Highlights

Optimization of Access Control Policies

Sascha Kern,Thomas Baumer,Sebastian Groll,Ludwig Fuchs,Günther Pernul

- Derivation of ACP optimization based on data quality and 16 well-established criteria
- Literature survey on ACP optimization based on six optimization objectives
- Discussion and findings for ACP optimization

Optimization of Access Control Policies

Sascha Kern[a,*], Thomas Baumer[a], Sebastian Groll[a,b], Ludwig Fuchs[a] and Günther Pernul[b]

[a] *Nexis GmbH, Franz-Mayer-Straße 1, Regensburg, 93053, Bavaria, Germany*
[b] *University of Regensburg, Universitätsstraße 31, Regensburg, 93053, Bavaria, Germany*

ARTICLE INFO

Keywords:
Access Management
Data Quality
Policy Optimization
Policy Maintenance
Role-Based Access Control
Attribute-Based Access Control

ABSTRACT

Organizations undertake complex and costly projects to model high-quality Access Control Policies (ACPs). Once built, these policies must be maintained and managed in an ongoing process to keep their quality high. Insufficient maintenance leads to inaccurate authorization decisions and increases the policies' administrative effort and susceptibility to errors. While the initial modelling of ACPs has received significant research interest, their optimization is not yet covered as broadly. This work provides a theoretical foundation for ACP quality and its optimization. Furthermore, it analyzes how existing research addresses optimization of ACPs with regard to six crucial optimization dimensions. It presents a structured literature survey tracing these optimization dimensions, the contributed research artifact and data requirements. Building on this literature catalogue, this work elaborates on inaccuracies for user permission assignments, data availability, minimal perturbation and recommendation-based optimization.

1. Introduction

The organizational structures and IT infrastructures of modern companies are subject to constant change. Routine operations like departmental changes of employees, changing responsibilities or the integration of new application systems into the IT landscape require an adaptation of IT security configurations. This includes updating Access Control Policies (ACPs), machine-processable rules that define authorizations and can be evaluated in a fully automated manner to determine which accesses an employee is allowed to make (Samarati and de Vimercati, 2001). Due to changing environmental conditions, ACPs that were once of high quality lose accuracy over time (Xia et al., 2014; Hu et al., 2011). Moreover, ACPs proliferate over time, as policy administrators may over-grant access to conform with immediate business needs (Xu et al., 2017; Xiang et al., 2019) or update policies in an erroneous or non-optimal way. Besides hard errors, ACP proliferation leads to lower comprehensibility and maintainability, which increases the ACPs' administrative cost and their proneness for further errors (Bauer et al., 2009; Beckerle and Martucci, 2013).

Incorrect or overly permissive access decisions leave companies vulnerable to insider threats. A malicious or careless insider can harm an organization severely, with consequences spanning from unintentional incidents to sabotage, fraud or espionage (Tsiostas et al., 2020). In contrast, overly restrictive access decisions prevent employees from doing their work, leading to costly interruptions in operations and task backlogs. Recent studies estimate that the average annual cost of insider threats for companies reach $11.45 million in 2020 (Gilbert, 2021; Tsiostas et al., 2020). The implementation of effective Identity and Access Management (IAM) measures, which follow the principle of least privilege (Horne, 2011), is hence mandated by major regulatory frameworks and IT security standards[1].

Maintenance measures, which optimize the quality of ACPs in a continuous manner, are a fundamental requirement for providing an accurate level of security with reasonable administrative effort over a longer period of time, and for maintaining the investment made in the initial modelling of high quality ACPs (Molloy et al., 2010; Parkinson et al., 2020; Kunz et al., 2015a). As the initial modeling of ACP sets with high quality requires high time and financial effort (Jaferian et al., 2014), maintenance processes aim to improve the quality of an ACP set by applying updates that leave the existing state intact. ACP maintenance is commonly approached in two types of processes: First, access reviews are a process where responsible humans (such as a department head) review ACPs for entities in their responsibility (for example roles assigned to their employees) and try to find and rectify inaccuracies. The effectiveness of access reviews is limited, since reviewers have to check large amounts of data in a largely manual process and have limited information to make a qualified decision (Groll et al., 2021; Pan et al., 2018; Jaferian et al., 2014). Second, ACP refinement processes aim to improve the quality of an ACP set by updating it in a (semi-)automated manner without deconstructing it (Xia et al., 2014). Both approaches will be considered in the course of this work.

This work contributes to an optimization of ACPs by addressing the following **research question**: *Which methods for the optimization of roles and Attribute-Based Access Control (ABAC) policies are present, and what are their*

*Corresponding author: S. Kern
sascha.kern@wiwi.uni-regensburg.de (S. Kern)
https://nexis-secure.com/ (L. Fuchs); https://www.uni-regensburg.de/wirtschaftswissenschaften/wi-pernul/startseite/index.html (G. Pernul)
ORCID(s): 0000-0002-7082-000X (S. Kern); 0000-0003-0157-3057 (T. Baumer); 0000-0003-1338-9003 (G. Pernul)

[1] Such as the Sarbanes-Oxley Act (One Hundred Seventh Congress of the United States of America, 2002), the Basel Accords (Basel Committee on Banking Supervision), the European General Data Privacy Regulation (The European Parliament and the Council of the European Union, 2016), the ISO 27000 standards (International Organization for Standardization, 2013), or the BSI Grundschutz (Bundesamt für Sicherheit in der Informationstechnik, 2019).

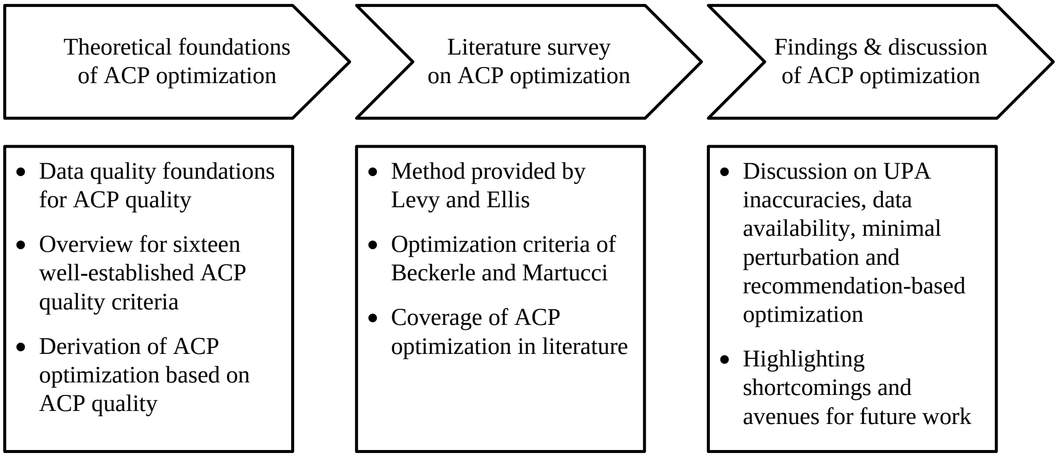
Theoretical foundations of ACP optimization
Literature survey on ACP optimization
Findings & discussion of ACP optimization
• Data quality foundations for ACP quality
• Overview for sixteen well-established ACP quality criteria
• Derivation of ACP optimization based on ACP quality
• Method provided by Levy and Ellis
• Optimization criteria of Beckerle and Martucci
• Coverage of ACP optimization in literature
• Discussion on UPA inaccuracies, data availability, minimal perturbation and recommendation-based optimization
• Highlighting shortcomings and avenues for future work

Figure 1: General approach of the study

advantages and limitations?. The contribution of this work is threefold: (i) We provide a definition of ACP quality as an instance of the data quality concept and supplement it with a collection of 16 ACP properties that are repeatedly used in literature to determine ACP quality. Building on this, we define ACP optimization as an improvement of the quality of existing ACPs. (ii) We conduct a structured literature survey based on the methodology proposed by Levy and Ellis (2006). Our scope is set for scientific publications describing means for optimization of ACPs that satisfy at least one of the six optimization objectives of Beckerle and Martucci (2013). After obtaining a literature catalogue for publications on optimization of ACPs, we categorize and analyze the literature catalogue. (iii) Finally, we build on findings from the literature survey and discuss important aspects of ACP optimization in more detail. At first, we discuss prototypical approaches to identify User Permission Assignment (UPA) inaccuracies, their advantages and disadvantages and their data requirements. Subsequently, we discuss the availability of three classes of data on which ACP optimization methods commonly rely in order to analyze the consequence of these data requirements. We then discuss the concepts of minimal perturbation and recommendation-based optimization and their addressing in existing literature. In addition, current shortcomings and research gaps are identified and avenues for future work can be highlighted. The discussion is presented in section 5. Figure 1 gives an overview of the general approach.

This work is focused on the Access Control Models (ACMs) Role-Based Access Control (RBAC) (Sandhu, 1998) and ABAC (Hu et al., 2014) since these are the most common ACMs. The standard RBAC model defines ACPs in the form of roles, which are bundles of permissions that can be assigned to subjects (e.g. employees) in a well-organized way. RBAC also offers the definition of constraints, which are statements that express negative authorizations, i.e. authorizations that must not be granted by the role set. ABAC ACPs in contrast are modeled as (dynamic) policies which make authorization decisions based on attributes of subjects (e.g. employees), objects (permissions) or the execution environment (e.g. the execution time). The term ACP hence refers to (constrained) RBAC roles as well as to ABAC policies (Samarati and de Vimercati, 2001). The Organization for the Advancement of Structured Information Standards (OASIS) defined the eXtensible Access Control Markup Language (XACML) standard, which provides a notation for expressing ABAC policies in an XML-based format as well as a reference architecture for a policy evaluation mechanism (Godik and Moses, 2003). XACML is the most important technical standard for ABAC and is explicitly addressed in many ABAC-related publications. Note that ABAC was often proclaimed as the successor of RBAC[2] since ABAC has the descriptive strength to express RBAC along with other attribute-based policies. E.g. Cheminod et al. (2018) showcase this behavior for an industrial use case. However, ABAC has still not reached the maturity of RBAC and has difficulties with practical adoption (Servos and Osborn, 2017; Puchta et al., 2019). Most issues of ABAC are rooted in the raised flexibility of attributes and ABAC policies which backfire as increased complexity and thus are less comprehensive for administrators and policy engineers (Servos and Osborn, 2017). Additionally, spill-over effects from RBAC to ABAC and vice versa can be observed in literature (Gardiyawasam Pussewalage and Oleshchuk, 2017; Qi et al., 2018; Nazerian et al., 2019). Further well-known ACMs, like Discretionary Access Control (DAC) and Mandatory Access Control (MAC) (Samarati and de Vimercati, 2001), or very recent approaches like Attribute-aware Relationship-Based Access Control (AReBAC) (Cheng et al., 2014; Chakraborty and Sandhu, 2021b,a) are not in the scope of this work.

The remainder of this paper is structured as follows. Section 2 presents work that is related to the quality of ACPs and their optimization. Section 3.2 defines ACP quality and ACP optimization and presents common quality criteria.

[2] In 2013 Gartner stated that "*by 2020 70% of enterprises will use attribute based access control [...] as dominant mechanism to protect critical assets [...]*" `https://www.gartner.com/en/documents/2607617`

Section 4 presents a literature survey on optimization of ACPs. Building on the findings of the survey, further aspects of ACP optimization are discussed in section 5. Finally, section 6 sums up the results and concludes this work.

2. Related Work

While existing studies present a comprehensive picture of RBAC and ABAC research and open challenges (Fuchs et al., 2011; Servos and Osborn, 2017), to the best of our knowledge no literature study has examined the more specific field of ACP optimization yet. Since publications that contribute to this field directly are presented and analyzed in the literature survey in chapter 4, this chapter serves to present research areas that are closely related to ACP quality optimization.

The assessment of ACP quality is a fundamental building block for their optimization. Since ACP quality comprises many dimensions, the scientific literature is heterogeneous and proposes many ways to assess ACP quality. This includes the definition of quality metrics (Jabal et al., 2019; Beckerle and Martucci, 2013) and distinct research realms that aim at individual quality-related objectives. Examples are ACP anomaly analysis, which aims at the identification of conflicts and redundancies, or XACML evaluation runtime analysis, which aims to find causes for a slow evaluation runtime. Moreover, ACP quality research is concerned with the question of how the quality of ACPs develops in real environments and which structural reasons are responsible for this. Researchers have documented that the quality of ACPs gradually deteriorates without targeted countermeasures, and have identified structural causes such as structurally determined over-granting (Xiang et al., 2019) or the role explosion problem (Elliott and Knight, 2010).

The initial creation of ACPs with high quality is addressed by ACP modelling approaches. Existing approaches differ greatly in terms of modelling objectives and their definitions of *optimal* ACPs (Kunz et al., 2015b; Mitra et al., 2016). Existing proposals typically focus on policy mining, i.e. the automatic generation of new ACPs based on *existing* authorization structures, or policy engineering, i.e. the (mostly manual) definition of new ACPs based on an *ideal* authorization structure. Hybrid approaches aim to utilize the advantages of mining and engineering (Cotrini et al., 2018; Fuchs and Pernul, 2008). While the initial creation of high-quality ACPs does not constitute an optimization, some RBAC or ABAC mining algorithms provide a maintenance mode that works on existing ACPs and were included in the literature survey. The repeated re-generation of an entire policy set is an alternative approach to keep ACPs up-to-date (Calo et al., 2019; Verma et al., 2019). It is based on the assumptions that policies remain unchanged during their whole life cycle and can be generated on a sufficiently high quality level without any human interaction. Instead of maintaining existing ACPs, this approach aims to ensure a sufficient quality be re-generating all policies from scratch on a regular basis.

