

Semantic Web Policies for Security, Trust Management and Privacy in Social Networks

Juri L. De Coi¹, Philipp Kärger¹, Daniel Olmedilla², and Sergej Zerr¹

¹ L3S Research Center & Leibniz University of Hannover, Germany

² Telefónica Research & Development, Madrid, Spain

Abstract. The ability of defining privacy preferences in the current social platforms is very restricted. Typically, the user is provided with only some predefined options to select from. Semantic Web policies can be exploited in order to allow users to control privacy in social web applications. Such policies are generally considered statements that define the behavior of a system.

However, although Semantic Web policies gained a lot of interest in recent years and policy languages became more and more complex, suitable and easy-to-use solutions for highly dynamic social platforms are still needed. This paper presents a Semantic Web policy framework that not only allows for a fine-grained policy language and access and privacy control, but which also addresses its easy specification by non-computer experts and which explains both the policies and the decisions made when reasoning over them.

1 Introduction

Open distributed environments such as the World Wide Web offer easy sharing of information, but provide few options for the protection of sensitive information and other sensitive resources. Since the Web became a place where people are not only consuming but creating, publishing and sharing content, it is needed to allow people to exactly define who is allowed to access which part of the content they provide. Unfortunately, there is currently no simple way to restrict access to some content to only a set of trusted parties not previously known or to decide who is allowed to access some part of a profile [1]. In Flickr for instance, a user is allowed to define other users as friends or family members and, based on those assignments, it can be decided who is allowed to see what picture. Still more fine-grained access control management is lacking. Thus, if both, colleagues and close friends, are in the same group, there is no way to restrict access to a subgroup for a particular picture [1]. Other web applications for sharing data or social platforms do also allow only for similar limited access control features leading to a high potential of leaking private data [2].

However, the protection of sensitive information and other sensitive resources plays a crucial role in raising the level of trust in web resources and hence in enabling the potential of the Web. For example, even though latest developments such as Web 2.0 have demonstrated that many users are willing to participate and therefore share information publicly, recent experiences with Facebook's "beacon" service [3] and Virgin's use of Flickr pictures [4] have also shown that users are not willing to accept every possible use (or abuse) of their data. Therefore, the application of suitable policies for protecting services and sensitive data may determine success or failure of a new service.

Semantic Web policies have gained a lot of interest in the last years in order to fill, among others, this gap, and promising approaches show their applicability to real-life applications, be they in the area of web personalization, agent and network control, or access control on the Web. For this reason a number of policy languages have been defined in the last years such as Cassandra [5], EPAL [6, 7], KAoS [8], PeerTrust [9], Ponder [10], PSPL [11], Rei [12], *RT* [13], TPL [14], PROTUNE [15], WSPL [16] and XACML [17, 18]. Unfortunately, a major hindrance to widespread adoption of policy languages are their shortcomings in terms of understandability and usability: in order to be machine-understandable all of them rely on a formal syntax, which common users find unintuitive and hard to grasp. Therefore, in most such applications, policies are required to be authored and handled by system administrators or specifically trained people, and common (non-computer expert) users are not able to understand and personalise them.

In this paper, we present the PROvisional TRust NEgotiation framework PROTUNE [15] which extends two previous languages: PAPL [11] and PEERTRUST [9]. As its predecessors, PROTUNE supports a variety of evidences—strong evidences like digital credentials and weak evidences such as declarations—but, it can be extended by user-defined actions. Additionally, there are two main features that will be briefly described, and which aim to increase user awareness and to ease policy authoring by non-computer experts: natural language policy specification and explanations.

The former allows users to exactly define their policies, that is, with whom they want to share their data, by writing English sentences (in a controlled natural language). Such natural language sentences are then mapped into the formal PROTUNE syntax, which can then be used during the policy evaluation process.

On the other hand, explanations address the problem of a user not understanding the implications of a policy. In order to make policies effective, a first prerequisite is that users must be able to understand them. However, producing accurate documentation is difficult and expensive, as well as keeping documentation aligned with the current policy. Furthermore, the documentation needs to be contextualized: the decisions taken during an access request must be explained w.r.t. that particular request. For instance, receiving a simple “Authorization denied” is generally not satisfactory, and may affect the image and revenues of the service provider. Our approach automates the documentation with a second generation explanation facility (PROTUNEX) which produces controlled natural language explanations [19]. PROTUNEX exploits a general algorithm to generate explanations, and makes use of annotations to improve readability by indicating how to render specific parts of the policy.

