Authorisation in context: incorporating context-sensitivity into an access control framework.

FAILY, S., LYLE, J., FLÉCHAIS, I., ATZENI, A., CAMERONI, C., MYRHAUG, H., GÖKER, A. and KLEINFELD, R.

2014

Authorisation in Context: Incorporating Context-Sensitivity into an Access Control Framework

Shamal Faily
Bournemouth University
sfaily@bournemouth.ac.uk

John Lyle, Ivan Fléchais
University of Oxford
firstname.lastname@cs.ox.ac.uk

Andrea Atzeni, Cesare Cameroni
Politecnico di Torino
firstname.lastname@polito.it

Hans Myrhaug
AmbieSense Ltd
hans@ambiesense.com

Ayşe Göker
AmbieSense Ltd / City University London
ayse@ambiesense.com

Robert Kleinfeld
Fraunhofer FOKUS
robert.kleinfeld@fokus.fraunhofer.de

With sensitive information about ourselves now distributed across personal devices, people need to make access control decisions for different contexts of use. However, despite advances in improving the usability of access control for both developers and users, we still lack insights about how the intentions behind policy decisions in different contexts of use are shaped. In this paper, we describe how *context* was incorporated into an access control framework using a study of how context influences access control decision making. We describe how the main recommendations arising from this study were used to build context into a policy editor for this access control framework.

HCI-Security, Access Control, Policy, Affinity Diagram

1. INTRODUCTION

Personal information is no longer locked down in specific locations, where access control can be succinctly described. Instead, information about our habits or preferences is now dissipated across a variety of devices, such as mobile phones, cars, and smart TVs. Because our relationship with these technologies is still inchoate, many unanswered questions remain about how developers should build end-user access control management tools for these device federations, especially given the variety of physical and social contexts in which these collections of devices are used.

Improving the usability of policy authoring tools for both developers and users has been a popular line of HCI-Security research in recent years. Although past work has cast light on some of the challenges in using and developing policy management tools, reflecting on the pervasiveness of mobile and web applications uncovers further challenges. For example, consider the scenario below:

Alice is on her way to meet Justin, an old school friend. Alice's directions to Justin's house are confusing; to avoid getting lost she requests Justin's GPS location, and her in-car navigation computer to picks the best route. Justin receives the permission request and, seeing that Alice is running late, authorises Alice to view his location. A few days later, Alice is visiting friends in London and again uses a navigation app to find the best route to their proposed meeting place. However, because Justin's location was the last she used, the application starts to route towards him instead. By coincidence, Justin is also in London, and Alice follows the seemingly-plausible directions. Alice arrives at her destination and is shocked to find Justin walking out of a building that, to Justin's embarrassment, she recognises as one of central London's seedier establishments. Alice's application, recognising that she has reached her destination, then automatically checks her into her new location via Foursquare.

This scenario highlights several of these aforementioned challenges. First, the situations where such tools might be used are as much about enabling freedom of action as they are applying constraints. Many policy tools have been implicitly designed for organisational settings where a certain degree of compliance with organisational norms can be expected. When people make personal use of applications, the values and norms influencing policy decisions may be shaped around personal freedom rather than information security compliance. Second,

our access control decisions are influenced not only by the objects and different (human or machine) subjects, but also by the contexts within which objects and subjects interoperate. People may make access control decisions for devices on the move when the context of enforcement is at home, and vice-versa. Third, presenting the ability to react to all variables in a context leads to complexity overload. Too many or too few options may lead to habituation, and there is already evidence that such habituation leads to malware being inadvertently installed on devices. Yet, at the same time, there are few examples of how these challenges might be addressed. Designers need insights about how to tackle the process of initially deriving a context sensitive policy, and how such policies might evolve as people's understanding of their contexts of use develops.

These challenges suggest that designing policy tools to meet the security and privacy expectations of end-users, without compromising their freedom of action, is easier said than done. Consequently, it would be useful to understand how people develop these expectations in realistic situations. By doing so, we can elicit concepts that influence this decision making process, and help developers build tools that help policy authors make context-sensitive access control decisions wherever they are. To this end, this paper describes how *context* was incorporated into an access control framework. We briefly describe related work in HCI-Security towards addressing the problem this paper addresses, before describing how three factors influencing the elicitation and categorisation of context-sensitive security policies were elicited. We conclude by illustrating the implications of these factors for the design of policy management tools using a policy editor for this access control framework.