3. Theoretical Foundations of ACP Optimization

Before presenting the methodology and findings of the survey, this chapter introduces the theoretical foundations for ACP quality and its optimization. Section 3.1 demonstrates that ACP quality is an instance of the data quality concept as defined by Wand and Wang (1996) and provides a definition of ACP quality based on it. To complement this definition, section 3.2 presents a collection of 16 well-established ACP quality criteria. Building on these preliminaries, section 3.3 concludes the theoretical foundation with a definition of ACP optimization.

3.1. Quality of Access Control Policies

Present research on ACP quality shows similarities to data quality research: There is consensus in both fields that quality is a multidimensional concept and cannot simply be assessed as universally *good* or *bad*. As in the field of data quality research, IAM research also applies different criteria to assess the quality of ACPs. While some of them only make sense in the context of ACPs (like the grade of automation or evaluation runtime), others are equivalent to data quality dimensions that are well-established outside the research realm of IAM (e.g. accuracy, understandability, completeness or redundancy). To the best of our knowledge, present research did yet not formally show that ACP quality constitutes an instance of the data quality concept. In this section, we show that the widely acknowledged data quality model by Wand and Wang (1996) is applicable to ACPs and provide a definition of ACP quality based on it.

The basis for an organization's access control decisions is its security policy (Sandhu and Samarati, 1994; Samarati and de Vimercati, 2001). A security policy is a collection of (often informal) requirements that define the authorizations of an organization. One of the most commonly quoted security policy requirements is the principle of least privilege, which states that a subject should not inherit more permissions than it needs to perform its tasks (Horne, 2011). A security policy may also comprise requirements that are not directly related to the organization's security, for example to fulfill organizational or regulatory needs. One example are Segregation of Duty (SoD) requirements, which define mutually exclusive authorizations and are commonly employed to avoid conflicts of interest. Access Control Policies (ACPs) are machine-processable rules that can be evaluated in a fully automated manner to determine which accesses a subject is allowed to make (Samarati and de Vimercati, 2001). Following these established definitions, ACPs are a data representation of the authorizations that are specified by an organization's security policy.

The data quality model by Wand and Wang (1996) defines an information system as an entity which exists parallel to a real-world system. The data stored in the information system is a representation of a perception of the real-world system. Through interpretation of this data, a user perceives a view of the real-world system as inferred from the information system. The process of creating data that represents real-world entities is called the representation transformation. The process of creating an interpretation of the representational data which resembles the original real-world entities is called interpretation transformation. If both, representation transformation and interpretation transformation, work correctly, the view of the real-world system as inferred from the information system is identical to the view of the real-world system gained from direct observation. Any disparity in between these views represents a data deficiency. The authors define four assumptions which need to be met for the model to be applicable:

(i) *The Representation Assumption: An information system is a representation of a real-world system as perceived by users.* The authorizations defined by an organization's security policy exist outside the information system and are hence (abstract) real-world entities. The ACPs that express them are consequently a data representation of real-world-entities, and the information system that stores them is a representation of a real-world system.

(ii) *The Interpretation Assumption: An information system is built for use by the user whose view of the real-world system is captured in the design of the system.* Wand and Wang explain that this assumption serves to ensure that the interpretation transformation (i.e. the process of transforming data back into perceivable real-world entities) will be able to map the data representation back to the original real-world entities. In the instance of ACPs, the design of the information system equals the view of the information system, because the representation transformation and the interpretation transformation are based on the same access control model: Any set of authorizations can be represented as a user-permission matrix. Both access control models in scope, RBAC and ABAC, allow to define an ACP set for every possible user-permission matrix, that will be mapped back the the exact same user-permission matrix.

(iii) *The Inference Assumption: The information system can create a perceptible representation from which the user can infer a view of the real-world system as represented in the information system.* Since every ACP set can be represented (and also visualized) as a user-permission matrix, an information system can always infer a view of the original authorizations that are represented by the ACP set.

(iv) *The Internal View Assumption: Issues related to the external view such as why the data are needed and how they are used are not part of the model.* This assumption is self-fulfilling as it is merely states that the model does not deal with external issues.

By showing that ACPs fulfill these four assumptions, we show that the data quality model by Wand and Wang (1996) is applicable to ACPs and that the concept of ACP quality constitutes an instance of the data quality concept. Present research widely agrees that the quality of data is best described as its "fitness for use" (Tayi and Ballou, 1998). In accordance with this definition, we define the quality of ACPs as their fitness for use with regard to one or more quality dimensions that reflect the application context of access control.

3.2. Established Quality Criteria

Present research applies many different criteria to evaluate the quality of ACPs, many of which include a concept of optimal quality. Beckerle and Martucci (2013) developed six criteria to determine well-usable ACPs sets and developed metrics to quantify these. Both Kunz et al. (2015b) and Mitra et al. (2016) present surveys on role mining approaches and point out objectives that role mining algorithms apply to achieve high-quality roles. Jabal et al. (2019) define a list of policy analysis criteria with implications to the policy quality. Besides that, a large number of both RBAC and ABAC related publications define quality criteria "on the fly" and often also define metrics for these quality criteria to approach a particular objective at hand.

In order to complement the definition of ACP quality, the remainder of this section presents a collection of properties of ACPs that are commonly applied in existing literature to evaluate the quality of ACPs. The collection comprises 16 properties, 14 of which have an optimum in terms of ACP quality. The remaining two criteria (usage and relevance) are often used in the context of quality assessment, but are not an expression of ACP quality themselves. Furthermore, seven of the presented criteria affect the evaluation of ACPs by the access control mechanism directly, while the remaining nine only implicitly affect their correct evaluation through factors such as error-proneness during policy administration or maintenance efficiency. Please note that this is not a complete list, as creating a full list of established quality criteria would require a structured, reproducible literature survey on its own. Individual quality criteria may be positively or negatively correlated, or not influence each other at all: For example, adding new rules to an ABAC policy will likely increase its UPA coverage and bring it closer to being complete. At the same time, the complexity of the policy is increased, which suggests a negative correlation between the objectives of minimal complexity and maximal completeness. Kunz et al. (2015b) present a dependency analysis of quality criteria applied during role mining. Despite that, to the best of our knowledge, the interaction of ACP quality criteria has not been analyzed by scientific research yet. Table 1 provides an overview of all 16 properties, their optima and whether or not the criteria affect the ACP evaluation directly.

Accuracy is the most important quality dimension for ACPs as it expresses the effective correctness of their access control decisions. It is hence directly related to the correct evaluation of the policies. The accuracy of an ACP set defines, how accurately it represents the authorizations defined by the security policy[3] (Beckerle and Martucci, 2013). There are two types of errors that decrease the accuracy of an ACP set: If an ACP set grants excessive UPAs, subjects inherit more permissions than they require. In contrast, missing UPAs mean that subjects require particular permissions, but are not

[3] Beckerle and Martucci (2013) refer to the Security Policy as "Access Control Policy". To avoid confusion, this work uses the more common term "Security Policy". The term "Access Control Policy" refers to the data representation of the authorizations defined by the security policy, which is subject to optimization.

Table 1
Common quality-related ACP properties

Property	Optimum	Affects Evaluation
Accuracy	max	yes
Excessive UPAs	min	yes
Missing UPAs	min	yes
Maintainability	max	no
Understandability	max	no
Sem. meaningfulness	max	no
Complexity	min	no
Redundancy	min	no
Conflicts	min	yes
Grade of automation	max	yes
Evaluation runtime	min	yes
Similarity to opt. state	max	no
Risk	min	no
Completeness	max	yes
Usage	ambiguous	no
Relevance	ambiguous	no

granted them by the ACP set. The challenge in identifying inaccurate UPAs lies in determining which UPAs *should* be granted. If the entire set of correct UPAs was known, an ACP set could be optimized for perfect accuracy in a fully automated manner.

Excessive UPAs are a violation of the principle of least privilege and can cause a security vulnerability. An example of how harm can be done by excessive authorizations is when a hospital employee publishes sensitive patient data, either accidentally or in malicious intent. Excessive authorizations can also be abused by external attackers, for example, if an authorized employee is blackmailed or his or her user account is hijacked. The minimization of excessive UPAs is the primary objective when modelling and administering ACPs and is required to enable an acceptable level of security. Finding excessive UPAs however poses a greater challenge, and over-allocated privileges often go unnoticed until they are misused for a malicious act. For this reason, excessive UPAs tend to accumulate over time, making targeted countermeasures necessary (Fuchs et al., 2014). The amount of excessive UPAs that are granted by an ACP set is hence a crucial indicator for its quality.

Missing UPAs keep subjects from doing their work and hence conflict with business continuity. For example, a company's supply chain could suffer outages because an employee lacks the authorization to post a goods receipt in the enterprise resource planning system. Missing UPAs have a higher visibility than excessive UPAs because their damage occurs relatively quickly: A subject who has been wrongly deprived of an entitlement can immediately thereafter no longer perform a certain task. The impact can be substantial since this can include very basic privileges, such as authorization to enter an organization's premises or to log into their work station.

Maintainability describes how well an ACP set can be administered and kept up-to-date (Benedetti and Mori, 2019; Cheng et al., 2019). Low maintainability makes an ACP set prone to errors and leads to a higher administration effort. Increasing an ACP set's maintainability is hence a prime objective of ACP optimization. The maintainability of an ACP set is influenced by several properties including its understandability, its complexity, its redundancy or the amount of conflicts that it contains, which can be assessed and optimized individually. Tool-supported administration of ACP also helps early on in keeping desired properties up-to-date (Seifermann et al., 2022).

Understandability expresses how well an ACP set can be understood by humans. It is closely related to maintainability and is often cited together (Cheng et al., 2019; Hummer et al., 2015). A cryptic ACP set that is hard to understand is also hard to administer or maintain: For example, a policy administrator could misunderstand the meaning of an ABAC policy and make erroneous changes that cause inaccuracies. Alternatively, an administrator could decide not to make changes at all to a policy that he or she does not understand, leading to fast obsolescence of that policy. Maintaining a good understandability is hence considered a prime challenge by several authors (Kunz et al., 2019; Meier et al., 2013). A key factor for understandable ACPs is semantic meaningfulness. Moreover, several authors proposed approaches for visualizing ACPs, which aim to improve the understandability of an ACP set without applying any changes to it (Puchta et al., 2019).

Semantic meaningfulness means that ACPs represent a human-understandable real-world concept. It is often argued to be crucial for ACP understandability (Xu, 2014; Molloy et al., 2010). Since semantic concepts can be described with attribute values, the semantic meaningfulness of an ACP can be assessed by measuring its accordance with semantically meaningful attributes (Xu, 2014). For example, a role that can be described as "This role grants all permissions that are required for all software developers" would accord to 100% to the value "software developer" of the semantically meaningful employee attribute "job title" and hence has a very high semantic meaning. Note that this definition does not require the ACP to be defined based on attributes itself. Moreover, an attribute-based ACP does not automatically have a higher semantic meaning than a role, since it can define a long list of permitted or denied UPAs which share little or no semantic meaning.

Complexity expresses the amount of elements that an ACP set consists of. For example, an ABAC policy that comprises 200 statements is likely more complex than one that contains only 5 statements. Similarly, a set of 50 roles with hierarchies among them and numerous permission assignments is likely more complex than a set of 5 roles with few permission assignments and no role hierarchies. Low complexity improves the maintainability of an ACP set and reduces the computational effort required for its evaluation. Existing research assesses the complexity of an ACP set in numerous ways, including the amount of roles and ABAC policies contained in it, the size of roles and ABAC policies and more specific measurements. The most generic definition of ACP complexity is the Weighted Structural Complexity (WSC), which is a weighted sum of all elements defined by the underlying ACM (Molloy et al., 2008; Xu, 2014). The complexity of an ACP set is among the most commonly cited quality indicators. Unlike its understandability or semantic meaningfulness, the complexity of an ACP set can be quantified objectively without requiring any further data.

Redundancy occurs if an ACP set defines positive or negative authorizations more than once. For example, an employee could inherit the permission to close customer requests in a ticketing system twice because he has the role "customer support employee" and the role "administrator of ticketing system". Similarly, a redundant negative authorization could occur for a bank employee if an ABAC policy defined that no bank employee who serves private customers may approve lending, and that no employee who is still in training may approve lending. Redundancy leads to an unnecessary bloating of ACP complexity. Moreover, ACP redundancy is a possible cause for administration errors, as a redundantly defined positive or negative authorization must be removed more than once for the change to become effective. If one of the redundant definitions is overlooked in the process, the ACP set obtains an inaccuracy. The impact can be substantial, for example when an emergency permission revocation process (i.e. a process where a subject is immediately stripped of all permissions, for example because the digital identity was stolen) fails because a redundant permission assignment is overlooked.

