Formal Definitions for Usable Access Control Rule Sets From Goals to Metrics

Matthias Beckerle
Technische Universität Darmstadt
Darmstadt, Germany
beckerle@tk.informatik.tu-darmstadt.de

Leonardo A. Martucci
University of Karlstad
Karlstad, Sweden
leonardo.martucci@kau.se

ABSTRACT

Access control policies describe high level requirements for access control systems. Access control rule sets ideally translate these policies into a coherent and manageable collection of ALLOW/DENY rules. Designing rule sets that reflect desired policies is a difficult and time-consuming task. The result is that rule sets are difficult to understand and manage. The goal of this paper is to provide means for obtaining *usable access control rule sets*, which we define as rule sets that (*i*) reflect the access control policy and (*ii*) are easy to understand and manage. In this paper, we formally define the challenges that users face when generating usable access control rule sets and provide formal tools to handle them more easily. We started our research with a pilot study in which specialists were interviewed. The objective was to list usability challenges regarding the management of access control rule sets and verify how those challenges were handled by specialists. The results of the pilot study were compared and combined with results from related work and refined into six novel, formally defined metrics that are used to measure the security and usability aspects of access control rule sets. We validated our findings with two user studies, which demonstrate that our metrics help users generate statistically significant better rule sets.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Models and Principles—*user / machine systems*; D.4.6 [**Operating Systems**]: Security and protection—*access control*; D.2.8 [**Software Engineering**]: Metrics—*complexity measures*.

General Terms

Human factors; Security; Experimentation; Design.

Keywords

Access control; Usability; Security; Metrics; Formal logic.

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
Symposium on Usable Privacy and Security (SOUPS) 2013, July 24–26, 2013, Newcastle, UK.

1. INTRODUCTION

Access control mechanisms are used to ensure that

(*a*) access rights to resources are granted only to the authorized parties and

(*b*) access rights to resources are not denied to the authorized parties.

Access control mechanisms are used for controlling physical and logical access rights to shared resources, such as specific areas in an enterprise or data in a computer system. In a computer-based system, files and directories have rules associated with them that define a user's access rights, e.g., permissions for reading, writing or executing. An access control rule can be defined as a Boolean decision (ALLOW or DENY), which is taken upon the arrival of an access request. Access control mechanisms take as input a collection of access control rules, i.e., an access control rule set.

Access control policies are collections of high-level statements [12] that are expressed as access control rules. The authoring of access control policies, their management and implementation is not restricted to specialists in computer security. These activities are now expected even from less experienced users [4]. However, the task of generating and managing access control rule sets is not trivial [1, 4, 13]. Errors in access control rule sets can lead to unintended results, such as sharing more (or less) data than desired and the generation of too complex access control rule sets [13]. In addition, complex access control rule sets are difficult to manage and tend to have errors and inconsistencies, such as conflicting or duplicated rules.

The goal of our work is to provide a means for obtaining usable access control rule sets. We define usable access control rule sets as rule sets that fulfill the following two objectives:

(*i*) Usable access control rule sets reflect the access control policy (including items (*a*) and (*b*) above).

(*ii*) Usable access control rule sets are easy to understand and manage.

To accomplish the aforementioned objectives, we present a novel approach to support generating sound and manageable access control rule sets. This is achieved by defining and formalizing a set of goals for building usable access control rule sets. Our formalization makes it possible to compare and analyze access control rule sets *automatically*. The sets are analyzed in an automatic way, with regards to the enforcement of policies and the manageability of the rule set.

The access control model considered in our work is attribute-based access control (ABAC) [2, 14]. ABAC uses attributes to associate access rights with users, while role-based access control (RBAC) [5] uses roles. ABAC is more flexible than RBAC since dynamic conditions can be encoded in attributes. Moreover, ABAC can be used to implement other mechanisms, such as RBAC. Therefore, a formalization based on ABAC can also cover other dominant access control models, such as MAC (mandatory access control) and DAC (discretionary access control) [6].

The research method was structured into three parts. The first part consists of a pilot study with system administrators (using semi-structured interviews) and an analysis of papers mainly presented at CHI and SOUPS. This led to the definition of six goals for building usable and secure access control rule sets. We formalized these goals and assigned a set and a metric to each. The introduced metrics allow us to attribute a weighted score to each goal. Designating scores to rule sets allows users to evaluate, identify weaknesses, and compare alternative rule sets. The second and third parts of our investigation consist of user studies. In the second part we evaluated how helpful the metrics were to users in creating rule sets. In the third part we evaluated if our metric correlates with the opinion of IT support professionals.

In the remainder of this paper, we begin with the pilot study and continue with the summary of the background. We then describe our set of goals for usable access control, formalize the goals and the related metrics, and present a test scenario for exemplifying our analysis. Finally, we validate our results with the help of user studies, discuss our findings, and present our conclusions.

2. PILOT STUDY

We started with a pilot study, which consisted of semi-structured interviews with IT support professionals, i.e., experts. The objectives were to list the usability challenges regarding the management of access control rule sets and to look at how the participants handled those challenges.

2.1 Method

The participants were all IT support professionals (system administrators). They were recruited from business and public sectors (universities). Seven IT support professionals from four different organizations were interviewed. All of them managed Linux- or Windows-based access control mechanisms, using tools and services like Active Directory, `iptables`, and firewalls. No financial incentive was offered to the participants.[1]

We used semi-structured interviews as our method of inquiry in the pilot study. This method provided us the flexibility to ask for details regarding the challenges faced when managing access control rule sets. The interviews were individual and carried out under the condition that anonymity would be preserved (access control rule set details are usually confidential). We started with questioning the participants about their position in the organization hierarchy and about the main tasks related to access control management. All interviews were digitally recorded.

We asked about potential problems that occur when new access control rules are defined and when existing rule sets have to be changed. Furthermore, we asked what types of errors can occur in these processes and how they are avoided or circumvented.

2.2 Results

All participants of our pilot study reported strict procedures for managing user rights. Changes or adjustments in the access control rule set were discussed in meetings with other system administrators. A system administrator, from an organization with about 1 000 employees, estimated that one full work day is spent on such meetings every month. The administrator reported that these regular meetings were considered to be of high importance for the organization and the main objective was to guarantee the understandability and manageability of the access control rule set.

The participants also stressed the existence of two general kinds of challenges regarding the management of access control rule sets.

First, rule sets needed to be secure but allow legitimate accesses at the same time, i.e., all allowed accesses should be authorized and no security gaps should exist:

- (G1) Rule sets have to deny unauthorized access.
- (G2) Rule sets have to grant authorized access.

Second, rule sets needed to be understandable and manageable to help system administrators verify the correctness of the implementation of the stated policies.[2] The participants reported a series of potential sources of problems in access control rules sets that resulted in poor manageability. We organized those sources into the following goals:

- (G3) Redundant rules need to be removed.
- (G4) Contradicting rules need to be removed.
- (G5) Concise rule sets are better than large rule sets.
- (G6) Rule sets that are designed to facilitate the administrators' work to add/remove users to/from rule sets are easier to manage than rule sets that are not designed to facilitate the administrators' work.