2. RELATED WORK

(Zurko and Simon 1996)'s seminal work on user-centered security was, in part, motivated by the need for usable least-privilege access control. Since this time, the HCI-Security community has taken a keen interest in closing what (Norman 1988) describes as the *gulf of execution* between the intentions of users, and the system's means of implementing them. Studies into enterprise policy management tools have gone some way towards closing this gap. In particular, work by (Reeder et al. 2007) identified several general policy authoring challenges that need to be addressed by policy management tool developers; these include enforcing consistent use of terminology, making the concept of default rules and their rationale clearer, and facilitating the grouping of objects.

(Kelley et al. 2011) highlighted the difficulties people have devising a priori categories of objects that remain useful when making policy decisions. Earlier work on evaluating privacy preference tools identified similar problems between a priori policy specification and usage (Lederer et al. 2004). Because they need to understand the privacy implications of situated use, Lederer et al. argue that users prefer to carry out actions with imperfect default settings, rather than semi-intuitively configuring data on an a priori basis. The idea that people work off a general access control policy and vary this by different contexts was also identified by (Smetters and Good 2009).

Because much of this related work is framed from the perspective of using policy authoring tools, there has been comparatively little work on how the intentions behind policy decisions are formulated, and how these might be influenced. To glean an understanding of these intentions, it is useful to observe how people formulate and decide policy related actions using both scenarios and contexts specific to their day-to-day lives. To understand what these concepts and factors might be then examining how ordinary users respond to contrived, non-specific usage scenarios is not sufficient. Instead, we need to examine how representative stakeholders make access control decisions based on contexts of use that are meaningful to them.

3. APPROACH

3.1. Contextual access control in *webinos*

webinos is a software infrastructure for running web apps across mobile, PC, home media, and in-car devices (Fuhrhop et al. 2012). Its software architecture includes policy management components for facilitating cross-device access control (Lyle et al. 2012). Because *webinos* policy management is based on the XACML attribute-based access control model, it is theoretically possible to make fine-grained access control decisions based on environment attributes. Unfortunately, while the use cases upon which these components were based describe the flow of data between end users and system components, these are generic and framed in terms of whether or not users can access device features via applications running on different devices, rather than where these devices might be located.

As part of the project, a collection of personas – behavioural specifications of archetypical users (Cooper et al. 2007) – were developed to provide a voice for users and developers impacted by *webinos*. While useful for envisaging perceptions these stakeholders might have about *webinos*, it was unclear how their expectations about access

control might change in line with subtle changes in physical or social contexts within which *webinos*-enabled apps might be used. Without knowing these expectations, it would be difficult to formatively evaluate tools for creating and managing context-sensitive access control policies. When it became apparent that team members had difficulty even envisaging user interfaces for context-sensitive policy management, we decided to explore the impact of the concept of *context* on *webinos* policy specification and management.

3.2. Methodology

To understand how representative users would experience access control decisions across multiple devices and contexts, we ran nine persona-based participatory design workshops; each workshop was situated around the characteristics and activities of a particular persona. Following discussions within the project team, we were interested in three particular representative users. The first of these was a web application developer (Jimmy). *webinos* would not be successful if developers did not adopt it, so their stake in access control decisions would be critical. The second persona (Clara) represented younger, teenaged users, because we felt such users were more likely to adopt new technology. The final person represented the parent of a young child (Helen); this persona represented users that had made lifestyle choices making them sensitive to security & privacy concerns. These personas are described in more detail by (webinos Consortium 2011).

Workshop participants were recruited based on how closely they matched the characteristics of the three different personas. Participants were presented with a scenario meaningful to them, and asked to elicit and categorise types of data that would need to be subject to access control. For example, Helen workshops were structured around making decisions about the security and privacy implications of a networked in-car entertainment system being used by her young son while on a long car journey to see her parents.

Each workshop involved 3-4 participants, and lasted approximately 1.5 hours. The participatory design activity revolved around an affinity diagramming exercise. Affinity diagramming involves participants eliciting and organising data items, and consolidating these into groups that are meaningful to participants. As such, affinity diagramming allows participants to understand and make sense of data subject to access control without constraining thinking around any specific tool or technology.