Conflicts exist within an ACP set if it defines both positive and negative authorizations for the same user-permission pairs. This means that a particular permission is both allowed and forbidden for the same user. For example, an ABAC policy could state that IT administrators have file access to an application server, while at the same time denying access to personnel data files for anyone outside the human resource department. While a static conflict is present within the effective UPAs that an ACP set realizes at a given time, dynamic conflicts are *potential* conflicts, i.e. conflicts that could arise due to the dynamic nature of the ACPs, but did not necessarily generate a contradiction yet (Dunlop et al., 2003). While ABAC or XACML ACP sets define both positive and negative authorizations by default, RBAC ACP sets can only contain conflicts if used with RBAC constraints (Sandhu, 1998) since unconstrained roles define only positive authorizations. If a conflict exists within an ACP set, the access control mechanism must resolve it in order to make an unambiguous authorization decision. This is achieved by applying a conflict resolution strategy which defines how to make authorization decisions if a conflict occurs. The XACML standard defines basic conflict resolution strategies (Moses, 2005) and many more sophisticated conflict resolution algorithms were proposed by researchers. However, conflict resolution only aims to enable the access control mechanism to make a deterministic decision *despite* the presence of conflicts. It does not update the ACP set to remove the conflict and hence does not constitute a quality optimization. Since conflicts make an ACP set's authorizations ambiguous, they are a possible cause for inaccuracies and reduce the ACP set's understandability. Moreover, real-time conflict resolution reduces the evaluation performance of the ACP set.

The **grade of automation** of an ACP set determines, to which extent the authorizations defined by it adapt to new situations dynamically without requiring manual updates. For example, an ABAC policy that defines authorizations based on an employee's department affiliation requires no updating of a policy definition if an employee moves into another department, since the employee's department attribute value would change, thus leading to an updated result in the evaluation of subsequent authorization requests. Attribute-based ACPs inherently offer the possibility of dynamic rule definitions, since changes in referenced attribute values also change the authorization decisions resulting from ACPs evaluation. The standard RBAC model (Sandhu, 1998) in contrast is static and cannot update authorizations automatically unless it is extended with an automation mechanism (such as Kern and Walhorn (2005); Al-Kahtani and Sandhu (2004); Aftab et al. (2015)). While ACP automation is is directly related to the policy evaluation, it is also a critical factor in ACP maintenance as it reduces administrative effort and prevents excessive and missing UPAs before they occur (Fuchs et al., 2014; Kunz et al., 2015a). However, it cannot make ACP maintenance obsolete, since an automation mechanism operates with a limited scope, and dynamic ACP definitions can out-date or be erroneous like static ones (cmp. section 4.3.6). Additionally, automation also eases other IAM processes like policy refinement, policy verification or conflict resolution (Cheminod et al., 2017, 2019).

The **evaluation runtime** of an ACP set determines how quickly it can be loaded and evaluated by an access control mechanism to answer an authorization request (Turkmen and Crispo, 2008; Miseldine, 2008). A sufficiently low evaluation runtime is critical if ACPs must be evaluated in real-time, since a pending authorization decision is a performance bottleneck for all relying application systems. As a result, users could spend considerable time waiting for simple button clicks to be executed, or performance-critical operations in an organization's IT infrastructure such as large data processing tasks could pile up. The real-time evaluation of authorization requests by a central access control mechanism is a requirement specified in the OASIS XACML reference architecture (Hu et al., 2014). The evaluation runtime of an ACP set can be influenced by many factors, like the amount of contained ACPs, or the amount, size, order of the rules that an ACP contains or the algorithm used (Marouf et al., 2011; Deng et al., 2021).

Similarity expresses how similar one ACP set is compared to another. For example, a set of 10 department roles is likely more similar to a set of 5 department roles and 5 roles that represent an employee's job title than it is to a set of 50 roles which represent employees' job titles. Similarity is often used as a quality criterion by measuring how similar the assessed ACP set is to another ACP set which is considered optimal (Narouei and Takabi, 2019; Hadj et al., 2017). By maximizing its similarity to the optimal ACP set, a new ACP set can be generated that resembles the structure of an existing, well-structured ACP set, but includes the results of the newly applied generation algorithm. Moreover, similarity can be used to estimate the amount of updates that is required to migrate from one ACP set to another (Vaidya et al., 2008). By applying high similarity to the original ACP set as an objective, an ACP optimization method can produce updates with *minimal perturbation*, which is expected to reduce the administrative effort required for implementing the changes.

Risk determines the impact of excessively assigned permissions for an ACP or an ACP set. For example, a policy that grants subjects full access to a banking system is more risky than a policy that grants subjects access to a WiFi hotspot. The risk of an ACP reflects the aggregated risk of the permissions that are permitted by it. ACP modelling or optimization methods can use risk minimization as an objective to reduce the impact of excessive permission assignments (Jin et al., 2016; Dos Santos et al., 2014). High risk is also an indicator for high maintenance priority and can serve as context information for policy engineers and reviewers, based on the assumption that a high risk value suggests a more restrictive handling than a low one (Fuchs et al., 2014).

Completeness determines the amount of UPAs that are covered by an ACP set. Examples for incomplete UPA sets could be a set of roles which contain only a subset of the permissions managed by an organization, or an ABAC policy that only contains attribute definitions to make authorization decisions for a subset of the requested subjects. If an ACP set does not cover UPAs, an access control mechanism is unable to determine an access control decision for the corresponding user-permission pair during policy evaluation (except for a standard fallback decision). The criterion of completeness is also often used in ACP modelling to determine to which extent a newly modelled ACP set expresses the UPAs defined by an input state (Kunz et al., 2015b). The term coverage is closely related and often used interchangeably, but does not necessarily contain a quality indication. The coverage of an ACP is an indicator for its relevance, based on the assumption that a policy that defines many authorization decisions is more important than a policy that defines few ones.

Usage determines how often an ACP was invoked, i.e. how often the authorizations defined by an ACP were requested and executed. For example, an employee in the goods receiving department of a company might use the authorization to debit a delivery every day. An example of an infrequently used UPA could be that a back office employee needs to submit a balance sheet only once a year. If an UPA is never executed, this is an indicator that it is not needed, i.e. it is excessively assigned according to the principle of least privilege. The usage of an ACP can be reconstructed via access logs (as defined in section 4.2.3), by counting the invocations of the UPAs that are covered by the ACP over a defined timespan. ACP usage can be used to determine UPA inaccuracies (cmp. section 5.1.2) and is hence an important tool for ACP quality assessment. Moreover, the usage of an ACP is an indicator of its relevance, assuming that an ACP that is often invoked is more important than one that is scarcely invoked (Pan et al., 2018; Hadj et al., 2018a).

Relevance expresses how strongly an ACP influences authorization decisions in productive operations. For example, a basic role that allows every employee to access their work stations is likely more relevant than a specialist role that allows few employees to create a new email distribution list. ACP relevance is commonly analyzed to determine priorities when assessing or optimizing ACP quality. In case of conflicting policies, relevance can be used to prioritize a policy that should overrule another (cmp. section 4.3.5). The relevance of a policy can also be an indicator of timeliness (Bauer et al., 2011), and ACPs with low relevance can be interpreted as a security risk since they are likely to be over-permissive (Jabal et al., 2019). Moreover, policies with higher relevance can be maintained with higher priority in order to improve the effectiveness of ACP maintenance (Hadj et al., 2019). The properties *Risk*, *Completeness* and *Usage* are closely related to the relevance of an ACP and often used as an indicator.

3.3. Optimization of Access Control Policies

Building on the definition of ACP quality, we define ACP optimization as an improvement of the quality of existing ACPs with regard to specified quality criteria. This definition has two important implications: First, since ACP quality is multidimensional, ACPs cannot be optimized towards a universal optimum, but only towards a particular optimization objective. For example, consider an update operation that assigns a new permission to a role in order to provide this permission to the employees which inherit the role. This optimization may reduce the amount of missing UPAs and hence constitutes an optimization with regards to accuracy. At the same time, adding a new assignment to the role set increases its complexity and hence reduces its quality with regards to this quality dimension. Second, ACP optimization requires that an existing ACP set is updated rather than a new ACP set being created from scratch. This implication seems obvious, as the quality of a data state can only be improved in comparison with a reference state, i.e. the original data state that existed before the optimization. However, since every update operation generates a new data state, every optimization of an existing ACP set could also be interpreted as the creation of an new ACP set. The optimization of an existing ACP set differs in its goal from modeling a

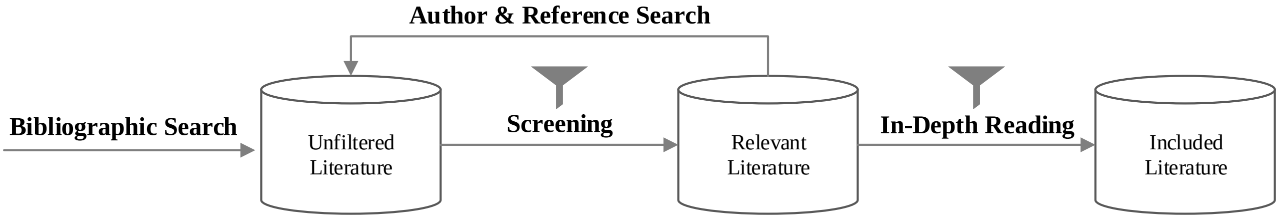

Figure 2: Applied Literature Research Process in accordance with Levy and Ellis (2006)

new one in that it aims to leave the existing set structurally intact. This goal stems from the need to maintain ACPs at a high quality level over time at a reasonable cost: Daily operations, such as an employee's department change, or the integration of new application systems into an organization's IT infrastructure, require frequent updates to an ACP set. Both neglecting such updates and sub optimal updates have a negative impact on the quality of the ACP set. Without continued quality measures, it is common for ACP sets to proliferate over time, while changing real-world conditions and proprietary policy updates cause their quality to degrade (Xia et al., 2014; Hu et al., 2011; Xu et al., 2017; Xiang et al., 2019). Unlike ACP modelling, ACP optimization aims to make partial changes to a (possibly very large) ACP set while leaving the remainder of the ACP set unchanged. As a result, an ACP set that was modelled with high quality can be kept on high quality with significantly lower effort than was required for its initial modelling. Moreover, by leaving an ACP set structurally intact, ACP optimization is able to retain the (often informal) semantic meaning of the existing ACPs that may be known only to human policy engineers. One possible approach to ensure that an ACP set remains structurally intact during optimization is the concept of "minimal perturbation", which is discussed in detail in section 5.3.

4. Literature Survey on Access Control Policy Optimization

4.1. Survey Methodology

The literature survey follows the methodology proposed by Levy and Ellis (2006). It aims to examine the existing body of research on ACP optimization (as defined in section 3.3) and provide a structured analysis of the research field. Since ACP optimization has to follow defined optimization objectives, we worked out six optimization objectives that are relevant and well established in existing literature. These six optimization criteria are presented along with further categorization criteria in section 4.2. We define the scope of literature included in the survey as *"Scientific publications that propose means for optimization of roles or attribute-based ACPs with regard to at least one of the six defined optimization objectives"*.

In accordance with the applied survey methodology, the literature research was started with a bibliographical database search. We used combinations of generic keywords like "maintenance", "improvement", "optimization", "correction" and more specific keywords for the distinct optimization objectives, such as "redundant" or "redundancy". Structured permutations of these keywords were entered in the online databases ACM Digital Library[4], IEEE Digital Library[5] and Google Scholar[6]. All publications that were not obviously related to another topic were added to an initial list of "unfiltered literature". We also included all publications from the ACM Symposium on Access Control Models and Technologies (SACMAT)[7] conference from the years 2001 to 2021 in this list. Every publication in the "unfiltered literature" list was then screened, which means that we read it superficially do determine whether it is relevant with regards to the survey topic. In this step, we read the title, abstract, introduction and conclusion and used the document search function to determine how the keywords applied during the bibliographical search were used. Doing so, we narrowed down the list of unfiltered literature to a second list of "relevant literature", which needs to address either the quality of ACP in general or one of the optimization objectives. All publications that were regarded relevant for the topic were then read in depth to determine whether they fit the survey scope and could be added the final list of "included literature". This list formed the ACP optimization literature catalogue which was categorized and analyzed. Since a bibliographical search could only serve as an entry point, we conducted author and reference search for all publications in the list of "relevant literature" and added the resulting publications back into the initial list of "unfiltered literature". Repeating the screening process for these publications, we executed a recursive search that allowed for a deep exploration of relevant research realms. A schematic overview of the applied literature research process is given in Figure 2.

Due to its broad scope, the survey comprised a heterogeneous literature base. As a result, the literature was hard to grasp with keywords. The majority of relevant results was yielded via author and reference search. Many publications in the "included literature" list do not explicitly define ACP optimization as maintenance or quality optimization, but rather define distinct objectives with a narrow scope which are semantically equivalent. Moreover, many relevant

[4] http://dl.acm.org/
[5] http://www.computer.org/
[6] http://scholar.google.com/
[7] http://www.sacmat.org/

Table 2
Coverage of selected quality criteria in literature

Quality criterion	Coverage
Excessive UPAs	Fuchs et al. (2014); Hill (2006); Jaferian et al. (2014); Hummer et al. (2015); Puchta et al. (2019)
Missing UPAs	Benedetti and Mori (2018, 2019); Fuchs and Pernul (2010); Colantonio et al. (2012); Meier et al. (2013)
Redundancy	Guarnieri et al. (2013); Shamoon et al. (2012); Hu et al. (2013); Mitra et al. (2016); Kunz et al. (2015b)
Conflicts	Hounder (2010); Deng and Zhang (2017); Dia and Farkas (2012); Shamoon et al. (2012)
Complexity	Mitra et al. (2016); Kunz et al. (2015b); Servos and Osborn (2017); Currey et al. (2020); Fuchs et al. (2014); Molloy et al. (2010)
Grade of automation	Fuchs et al. (2014); Kunz et al. (2015a); Hu et al. (2010a); Kern and Walhorn (2005); Al-Kahtani and Sandhu (2004); Aftab et al. (2015)

research realms used other keywords than we would have expected: For example, the realm of "XACML anomaly analysis" aims at the identification (and sometimes removal) of anomalies in XACML policies, which can be either conflicts or redundancies and hence fit the survey scope. Since we did not know of this research realm beforehand, we hardly could have found it using keywords.