A more detailed and refined description of G1 to G6, that takes related work into account, can be found in Section 4.1.

The participants were also asked about the usability of different access control mechanisms. They all pointed out that indirect access control mechanisms (like RBAC and ABAC) are more usable than direct access ones (like discretionary access control (DAC) or mandatory access control (MAC) [3]). However, they acknowledged that the task of translating entity-file access decisions (e.g., user x is allowed to access file y) is more difficult in RBAC and ABAC than in the other access control mechanisms.

3. BACKGROUND

Recent studies presented at CHI and SOUPS describe challenges and discuss solutions for managing access control mechanisms. They stress that usability is fundamental for setting up manageable and secure access control rule sets. In this section we summarize the findings of these studies.

[1]We did, however, promise to inform them first-hand about our findings and conclusions.

[2]The distinction between policy makers and implementers identified by Bauer et al. [1] maps directly to these two challenges. Their findings are summarized in Section 3.

Bauer et al. [1] list a series of real life challenges in access control management that are part of the quotidian work of system administrators from different organizations. More importantly, Bauer et al. pinpoint general causes that lead to unmanageable access control rule sets. They identify two groups in their study: policy makers and policy implementers. Policy makers create access control policies and policy implementers implement the policies designed by policy makers. This separation of roles leads to problems as policy makers do not often get to see the access control rules and policy implementers do not know the real intentions behind the policies. Bauer et al. also pinpoint problems that arise from having multiple policy makers and implementers working on the same system. It often results in access control rule sets that are hard to maintain and understand. Exceptions in the access control rules are particularly hard to manage as they demand notifications of changes between policy implementers. Furthermore, documentation needs to be kept up-to-date.

Smetters and Good [13] study the level of control necessary for users by examining access control policies created by users in a medium-size corporation. The access control policies regulate access to data files that are stored in a document sharing system. The system supports the creation of groups of users and implements RBAC. Smetters and Good conclude that users rarely change access rights of files or folders, and tended to store files in folders that had the appropriate access control policy as the files would inherit the folder's access rights. Furthermore, the creation of access control rules with effects other than expected and of redundant rules that could be made much simpler results in complex access control policies [13].

The particular needs and practices of access control in home environments were analyzed by Mazurek et al. [7]. Home environments are usually managed by users with limited or no knowledge regarding access control mechanisms. Hence, they describe a contrasting scenario in comparison to Bauer et al. [1] as participants have no previous theoretical or practical experience with access control mechanisms. We highlight two conclusions of Mazurek et al. regarding home users. First, home users desire access control mechanisms with greater granularity (complexity) than just names associated with files. Second, users wish for short and simple rule sets. In this paper, we discuss those two apparently conflicting goals and how they could both be achieved.

Errors in access control settings were evaluated by Egelman et al. [4]. In their paper, they examine how users implement access control policies with the limited settings offered by Facebook. The participants were Facebook users recruited from a higher education institution. The paper demonstrates that users are likely to introduce errors in their access control rule sets which often results in less restricted access control policies. Egelman et al. emphasized the importance of offering feedback followed by guidance on how to correct access control rule sets. Feedback with no guidance was proven to result in an increased number of incorrect rules [4].

Detection and resolution of conflicting access control rules were studied by Reeder et al. [11]. In particular, they targeted the problems of visualizing conflicts in access control rule sets in Windows-based operating systems. They pointed out two particular weaknesses in the Windows conflict resolution method arising from *deny precedences*[3] and *two-dimensional conflicts*.[4] Reeder et al. propose more suitable methods to solve the aforementioned weaknesses along with a grid-like user interface [10]. The interface was used to show and manipulate permissions in a more intuitive way than the Windows standard interface [11].

Dynamic creation of access control rules for computer file access was analyzed by Mazurek et al. [8]. Their objective was to evaluate the usability and general interest in a reactive access control mechanism, where users who own data files receive email requests from others wanting to access these files. Ad hoc decisions were taken by the file owners. Decisions were either to *ignore*, *allow* or *deny* the received request. File owners could also make ALLOW and DENY decisions permanent or temporary for the current request. Reactive access control can be potentially annoying, as pointed out by Mazurek et al. [8]. The (albeit limited) monetary incentive ($0.25/answer) and, more importantly, the limited time period (one week) and relative low and constant load of requests used in the evaluation (15 requests/day) may have masked some results regarding the true annoyance of a reactive access control mechanism. Furthermore, the creation of ad hoc access control rules resulted in unmanageable access control rule sets with the same limitations of discretionary access control mechanisms, like the determination of the unique ownership for each data file in a system.

4. USABLE ACCESS CONTROL: GOALS

In this section we summarize the limitations, problems and findings identified in our pilot study (Section 2) and in the background (Section 3) and organize them in a set of six goals for building usable access control rule sets. These goals are then formalized in Section 5 using formal logic.

4.1 Definition of Goals

We define the goals in terms of ownership, objects and access control rules. Owners can grant or deny access to objects using access control rules. Objects are resources such as data files, data folders, or physical rooms. Access control rules are written in terms of ALLOW or DENY decisions. The six goals identified are:

(G1) Allow no more than the owner wants to be allowed. This goal defines that a resource should be accessed only by people that are intended to have access to it. Allowing more than intended is the result of less restrictive or missing access rules. Less restrictive access rules are a likely consequence of errors introduced by owners as shown by Egelman et al. [4] in their study with Facebook users. Smetters and Good [13] also identified this problem in their analysis of documents with public access.

(G2) Allow everything the owner wants to be allowed. This goal states that a resource must be available to the people that are intended to have access to it. This goal basically complements *G1*. Allowing less than the intended access is the result of too restrictive access rules. Too restrictive access rules occur when the initial access control policy is insufficient as shown by Mazurek et al. [7].

[3] DENY rules take precedence over ALLOW rules.

[4] Conflicts that cannot be solved using the *specificity precedence* method. This method states that rules applied to more specific entities have precedence over rules applied to less specific entities, i.e., user-related rules have precedence over group-related rules [11].

(G3) A rule must not be fully covered by another rule of the same rule set. Redundant rules augment the complexity of an access control rule set by introducing new rules that are already covered by existing rules, thereby reducing the manageability of the access control system. Redundancies account for one of the reasons leading to errors in access control decisions [13].

(G4) Two rules belonging to the same rule set must not conflict. Conflicting access control rules impair the understandability of a rule set and often increase its complexity. Moreover, the resulting action from conflicting access control rules will depend on the implementation of the access control mechanism's conflict-resolution method. *Deny precedence* implies that DENY rules take precedence over ALLOW rules. *Allow precedence* implies the opposite. The order of appearance in the rule set can be used to define the precedence too, i.e., the first fitting rule is picked. Conflict-resolution in Windows-based systems was studied by Reeder et al. [11] who propose a new conflict-resolution method. Reeder et al. conclude that methods have inherent trade-offs as no method is able to always deliver the desired set of permissions. In our pilot study, we confirmed the findings of Reeder et al. The IT support professionals interviewed in our pilot study stated that conflicting rules were the most annoying issue in terms of maintainability.