The affinity diagramming exercise followed a specific structure. After introducing the scenario and providing a brief overview of affinity diagramming, each session was divided into three stages.

In the first stage (object clustering), participants spent 20 minutes eliciting data objects subject to access control, writing these on post-it notes, and affixing the notes to a wall or whiteboard; these object post-it notes were then grouped under categories the participants found useful for making access control decisions.

In the second stage (subject clustering), participants were required to elicit people (subjects) that should or should not have access to the objects elicited in the first stage. Red and green post-it notes designated parts of whiteboard where denied and allow objects are organised respectively. The participants then re-categorised the objects depending on what each subject should be allowed or denied access to; this stage took 30 minutes.

After a 10 minute break, participants spent 20 minutes on the final stage (context clustering). This involved identifying different contexts associated with the scenario and, for selected subjects, repeating the subject clustering stage based on each context. Following this stage, the participants walked through the affinity diagrams created, after which a short debrief session was held to find out how the participants found the exercise.

Audio and visual data was captured for each workshop. Following each workshop, the workshop organiser prepared a short report summarising the event's outcome and the main themes emerging from the affinity diagrams and the associated discussions.

We devised this approach because team members were already familiar with affinity diagramming from their previous work developing personas. This previous work also helped them recruit suitable participants, based on the workshops they were organising; these workshops were held in the UK, Italy, and Germany. Because the experiments were concerned with the subjects' behaviour rather than the affinity diagrams themselves, both the scenarios could be explained and sessions run in the local language, as only the session report needed to be written in English. The report structure guided participants towards the sort of observations that needed to be made, and subsequent telephone conferences helped validate the research being carried out because it gave team members an opportunity to present and discuss their results.

4. RESULTS

Once all the sessions had been completed, the workshop reports and transcripts were subject to open and axial coding (Corbin and Strauss 2008) by team members with experience in qualitative data analysis. From this coding exercise, 14 refined thematic concepts were identified. On investigating the relationships between these concepts and their grounding in the empirical data, we identified three main factors that influenced the elicitation and categorisation of context-sensitive access control policies: shared working contexts, pre-existing biases, and expectation based decision making.

4.1. Shared working contexts

Although not formally acknowledged by the participants, each workshop appeared to elicit and categorise data within the frame of a working context. The ability of participants to frame data was mediated by three factors.

The first of these were the nuances in the working context; examples of these range from varying the time-frame of a working context through to changing the family relationship of subjects in a context. Exploring these shed new light on pre-existing objects, but also led to considerable discussion, slowing down the rate of progress.

The second factor was general fatigue. Framing and re-framing objects and categories within a working context was time-consuming. In some cases, a rigorous exploration of objects and subjects in the working context left participants so tired that contexts were specified around pre-existing subjects or locations closely related to the working context.

The third factor was the use of supporting tools - in particular sketches and check-lists. Sketches were used in one workshop as a supplement to the context clustering affinity diagramming to explain particular subtleties of a context. Mental checklists and matrices were in a number of workshops to check the relevance of concepts, or validate a concept's inclusion under a particular category.

Although primarily used for concept elicitation and categorisation, framing was also useful for identifying concepts and categories forming the basis of innovation within the general domain. Examples included the elicitation of commercialisation and regulatory concepts that might foster improved security and privacy.

4.2. Pre-existing biases

The ability of participants to frame concepts and categories was influenced by pre-existing biases. In some cases, biases led to restrictive thinking about concepts because of pre-existing domain knowledge. This was most obvious in the Jimmy workshops, which centred on specifying policies for a training course website. Pre-existing knowledge about how the website was used, or assumptions about how hardware was setup appeared to unnecessarily dismiss objects as disallowed to particular subjects and context. Pre-existing biases were most evident when considering the implicit working context during object and subject clustering. Sometimes, these biases were re-enforced by participants when discussions were grounded around a particular frame; these frames were based on anecdotal experiences of the working context or when prompted by the facilitating workshop organiser.

As well as restricting thinking, biases and grounding also facilitated the elicitation of concepts which might otherwise have been missed. In almost all workshops, grounded discussions around particular working concepts led to the identification of concepts which were missed during the initial context-free object clustering stage.