2 Motivation

Preserving privacy in social platforms is a known challenge (see [3] for an information leakage on Facebook or [4] for a similar case on Flickr). Existing social platforms provide only limited means to define access control rights [20]. The problem is that privacy preferences have to be easy to define but still expressive enough to capture all possible cases and combinations of a user’s wishes to define who is allowed to access what. An additional challenge is the dynamics of nowadays social networks. We identified the following requirements for a solution aiming at preserving privacy on a social platform.

Adaptive to social network dynamics. Since nowadays social networks change rapidly, any solution should be able to easily adapt itself to changes in the data of requesters.

Fine-grained. Privacy preferences can be arbitrary fine-grained. If a picture shall be shared only with a restricted set of people (maybe not even known in advance), it should be easy to express such requirement.

Extensible. Privacy preferences should be easily extensible: imagine a user uploads a new set of pictures to a social platform and she wants them to be shared with attendees of a conference. The user's current privacy preferences should be easily extensible without requiring all conference attendees to be registered at the social platform.

Natural language interface and feedback. Defining privacy preferences has to remain a simple and straightforward process. Access control decisions should be transparent and well explained to users. Similarly, the specification of privacy preferences has to protect users from collections of check boxes defining which friends is allowed to access which file or from similar complicated policy definitions.

Security mechanisms. Last, but not least, any solution must fulfill basic security and privacy requirements, such as reliability, support to authentication, delegation of rights and evidences (such as credentials and declarations), etc.

3 The Policy Framework Protune

In our approach we used the PROTUNE³ (PROvisional TrUst NEgotiation) framework for policy evaluation and enforcement. PROTUNE [15] aims at combining distributed trust management policies with provisional-style business rules and access control-related actions. PROTUNE features an advanced policy language for policy-driven negotiation and supports distributed credentials management and flexible policy protection mechanisms. The PROTUNE policy language is Logic Programming-based and as such a PROTUNE policy has much in common with a Logic Program. As a quick overview, PROTUNE provides a framework with the following features:

- A trust management language supporting general provisional-style⁴ actions (possibly user-defined).
- An extensible declarative meta-language for driving decisions about request formulation, information disclosure, and distributed credential collection.
- A parametrized negotiation procedure, that gives a semantics to the meta-language and provably satisfies some desirable properties for all possible meta-policies.
- General, ontology-based techniques for importing and exporting meta-policies and for smoothly integrating language extensions.
- Advanced policy explanations in order to answer why, why-not, how-to, and what-if queries [19].

A live demo of PROTUNE in a Web scenario is publicly available⁵ as well as a screencast⁶.

³ <http://policy.L3S.uni-hannover.de/>

⁴ Authorizations involving actions and side effects are sometimes called provisional.

⁵ <http://policy.l3s.uni-hannover.de/>

⁶ <http://www.viddler.com/olmedilla/videos/1/>. Recommended in full screen.

4 Controlled natural language for easy specification of privacy policies

As introduced previously, a major hindrance to widespread adoption of policy languages are their shortcomings in terms of usability: in order to be machine-understandable all of them rely on a formal syntax, which common users find unintuitive and hard to grasp.

Controlled natural languages (CNLs) are formal languages which have been designed in order not to look like formal languages but to be more user-friendly. As formal languages they are described by a formal grammar each well-formed sentence must comply to. Moreover, they embed disambiguation rules in order to deterministically disambiguate sentences which in full natural language have different readings. On the other hand, any controlled natural language sentence is also a natural language sentence what makes them easy to create, understand and modify by common users.

The use of controlled natural languages can improve usability of policy languages. It is possible to use such a controlled natural language (we used a subset of ACE [21]) in order to express policies and describe the mapping between such policies and the policies written in a formal policy language such as PROTUNE.

This way, formal PROTUNE policies do not need to be specified following the logic-programming based syntax, but could simply be written as

- If the requester is a friend and the resource is a family-picture then the requester can access the resource.
- If the company of a credit-card is "VISA" then the credit-card is accepted.
- If the requester is older than 18 and she is Bob's friend then she can access everything which is in "adult-content-folder".

These examples show which features of (ACE and therefore of) the English language can be used in order to define PROTUNE policies: common nouns (like `requester`), adjectives (like `accepted`), verbs (like `access`), genitives (like `in company of a credit-card`) and strings (like "VISA"). Additionally, proper names (e.g., Bob), adjectives in comparative (e.g., `older than`) as well as in superlative form, integers (e.g., `18`) as well as reals, prepositional phrases (e.g., `in "adult-content-folder"`), saxon genitives (e.g., `Bob's`) and relative pronouns (e.g., `everything which is`) are supported too. The full details of the ACE→PROTUNE translation are available in [22].