(G5) Minimize the number of rule set elements. Minimizing the size of rule sets reduces their complexity and facilitates visual inspection. Complexity was identified as a major problem in the manageability of access control rule sets in the user studies of Smetters and Good [13] and Mazurek et al. [7] who evaluated distinct test environments (a medium-size corporation and home settings, respectively). After removing redundancies (*G3*) and (in some cases) eliminating conflicts (*G4*), the size of a rule set can be further optimized. One way to further optimize according to *G5* is to grant rights based on attributes instead of unique identifiers (granting access rights for STUDENTS is one access rule – granting access right for individual students by using the matriculation number leads to number-of-students access rules), by reducing the amount of attributes per rule and avoiding unnecessary rules. But contrary to *G3*, this procedure can lead to other conflicts, e.g., opening gaps for intruders.

(G6) Minimize maintenance effort in a changing system. Minimizing maintenance effort of an access control rule set whose access control policies are constantly changing requires a manageable and understandable rule set. Most of the changes in the rule set happen when access control policies are modified, or when users are added to or removed from the system. Overfitting rule sets results in increased maintenance effort.[5]

4.2 On Goals and Derived Metrics

G1 and *G2* are security related goals as they express access control decisions. The manageability of rule sets is reflected in goals *G3* to *G6*. All six goals for building usable access control rule sets need to be taken into account when creating new or evaluating existing rule sets. The need to evaluate all goals is a result of the non-orthogonality between the goals. Optimizing one goal might lead to a degradation of other goals in some cases, or might have a positive correlation in other cases. An example of trade-offs between goals was presented in Section 4.1 on the relation between *G5* and *G3*.

This relationship between goals can be illustrated as follows. *G2* can be maximized by defining a general ALLOW decision for every request. This solution conflicts with *G1*, as it may allow more than the owner wants to be allowed.

Reactive access control [8] is another example that showed the relationship between our stated goals. It allows changes to be made in the access control list according to the most current access control policy. Access control policies are defined by the owner on an ad hoc basis. Thus, *G2* is influenced positively as everything the owner wants to be allowed is allowed.[6] However, drawbacks in reactive access control, such as the lack of consistency checks in the resulting access control rule set, the probable creation of redundant and conflicting rules, and the potential annoyance of making ad hoc decisions regarding access control requests would result in negative effects on goals *G3*, *G4*, *G5* and *G6*.

The fulfillment of the goals can also be used to reduce mismatches between people's mental models regarding access control mechanisms and how they are actually implemented, which is a problem identified by Mazurek et al. [7]. Such mismatches can be reduced if users are able to verify the implemented policies and compare the actual implementation with the desired policies.

5. FORMALIZATION

In this section, we formalize the goals *G1* to *G6* and define the mathematical foundations of our approach. We first describe the building blocks that are needed to formalize ABAC, which is used as a reference system for further definitions. The formalization provides the sets, metrics and optimization criteria that are used to evaluate how usable an access control rule set is.

In this section, the following set operators are used:

- $|X|$ is the cardinality of X.
- $\wp(X)$ is the powerset of X.
- $X \setminus Y$ is the relative complement of Y in X.
- $X \triangle Y$ is the symmetric difference (XOR) of X and Y.

5.1 Basic Building Blocks

The basis for the formalization is given with the following definitions. We follow the general set nomenclature, where capital letters refer to sets and non-capital letters to single elements. All sets are assumed to be finite.

Definition 1. Entities. An entity is a subject, e.g., a person, that could be granted access to an object. The set of all entities is referred to as W (all possible entities, i.e., "the World"). The set E describes all entities in a system S, where $W \supseteq E$. The set B describes the group of owners of a system, where $W \supseteq B$.

Definition 2. Attributes. Attributes are properties of entities such as ID number, age, gender, roles or security level.

[5]We use the term *overfitting* according to its machine learning definition. In the scope of this paper, it means that rule sets that perform well at the current state of the system may perform poorly if the system is modified.

[6]There are no guarantees that reactive access control maximizes *G2* as DENY decisions may have permanent effects.

The set of all attributes is referred to as Å and a subset of Å is called A.

Definition 3. Objects. Objects are anything that access rights can be assigned to, e.g., a file or directory. The set of all objects is referred to as O, the set of all objects in a system is called D (e.g. "Data" in form of all files of an information system), and the subsets of D are named H (e.g. a subdirectory or "hierarchy" in a Windows-based system). In short: $O \supseteq D \supseteq H$.

Definition 4. Access decisions. There are two possible outcomes for an access request: allow or deny. We refer to the set of access decisions as $Z = \{-1, 1\}$, where -1 means DENY and 1 means ALLOW.

5.2 Derived Building Blocks

The following building blocks are constructed using the basic building block introduced above.

Definition 5. Rule and Rule Set. A rule describes the relation between single attributes, objects, and access decisions. For instance, the rule ({*Students*},{*Printer*},1) states that entities with the attribute *Students* are *allowed* to access the object *Printer*. A list R of n rules is called a rule set. We use the following notation: $R = (r_1, \ldots, r_n)$, where $r_i = (A_i, H_i, z_i)$ and r_i refers to the i^{th} rule of the rule set R. H_i refers to subsets of D, and A_i refers to subsets of $\mathring{A}$.

Definition 6. System. A system S is an environment described by sets of entities E, objects D, and access control rules R. It is defined as:

$$S = (B, E, D, R) \in \wp(W) \times \wp(W) \times \wp(O) \times (\wp(\mathring{A}) \times \wp(D) \times Z)^n$$

where B denotes the set of owners of the system, i.e., the entities that define the access control rules for this system and $n = |R|$.

Further, we define two functions. Function f_A is used for extracting all attributes from an entity. Function $f_{request}$ provides access decisions.

Definition 7. Attribute Extraction Function f_A. The attribute extraction function is defined as:

$$f_A : W \to \wp(\mathring{A}), w \mapsto f_A(w) := A_w.$$

It returns all attributes A_w, that belong to entity w.

Definition 8. Access Decision Function $f_{request}$. Let w be the requesting entity and H_w be the requested information. Let $z_{default}$ denote the default access decision if no rule is applicable to an access request, n denotes the number of rules in R and $i \in \mathbb{N}$, $1 \leq i$. The access decision function $f_{request}$ is defined as follows:

$f_{request}(w, H_w) := f_{req}(f_A(w), H_w, 1)$ or
$f_{request}(A_w, H_w) := f_{req}(A_w, H_w, 1)$, where
$f_{req} : (\wp(\mathring{A}) \times \wp(D) \times \mathbb{N}) \to Z, (A_w, H_w, i) \mapsto f_{req}(A_w, H_w, i)$

$$f_{req}(A_w, H_w, i) := \begin{cases} z_{default} & if\,(i > n), \\ z & if\, A_w \supseteq A_i, H_w \subseteq H_i \\ & .(A_i, H_i, z_i) = r_i, \\ f_{req}(A_w, H_w, i+1) & else. \end{cases}$$

For most systems, one would typically use $z_{default} = -1$.