4.3. Expectation-based decision making

The ability to frame concepts was also influenced by expectations held about the behaviour of particular subjects. Some of these were formed by pre-existing biases because participants felt they were proxy users for subjects under discussion. In others, participants espoused opinions they believed subjects might find important, irrespective of whether *they* found it important. For example, in one of the Clara workshops, participants proposed Digital Rights Management restrictions that they felt content providers would find useful. In some workshops, participants felt confident enough in their knowledge of subject expectations that concepts would be moved between allowed and denied sections of the white board or wall by category and, in some cases, categories would be elicited before concepts, and subsequent concepts grouped by static, pre-existing categories.

Expectations were important for envisaging the impact of policy decisions but, in certain cases, this also led to scope creep when participant knowledge of subjects or the domain appeared to be deficient. One of the factors contributing to scope creep was generalisation about the problem domain; this arose due to lack of knowledge or only a superficial treatment of concepts. The other contributing factor was generalisation due to perceived lack of relevance. Examples of this included collectively grouping "interface" or "service" technology because they seemed equally relevant

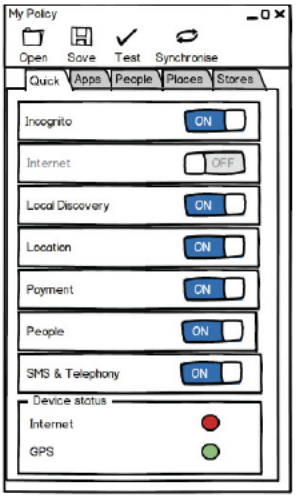

***Figure 1:** Prototype for coarse-grained access control settings interface*

to all subjects and contexts, and generalisation categories for concepts that were disallowed for particular subjects and contexts. In some cases, relevance generalisation led to an implicit transfer of trust onto objects or subjects related to generalised objects or categories. For example, "administrator" subjects were, in some cases, given untethered access to concepts. Similarly, in one workshop, participants considered a context where a "secure" terminal might be used by an administrator in an untrusted location.

5. BUILDING CONTEXT INTO THE WEBINOS POLICY EDITOR

These three factors informed the design and development of access control interfaces for a webinos *hub* dashboard application. This is not unlike home dashboards used for broadband router home hubs. The application was explicitly designed for what Cooper et al. (Cooper et al. 2007) describe as a "sovereign posture", where as much screen space as possible can be devoted to policy authoring and editing. This is because, based both on past work and our results, making sensible access control decisions for multiple contexts of use can monopolise users' attention for long periods of time.

Policy interfaces were sketched using Balsamiq to explore different interface possibilities. We used the workshop data to inform information architecture decisions. For example, when prototyping possible interfaces for an access control policy editor, we wanted to create coarse-grained access control settings, such as that shown in Figure 1. When considering what these items should be, we used the categories of objects elicited in the affinity diagrams created in the workshops. This secondary use of

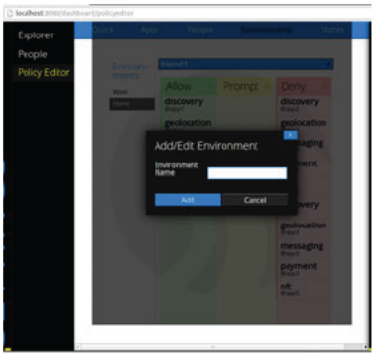

***Figure 2:** Adding Environments*

design data suggests the value such research can have given the increasing prominence placed on research where agility and lightweight techniques are encouraged, and where collaborating engineers are encouraged to *fake* process and focus on a final product (Parnas and Clements 1986).

As the sections below describe, these editor changes influenced the design of the underlying architecture in different ways. For example, the need to be flexible about incorporating context into the access control interfaces led to orthogonality between the API for modifying the access control policies, and the interfaces users used to interact with it.

5.1. Context-driven policy editing

We believe the usual order in which policies are developed – identifying *objects* and *subjects* first and then creating rules based on how they are allowed to interact in a particular *context* – should be reversed. While the workshops were designed to initially stimulate thinking about objects which needed to be secured, framing them with a working context meant that context could not be divorced from the policy specification processes. For this reason, we believed it would be more fruitful to begin the process by eliciting contexts of use before objects within them, or subjects that use them.