5 Protune Policy Explanations

Even with a policy with relatively few rules it could be hard for a common user —with neither a general training in Computer Science nor a specific knowledge of mechanisms and formats of the system — to understand what is actually required to access a certain service.

PROTUNEX, the explanation facility of PROTUNE [19], plays an essential role in improving policy understanding and *cooperative enforcement*: the explanation system enriches the denials with information about how to obtain the permissions (if possible) for the requested service or resource.

PROTUNEX provides explanations in natural language based on verbalization metarules and supports four kinds of queries: *How-to* queries provide a description of a policy and may

help a user in identifying the prerequisites needed for fulfilling the policy. How-to queries may also be used to verify a complex policy. *What-if* queries are meant to help users *foresee* the results of a hypothetical situation, which may be useful for validating a policy before its deployment. Finally, *why* and *why-not* queries explain the outcome of a concrete negotiation (i.e., provide a *context-specific* help). Why/why-not queries can be used both by end users who want to understand an unexpected response, and by policy administrators who want to diagnose a policy.

A full demo of PROTUNEX is publicly available⁷

6 Policy Enforcement for Preserving Privacy in Social Platforms

Using policies and the PROTUNE policy framework for access control allows the user for a fine-grained specification of his privacy preferences in a declarative manner. This section summarizes how enforcing those policies preserves privacy in social data sharing platforms.

Since policies are declarative statements and by using a current state-of-the-art policy engine, we allow advanced control of data sharing. We will now shortly name those features [23]—instantiated as the features of PROTUNE—and describe their benefits for social software:

Negotiations: In traditional access control or authorization scenarios only one party is able to specify policies which the other one has to conform to. Typically only one of the interacting parties is enabled to specify the requirements the other has to fulfill, whereas the other has no other choice but satisfying them (and thereby being authorized) or not (and thereby not being authorized). Therefore, a more expressive approach allows both parties to discuss (i.e., negotiate) in order to reach an agreement. Trust Negotiations [24], for example, allow both parties to incrementally disclose information in order to get or grant access to some data on a social platform.

Evaluation and Actions: In order to evaluate a policy, some actions performed by the policy infrastructure might be required. Examples for such actions are sending credentials or queries to external systems, for example in order to decide if the requester is a registered friend on some other social platform. In this case, the PROTUNE policy enforcement infrastructure can automatically query this platform to get the required information. Another common action is the retrieval of environmental properties like the current system time (e.g., if access is allowed only in a specific time frame) or location (e.g., share files only with people in the same meeting room). By doing this, privacy control in social platforms becomes extensible and information from other accessible social platforms can be exploited for security decisions.

Explanations: As stated before, with PROTUNE it is possible to generate explanations [19] out of the policies and the decisions they make. On the one hand, this helps a user to check whether the policies she created are correct and, on the other hand, they inform other users about why a decision was made (or how the users can change the decision by performing a specific action). For example, whenever access to a certain data or resource is denied on a data sharing platform, an explanation informing about why the request failed can be provided. Together with the authoring of policies in CNL, such natural language explanations encapsulate the formal policy evaluation completely from the user.

⁷ <http://cs.na.infn.it/reverse/demos/protune-x/demo-protune-x.html>.

Strong/lightweight evidences: The result of a policy’s evaluation may depend on the identity or other properties of a requester such as age, membership in a certain club, etc. PROTUNE provide a means to communicate such properties.

Ontologies: In trust negotiations policies have to be exchanged among entities within the social platform in order to transfer information about what is needed to fulfill a policy. Although the basic constructs may be defined in the policy language (e.g., rule structure and semantics), policies may be used in different applications and even define new concepts. The ontology support of PROTUNE helps to provide well-defined semantics for new concepts to be understood by different entities[25, 26].

All these features of our approach meet the requirements identified in Section 2 as follows. Using the PROTUNE language makes our privacy solution adaptive and fine-grained. PROTUNE’s ability to add external data sources allows to extend our solution towards data sources that lay outside of the actual social platform. The natural language policy definition and PROTUNE’s explanation facility makes our approach talk natural language to the user and vice versa. Last but not least, our solution comprises the common security features. The logic programming nature of the policy language behind makes the privacy enforcement reliable.