Up to this point, we have provided a formalization of ABAC. In order to be able to evaluate whether a given rule set actually fits the system owners' intention, we provide a notation with regards to the intended behavior of the access control mechanism.

Definition 9. Owners' Intention. The function $f_{intended}$ specifies the owners' intention with regard to access control decisions and is given as:

$f_{intended} : (W \times \wp(D)) \to Z, (w, H) \mapsto f_{intended}(w, H)$.

In practice, it can be challenging to acquire the function $f_{intended}$. A possible solution could be to observe system usage over a period of time and use this information to approximate $f_{intended}$.

$$f_{intended}(w, H) := \begin{cases} 1 & if\, B\, wants\, f_{request}(w, H) = 1, \\ -1 & else. \end{cases}$$

5.3 Access Decision Sets

Before we formalize the goals *G1* to *G6* in a concise manner, we need to define eight access decision sets divided into two collections of sets, each containing four sets. The first collection relates to the entities that are known as part of the system S, whereas the second collection is required to address the problem of generalization of rule sets.

Definition 10. System Access Decision Sets.

$$\begin{aligned} M_{E_{Allow}} &= \{(e,d) | f_{Request}(f_A(e), \{d\}) = 1\}, \\ M_{E_{Deny}} &= \{(e,d) | f_{Request}(f_A(e), \{d\}) = -1\}, \\ M_{E_{Wanted}} &= \{(e,d) | f_{Intended}(f_A(e), \{d\}) = 1\}, \\ M_{E_{Unwanted}} &= \{(e,d) | f_{Intended}(f_A(e), \{d\}) = -1\}. \end{aligned}$$

Definition 11. World Access Decision Sets.

$$\begin{aligned} M_{W_{Allow}} &= \{(w,d) | f_{Request}(f_A(w), \{d\}) = 1\}, \\ M_{W_{Deny}} &= \{(w,d) | f_{Request}(f_A(w), \{d\}) = -1\}, \\ M_{W_{Wanted}} &= \{(w,d) | f_{Intended}(f_A(w), \{d\}) = 1\}, \\ M_{W_{Unwanted}} &= \{(w,d) | f_{Intended}(f_A(w), \{d\}) = -1\}. \end{aligned}$$

5.4 Security and Usability Metrics

The Definitions *1* to *11* are used to formally define the sets S_{Gi}, (where $1 \leq i \leq 6$). The sets S_{Gi} correspond to the security and usability metrics related to the goals Gi. The elements of a set S_{Gi} are the rules that contradict a goal Gi.

The criterion to achieve a goal Gi is therefore to minimize the number of elements in S_{Gi}: $minimize(|S_{Gi}|)$. The following definitions can be used to rate the usability of an access control rule set or to compare two different rule sets. The formalized definitions for S_{Gi} are:

(S_{G1}) Cases where too much is allowed (allow not more than the owners want to be allowed):

$$S_{G1} = M_{E_{Allow}} \setminus M_{E_{Wanted}}.$$

(S_{G2}) Cases where too little is allowed (allow everything the owners want to be allowed):

$$S_{G2} = M_{E_{Wanted}} \setminus M_{E_{Allow}}.$$

(S_{G3}) Unnecessary rules (a rule must not be fully covered by another rule of the same rule set):

$$\begin{aligned} S_{G3} = \{&(r_i, r_j).\ 0 \leq i \leq n-1 \ \wedge\ i < j \leq n \\ &\wedge\ A_j \supseteq A_i \ \wedge\ H_j \subseteq H_i \ \wedge\ z_j = z_i\}. \end{aligned}$$

(S_{G4}) Contradicting rules (two rules belonging to the same rule set must not conflict):

$$S_{G4} = \{(r_i, r_j).\ 0 \leq i \leq n-1 \ \wedge\ i < j \leq n \ \wedge\ \exists(e,h) .(f_A(e) \supseteq A_i, h \subseteq H_i, z)\ \nexists(f_A(e) \supseteq A_j, h \subseteq H_j, -z)\}.$$

The default access decision $z_{default}$ is not considered as a contradiction as it is not part of the rule set itself.

(S_{G5}) Number of elements in the rule set (minimize the complexity of rules and rule set):

$$S_{G5} = R \quad and$$
$$|S_{G5}| := \sum_{(A_i,H_i,Z_i)\in R} |A_i| + |H_i| + 1.$$

(S_{G6}) Cases that will lead to wrong access decisions in the future (minimize the maintenance effort in a changing system):

$$S_{G6} = M_{W_{Allow}} \triangle M_{W_{Wanted}} \cup M_{W_{Deny}} \triangle M_{W_{Unwanted}}.$$

In practice, it is very difficult to build the set S_{G6}, since it takes into account a future state as it considers entities that are not yet part of the system but will join it at a future time. The cross-fertilization between the fields of knowledge engineering and machine learning refers to this problem as a generalization or overfitting problem [9] and it can provide a solution for S_{G6}. In addition, the interviews with IT support professionals in the pilot study (and informal discussions with scientists from the knowledge engineering field) indicate that an optimized $|S_{G3}|$, $|S_{G4}|$, and $|S_{G5}|$ would have a positive effect on $|S_{G6}|$.

5.5 The Cost of Wrong Access Decisions

The two types of failures related to access control decisions are: decisions that should have been denied but were not, i.e., the elements in S_{G1}; or decisions that should have been allowed but were not, i.e., the elements in S_{G2}. Naturally, the consequences of failures vary. Granting access to a confidential file carries a higher cost than granting access to a non-critical system file. To capture such distinctions between different failures regarding their impact on the system or its users, the functions $f_{S_{G1}}$ and $f_{S_{G2}}$ are used.

The value $cost_{S_{G1}}$, which is related to S_{G1} and attributed to an access control rule set, is

$$cost_{S_{G1}} = \sum_{d\in X} f_{S_{G1}}(d),$$

where $X = \{d|(e,d) \in M_{E_{Allow}} \setminus M_{E_{Wanted}}\}$.

The value $cost_{S_{G2}}$, which is related to S_{G2} and attributed to an access control rule set, is

$$cost_{S_{G2}} = \sum_{d\in Y} f_{S_{G2}}(d),$$

where $Y = \{d|(e,d) \in M_{E_{Wanted}} \setminus M_{E_{Allow}}\}$.

And the *total cost* $= cost_{S_{G1}} + cost_{S_{G2}}$.

6. EXAMPLE

In this section we provide a scenario to illustrate how the security and usability metrics presented in the previous section can be used to measure, compare and optimize rule sets in order to construct usable access control rule sets, i.e, rule sets that are easy to understand and manage and that reflect the access control policy. The scenario presented in this section is the same one used in User Study 1 presented in the next section. The scenario is described by:

- a table of entities and their attributes,
- a table with the description of a file system,
- a graphical representation of the same file system,
- two tables describing access control rule sets.

Table 1: Entity–Attribute–Relationship Table. The 'x' markings indicate that a given attribute (column) is associated with a given entity (row), e.g., entity 1 has attributes *A3*, *A4* and *A7*.