4.2. Research and Categorization Criteria

As seen in chapter 3, ACP optimization is not universal, but can only improve the quality with respect to defined optimization objectives. To define the research scope of the survey, we screened existing literature and selected six optimization objectives that are central for ACPs' fitness for use. These six optimization objectives serve as criteria for selecting relevant literature for this study (cmp. section 4.1) and are presented in detail in the following section 4.2.1.

Beside a textual analysis, the applied survey methodology suggests to categorize the literature catalogue. For a meaningful categorization, the selected criteria need to be concrete enough for in-depth insights while covering a heterogeneous literature catalogue. This leads us to three categories within this survey: (i) The first category is the targeted ACM of a publication. The majority of analyzed publications (except Hummer et al. (2015, 2016)) can be categorized as either RBAC or ABAC related. (ii) The next category is the optimization objective which serve simultaneously as research criteria. (iii) Another category is the contributed ACP optimization research artifact of the publication. (iv) Finally, the used data of the optimization method is analyzed as the last category. While the distinction between RBAC and ABAC is self-explanatory, the remaining categorization criteria are presented in sections 4.2.1 to 4.2.3.

4.2.1. Optimization objective

Existing literature defines several quality criteria on the basis of which ACPs can be optimized (cmp. section 3.2). To clearly define and narrow the scope of the survey, we searched the literature for optimization goals that are critical to the fitness for use of ACPs. Beckerle and Martucci (2013) conduct semi-structured expert interviews and a literature analysis to identify critical requirements for obtaining usable ACP sets. Based on their results, they argue that the main aim of ACP optimization should be to improve the accuracy and maintainability of ACPs. They specify these requirements and work out six optimization objectives that serve these two goals. The authors also develop metrics for the quantification of these criteria and conduct two user studies to evaluate them. To the best of our knowledge, this is the only scientific publication that documents a structured research process for developing ACP optimization objectives and provides a conclusive evaluation. The optimization criteria are: (i) "Allow no more than the owner wants to be allowed.", (ii) "Allow everything the owner wants to be allowed.", (iii) "A rule must not be fully covered by another rule of the same rule set.", (iv) "Two rules belonging to the same rule set must not conflict.", (v) "Minimize the number of rule set elements." and (vi) "Minimize maintenance effort in a changing system.". Each of these six optimization criteria can be mapped to one ACP quality dimension that was presented in section 3.2. We reformulate them to accord with the common literature terminology based on the addressed quality dimensions: *(i) Reduce excessive UPAs*, *(ii) Reduce missing UPAs*, *(iii) Reduce redundancy*, *(iv) Reduce conflicts*, *(v) Reduce complexity* and *(vi) Increase grade of automation*. We conducted a literature search to confirm the relevance of these six quality dimensions. Table 2 lists further publications that underline their importanc for ACPs' fitness for use. Note that this list is not exhaustive. Due to their thorough foundation in existing literature and their frequently argued importance for optimization, we selected these six optimization objectives as the basis for the literature survey.

4.2.2. Research Artifact

The analyzed publications on ACP optimization are heterogeneous not only in terms of the addressed optimization objectives, but also in terms of their contributions. We identified four types of research artifacts that are repeatedly presented to contribute to the optimization of ACPs: (i) Optimization process models analyze ACP optimization from a business perspective. They define process steps, roles and responsibilities and analyze how the technical optimization can be embedded into a changing real-world environment. (ii) Optimization algorithms define formal ways to modify

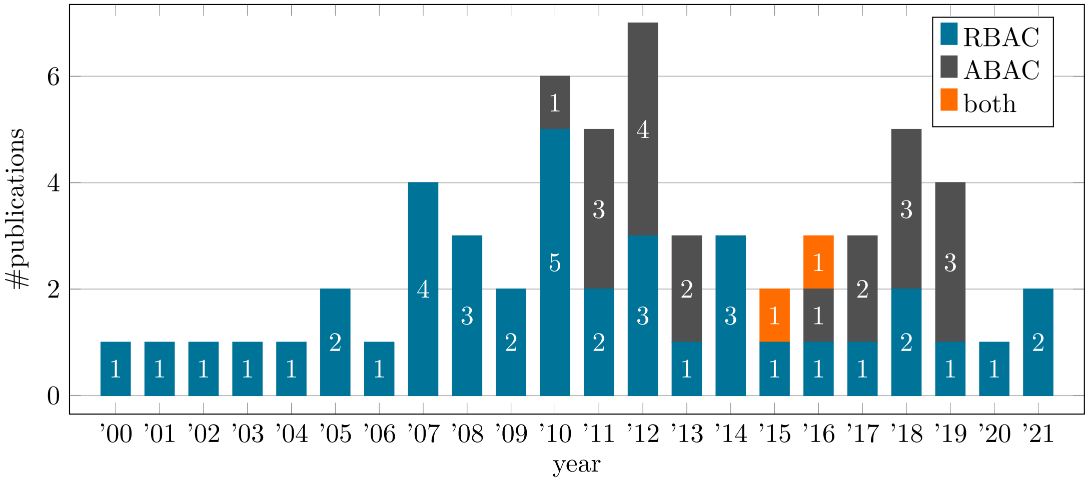

Figure 3: Publications over time

ACPs in order to improve their quality. They receive an existing ACP set (and possibly supplemental data) as input and generate an optimized ACP set as output. (iii) Optimization tools aid humans in the semi-automated optimization of ACPs. Publications with this kind of contribution present optimization tools to demonstrate how tool-supported ACP optimization can be done. (iv) ACM extensions propose ways to enhance ACMs to improve ACP quality. In the analyzed literature catalogue, this type of research artifact is only present in the form of RBAC extensions that enhance role definitions in order to provide automation. Note that we were not able to categorize the research artifact of a publication into one of these standardized categories in 5 cases.

4.2.3. Data usage

For the last criteria group, the survey literature was analyzed for the kind of data that was used to perform ACP optimization. We applied the conceptual IAM data model proposed by Kunz et al. (2019) to achieve an integrated view on the processed data. The model defines the central data entities that are processed in IAM and their relations towards one another. In addition to RBAC and ABAC, it integrates the conceptual entities defined in six central IAM technology standards[8] and proposes a terminology that is suited to cover the integrated concepts. According to this definition, a digital identity is a representation of a human user, which can be a record of an employee that is stored and processed in a human resource management system. Within an application system, digital identities possess accounts, to which permissions are assigned. Permissions can be hierarchically nested, in which case they are inherited transitively. Parallel to the definitions within an application system, digital identities can be assigned roles (as defined by RBAC), which inherit permissions through assignment and can also be hierarchically nested. Beside that, permissions can be granted via policies, which are ABAC or XACML policies in the context of this work. Note that in our terminology, both roles and policies are ACPs. The model defines a context entity, which is a scenario that can be evaluated by a policy (for example environmental conditions in an ABAC policy). At last, the model defines an attribute entity, which expresses a property of a digital identity, an account or a permission and can be evaluated by policies.

We identified a total of five types of data that were repeatedly used for ACP optimization: The *ACP set*, a *user-permission matrix*, *entity attributes*, *access logs* and *update logs*. From this set, we chose the use of entity attributes, access logs and update logs as categorization criteria. Processing of the ACP set and the user-permission matrix was not analyzed since these data types are trivial: The existing ACP set is the most basic type of data and must be known in order to be optimized. It comprises the roles or ABAC policies that are in effect and are updated in the course of the optimization. A user-permission matrix (also called access control matrix or UPA set) is the most basic representation of the permission assignments that an ACP set grants (Molloy et al., 2010). It is a boolean matrix which holds a true of false value for every possible user-permission combination. A user-permission matrix is commonly visualized in the form of an access grid and is not limited to a particular ACM (Meier et al., 2013). An optimization that updates an ACP set needs to verify that it did not create inaccuracies. Since it is the most basic permission assignment information available, a user-permission matrix is commonly assumed to be available for any ACP optimization effort.

Entity attributes are properties of entities other than the optimized ACPs themselves. As defined by Kunz et al, commonly expressed entities are digital identities, accounts and permissions. Attributes are often used to give ACPs semantic meaning: Since an attribute reflects a real-world property, binding an ACP to a given attribute can bind it to

[8] Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAML)/Shibboleth, Service Provisioning Markup Language (SPML), Open Authorization (OAuth), System for Cross-domain Identity Management (SCIM) and XACML

its semantic meaning (Molloy et al., 2010). Optimization approaches that require entity attributes need to process actual value expressions of concrete entities. In contrast, a method that restructures a given XACML policy by reordering its attribute statements without processing any entity attribute values would not be classified into this category. The reliance on entity attributes is a limiting factor since their availability may be limited. Moreover, the use of entity attributes for ACP optimization means that the results are dependent on the value of said attributes at the time of the optimization, meaning that an optimization result may lose validity when the attribute values change.

Access logs express historic accesses of users to permissions. While their notations differ, access logs can be displayed as a tuple <*S,O,A,R*> that represents a historic access request, with *S* being the requesting subject, *O* being the requested object, *A* being the requested action and *R* the result of the request, i.e. *permit* or *deny* (Xiang et al., 2019). Note that some publications only consider successful permission invocations, thus reducing access logs to a tuple <*S,O,A*> of permitted access requests. Access logs provide valuable insight on the actual need of permissions and can help to identify missing or excessive permission assignments.

Update logs are the second type of historic data used for ACP optimization. Update logs contain information on past changes of IAM related entities, for example a modification of an ACP, the creation of a new user account or the change of an employee's department affiliation. Update logs can be used to identify real-world events that provide important ACP update information (for example, the job change of an employee might require a change of his or her permissions) and can provide insight on the development of an ACP set over time. To the best or our knowledge, no scientific publication exists that defines the structure of IAM update logs.

4.3. Criteria-Based Analysis

4.3.1. Overview

The literature survey yielded 61 publications that provide means for optimization of existing RBAC and ABAC ACPs since the year 2000. Out of these publications, 42 address the optimization of roles, 21 address the optimization of attribute-based policies, and two address both ACMs. The optimization of ACPs has been addressed continuously over the past 20 years. While the peak of interest for RBAC optimization occurred between 2007 and 2014, the topic receives steady attention to this day, underlining that role optimization remains a relevant research subject. The first analyzed ABAC optimization paper was published in 2010. The tables 3 and 4 present all analyzed publications and their categorization according to the survey criteria defined in section 4.2. Figure 3 depicts all analyzed sources ordered by their year of publication.

4.3.2. Considered Optimization Scenarios

To connect ACP optimization with real-world scenarios, it is helpful to consider optimization scenarios. While various themes for optimization scenarios are possible, e.g. IAM goals (Hummer et al., 2018) or usage of popular techniques like access reviews (Groll et al., 2021), it is reasonable to take a closer look at maturity and automation as its driver. Like shown by Schrimpf et al. (2021) a driver for higher maturity is automation which requires or builds upon underlying optimization methods. In the sense of automation, ACP optimization scenarios can thus be considered as manual, semi-automated and automated.

In manual optimization, a human administrator or policy engineer updates part of an ACP set based in their individual context knowledge. Manual adjustments to an ACP set are commonly done in daily operations, e.g. to grant employees new permissions after their responsibilities changed. Another example are access reviews, where a responsible human (e.g. a department head) tries to find excessively assigned permissions manually and marks them for revocation (Jaferian et al., 2014). Due to the lack of automation, manual optimization is limited to small ACP sets or subsets of larger ACP sets. Another optimization scenario is the fully automated updating of an ACP set. In this scenario, an ACP optimization algorithm generates optimization steps for an ACP set. The resulting changes will be implemented in the underlying applications automatically. This approach has the disadvantage that an algorithm has no understanding of the semantic structure of an ACP set. As a result, the structure of the optimized ACPs may differ greatly from their original structure. In addition, fully automated methods can hardly be integrated into established change management processes, which require that changes to ACPs must be confirmed by a responsible employee. The third scenario is semi-automated optimization. Semi-automated processes try to bridge the gap between manual and automatic optimization by enabling humans with technical support to optimize a very large set of ACPs. A common example are recommendation-based optimization methods, which generate possible ACP updates automatically, and delegate them to responsible humans for decision. Updates will only become effective if the responsible human agrees to them. Semi-automated optimization can be supplemented by ACP visualization procedures and other data analysis techniques which are not optimization methods themselves. The concept of recommendation-based optimization is discussed in section 5.4.