References

1. Passant, A., Kärger, P., Hausenblas, M., Olmedilla, D., Polleres, A., Decker, S.: Enabling trust and privacy on the social web. In: W3C Workshop on the Future of Social Networking, Barcelona, Spain (January 2009)
2. Gross, R., Acquisti, A., Heinz, III, H.J.: Information revelation and privacy in online social networks. In: WPES ’05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, New York, NY, USA, ACM (2005) 71–80
3. Nakashima, E.: Feeling betrayed: Facebook users force site to honor their privacy. The Washington Post (November 30, 2007)
4. AP with Asher Moses: Virgin sued for using teen’s photo. The Sydney Morning Herald (September 21, 2007)
5. Becker, M.Y., Sewell, P.: Cassandra: Distributed access control policies with tunable expressiveness. In: 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), Yorktown Heights, NY, USA, IEEE Computer Society (June 2004) 159–168
6. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise privacy authorization language (epal 1.2). Technical report, IBM (November 2003)
7. Backes, M., Karjoth, G., Bagga, W., Schunter, M.: Efficient comparison of enterprise privacy policies. In: Proceedings of the 2004 ACM symposium on Applied computing, ACM Press (2004) 375–382
8. Uszok, A., Bradshaw, J.M., Jeffers, R., Suri, N., Hayes, P.J., Breedy, M.R., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: Chaos policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In: 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), Lake Como, Italy, IEEE Computer Society (June 2003) 93–96
9. Gavriiloae, R., Nejd, W., Olmedilla, D., Seamons, K.E., Winslett, M.: No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In: 1st European Semantic Web Symposium (ESWS 2004). Volume 3053 of Lecture Notes in Computer Science., Heraklion, Crete, Greece, Springer (may 2004) 342–356

10. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: 2nd IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), Springer (February 2001) 18–38
11. Bonatti, P., Samarati, P.: Regulating Service Access and Information Release on the Web. In: Conference on Computer and Communications Security (CCS'00), Athens (November 2000)
12. Kagal, L., Finin, T.W., Joshi, A.: A policy language for a pervasive computing environment. In: 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), Lake Como, Italy, IEEE Computer Society (June 2003) 63–
13. Li, N., Mitchell, J.C.: Rt: A role-based trust-management framework. In: Third DARPA Information Survivability Conference and Exposition (DISCEX III), IEEE Computer Society (April 2003)
14. Herzberg, A., Mass, Y., Michaeli, J., Ravid, Y., Naor, D.: Access control meets public key infrastructure, or: Assigning roles to strangers. In: 2000 IEEE Symposium on Security and Privacy, IEEE Computer Society (May 2000) 2–14
15. Bonatti, P.A., Olmedilla, D.: Driving and monitoring provisional trust negotiation with metapolicies. In: 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005), Stockholm, Sweden, IEEE Computer Society (jun 2005) 14–23
16. Anderson, A.H.: An introduction to the web services policy language (wspl). In: 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), IEEE Computer Society (June 2004) 189–192
17. Lorch, M., Proctor, S., Lepro, R., Kafura, D., Shah, S.: First experiences using xacml for access control in distributed systems. In: Proceedings of the 2003 ACM workshop on XML security, ACM Press (2003) 25–37
18. Simon Godik, T.M.: Oasis extensible access control markup language (xacml) version 1.0. Technical report, OASIS (February 2003)
19. Bonatti, P.A., Olmedilla, D., Peer, J.: Advanced policy explanations on the web. In: 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, IOS Press (Aug-Sep 2006) 200–204
20. Chew, M., Balfanz, D., Laurie, B.: (under)mining privacy in social networks. In: Web 2.0 Security and Privacy (in conjunction with IEEE Symp.on Security and Privacy). (2008)
21. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In Baroglio, C., Bonatti, P.A., Maluszyński, J., Marchiori, M., Polleres, A., Schaffert, S., eds.: Reasoning Web, Fourth International Summer School 2008. Number 5224 in Lecture Notes in Computer Science, Springer (2008) 104–124
22. De Coi, J.L.: A possible ACE → Protune mapping. Technical report, Forschungszentrum L3S, Appelstr. 9a, 30167 Hannover (D) (July 2008)
23. De Coi, J.L., Kärger, P., Koesling, A.W., Olmedilla, D.: Control your elearning environment: Exploiting policies in an open infrastructure for lifelong learning. *IEEE Transactions on Learning Technologies* **1**(1) (2008)
24. Winslett, M.: An introduction to trust negotiation. In: *iTrust*. (2003) 275–283
25. Leithead, T., Nejd, W., Olmedilla, D., Seamons, K.E., Winslett, M., Yu, T., Zhang, C.C.: How to exploit ontologies for trust negotiation. In: *ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*. CEUR Workshop Proceedings (2004)
26. Nejd, W., Olmedilla, D., Winslett, M., Zhang, C.C.: Ontology-based policy specification and management. In: *ESWC*, Heraklion, Greece, Springer (2005)