Entity	Attributes					
	A3	*A4*	*A5*	*A6*	*A7*	*A8*
1	x	x			x	
2	x	x			x	
3			x	x	x	
4	x		x		x	
5	x	x	x		x	
6		x	x		x	
7	x		x		x	
8			x		x	x

In the scenario, each entity has an arbitrary number of attributes assigned to it. There are eight entities (1 to 8) and six attributes (*A3* to *A8*). Table 1 illustrates the relationship between entities and attributes.

The scenario describes a file system. It defines which files an entity should or should not have access to. The file system mimics a MS-Windows file system with 'C:' as its root. The directories are associated with the letters 'a', 'b', and 'c'. All files have a '.txt' extension. The file system is presented in Table 2. Table 2 also includes the $cost_{S_{G1}}$ associated with each file.

A graphical representation of the file system is illustrated in Figure 1. It also depicts the $cost_{S_{G1}}$ for files 'd.txt', 'f.txt' and 'j.txt', which are attributed values that differ from the default value. The $cost_{S_{G1}}$ of each file (except 'd.txt', 'f.txt' and 'j.txt') is 10 points and the $cost_{S_{G2}}$ of each file is 5.

Tables 3 and 5 present a rule set each. The rule sets are two different implementations of the access control policy represented in the entity-attribute relationship presented in Table 1 regarding the file system described in Table 2.

The compilation of the scores $|S_{Gi}|$, $cost_{S_{G1}}$ and $cost_{S_{G2}}$ (associated with S_{G1} and S_{G2} respectively) in Tables 4 and 6 represent the results obtained from each rule set and take into account the file system and the desired entity-attribute relationship of the scenario.

It is clearly more difficult to analyze the two rule sets and decide which one better fits the scenario without considering the scores $|S_{Gi}|$. With the $|S_{Gi}|$ scores, it is much easier to compare both rule sets, as they provide a clear indication of the quality of each rule set regarding the defined goals for security and usability of a rule set. The values of $cost_{S_{G1}}$ and $cost_{S_{G2}}$ are the most important values to compare when looking at the accuracy of the rule sets, i.e., how accurate are they when making a correct access control decision. $|S_{G3}|$,$|S_{G4}|$, and $|S_{G5}|$ are related to the manageability of the access control rule set.

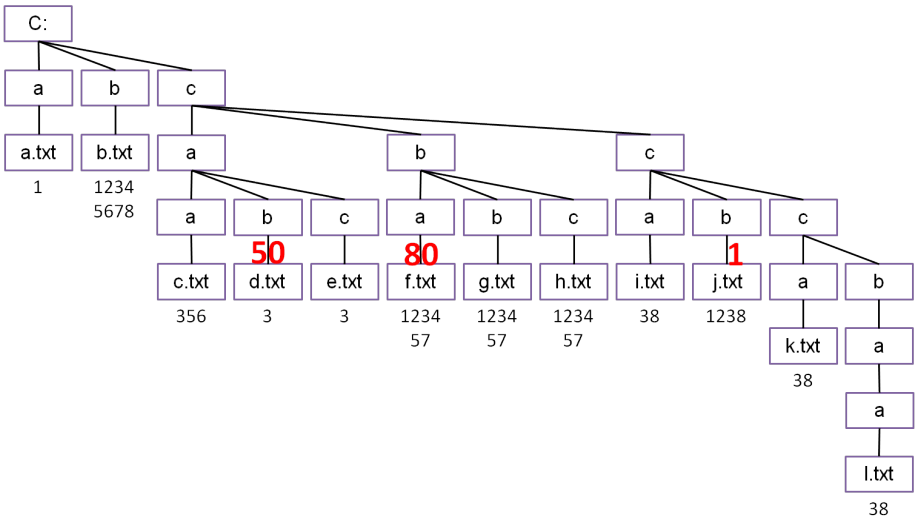

Figure 1: A graphical representation of the file system. It shows the files an entity should have access to and the non-default value $cost_{S_{G1}}$. The default cost value is 10.

Table 2: A description of the file system showing which files an entity should have access to and the $cost_{S_{G1}}$, i.e., too much is allowed, attributed to each file. The $cost_{S_{G2}}$, i.e., too little is allowed, is 5 for each file.

$cost_{S_{G1}}$	File Name	Entities that should have access
10	C:\a\a.txt	1
10	C:\b\b.txt	1, 2, 3, 4, 5, 6, 7, 8
10	C:\c\a\a\c.txt	3, 5, 6
50	C:\c\a\b\d.txt	3
10	C:\c\a\c\e.txt	3
80	C:\c\b\a\f.txt	1, 2, 3, 4, 5, 7
10	C:\c\b\b\g.txt	1, 2, 3, 4, 5, 7
10	C:\c\b\c\h.txt	1, 2, 3, 4, 5, 7
10	C:\c\c\a\i.txt	3, 8
1	C:\c\c\b\j.txt	1, 2, 3, 8
10	C:\c\c\c\a\k.txt	3, 8
10	C:\c\c\c\b\a\a\l.txt	3, 8

Table 3: Access Control Rule Set One

#	Path	Attributes	Decision
1	c:\a\	*A6*	DENY
2	c:\	*A6*	ALLOW
3	c:\b\	*A7*	ALLOW
4	c:\c\a\	*A4, A5*	ALLOW
5	c:\c\b\	*A7*	ALLOW
6	c:\c\c\	*A8*	ALLOW
7	c:\c\c\c\	*A8*	ALLOW

Table 4: Metric Scores of Rule Set One

Goal	$\lvert S_{Gi}\rvert$	$cost_{S_{Gi}}$
G1 (Too much allowed)	10	320
G2 (Too little allowed)	3	15
G3 (Unnecessary rules)	1	-
G4 (Contradicting rules)	1	-
G5 (Elements in rule set)	22	-

Regarding accuracy, Rule Set Two (Table 5) is superior to Rule Set One (Table 3), as it has better (lower) scores for $|S_{G1}|$, $|S_{G2}|$, $cost_{S_{G1}}$ and $cost_{S_{G2}}$, as shown in Tables 4 and 6. Only regarding $|S_{G5}|$ does Rule Set One have a better score than Rule Set Two, which has (three) more elements in its rule set. In all other categories, Rule Set Two is superior or equal to Rule Set One.

When optimizing a rule set, the provided values are very helpful. For instance, regarding Rule Set One, the metric S_{G1} indicates 10 errors with an associated $cost_{S_{G1}}$ of 320. The $cost_{S_{G2}}$ of denying authorized accesses is 15 (in row G2). Inspecting the 10 elements in set S_{G1}, the following information can be extracted: $\{(E5, d.txt), (E5, e.txt), (E6, d.txt), (E6, e.txt), (E6, f.txt), (E6, g.txt), (E6, h.txt), (E8, f.txt), (E8, g.txt), (E8, h.txt)\}$. The cost function $f_{S_{G1}}$ shows that $(E6, f.txt)$ and $(E8, f.txt)$ both have cost values of 80 and are the most critical errors, i.e., the errors with the highest possible cost. These two errors can be eliminated by changing the attribute *A7* to the attribute *A3* in rule #5. Then, by recalculating the results we obtain: $|S_{G1}| = 4$, $|S_{G2}| = 3$, $|S_{G3}| = 1$, $|S_{G4}| = 1$, $|S_{G5}| = 22$ with a $cost_{S_{G1}}$ of 120 and a $cost_{S_{G2}}$ of 15. These values indicate a significant improvement over the former version of the rule set.