The publications analyzed during the survey cannot always be clearly assigned to one of these scenarios. We observe that publications that propose an optimization algorithm typically assume fully automated optimization. Still, some optimization algorithms allow that the resulting change steps are delegated to humans in the form of recommendations, thus enabling them for semi-automated optimization (Han et al., 2012; Hu et al., 2010a; Rao et al., 2021; Benedetti and Mori, 2018, 2019). Publications that present an optimization tool aim to support humans in the semi-automated optimization of ACPs by definition. Of

Table 3
Categorized literature for ABAC optimization

Publication	Optimization Objective						Research Artifact				Data Usage		
	Reduce Excessive UPAs	Reduce Missing UPAs	Reduce Complexity	Reduce Redundancy	Reduce Conflicts	Increase Grade of Automation	Optimization Process Model	Optimization Algorithm	Optimization Tool	ACM Extension	Entity Attributes	Access Logs	Update Logs
Argento et al. (2018)	■							■			■	■	
Benkaouz et al. (2016)			■					■					
Cheng et al. (2019)					■								
Deng and Zhang (2017)					■			■					
Dia and Farkas (2012)					■			■					
Guarnieri et al. (2013)				■				■					
Hadj et al. (2017)			■	■				■					
Hadj et al. (2018a)				■	■			■				■	
Hadj et al. (2018b)				■	■			■					
Hadj et al. (2019)					■			■				■	
Hein et al. (2011)					■			■					■
Hounder (2010)					■			■					
Hu et al. (2011)				■									
Hu et al. (2012)				■	■			■					
Hu et al. (2013)				■	■			■					
Hummer et al. (2015)	■	■					■				■	■	
Hummer et al. (2016)	■	■					■				■	■	
Narouei and Takabi (2019)			■					■			■		
Oberholzer (2011)			■		■			■					
Shamoon et al. (2012)					■			■					
Stepien et al. (2012)			■	■				■					

the publications analyzed that present a process model, all except Strembeck (2010) assume that changes to an ACP set can be generated by algorithms, but must be delegated to a human for decision before becoming effective. It is noticeable that publications that aim to embed ACP optimization into an organization's processes largely reject fully automated optimization (except Strembeck (2010)). Publications that propose an extension of an ACM define ways to extend the structure of a role set with automation logic. However, they do not define a concrete optimization scenario in which a role set is updated to inherit this automation logic. ACM extensions are thus rather a blueprint for ACP optimization and not limited to a concrete optimization scenario.

4.3.3. *Reduce Excessive & Missing UPAs*

We identified nine publications that aim at identifying and rectifying inaccurate UPAs. Out of these nine publications, six address both missing and excessive permission assignments. Another two exclusively aim at excessive permission assignments and one addresses only missing permission assignments. Fuchs et al. (2014) propose a process model for RBAC optimization. Hummer et al. (2015, 2016) present a process model for both RBAC and ABAC optimization. Benedetti and Mori (2018) present both an RBAC optimization process model and a Max-SAT algorithm that uses access logs to identify missing permission assignments and adjust roles while minimizing their complexity. They expand it in a follow-up publication to address excessive permissions as well (Benedetti and Mori, 2019). Baumgrass (2011) and Zhang et al. (2013) both use access logs to identify missing or excessive UPAs and adjust roles accordingly. Argento et al. (2018) use access logs to identify excessive permission assignments and update ABAC policies. Groll et al. (2021) propose to use negative access review decisions, i.e. decisions in which a human reviewer identified excessive UPAs, to identify similar UPAs and generate revocation recommendations for them, thus amplifying the impact of manual identification of excessive permissions. To the best of our knowledge, this is the only publication proposing a method that uses a data source other than access logs to identify excessive permissions automatically. Note that approaches which provide automation for RBAC updating are also relevant for this optimization objective as they aid the closely related objective of *preventing* UPA inaccuracies. These approaches are analyzed in section 4.3.6.

The key challenge for correcting missing or excessive UPAs is to identify them. Once found, such inaccuracies can be corrected in a fully automated manner. Throughout the literature analysis we identified three basic approaches for identifying missing or excessive permission assignments: First, manual identification requires that a human overlooks

Table 4
Categorized literature for RBAC optimization

Publication	Optimization Objective						Research Artifact				Data Usage		
	Reduce Excessive UPAs	Reduce Missing UPAs	Reduce Complexity	Reduce Redundancy	Reduce Conflicts	Increase Grade of Automation	Optimization Process Model	Optimization Algorithm	Optimization Tool	ACM Extension	Entity Attributes	Access Logs	Update Logs
Aftab et al. (2015)						■				■	■		
Al-Kahtani and Sandhu (2002)						■				■	■		
Al-Kahtani and Sandhu (2003)						■				■	■		
Baumgrass (2011)	■	■										■	
Benedetti and Mori (2018)		■	■				■	■				■	
Benedetti and Mori (2019)	■	■	■				■	■				■	
Chakraborty and Ray (2006)						■				■	■		
Fuchs et al. (2014)	■	■	■				■					■	■
Gal-Oz et al. (2011)				■				■					
Groll et al. (2021)	■										■		■
Guo et al. (2008)			■					■					
Han et al. (2012)						■		■		■	■		
Herzberg et al. (2000)						■				■	■		
Hu et al. (2010a)						■	■	■	■				
Hu et al. (2010b)						■		■					
Hu et al. (2016)						■		■					
Huang et al. (2012)						■		■		■	■		
Hummer et al. (2015)	■	■					■				■	■	
Hummer et al. (2016)	■	■					■				■	■	
Kern and Walhorn (2005)						■				■	■		
Lu et al. (2014)						■							
Lu et al. (2017)						■				■			
Molloy et al. (2010)			■	■				■			■		
Ni et al. (2009)						■				■	■		
Pan et al. (2018)			■					■				■	
Pang et al. (2007)				■	■			■					
Pang et al. (2008)			■	■				■					
Parkinson et al. (2020)						■				■			
Rao et al. (2021)						■		■		■		■	
Saffarian et al. (2009)						■				■	■		
Shafiq et al. (2012)					■			■					
Sheng and Osborn (2004)						■				■	■		
Strembeck (2005)				■					■				
Strembeck (2010)				■			■					■	■
Takabi et al. (2007)						■		■		■	■		
Takabi and Joshi (2010)			■					■					
Vaidya et al. (2008)						■		■					
Xia et al. (2014)			■					■					
Yi-qun et al. (2007)						■				■	■		
Zhang et al. (2007)			■					■					
Zhang et al. (2013)	■	■						■				■	
Zong et al. (2011)						■				■	■		

an ACP set and tries to find assignments which he or she knows to be inaccurate. This task is commonly executed in the form of access reviews, which are the de-facto standard process for this task and mandated by central regulation frameworks. Second, usage-based approaches analyze access logs for information on historic permission invocations and can thus be used to determine the set of actually needed UPAs. Third, update history based approaches analyze historic updates of the ACP set to identify missing or excessive permission assignments. The advantages and data requirements of these basic concepts are discussed in section 5.1.

4.3.4. *Reduce Complexity*

The survey yielded ten publications that aim to reduce the complexity of an RBAC state and five publications that aim to simplify an ABAC state. Out of the RBAC related publications, Fuchs et al. (2014) are the only one that addresses the problem exclusively at the process level. They propose a process model for the optimization of roles regarding accuracy and complexity. Benedetti and Mori (2018, 2019) define an RBAC maintenance process model and an algorithm that optimizes the complexity of an RBAC state. Xia et al. (2014) define the Role Refinement Problem and provide an algorithmic analysis, but also define process-level requirements for role refinement. The remaining six RBAC-related publications address complexity optimization strictly on a technical or algorithmic perspective. Takabi and Joshi (2010), Pan et al. (2018) and Benedetti and Mori (2018, 2019) aim to minimize the amount of updates required to perform the generated optimizations. Molloy et al. (2010) use user attributes to increase the semantic meaning of the roles that are either mined or maintained in their approach. They also argue for the use of access logs and update logs, but do not use them themselves. Pan et al. (2018) use access logs to calculate the usage of roles, which they use to determine its relevance as one indicator of its quality. The five ABAC-related publications propose algorithms to reduce the complexity (Oberholzer, 2011; Benkaouz et al., 2016; Narouei and Takabi, 2019) or both complexity and redundancy (Stepien et al., 2012; Hadj et al., 2017) of an existing ABAC state. Hadj et al. (2017) and Narouei and Takabi (2019) calculate the similarity of input and output state to minimize the required updates. Narouei and Takabi (2019) require user and permission attributes for the refinement of an ABAC policy set. Altogether, ACP optimization for complexity requires no data other than the ACP set and the user-permission matrix since it is a mere algorithmic problem that can be solved without semantic knowledge. As long as the resulting UPAs remain unchanged, the ACP set can be rearranged for a lower complexity in a fully automated manner. However, context knowledge (for example provided by access logs) can provide further inside on the meaning of ACP and benefit the optimization efforts.

4.3.5. *Reduce Redundancy*

We identified 14 publications that propose methods for reducing redundancy in existing roles or attribute-based policies. Strembeck (2005) define a high-level maintenance process for roles that aims at reducing redundancy. Strembeck (2010) present a role engineering tool that supports humans in finding and removing redundant role-permission assignments. The remaining twelve publications propose algorithms for the reduction of redundancy in roles or XACML policies. We observe that optimization methodologies often consider redundancy reduction a secondary optimization, as four of the analyzed publications aim to reduce complexity and five rectify conflicts as their primary objective. While the reduction of redundancy is a non-trivial algorithmic problem, it can be addressed very well with algorithms since solving it at core requires no data other than the optimized ACP set and its UPA set. Consequently, most of the analyzed approaches use no further data. However, Molloy et al. (2010) use entity attributes to mine new and optimize existing roles with redundancy as one optimization objective. Strembeck (2005, 2010) propose the use of context constraints during role optimization and define a process for engineering it, using access logs and update logs as data sources.

4.3.6. *Reduce Conflicts*

A large research area aims at the resolution of ACP conflicts. However, most of these publications propose methods to generate a permit or deny decision during the evaluation of ACPs despite the existence of a conflict. They do not modify the underlying ACPs to correct the error and hence to not fit into the scope of this survey. Altogether, we identified 14 publications that propose methods for the reduction of ACP conflicts. Cheng et al. (2019) define a methodology for removing inconsistencies (which include conflicts) in rule-based ACPs such as ABAC policies. Despite their method being named "removing process", we chose not to categorize it as a process model as it is a technical methodology that is closer to defining an algorithm than a (business) process model. The remaining 13 publications define conflict reduction algorithms. Both Pang et al. (2007) and Shafiq et al. (2012) propose graph optimization algorithms to remove constraint conflicts on an RBAC state. The remaining eleven publications propose algorithms for the reduction of rule conflicts in an XACML policy set.

A crucial challenge with direct impact to the ACP quality is to decide whether a given conflict should be converted into a permit or deny decision before updating the ACP set accordingly. A wrong correction decision would still remove a conflict in the ACP set, but lead to a UPA inaccuracy in return. The analyzed literature offers different approaches to make this decision. Hu et al. (2011, 2012) and Deng and Zhang (2017) embed existing conflict resolution strategies in their approach. Hu et al. (2013) refine the previously proposed approach to apply conflict resolution strategies for individual segments of an ACP set, and apply removability constraints to individual ACPs in order to achieve a more fine-grained correction decision. Both Dia and Farkas (2012) and Hadj et al. (2018b) extend ACPs with scope constraints to support the correction decision. Dia and Farkas (2012) also generate recommendations for decisions on conflict removal which can be delegated to responsible entity owners if the correction algorithm generated a non-mandatory update. Hein et al. (2011) generate update logs that track the changes to an ACP and use these in order to support administrators and allow them to restore a previous state which is known to be correct. Shafiq et al. (2012) apply priorities to role constraints and evaluate these during the conflict correction. At last, Hounder (2010) defines algorithms to aid human policy administrators in conflict correction, thus proposing a semi-automation of conflict correction which can incorporate the semantic knowledge of a human in the resolution decision.

We observed that conflict correction research primarily focuses on attribute-based policies with only two publications for role conflict correction yielded by the survey. We explain this by the fact that attribute-based ACPs are prone to conflicts since they define both positive and negative authorizations by default. Roles themselves in contrast define only positive authorizations and can only be conflicted if used with constraints (Ahn and Sandhu, 2000). Nevertheless, role constraints are crucial part of role-based access control and are required to express role restrictions for fundamental process-level requirements such SoD policies. The current state of research on RBAC conflict correction does not reflect this.

4.3.7. Increase Grade of Automation

Since ABAC is dynamic by nature, providing automation is only relevant for RBAC in the scope of the survey. Out of 22 publications that provide means for RBAC automation, we group 16 publications into three classes: Rule-based automation, learning-based automation and trust-based automation.

Rule-based automation extends roles with rules which evaluate attributes to automatically update employee-role assignments (Al-Kahtani and Sandhu, 2002; Kern and Walhorn, 2005; Yi-qun et al., 2007; Han et al., 2012; Huang et al., 2012; Aftab et al., 2015), role-permission assignments (Han et al., 2012; Huang et al., 2012; Aftab et al., 2015; Parkinson et al., 2020) or role hierarchy assignments (Al-Kahtani and Sandhu, 2003). They can hence be interpreted ad hybrids of RBAC and ABAC. Since attributes can reflect a real-world property with semantic meaning, attribute-based role automation can help to improve the semantic meaningfulness of roles. A single rule can cover a multitude of role model updates and can hence greatly reduce administrative effort for a role set. Nevertheless, automation rules can outdate over time and need to be maintained together with the role set.

Learning-based automation approaches use existing entitlement information as input to find valid role updates. Sheng and Osborn (2004) use entity attributes to generate employee-role updates and Ni et al. (2009) use entity attributes to generate role-permission updates. Rao et al. (2021) utilize access logs to automatize employee role assignment. Learning-based automation works without static rules and hence does not necessarily require manual definitions. Unlike rule-based automation, learning-based automation may react to changes that the role engineers did not consider when modelling the role set. However, learning-based automation requires learning data as input, and the quality of its updates is limited by the quality of the input data. As a result, we argue that learning-based optimization might be better suited to maintain a role set with an already high quality, than to optimize a role set with low quality from scratch.

