individual's data for any purpose.
3. Auditing is a core feature. All submissions and accesses to PII are stored in the Privacy Manager database. A comprehensive reporting tool allows for various reports to be created based on this data. For example, a report can show all accesses to an individual's PII if that individual requests it.
4. Authorization of submissions and accesses to PII. Privacy Manager can provide an authorization decision about whether the data accessor or submitter is allowed to do so based on the privacy policy and user consent and preferences.

What are the Privacy Manager components?

Privacy Manager consists of two components:

A **Privacy Manager Server**. This server has a number of roles:

1. To define the Privacy Policy
2. To map the policy to IT resources
3. To create the audit trail
4. To provide the reporting tools.

Privacy Manager Monitors. These are the integration points between the Privacy Manager Server and the application environment. They have a number of roles:

1. To learn and understand the data schema of the storage system to be monitored.
2. To register details of the storage system with Privacy Manager.
3. To intercept submission and access activity to the storage and report this to Privacy Manager for auditing.
4. To enforce applicable privacy policy following a request of Privacy Manager for an access conformance check.

5. To supply Privacy Manager with values from the monitored storage system related to conditions attached to policy rules.

Practical Experiences with Using P3P for an Authorization Language

Privacy Manager uses P3P as its privacy policy language. P3P was designed as a privacy policy declaration language and its use as an enforcement language is unique. We have worked with a number of customers with Privacy Manager and this has allowed us to validate the use of P3P as a privacy authorization language.

Why is a Privacy Policy Different to an Access Control Policy?

Our background in Tivoli Security involves Access Management. We have a product IBM Tivoli Access Manager that provides access management to the web, operating systems, messaging systems and other application environments.

We have found that Privacy Policies are more complex than typical access control policies. The main differences are:

1. Privacy policies use the **Purpose** for use of the data in making access decisions.
2. Privacy policies list **Data-Categories** in policy statements and not individual resources.
3. Privacy policies may check the user **Consent** before allowing access. So even if the privacy policy statements allow access, if a user consent to the policy is not recorded, the data will not be released.
4. Privacy policy **Conditions** need to be more flexible. For example, the privacy policy may state that the system needs to check the user's opt-in or opt-out choice. Or it may need to check the individual's age or other information.

Privacy policy statements in Privacy Manager are based on the P3P standard and are of the form:

> ALLOW USERS to USE PII_TYPES for PURPOSES [if CONDITIONS] [if CONSENT][4]

e.g

> Allow General_Practioners to use medical_records for diagnosis [if General_Practioner treated patient] and [if patient opt-in]
>
> Allow General_Practioners to use medical_records for diagnosis [if General_Practioner is patient's Primary_Care_Physician]
>
> Allow General_Practioners to use medical_records for emergency

[4] Recording and enforcing consent is a Privacy Manager function and not part of P3P. In Privacy Manager this consent enforcement is decided at the policy level and is not actually part of a statement.

Note that the statement refers to data categories (PII_Types) and not individual resources.

The privacy policy is intentionally left at a high level so that it can be created by a Company Chief Privacy Officer (CPO) or their staff in a policy creation role. These people do not need to understand the underlying IT infrastructure. This is left to the IT staff who integrate Privacy Manager into the application environment and perform mapping between high level statements and resources.

What were the shortcomings in using P3P as an Authorization Language in Practice?

We have found a number of short comings in P3P working with customers:

1. The use of pre-defined types
2. The only action is USE
3. No obligations
4. No disallow rule
5. Limited Conditions

The use of pre-defined types

P3P pre-defines a set of types. This makes sense when it is used as a declaration language for interoperability, but as an authorization language it is not as useful.

For example, for PURPOSE P3P defines a number of standard types: current, admin, develop, tailoring etc.

In a customer environment we have found that these are not useful. Customers want to define their own PURPOSES and other values based on the operating environment they are working. For example, for one of our health care customers the following purposes were useful:

```
medical_diagnosis
blood_research
statistical_analysis
billing
```

Not being able to use the pre-defined types does not just apply to PURPOSE but all pre-defined types in P3P. Customers in each case wanted to define their own. Our console provided the P3P types for the customers to use but in all cases customers avoided using them.

The only action is USE

P3P does not allow a set of actions on data. Our customers want to write policies for different actions on data. For example,

read
write
delete

are actions that we have required with our customers. However, because we were using P3P we were not able to define a policy with these different actions.

No obligation

P3P does not allow the use of an obligation in a policy. This has meant that we have not been able to implement some of the current policies with our customers. For example, our health care customers wanted to write a policy statement of the form:

> ALLOW general_practioners to READ medical_records if {some conditions} with obligation {if patient is of VIP category flag alert}

Another example is:

> ALLOW sales to WRITE customer_data if {conditions} with obligation {if customer < 18 then get parent approval or delete data within 7 days}

We were unable to implement these policies with our customers.

No disallow rule

P3P does not have disallow rules. This has meant that we have had to create policies much more complicated than necessary.

An example might be where we have a set of groups in a hierarchy:

engineering	e_assistants
	e_managers
	e_contractors
	e_architects
	e_administrative

A customer required a set of rules:

```
ALLOW  engineering to READ customer_engineering_data
DISALLOW e_contractors to READ customer_engineering_data
```

Not having an ALLOW rule means that this would have to be rewritten as

```
ALLOW  e_assistants to READ customer_engineering_data
ALLOW  e_managers to READ customer_engineering_data
ALLOW  e_architects to READ customer_engineering_data
ALLOW  e_architects to READ customer_engineering_data
```

The policies that we created for our customers were not as efficient concise as they could have been with a DISALLOW statement.

Limited Conditions

We have found with our customers that we really need a generalized condition language to express the kind of policies that are required in practice.

Conclusion

P3P is well-suited for formalizing privacy promises that are communicated to end-user. In our practical experiences, privacy statements like the following has to be formalized:

> Blood Disorder Researcher may access all patient medical information for the purpose of medical research if the Blood Disorder Researcher is from the same post code as the patient, and the patient has consented and the patient's General Practioner has consented and the treatment was a blood test and the purpose of the blood treatment was not STD related.

It turned out that many of the policy statements from our customers required conditions to be evaluated. So we need a generalized condition language that can assess boolean type rules as conditions on the base statements. As a consequence, we feel that P3P is too coarse-grained and lacks some features for enterprise-internal privacy enforcement. We believe that the requirements for a privacy declaration language and privacy authorization language are different. As a consequence, we feel that there should be a P3P-compatible language to actually enforce the P3P promises made.