7. VALIDATION

We validated the sets, metrics, optimization criteria and their usefulness to support users in generating usable access control rule sets by testing three hypotheses:

Hypothesis H1: The sets, metrics and optimization criteria help users produce better rule sets.

Hypothesis H2: The metric scores correspond to rankings obtained from IT support professionals when evaluating the translation of policies into access control rule sets (related to *G1* and *G2*).

Hypothesis H3: The metric scores correspond to rankings

Table 5: Access Control Rule Set Two

#	Path	Attributes	Decision
1	c:\b\	*A7*	ALLOW
2	c:\c\	*A6*	ALLOW
3	c:\c\a\a\	*A4*, *A5*	ALLOW
4	c:\c\b\	*A3*	ALLOW
5	c:\c\c\	*A8*	ALLOW
6	c:\c\c\b\	*A3*, *A4*, *A5*	DENY
7	c:\c\c\b\	*A3*, *A4*	ALLOW

Table 6: Metric Scores of Rule Set Two

Goal	$\|S_{Gi}\|$	$cost_{S_{Gi}}$
G1 (Too much allowed)	0	0
G2 (Too little allowed)	1	5
G3 (Unnecessary rules)	0	-
G4 (Contradicting rules)	1	-
G5 (Elements in rule set)	25	-

obtained from IT support professionals when evaluating the understandability and manageability of access control rule sets (related to *G3*, *G4* and *G5*).

We tested these hypotheses with the help of two user studies. User Study 1 aimed to gather data from both non-experts and IT support professionals regarding the creation of rule sets that match the system owner's intention with and without the support of our proposed sets, metrics and optimization criteria. The outcome from User Study 1 was used as input to User Study 2. The output of the user studies was analyzed in Section 7.3 and the limitations of our user studies are listed and discussed in Section 7.4.

7.1 User Study 1

In User Study 1, participants were asked to complete a computer-assisted task regarding the optimization of an access control rule set. Two test conditions were used for completing the task: *without the sets, metrics and optimization criteria* (*WOS*) and *with support of the sets, metrics and optimization criteria* (*WS*). Participants were randomly assigned to one or the other test condition.

7.1.1 Method

Twelve participants took part in the study. Two-thirds were non-experts regarding access control configuration and management. The other four participants were IT support professionals, who manage access control mechanisms on a regular basis. One of the IT support professionals had also participated in the pilot study. The age of the participants ranged between twenty and fifty-five ($\mu = 34.5$, $\sigma = 8.1$) and four participants were female. Seven of the participants were graduate students, one had a PhD degree, three held degrees from universities of applied sciences, and one had no university degree. No financial incentive was offered to the participants for taking part in the study.

A between subject design was applied in this user study. The study was designed as a laboratory experiment. The experiment was individual, i.e., one participant at a time. Participants had the task explained by a supervisor (the task was described in print, which was handed out at the beginning of the experiment). The supervisor answered questions regarding the task description, informed the participants about the maximum time allowed and enforced this time limit. The time allowed was 20 minutes (plus the time required to explain the task). Participants were encouraged to vocalize their line of thought.

The task was to minimize the cost associated with the given rule set by changing, adding or deleting rules from an existing access control rule set. The rule set was given to the participants in the form of an 'MS Excel spreadsheet' to eliminate possible bias, as all participants were familiar with this spreadsheet application.

There were two conditions used in the laboratory experiment: *without support of the sets, metrics and optimization criteria* (*WOS*) and *with support of the sets, metrics and optimization criteria* (*WS*). Participants were randomly assigned to one of the two conditions. The IT support professionals were equally distributed between the two conditions to avoid impact of their expertise on the results.

In *WOS*, participants were asked to optimize the rule set without additional support by any sets, metrics and optimization criteria (apart from the spreadsheet application). In *WS*, the spreadsheet application was programmed to return all sets and metrics provided by our formalization, including the *total cost* ($= cost_{S_{G1}} + cost_{S_{G2}}$) associated with the rule set, which was displayed when the participant clicked a button labeled UPDATE in the spreadsheet application interface.

The participants were informed what rule sets are, how rules are expressed (in terms of ALLOW/DENY decisions), and how they are processed (from top to bottom). In particular, the participants were informed about the following: DENY rules having precedence over ALLOW rules, there is a default DENY ALL rule at the end of the rule set, and if a rule is defined to a directory then all its sub-directories and files inherit that same rule.

The task description contained: Table 1, Table 2 and its graphical representation (Figure 1), and Table 3, which presented the initial rule set to be modified by the participant to adhere to the desired policy.

At the end of the experiment, participants handed in the access control rule sets that they produced. Twelve rule sets were obtained. Participants in the condition *WOS* were asked, after handing in their rule sets, to redo the experiment with the support of the sets, metrics and optimization criteria, i.e., following the *WS* test condition, and produce six new sets of rules. The six additional rule sets were used to increase the size of the input to User Study 2 and used only to test *Hypothesis H2* and *Hypothesis H3*. Naturally, the additional rule sets were not used to test *Hypothesis H1* as they were affected by order and learning effects. Order and learning effects of the additional rule sets are not relevant to the objectives of User Study 2.

7.1.2 Acquired Data

The outcome of User Study 1 was three times six access control rule sets (six from test condition *WOS*, six from test condition *WS*, and six additional ones). These rule sets were used as input for User Study 2.

7.2 User Study 2

In User Study 2, the participants were IT support professionals. They were asked to evaluate and rank the rule sets that were obtained from User Study 1 based on their own experience and knowledge. Two evaluation criteria were de-

fined: (a) *how accurately the rule sets implement the access control policy* and (b) *how easily the rule sets can be understood and managed.*

7.2.1 Method

The 18 rule sets generated in User Study 1 were tested 4*2 times by IT support professionals. So 8 sub-experiments each with $N = 18$ were performed. Four IT support professionals took part in the evaluation according to criterion (a) and four took part in the evaluation according to criterion (b). Each expert processed all 18 rule sets. The IT support professionals were recruited from business and public sectors (universities). One of the participants had taken part in the pilot study and User Study 1. Two of the IT support professionals had taken part in the pilot study but not in User Study 1. All of them managed access control mechanisms on a regular basis and worked several years in positions related to IT support. Again, no financial incentive was offered to the participants.[7]

The collection of access control rule sets was sent to the IT support professionals by electronic mail. The ordering of the rule sets was randomized before being sent to the participants. The participants were asked to provide a short description of their approach for evaluating the rule sets regarding criteria (a) and (b). No time limit was set to complete the ranking.