Trust-based automation assigns roles to users based on the users' trustworthiness (Herzberg et al., 2000; Zong et al., 2011; Chakraborty and Ray, 2006; Takabi et al., 2007; Saffarian et al., 2009). Trust-based approaches differ from the two previously presented ones as they do not assign roles based on the tasks that a user has to perform (as required by the principle of least privilege), but try to assess a set of maximum permissible permissions. This is typically done by calculating trust scores for users and defining minimum trust scores that a user needs to have in order to be granted certain roles or permissions. Trust-based automation approaches are designed for open environments where the users are not completely known (for example collaborative platforms like Wikis). They are not designed for classical inhouse identity management environments where employees with a defined task range are managed (cmp. Fuchs and Pernul (2007); Fuchs et al. (2009)), but aim to mitigate risk when little user information is available.

We identified another six publications that provide means for automatic role updating, but do not try to identify possible updates themselves: Hu et al. (2010b) generate migration paths to automatize the implementation of role model updates in application systems. Hu et al. (2010a) define a tool and a process which aid the automatic updating of role-permission assignments by checking whether an update is achievable with a given set of constraints. Lu et al. (2014) and Hu et al. (2016) evaluate role updating algorithmically to determine the complexity of automatic checking for role-permission and role-role assignments. Lu et al. (2017) propose a role generalization algorithm that aims to optimize roles for automatic assignment via user authentication queries. Vaidya et al. (2008) aim to enable role mining algorithms for optimization of an existing role set by generating a state that is as similar as possible to an existing role set and an optimal one.

Altogether, we conclude that RBAC automation is thoroughly covered by research. It can help to reduce administrative effort and maintain a high role model quality. However, it does not make role maintenance obsolete, since all types of automation have limitations and cannot be expected to react to all future changes adequately. Beside a limited scope, RBAC automation configurations themselves can be erroneous or outdate over time just like a classical role set.

5. Discussion

Building on the results of the literature survey, this chapter discusses important aspects of ACP optimization. In doing so, we analyze several concepts that are repeatedly addressed in the survey literature and play a critical role in the optimization of ACPs. Since the identification of UPA inaccuracies is the most critical challenge in correcting excessive and missing UPAs, we work out three prototypical approaches commonly found in the literature and discuss their advantages and limitations as well as their data requirements in section 5.1. We then examine the availability of the three types of data that were included in the literature analysis in section 5.2. Subsequently we discuss the concepts of minimal perturbation and of recommendation-based optimization in sections 5.3 and 5.4. At last the limitations of this work are discussed in section 5.5.

5.1. Identification of UPA Inaccuracies

Finding inaccurate UPAs is the key challenge when optimizing ACPs for accuracy. If the complete set of required UPAs is known, an ACP set can be optimized for perfect accuracy in a fully automated manner. However, such a set does not usually exist in an explicit form, and finding out which permissions any subject should be granted in accordance with an organization's security policy is difficult and time-consuming. ACP accuracy optimization methods hence face the primary challenge to identify as many inaccuracies as possible in order to be able to correct them. During the literature analysis we identified three prototypical approaches to identify UPA inaccuracies, which we present and discuss below.

5.1.1. Manual Identification

Manual identification primarily concerns excessive UPAs, because manual identification of missing UPAs can simply be initiated by the affected users, for example by ordering a missing permission in a structured process through an IAM entitlement shop (Hornsteiner et al., 2020). The most obvious approach to identify excessive UPAs is to have a human who knows an organization's security policy check the effective UPAs and search for inaccuracies. A process that works by this scheme is known by the name of *Access Reviews*. Access Reviews are a standard IAM process that is executed periodically and aims to identify and rectify excessive UPAs (and sometimes other data inaccuracies like inaccurate attribute values) (Jaferian et al., 2014). Their execution is strongly driven by external requirements raised by compliance frameworks or IT security standards (Fuchs and Pernul, 2007; Royer, 2008). During an Access Review, a responsible human reviews a list of entitlement assignments for the users, ACPs or permissions within their responsibility, and decides whether they are still necessary. Based on this decision, the reviewer will either confirm an assignment's correctness, or refuse to confirm it, in which case the assigned permissions will be revoked. Access Reviews are a central measure to prevent the accumulation of excessive permissions (Jaferian et al., 2014). However, they are error-prone and tend to overlook and confirm excessive UPAs due to a number of structural challenges (Groll et al., 2021). In particular, the sheer amount of data that has to be processed during Access Reviews can be overwhelming, and the decision whether a user requires a particular permission is difficult to make. In case of uncertainty, Access Review decisions are biased towards confirming an existing assignment for a number of reasons: (i) Since an existing UPA likely has been subject to an approval process before becoming effective, a reviewer has reason to believe that an existing assignment has a legitimate reason. (ii) If a reviewer makes a false decision, only a false revocation of an existing UPA would have an immediate consequence, because an employee would no longer be able to execute a certain task as a result. In contrast, an erroneous confirmation of an already granted UPA is unlikely to have an immediate effect as long as the resulting security vulnerability is not abused for malicious action. (iii) For the same reason, the visibility of an erroneous revocation within an organization is higher than the visibility of an erroneous confirmation, which adds a social incentive for reviewers to simply confirm existing UPAs as to avoid visible errors. As a result, the effectivity of Access Reviews is limited. While some research effort tries to aid users in the effective execution of Access Reviews (Jaferian et al., 2014; Bobba et al., 2005), to the best of our knowledge only one approach was proposed that aims to measure the quality of Access Review decisions in order to identify erroneous UPA confirmations automatically (Groll et al., 2021). Despite the importance of Access Reviews and the difficulties practitioners face in implementing them, the tasks of aiding companies and users in the execution of Access Reviews and measuring their effectively were scarcely addressed by research.

5.1.2. Usage Based Identification

Usage based identification of UPA inaccuracies evaluates historic permission invocations to determine which permissions a particular user actually needed in the past. It is hence directly related to the principle of least privilege, which states that any user should not inherit more permission than necessary to perform his or her tasks. Several ACP optimization approaches use access logs for usage based identification of UPA inaccuracies (cmp. section 4.3.3). If access logs are available, excessively assigned permissions can be found relatively easy by comparing historic permission usages with the actually granted permissions of an ACP set. Any permissions that are granted for a user, but were not used within a specified time frame (e.g. over the last year) can be interpreted as excessively assigned and are hence candidates for removal. This approach is well suited to detect large quantities of excessively assigned permissions that were overlooked during UPA maintenance. However, it can struggle to distinguish excessive UPAs from accurate, but rarely used ones (e.g. a yearly creation of a report). Moreover, this approach can only identify excessive UPAs as long as they are not used (possibly even with malicious intent).

If access logs do not only include historic permission invocations, but also access requests which have been denied, they can be used to identify missing UPAs: If a permission invocation is often requested and denied, this can be an indicator that the permission is handled too restrictively. However, the identification of missing UPAs leaves more room for interpretation than the identification of excessive UPAs and requires finding a balance between business continuity considerations and security considerations. We hence argue that the paramount value of access logs lies in identifying excessive UPAs. Especially UPA accumulation, i.e. the accumulation of permission assignments that were once granted but are no longer required by a user (for example because the user's responsibilities have changed since the permission assignment) is a major problem that can be addressed very well with the described approach. However, usage based correction is limited by the availability of access logs, which is discussed in section 5.2.

5.1.3. *Update History Based Identification*

The use of update histories for the optimization of ACP accuracy was proposed by several authors. Some also provided examples how update histories could be used for this cause. Fuchs et al. (2014) cite the update history of a role model as an important source of context information for optimizing a role model. However, they only consider it for manual evaluation by human role engineers. Strembeck (2010) names trace management as necessary requirement for maintaining complex models, but does not go into the details of ACP maintenance in this regard. Hein et al. (2011) propose an algorithm to generate update logs that track the changes made to an ACP set and use them to enable administrators to perform a rollback operation and restore previous states. Mitra et al. (2016) propose to use update logs to identify roles which decayed over time in order to select candidates for maintenance. While the authors do not elaborate on this, using historical data to assess timeliness is common practice in other areas of data quality research (Heinrich and Klier, 2009). Molloy et al. (2010) name the event pattern of a user losing several permissions, followed by being assigned several permissions within a short time frame as an indicator for a job change event and argue that such events provide valuable context information for the creation of high-quality roles. While not explicitly naming it as a role optimization use case, they also stress that the creation of high-quality roles is not a once-and-for-all effort, but must be succeeded by continuous role optimization efforts. They provide further examples for meaningful events within a role model's update history, arguing that permissions which are often assigned or unassigned together are likely related to the same real-world context, thus providing an example of using update logs to determine the semantic meaning of permissions. Molloy et al. (2010) also argue that historic update information provides evidence on legacy permissions which should be removed from the role model. To the best of our knowledge, no methods were proposed that use update logs for the correction of missing or excessive UPAs. However, Groll et al. (2021) propose an approach to find erroneous Access Review decisions (i.e. decisions where an excessive permission assignment was falsely confirmed) automatically. The proposed approach, which uses revocation decisions from the analyzed Access Review as input, performs learning-based outlier detection to find other Access Review decisions which are likely to be over-permissive. Although the method does not explicitly use historic update information, it can be abstracted as an analysis of UPA unassignment events, which could also be taken from update logs instead of Access Review decisions. Altogether we conclude that ACP update histories are a promising source of context information for identifying excessive or missing UPAs. To this day, existing research has made few attempts to make use of this information source.

5.2. Data Availability

To highlight the limitations resulting from the data requirements of the analyzed ACP optimization approaches, we will discuss the availability of the investigated data types. As defined in in section 4.2 we assume the conceptual IAM model of Kunz et al. (2019) for an integrated view on the processed data. The three classes of data for which we analyzed the survey literature catalogue are entity attributes, access logs and update logs. First of all, the use of all of these data sources requires an integrated view on the processed IAM data. This requirement is not trivial since an identity management infrastructure comprises a wide range of heterogeneous data sources. These data sources may either provide a centralized data view (e.g. a human resource system that stores the employee data for the entire organization or a directory system that serves as a centralized user account and permission data storage) or exist as numerous decentralized data sources that have different data schemes and data storages (e.g. application systems that manage individual user accounts and permissions for their own application context) (Fuchs and Pernul, 2007). While having an integrated data view on the identity management infrastructure is a common prerequisite for IAM measures, the creation of such a view requires significant effort and should therefore not be neglected (Fuchs et al., 2009).

5.2.1. *Availability of Attributes*

Attributes are properties of entities within the IAM data view other than the ACPs themselves. If an integrated view of the related entities exists, entity attributes are from a technical point of view easily obtainable. Due to the sensitivity of personal data however, the processing of attributes of digital identities may be restricted. While this means that sensitive attributes (like an employee's loan details or sick leave history) might not be available for ACP optimization, it should not prevent attributes of digital identities from being processed altogether, as many central business-related attributes (such as departmental affiliation and job title) do not fall into this category.

5.2.2. *Availability of Access Logs*

The availability of access logs is more complex. The OASIS XACML reference architecture presumes that authorization requests are sent to a central access control mechanism with a Policy Decision Point (PDP) at its core. If this is the case, access requests (both granted and denied ones) can be logged completely. However, we argue that this is a prototypical architecture which comes with several challenges: (i) Creating a central access control mechanism requires that all applications which are subject to IAM efforts delegate their authorization decisions to a PDP. This requires significant integration effort, which may not be economically feasible. (ii) The central evaluation of access requests requires, that every authorizable action of a user within an application (which may in case of highly configurable systems come down to every single click on a button) be sent to the PDP, evaluated and answered in real-time before the user action is executed. Since this has to happen during the run time of the application with negligible time delay, this requires a highly traffic resistant, performant and available IT infrastructure and almost immediate access request evaluation on the PDP side, again implying efforts that may very well exceed the benefits of centralizing authorization decisions. (iii) This approach requires that all applications which are subject to IAM are technically able to delegate every authorization decision to a PDP, which is currently not the standard.

The alternative approach to obtain access logs is by decentralized logging within the applications where the permission invocations occur. The decentralized logs must then be collected and integrated in a central log stream. While the decentralized approach does not require real-time decision delegation, it also requires high integration effort and is limited by the ability of all managed applications to log permission invocations: Although several industry software solutions do provide access logs or access statistics[9] , this is not a primary use case for many software products and hence not the standard.

Beside technical and economical limitations, access logs contain particularly sensitive personal data as they enable work monitoring, and their use may be restricted due to requirements of the legislator or employee representatives. This problem is also known in other fields which monitor IT infrastructure for security, for example in the context of Security Information and Event Management (SIEM) (Menges et al., 2021). Due to the sensitivity of the processed data, it may be necessary to limit surveillance to the most critical areas, for example, users with a particularly high number of privileges (like administrators) or critical applications (like a banking system). Overall, we conclude that access logs are difficult to develop as a data source. In addition to significant technical hurdles, the economic viability of monitoring activities across the board is not always given. Moreover, ethical and legal hurdles must be considered. For these reasons, we argue that the availability of access logs that include all user accounts and permissions from all the applications in the scope of an organization's IAM cannot be assumed to be standard. Approaches that rely on the availability of access logs for optimization of ACPs may have limited applicability in practice.

5.2.3. *Availability of Update Logs*

Update Logs contain the update history of the entities processed in IAM. All availability limitations that apply to entity attributes hence also apply to entity attributes within the update logs. Apart from that, update logs are easy to obtain: If an integrated view of the IAM data exists, then its changes can be monitored, too. The creation of update logs is a common functionality for industrial IAM systems, which have a central view over IAM entities since they are used to manage ACPs and to provision them to the related application systems. Moreover, legal regulations imply the (partial) existence of update logs: In order to check compliance with the principle of least privilege, an auditor must be able to investigate when and how an ACP set was updated to grant users new permissions. For this reason, we argue that update logs are a readily accessible data source. Nevertheless, to the best of our knowledge, no scientific model exists that describes how IAM update logs are structured. Foundation work is still missing for the scientific development of this data source and possible applications.