The attribute-based access control framework used by webinos made context-driven policy management a possibility by re-using the hitherto unused *environment* attributes of XACML. Similarly, implementing the interface controls for adding new environments shown in Figure 2 seemed trivial. However, in practice, implementing these features challenged many assumptions made by the designers that specified the architecture, and the developers who implemented it. For example, while adding an environment was non-trivial given the implementation of the XACML model. During unit testing of the policy editor, problems were found that exposed developer assumptions about context.

It was assumed that the definition of context used by the requestor and resource owner is shared. However, personas would name environments using short phrases based on the workshop, e.g. Peter's home. Not only would these environment names be different for the same context that two personas would share but, developers discovered the spaces have been implicitly disallowed in the implementation. This was only identified when it was observed that requests that should have been allowed were denied.

5.2. Use supplemental tools for policy creation and management

Additional tools are needed not only for editing policies, but also for creating supplemental information. The workshop results showed that techniques such as sketches, matrices, and checklists were useful. Previous work in the HCI-Security literature has also found that, if supplemented with contextual information about policy rules, matrices improve speed and accuracy when viewing and changing policies over traditional policy management tools (Reeder 2008). Based on both the literature and the workshop results, subject/object matrix controls for editing access control policies should support supplemental information to allow policy editors to reconstruct the frame used by policy developers when creating the additional policy. This supplemental information may include multi-dimension matrix cells such as (Reeder 2008), or additional controls to allow the attachment of image files or design rationale. With this in mind, interfaces for editing specific permissions were augmented to allow additional textual rationale to be appended to rules.

6. CONCLUSION

In this paper, we have presented three factors influencing the elicitation and categorisation of context-sensitive security and privacy policies. Rather than viewing our study through the lens of a specific tool, and drawing from a random user sample to carry out a general policy authoring scenario, we have instead used participatory workshops to understand the experiences associated with policy decisions. We have also based our study around scenarios that were specific to demographics of the participants engaged in these workshops. Based on these factors, we have illustrated how context was built into a policy editor to better understand the design implications of contextual access control.

REFERENCES

Cooper, A., R. Reimann, and D. Cronin (2007). *About Face 3: The Essentials of Interaction Design*. John Wiley & Sons.

Corbin, J. M. and A. L. Strauss (2008). *Basics of qualitative research : techniques and procedures for developing grounded theory* (3rd ed.). Sage Publications.

Fuhrhop, C., J. Lyle, and S. Faily (2012). The webinos project. In *Proceedings of the 21st international conference companion on World Wide Web*, pp. 259–262. ACM.

Kelley, P. G., R. Brewer, Y. Mayer, L. F. Cranor, and N. Sadeh (2011). An investigation into facebook friend grouping. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction*, pp. 216–233. Springer-Verlag.

Lederer, S., I. Hong, K. Dey, and A. Landay (2004, November). Personal privacy through understanding and action: five pitfalls for designers. *Personal Ubiquitous Comput. 8*, 440–454.

Lyle, J., S. Monteleone, S. Faily, D. Patti, and F. Ricciato (2012). Cross-plaform access control for mobile web applications. In *Policies for Distributed Systems and Networks, 2012 IEEE International Symposium on*, pp. 37–44.

Norman, D. A. (1988). *The design of everyday things*. Basic Books.

Parnas, D. L. and P. C. Clements (1986). A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering 12*(2), 251–257.

Reeder, R. W. (2008). *Expandable Grids: A user interface visualization technique and a policy semantics to support fast, accurate security and privacy policy authoring*. Ph. D. thesis, Carnegie Mellon University.

Reeder, R. W., C.-M. Karat, J. Karat, and C. Brodie (2007). Usability challenges in security and privacy policy-authoring interfaces. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction*, pp. 141–155. Springer-Verlag.

Smetters, D. K. and N. Good (2009). How users use access control. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pp. 15:1–15:12. ACM.

webinos Consortium (2011, February). User expectations on privacy and security. Downloadable from `http://webinos.org`.

Zurko, M. E. and R. T. Simon (1996). User-centered security. In *Proceedings of the 1996 New Security Paradigms Workshop*, pp. 27–33. ACM.