7.2.2 Acquired Data

The result of User Study 2 is two rankings for each expert. One reflects the opinion of the IT support professionals regarding how accurately the rule sets implement the access control policy and the other one reflects how easily the rule sets can be understood and managed in their opinion. The participants took up to several hours to complete the task and one stated that the analysis of some rule sets took close to one hour to analyze. The IT support professionals reported different approaches and methods used in their rankings. The main aspects reported when evaluating manageability of rule set were the following: the time needed to read and understand it, the number of elements in it, and the number of DENY rules. The translation of the defined policy into a rule set was evaluated according to the number of security gaps and wrongly denied accesses. Next, each outcome of the sub-experiments of User Study 2 is tested for correlation with the outcome obtained using our sets, metrics and optimization criteria.

7.3 Results and Evaluation

In this section we validate our three hypotheses. First, hypotheses *H2* and *H3* were validated by the strong correlation between the ranking produced by IT support professionals and the ranking obtained by using our metric scores. After validating *Hypothesis H2*, we validated *Hypothesis H1*.

To validate *Hypothesis H2* and *Hypothesis H3*, we compared the rankings produced by the IT support professionals in the User Study 2 and the rankings generated using our metric scores. For testing *Hypothesis H2*, we compared the list of the four rankings produced using criterion (a) and the rankings generated using the *total cost* metric ($cost_{S_{G1}} + cost_{S_{G2}}$). *Hypothesis H3* was tested by comparing the list of the four rankings produced using criterion (b) and the rankings generated using S_{G3}, S_{G4} and S_{G5}.

[7]We again promised to inform them first-hand about our findings and conclusions.

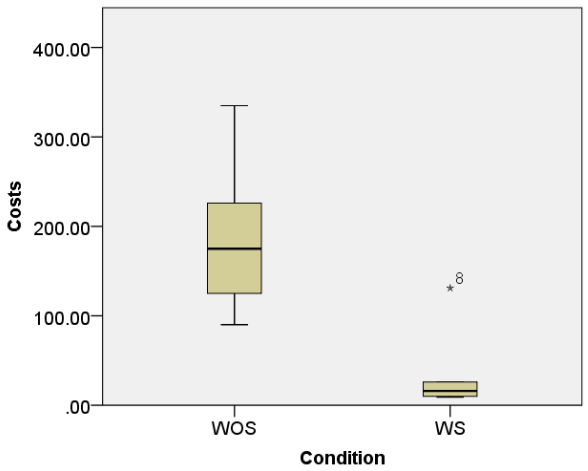

Figure 2: Box plot showing the results of User Study 1. They are presented with 0.95 confidence interval.

Spearman rank correlation coefficients were computed to assess the relationship between the rankings. Overall, there was a significant positive correlation between the ranking automatically produced and the rankings obtained from User Study 2, as shown in Table 7. The results from these tests validate both *Hypothesis H2* and *Hypothesis H3*.

The correlation was higher for *Hypothesis H2* than for *Hypothesis H3*. This was expected because there is a common methodology to evaluate how accurately a rule set implements an access control policy by analyzing the results for security gaps and non-granted legitimate access rights. The IT support professionals used similar methodologies to rank the rule sets according to criterion (a). Interestingly, all IT support professionals made small mistakes by overlooking some gaps. However, when ranking the rule sets according to their manageability, the IT support professionals used a wider variety of approaches, such as counting the number of DENY rules, the time spent to understand the rule set, or deciding intuitively.

User Study 2 aimed to evaluate whether the values $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$ can be used to provide results that are similar to results obtained from IT support professionals. The results from User Study 2 showed a strong correlation between the results obtained from the IT support professionals and the results that were automatically generated by a software that implements our proposed formalization. This result validated the expressiveness of $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$.

After validating *Hypothesis H2*, we were able to test *Hypothesis H1* by calculating the *total cost* metric of each access control rule set produced in the User Study 1 and comparing the results from the *WOS* and *WS* groups.

The Box plot in Figure 2 summarizes the results obtained from User Study 1. The mean total cost for condition *WOS* (no support) was significantly higher ($\mu = 187.7$, $\sigma = 36.7$) than the total cost for condition *WS* (with support) ($\mu = 34.7$, $\sigma = 19.5$). This difference in the results is also shown in Table 8, which compares the results for the two conditions using independent samples t-test for the test conditions *WOS* ($\mu = 187.7$, $\sigma = 36.7$) and *WS* ($\mu = 34.7$, $\sigma = 19.5$) for $t(3.692) = 7.621$ and $p = 0.007$.

Table 7: Spearman's rank correlation between the automatically produced rankings and the rankings obtained by the User Study 2. *Proposal* refers to the automatically produced ranking, i.e., the optimal outcome, and *Result 1* to *Result 4* to the results obtained from IT support professionals. $N = 18$ for all cases.

		Proposal	Result 1	Result 2	Result 3	Result 4
Spearman's rho	*Hypothesis H2* Correlation Coefficient	1.000	.908**	.967**	.971**	.955**
Spearman's rho	*Hypothesis H3* Correlation Coefficient	1.000	.922**	.820**	.874**	.777**

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8: Independent samples t-test. Input: *WOS* ($Mean = 187.7$, $SD = 36.7$) and *WS* ($Mean = 34.7$, $SD = 19.5$)

	Levene's Test for Equality of Variances		t-test for Equality of Means						
								95% Confidence Interval of the Difference	
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Equal variances assumed	3.005	.114	3.692	10	.004	153.00000	41.44581	60.65297	245.34703
Equal var. not assumed			3.692	7.621	.007	153.00000	41.44581	56.59155	249.40845

The participants in *WS* performed significantly better than the participants in *WOS*. The analysis of the results obtained from User Study 1 validated *Hypothesis H1* by showing that our sets, metrics and optimization criteria help users to produce significantly better rule sets.

7.4 Limitations

A real case regarding the management of an access control rule set can easily involve tens of thousands of objects and as many entities. Still, we deliberately designed User Study 1 with few objects (12) and entities (8). Our decision to limit the number of objects and entities was based on two points. First, a more complex scenario would be more difficult for participants to understand under the conditions and practical limitations of the study. Second, User Study 1 is close to a worst case scenario with respect to the performance of our approach as a more complex scenario would also increase the space for misconfiguration and errors. As our metrics are designed to allow identification of such cases, it is expected to produce significantly better results in a more complex and non-controlled environment.

The sample size of User Study 1 (twelve participants) is not large, but enough to obtain significant results from the statistical tests on the collected data. In User Study 2, four IT support professionals ranked the 18 rule sets produced in User Study 1. Increasing the number of participants in User Study 1 would result in a large sample of rule sets and it would also increase the number of rule sets each IT professional would need to rank. A practical limitation of our study is that all the participants were volunteers, and the amount of effort required from the experts was considerable. The four IT professionals in User Study 2 produced similar rankings, which suggests that four was sufficient for our evaluation. The IT professional volunteers were very positive about our studies and, following User Study 1, two of them independently asked the study supervisor about the possibility of integrating our tools into their workspace, as they strongly believed that it would facilitate their work.