5.3. Minimal Perturbation

The concept of minimal perturbation aims to optimize ACPs with as few changes as possible (Vaidya et al., 2008). ACP optimization algorithms often address this goal by defining maximal similarity to the original state as a secondary optimization objective. The ability to optimize ACPs with few changes is an important factor that can determine the practicality of an optimization method for two reasons: (i) The fewer changes needed to achieve quality improvement, the lower the administrative effort required to implement those changes. Since many organizations are mandated to execute approval processes for changes to the ACP set, an ACP update often requires the interaction of humans (e.g. a role owner or employee owner) before it can be enforced. Moreover, since access control is often enforced decentralized within the application systems managed in an identity management infrastructure, changes to the ACP set have to be provisioned into the related application systems in order to take effect. Minimizing the perturbation hence improves the economic viability of ACP optimization. (ii) The fewer changes needed to achieve quality improvement, the higher the degree to which an ACP set remains structurally intact. ACPs are typically modeled to reflect semantic concepts. This goes from single policies reflecting simple statements (like "any employee of this organization may access the WiFi hot-spot") up the whole ACP set, which can be designed, for example, to reflect the organizational structure of a company (Fuchs and Pernul, 2008; Xu, 2014). Furthermore, ACPs incorporate (often informal) contextual knowledge that may only be known to human policy engineers. By largely preserving the structure of an ACP set after its initial creation, the semantic meaning of the ACPs and the contextual knowledge that has gone into them are also preserved, which is beneficial for the quality of the resulting ACP set and prevents it from becoming unrecognizable after repeated execution of optimization methods. Since minimal perturbation can be critical for the practical viability of an optimization method, we argue that it is not adequately addressed by existing research: Out of 38 publications in the literature survey catalogue that propose an algorithm for ACP optimization, only 10 consider minimal perturbation when defining their optimization objectives (Benedetti and Mori, 2018, 2019; Hadj et al., 2017; Hu et al., 2010a; Pan et al., 2018; Narouei and Takabi, 2019; Rao et al., 2021; Takabi and Joshi, 2010; Vaidya et al., 2008; Zhang et al., 2013).

[9] For example, Microsoft Exchange (https://www.microsoft.com/de-de/microsoft-365/exchange/email) provides detailed logs on email distribution list usages and SAP ERP (https://www.sap.com/) provides aggregated statistics of transaction invocations.

5.4. Recommendation-Based Optimization

In this section, we discuss the concept of recommendation-based optimization. Recommendation-based optimization means that ACP update steps produced by an optimization method are not applied directly to the underlying ACP set, but bundled into optimization recommendations. An optimization recommendation represents one or more changes to the ACP set and can be delegated for decision to responsible human decision makers. Only when the decision maker agrees to the optimization, the associated update steps are applied to the ACPs set. If no manual decision is required, an optimization recommendation can alternatively also be executed automatically.

The use of recommendations addresses business-related constraints that can be crucial for the practical applicability of an optimization method: (i) Approval processes often require that changes to the policy set be approved by a responsible subject (like a department head, an application owner or policy owner) before being implemented. By bundling optimization decisions into human-decidable steps, organizations can use (semi-)automatized optimization methods while remaining compliant. (ii) Business constraints may exist (like regulatory requirements or practical hurdles) that prohibit some changes in the policy set, but are unknown to optimization algorithms. Moreover, automatic optimization methods struggle to understand the semantic structure and context meaning of an ACP set. As a result, updates may be proposed that are technically valid, but make little sense in the real world. With recommendations, a human decider can serve as a quality gate for optimization steps and prevent changes that are problematic or forbidden. Since domain experts are likely to have specific context knowledge, including them into the optimization process can improve the effectiveness of the optimization altogether. We argue that recommendation-based ACP optimization constitutes a hybrid model of fully-automated and manual optimization. In the closely related domains of RBAC and ABAC policy modelling, which face the closely related challenge to automate the creation of semantically meaningful ACPs with high quality, hybrid approaches have also been proposed and gained broad acceptance (Fuchs and Pernul, 2008; Das et al., 2018).

A number of optimization methods analyzed in the survey rely on recommendations. Fuchs et al. (2014) propose a process model for RBAC optimization that includes a mechanism for generating new role extensions (i.e. new assignments of roles to employees, permissions or other roles). They define that every role extension is delegated to a role owner for decision. Hummer et al. (2015, 2016) propose a process model for the optimization of RBAC or ABAC ACPs based on usage patterns. Similar to Fuchs et al., they define that every optimization step is processed via a recommendation mechanism. Benedetti and Mori (2018) define an RBAC maintenance process that identifies missing role-permission assignments algorithmically. In Benedetti and Mori (2019), this process is extended to also find excessive permission assignments. They stress that the found "violations" must be presented to a security administrator who may confirm or reject them before they are processed to generate RBAC model updates. This approach differs from the previous recommendation-based optimization approaches as the recommendation mechanism is active in a preliminary stage of the optimization process, not at the end of it. Groll et al. (2021) define a methodology that analyzes confirmation decisions of access reviews to find possible errors. The results are then delegated to a human policy analyst for inspection. If the analyst confirms the error, the underlying UPA is revoked, which means that every found possible error is equivalent to a recommendation to revoke the underlying UPA. Baumgrass (2011) propose a process-centric methodology for refining existing RBAC states. They use event logs to derive RBAC artifacts (i.e. components of a role model like employee-role, role-role or role-permission assignments) to extend a role set. The authors stress that these artifacts are merely update candidates and must be integrated into the role model manually by a human decider, e.g. with the help of a role engineering tool. Hu et al. (2010a) present a tool which generates update steps that migrate an ACP set into a target state. Their tool can recommend different migration paths to a human, who has to decide which (if any) of them might be suitable. Rao et al. (2021), Han et al. (2012) and Chakraborty and Ray (2006) propose RBAC extensions which generate recommendations for updating user-role assignments. While Rao et al. (2021) and Han et al. (2012). propose rule-based approaches, Chakraborty and Ray (2006) propose to recommend user-role-assignments on the basis of users' trustworthiness.

It is noticeable that especially those publications that take a process perspective in optimizing ACPs require that optimizations are recommended to human deciders before they are implemented. Publications that propose algorithms for optimization often leave this requirement out. As a result, many optimization methods are not applicable for recommendation-based optimization. In order to be used for recommendation-based optimization, an optimization method must fulfill three requirements: (i) An optimization method needs to create individually decidable ACP update steps. This means that it must be possible to discard one step without having to discard the remaining ones. (ii) The decidable update steps must be small enough to be meaningfully decidable by a single decider. (iii) The decidable update steps must be human-understandable. The requirements (ii) and (iii) indicate that small update steps are preferable to large ones. Furthermore, it is helpful if the optimization method has a concept of the semantic structure of the ACP set. For example, ACPs that affect specific application systems can be treated and recommended in a bundle that is delegated to the respective system owner for decision. These requirements show that not every optimization algorithm can be reasonably adapted for recommendation-based optimization: For example, the output of a graph optimization algorithm that takes a role set as input and generates a single optimized role set cannot easily be converted into recommendations, since the decider would have to accept or reject the entire optimized role set. Otherwise, if only one included update step were rejected, the remaining optimization would be incomplete and the resulting role set would likely be erroneous. In a real-world business environment with requirements for business continuity, change management processes and regulatory requirements, the ability to embed ACP optimization steps into the existing process landscape is a crucial factory for the practical applicability of an optimization method.

5.5. Limitations of this Work

The content of this study is limited by the selection of ACP quality and optimization criteria. The 16 quality criteria presented in section 3.2 provide a broad overview of the quality consideration of ACPs in the scientific IAM literature. Nevertheless, we do not claim completeness, as a representative summary would require a structured, reproducible literature survey. Moreover, the research scope of the literature survey is limited by the six selected optimization objectives defined in section 4.2.1. To ensure the relevance of the chosen criteria, we adopted the six optimization criteria developed by Beckerle and Martucci (2013) and confirmed them as central to ACPs' fitness for use in further literature research. Nevertheless, this represents a preselection that influences the literature unearthed in the survey and the findings based on it.

The literature research process of the survey also represents a possible limitation: We found that many publications use proprietary terminology, or aim at specific problems that are semantically equivalent to ACP maintenance or optimization, but not formulated as such (for example "ACP anomaly resolution"). The heterogeneity of the literature poses a challenge to structured literature research. Although we have adhered to the methodology presented in section 4.1, we cannot rule out with certainty that we have overlooked literature that fits the scope of the survey.

Due to its broad scope, this work cannot provide a detailed comparative evaluation of the various approaches that researchers proposed for optimization of ACPs. The analyzed publications differ in terms of their contributed research artifact, their grade of automation, and their optimization objective. It is difficult to compare a process model for ACP optimization with an algorithm for ACP optimization, or to compare an algorithm for reducing redundancies with an algorithm for correcting conflicts. We tried to address this challenge by formulating a generalized problem description, and by formulating analysis criteria are applicable among the heterogeneous literature base. Moreover, this work does not evaluate the analyzed approaches for their effectiveness or correctness, since a formal evaluation of the analyzed publications would have exceeded its scope.

Furthermore, the quality of ACPs is not well developed scientifically. While existing research proposes numerous proprietary definitions of *good* ACPs, there are few publications that address the topic holistically. The six optimization objectives on which the survey is based are well documented in the literature. Other quality-related ACP properties that are presented in section 3.2 are cited less frequently. Further groundwork on ACP quality is desirable.

6. Conclusion

This work studied the optimization of ACPs with the following contributions: (i) We show that the quality of ACPs constitutes an instance of the data quality concept as defined by Wand and Wang (1996) and provide a definition of ACP quality based on it. We give a broad overview of ACP quality as perceived in the IAM literature and provide a definition of ACP optimization. (ii) We present a structured literature survey that categorizes and analyzes existing methods for ACP optimization. We point out that the reduction of excessive and missing UPAs comes with the paramount challenge of *identifying* these inaccuracies. Once found, such inaccuracies can be corrected in a fully automated manner. Since identifying UPA inaccuracies relies heavily on context data, the availability of such data is a bottleneck. The reduction of complexity and redundancy in contrast are algorithmic problems which can be solved without any data other than the existing ACP set and its user-permission matrix. Approaches for conflict correction face the challenge of deciding whether to resolve a conflict by granting or permitting a particular UPA, and existing literature offers many strategies to address it. The issue of providing automation for role updating is thoroughly covered. Many approaches can be categorized into one of three classes: Rule-based automation, learning-based automation and trust-based automation. (iii) Building on the structured survey, we analyze important aspects of ACP optimization in more detail. In particular, we discuss three basic concepts for identifying UPA inaccuracies and their advantages and limitations. We analyze three prototypical types of data on which ACP optimization methods commonly rely and discuss their availability: While many ACP optimization methods require the existence of access logs, we point out that their availability may be limited in practice. Update logs, on the other hand, have not yet been precisely defined by researchers or incorporated into optimization methods. However, many authors emphasize their value as a possible source of information in optimizing ACPs. Furthermore, we analyze the concepts of minimal perturbation and recommendation-based optimization and argue their relevance for ACP optimization. Although both concepts are known to the research community, we point out that most optimization methods do not take them into account.

Future work has several starting points to contribute to the optimization of ACPs. (i) The quality of ACPs is often addressed in research, but it lacks theoretical foundations. There is no comprehensive literature survey that compiles existing ACP quality dimensions in a structured and reproducible process (with the exception of the quality criteria works named in section 3.2). Also, the correlations of quality dimensions, i.e. which dimension influences another positively, negatively or not at all, are not comprehensively studied. Furthermore, there is no model that analyzes which

ACP quality dimensions are subordinate or superordinate to others: For example, accuracy or maintainability seem to be aggregated dimensions that are based on different properties of ACPs. A structured overview, e.g. in the form of a topology, could provide clarity here and help standardize terminology in IAM research. (ii) Update logs are hardly developed as a data source. While especially the identification of UPA inaccuracies urgently needs other data sources than access logs, there is no definition of how update logs are structured or can be obtained in an IAM infrastructure. This groundwork could be used to develop methods for the (semi-)automated identification of UPA accuracies, which could be used to improve security and business continuity in organizations. (iii) While purely algorithmic or purely process-related optimization methods have been proposed by research, there is a lack of publications describing how algorithmic optimization can be embedded in existing process landscapes. Methodologies that implement optimization procedures, e.g. with tool support, could help to close this gap. Furthermore, practical reports or case studies would be a valuable aid for researchers trying to align theoretical ACP optimization methods with real-world needs. An analysis of the extent to which fully or semi-automated optimization is applicable in practical scenarios would also be helpful. (iv) Finally, there is very little research on access reviews: Although numerous organizations need to invest significant effort into the execution of access reviews on a regular basis, little research effort has been made to improve their limited effectiveness.

The optimization of ACPs is a relevant and ongoing research topic, and practitioners are left with major challenges that are yet to be solved. Existing research differs greatly in scope and terminology, which indicates that the research topic has yet to gain maturiy. We hope to contribute to its standardization with this work.

CRediT authorship contribution statement

Sascha Kern: Conceptualization, Methodology, Validation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Project administration. **Thomas Baumer:** Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Sebastian Groll:** Conceptualization, Writing - Review & Editing. **Ludwig Fuchs:** Supervision. **Günther Pernul:** Writing - Review & Editing, Supervision, Funding acquisition.