A limitation of User Study 2 is that it cannot individually validate the metrics $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$, but only the composition of all factors together. Hence, we were not able to evaluate the impact of each individual metric when testing rule sets for their manageability. It would be interesting to analyze the individual impact of each metric to obtain even better results.

8. DISCUSSION

In this section we discuss our findings, open challenges towards introducing new factors in our metrics and opportunities for future work.

The six goals for building usable access control rule sets presented in our work were derived from the pilot study. The goals formalize the metrics used by experts to evaluate rule sets. This set of goals is not comprehensive and is a subset of goals for building usable access control rule sets. Other metrics could be included to the set if they are found relevant in future studies. For instance, the design of the user interface was never mentioned during the interviews of the pilot study, but it may be an important aspect for less experienced users. Another factor that is not captured by our metrics is the indirect interdependency of rules, which may impact the usability of rule sets. Extending the set of metrics could lead to better rule sets, but to determine their importance would require further testing and evaluation.

A challenging aspect of our building blocks presented in Section 5.2 is the formalization of the owners' intention, $f_{intended}$. Obtaining the owners' intention is out of the scope of this paper but it is a key aspect to be considered in future work. Solutions would possibly involve direct interaction with the owner using tools, such as a reactive access control mechanism [8], psychological testing, questionnaires or observation of the owners' behavior in using and sharing data.

Another important aspect to be carefully analyzed is the use of cost functions. Attribution of costs is highly subjective and dependent on the nature of data. Costs are relevant for defining levels of importance for different objects (i.e., different objects with different costs) and goals (i.e., different costs for too much allowed and too little allowed). Nevertheless, the metrics presented in this paper are independent of the attribution of costs. An interesting extension of this work would be to introduce cost functions for the sets S_{G3}, S_{G4}, S_{G5} and S_{G6}. The additional cost functions would be an important step towards building a single metric instead of multiple metrics to rate a rule set.

Optmizing a criterion could affect other criteria, therefore it is important to evaluate dependencies between criteria in future work. For instance, eliminating contradictions (*G4*) can sometimes lead to a more complex rule set (*G5*) as shown in the following example:

RULE 1: *A*lice is denied access to file.
RULE 2: Everyone is allowed access to file.

Above we have a short rule set with one contradiction. A non-contradicting rule set that describes the same scenario could be implemented as following:

RULE 1: *B*ob is allowed access to file.
RULE 2: *C*hris is allowed access to file.
RULE 3: *D*ave is allowed access to file.
...
RULE 23: *X*ena is allowed access to file.
RULE 24: *Y*uri is allowed access to file.
RULE 25: *Z*ara is allowed access to file.

This rule set results in a rule set with more elements and no contradictions. Cost functions of the sets S_{G3} to S_{G6} would be able to detect the effects between multiple criteria.

9. CONCLUSIONS

In this paper we introduced security and usability metrics that quantify how usable access control rule sets are. We started from informal requirements and a minimal set of basic formal building blocks. We then obtained a set of six formal definitions for security and usability properties of access control rule sets. We provided tangible and simple values that indicate the characteristics and the number of errors in access control rule sets. The provided metrics were validated by user studies that resulted in statistically significant evidence for our hypotheses.

In conclusion, our approach offers a uniform and scientific method for comparing different rule sets. Moreover, our metrics can be used as optimization criteria to generate usable access control rule sets and to improve their manageability. Furthermore, a formalization is the first step towards the implementation of tools for measuring and comparing different rule sets automatically. Future and ongoing work aim to demonstrate that the implementation of the results presented in this paper can significantly improve rule sets. Another objective is to design a tool that can be integrated in the daily working environment to actively help users produce usable access control rule sets.

Acknowledgments

The authors would like to first thank all the volunteers that took part in their user studies, in particular the IT support professionals. The authors are also thankful to Sebastian Ries for helping them with the statistical tests and to all reviewers for their valuable and insightful comments. This work was partially funded by A4CLOUD, a project of the Seventh Framework Programme for Research of the European Community, grant agreement no. 317550.

10. REFERENCES

[1] Bauer, L., Cranor, L. F., Reeder, R. W., Reiter, M. K., and Vaniea, K. Real life challenges in access-control management. In *Proc. CHI 2009*, ACM (2009), 899–908.

[2] Bonatti, P. A., and Samarati, P. A uniform framework for regulating service access and information release on the web. *J. Comput. Secur. 10* (Sep 2002), 241–271.

[3] Brand, S. DoD 5200.28-STD Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). *National Computer Security Center* (1985).

[4] Egelman, S., Oates, A., and Krishnamurthi, S. Oops, I did it again: mitigating repeated access control errors on Facebook. In *Proc. CHI 2011*, ACM (2011), 2295–2304.

[5] Ferraiolo, D. F., and Kuhn, D. R. Role-based access controls. In *Proc. of the 15th National Computer Security Conference* (1992), 554–563.

[6] Jin, X., Krishnan, R., and Sandhu, R. A unified attribute-based access control model covering DAC, MAC and RBAC. In *Proceedings of the 26th Annual IFIP WG 11.3 conference on Data and Applications Security and Privacy*, DBSec'12, Springer-Verlag (Berlin, Heidelberg, 2012), 41–55.

[7] Mazurek, M. L., Arsenault, J. P., Bresee, J., Gupta, N., Ion, I., Johns, C., Lee, D., Liang, Y., Olsen, J., Salmon, B., Shay, R., Vaniea, K., Bauer, L., Cranor, L. F., Ganger, G. R., and Reiter, M. K. Access control for home data sharing: evaluating social acceptability. In *Proc. CHI 2010*, ACM (2010), 645–654.

[8] Mazurek, M. L., Klemperer, P. F., Shay, R., Takabi, H., Bauer, L., and Cranor, L. F. Exploring reactive access control. In *Proc. CHI 2011*, ACM (2011), 2085–2094.

[9] Mitchell, T. M. *Machine Learning*, 1 ed. McGraw-Hill, Inc., New York, NY, USA, 1997.

[10] Reeder, R. W., Bauer, L., Cranor, L. F., Reiter, M. K., Bacon, K., How, K., and Strong, H. Expandable grids for visualizing and authoring computer security policies. In *Proc. CHI 2008*, ACM (2008), 1473–1482.

[11] Reeder, R. W., Bauer, L., Cranor, L. F., Reiter, M. K., and Vaniea, K. More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proc. CHI 2011*, ACM (2011), 2065–2074.

[12] Samarati, P., and di Vimercati, S. D. C. Access control: Policies, models, and mechanisms. In *Foundations of Security Analysis and Design, Tutorial Lectures (FOSAD 2000)*, vol. 2171 of *Lecture Notes in Computer Science*, Springer (2000), 137–196.

[13] Smetters, D. K., and Good, N. How users use access control. In *Proc. SOUPS 2009*, ACM International Conference Proceeding Series, ACM (2009).

[14] Yuan, E., and Tong, J. Attributed based access control (ABAC) for web services. In *ICWS*, IEEE Computer Society (11–15 Jul 2005), 561–569.