Acknowledgement

The research leading to these results was supported by the German Federal Ministry of Education and Research as part of the DEVISE project (https://devise.ur.de/).

References

Aftab, M.U., Habib, M.A., Mehmood, N., Aslam, M., Irfan, M., 2015. Attributed role based access control model, in: 2015 Conference on Information Assurance and Cyber Security (CIACS), pp. 83–89. doi:10.1109/CIACS.2015.7395571.

Ahn, G.J., Sandhu, R., 2000. Role-based authorization constraints specification. ACM Trans. Inf. Syst. Secur. 3, 207–226. URL: https://doi.org/10.1145/382912.382913, doi:10.1145/382912.382913.

Al-Kahtani, M., Sandhu, R., 2002. A model for attribute-based user-role assignment, in: 18th Annual Computer Security Applications Conference, 2002. Proceedings., pp. 353–362. doi:10.1109/CSAC.2002.1176307.

Al-Kahtani, M., Sandhu, R., 2003. Induced role hierarchies with attribute-based rbac, in: Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 142–148. URL: https://doi.org/10.1145/775412.775430, doi:10.1145/775412.775430.

Al-Kahtani, M., Sandhu, R., 2004. Rule-based rbac with negative authorization, in: 20th Annual Computer Security Applications Conference, pp. 405–415. doi:10.1109/CSAC.2004.32.

Argento, L., Margheri, A., Paci, F., Sassone, V., Zannone, N., 2018. Towards adaptive access control, in: Data and Applications Security and Privacy XXXII. Springer International Publishing, pp. 99–109. URL: https://doi.org/10.1007/978-3-319-95729-6_7, doi:10.1007/978-3-319-95729-6_7.

Bauer, L., Cranor, L.F., Reeder, R.W., Reiter, M.K., Vaniea, K., 2009. Real life challenges in access-control management, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. p. 899–908. URL: https://doi.org/10.1145/1518701.1518838, doi:10.1145/1518701.1518838.

Bauer, L., Garriss, S., Reiter, M.K., 2011. Detecting and resolving policy misconfigurations in access-control systems. ACM Trans. Inf. Syst. Secur. 14. URL: https://doi.org/10.1145/1952982.1952984, doi:10.1145/1952982.1952984.

Baumgrass, A., 2011. Deriving current state rbac models from event logs, in: 2011 Sixth International Conference on Availability, Reliability and Security, pp. 667–672. doi:10.1109/ARES.2011.104.

Beckerle, M., Martucci, L.A., 2013. Formal definitions for usable access control rule sets from goals to metrics, in: Proceedings of the Ninth Symposium on Usable Privacy and Security, pp. 1–11.

Benedetti, M., Mori, M., 2018. Parametric rbac maintenance via max-sat, in: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 15–25. URL: https://doi.org/10.1145/3205977.3205987, doi:10.1145/3205977.3205987.

Benedetti, M., Mori, M., 2019. On the use of max-SAT and PDDL in RBAC maintenance. Cybersecurity 2. URL: https://doi.org/10.1186/s42400-019-0036-9, doi:10.1186/s42400-019-0036-9.

Benkaouz, Y., Erradi, M., Freisleben, B., 2016. Work in progress: K-nearest neighbors techniques for abac policies clustering, in: Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, Association for Computing Machinery, New York, NY, USA. p. 72–75. URL: https://doi.org/10.1145/2875491.2875497, doi:10.1145/2875491.2875497.

Bobba, R., Gavrila, S., Gligor, V., Khurana, H., Koleva, R., 2005. Administering access control in dynamic coalitions, in: Proceedings of the 19th Conference on Large Installation System Administration Conference - Volume 19, USENIX Association, USA. p. 23.

Calo, S., Manotas, I., de Mel, G., Cunnington, D., Law, M., Verma, D., Russo, A., Bertino, E., 2019. AGENP: An ASGrammar-based GENerative policy framework, in: Policy-Based Autonomic Data Governance. Springer International Publishing, pp. 3–20. URL: https://doi.org/10.1007/978-3-030-17277-0_1, doi:10.1007/978-3-030-17277-0_1.

Chakraborty, S., Ray, I., 2006. Trustbac: Integrating trust relationships into the rbac model for access control in open systems, in: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 49–58. URL: https://doi.org/10.1145/1133058.1133067, doi:10.1145/1133058.1133067.

Chakraborty, S., Sandhu, R., 2021a. Formal analysis of rebac policy mining feasibility, in: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, New York, NY, USA. p. 197–207. URL: https://doi.org/

10.1145/3422337.3447828, doi:10.1145/3422337.3447828.

Chakraborty, S., Sandhu, R., 2021b. On feasibility of attribute-aware relationship-based access control policy mining, in: Barker, K., Ghazinour, K. (Eds.), Data and Applications Security and Privacy XXXV, Springer International Publishing, Cham. pp. 393–405.

Cheminod, M., Durante, L., Seno, L., Valenza, F., Valenzano, A., 2017. Automated fixing of access policy implementation in industrial networked systems, in: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), pp. 1–9. doi:10.1109/WFCS.2017.7991947.

Cheminod, M., Durante, L., Seno, L., Valenza, F., Valenzano, A., 2019. A comprehensive approach to the automatic refinement and verification of access control policies. Computers & Security 80, 186–199. URL: https://www.sciencedirect.com/science/article/pii/S0167404818303870, doi:https://doi.org/10.1016/j.cose.2018.09.013.

Cheminod, M., Durante, L., Valenza, F., Valenzano, A., 2018. Toward attribute-based access control policy in industrial networked systems, in: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1–9. doi:10.1109/WFCS.2018.8402339.

Cheng, Y., Park, J., Sandhu, R., 2014. Attribute-aware relationship-based access control for online social networks, in: Atluri, V., Pernul, G. (Eds.), Data and Applications Security and Privacy XXVIII, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 292–306.

Cheng, Z., Royer, J.C., Tisi, M., 2019. Removing problems in rule-based policies, in: ICT Systems Security and Privacy Protection. Springer International Publishing, pp. 120–133. URL: https://doi.org/10.1007/978-3-030-22312-0_9, doi:10.1007/978-3-030-22312-0_9.

Colantonio, A., Di Pietro, R., Ocello, A., Verde, N.V., 2012. Visual role mining: A picture is worth a thousand roles. IEEE Transactions on Knowledge and Data Engineering 24, 1120–1133. doi:10.1109/TKDE.2011.37.

Cotrini, C., Weghorn, T., Basin, D., 2018. Mining abac rules from sparse logs, in: 2018 IEEE European Symposium on Security and Privacy (EuroS P), pp. 31–46. doi:10.1109/EuroSP.2018.00011.

Currey, J., McKinstry, R., Dadgar, A., Gritter, M., 2020. Informed privilege-complexity trade-offs in rbac configuration, in: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 119–130. URL: https://doi.org/10.1145/3381991.3395597, doi:10.1145/3381991.3395597.

Das, S., Sural, S., Vaidya, J., Atluri, V., 2018. HyPE: A hybrid approach toward policy engineering in attribute-based access control. IEEE Letters of the Computer Society 1, 25–29. URL: https://doi.org/10.1109/locs.2018.2889980, doi:10.1109/locs.2018.2889980.

Deng, F., Yu, Z., Liu, W., Luo, X., Fu, Y., Qiang, B., Xu, C., Li, Z., 2021. An efficient policy evaluation engine for xacml policy management. Information Sciences 547, 1105–1121. URL: https://www.sciencedirect.com/science/article/pii/S0020025520308148, doi:https://doi.org/10.1016/j.ins.2020.08.044.

Deng, F., Zhang, L.Y., 2017. Elimination of policy conflict to improve the PDP evaluation performance. Journal of Network and Computer Applications 80, 45–57. URL: https://doi.org/10.1016/j.jnca.2016.12.001, doi:10.1016/j.jnca.2016.12.001.

Dia, O.A., Farkas, C., 2012. A practical framework for policy composition and conflict resolution. International Journal of Secure Software Engineering 3, 1–26. URL: https://doi.org/10.4018/jsse.2012100101, doi:10.4018/jsse.2012100101.

Dos Santos, D.R., Westphall, C.M., Westphall, C.B., 2014. A dynamic risk-based access control architecture for cloud computing, in: 2014 IEEE Network Operations and Management Symposium (NOMS), IEEE. pp. 1–9.

Dunlop, N., Indulska, J., Raymond, K., 2003. Methods for conflict resolution in policy-based management systems, in: Seventh IEEE International Enterprise Distributed Object Computing Conference, 2003. Proceedings., IEEE. pp. 98–109.

Elliott, A., Knight, S., 2010. Role explosion: Acknowledging the problem., in: Arabnia, H.R., Reza, H., Deligiannidis, L., Cuadrado-Gallego, J.J., Schmidt, V., Solo, A.M.G. (Eds.), Software Engineering Research and Practice, CSREA Press. pp. 349–355. URL: http://dblp.uni-trier.de/db/conf/serp/serp2010.html#ElliottK10.

Fuchs, L., Broser, C., Pernul, G., 2009. Different approaches to in-house identity management - justification of an assumption, in: 2009 International Conference on Availability, Reliability and Security, pp. 122–129. doi:10.1109/ARES.2009.154.

Fuchs, L., Kunz, M., Pernul, G., 2014. Role model optimization for secure role-based identity management, in: European Conference on Information Systems (ECIS), pp. 1–15.

Fuchs, L., Pernul, G., 2007. Supporting compliant and secure user handling - a structured approach for in-house identity management, in: The Second International Conference on Availability, Reliability and Security (ARES'07), pp. 374–384. doi:10.1109/ARES.2007.145.

Fuchs, L., Pernul, G., 2008. HyDRo – hybrid development of roles, in: Information Systems Security. Springer Berlin Heidelberg, pp. 287–302. URL: https://doi.org/10.1007/978-3-540-89862-7_24, doi:10.1007/978-3-540-89862-7_24.

Fuchs, L., Pernul, G., 2010. Reducing the risk of insider misuse by revising identity management and useraccount data, in: 2nd Int. Workshop on Managing Insider Security Threats, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). Morioka, Iwate, Japan, pp. 14–28. URL: https://epub.uni-regensburg.de/15129/.

Fuchs, L., Pernul, G., Sandhu, R., 2011. Roles in information security – a survey and classification of the research area. Computers & Security 30, 748–769. URL: https://www.sciencedirect.com/science/article/pii/S016740481100099X, doi:https://doi.org/10.1016/j.cose.2011.08.002.

Gal-Oz, N., Gonen, Y., Yahalom, R., Gudes, E., Rozenberg, B., Shmueli, E., 2011. Mining roles from web application usage patterns, in: Trust, Privacy and Security in Digital Business. Springer Berlin Heidelberg, pp. 125–137. URL: https://doi.org/10.1007/978-3-642-22890-2_11, doi:10.1007/978-3-642-22890-2_11.

Gardiyawasam Pussewalage, H.S., Oleshchuk, V.A., 2017. Attribute based access control scheme with controlled access delegation for collaborative e-health environments. Journal of Information Security and Applications 37, 50–64. URL: https://www.sciencedirect.com/science/article/pii/S221421261730128X, doi:https://doi.org/10.1016/j.jisa.2017.10.004.

Gilbert, N., 2021. 31 crucial insider threat statistics: 2021 latest trends & challenges. URL: https://financesonline.com/insider-threat-statistics/.

Godik, S., Moses, T., 2003. URL: https://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf.

Groll, S., Kern, S., Fuchs, L., Pernul, G., 2021. Monitoring access reviews by crowd labelling, in: Trust, Privacy and Security in Digital Business. Springer International Publishing, pp. 3–17. URL: https://doi.org/10.1007/978-3-030-86586-3_1, doi:10.1007/978-3-030-86586-3_1.

Guarnieri, M., Arrigoni Neri, M., Magri, E., Mutti, S., 2013. On the notion of redundancy in access control policies, in: Proceedings of the 18th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 161–172. URL: https://doi.org/10.1145/2462410.2462426, doi:10.1145/2462410.2462426.

Guo, Q., Vaidya, J., Atluri, V., 2008. The role hierarchy mining problem: Discovery of optimal role hierarchies, in: 2008 Annual Computer Security Applications Conference (ACSAC), pp. 237–246. doi:10.1109/ACSAC.2008.38.

Hadj, M.A.E., Benkaouz, Y., Freisleben, B., Erradi, M., 2017. ABAC rule reduction via similarity computation, in: Networked Systems. Springer International Publishing, pp. 86–100. URL: https://doi.org/10.1007/978-3-319-59647-1_7, doi:10.1007/978-3-319-59647-1_7.

Hadj, M.A.E., Erradi, M., Khoumsi, A., Benkaouz, Y., 2018a. Validation and correction of large security policies: A clustering and access log based approach, in: 2018 IEEE International Conference on Big Data (Big Data), pp. 5330–5332. doi:10.1109/BigData.2018.8622610.

Hadj, M.A.E., Khoumsi, A., Benkaouz, Y., Erradi, M., 2018b. Formal approach to detect and resolve anomalies while clustering ABAC policies. ICST Transactions on Security and Safety 5, 156003. URL: https://doi.

org/10.4108/eai.13-7-2018.156003, doi:10.4108/eai.13-7-2018.156003.

Hadj, M.A.E., Khoumsi, A., Benkaouz, Y., Erradi, M., 2019. Efficient security policy management using suspicious rules through access log analysis, in: Networked Systems. Springer International Publishing, pp. 250–266. URL: https://doi.org/10.1007/978-3-030-31277-0_16, doi:10.1007/978-3-030-31277-0_16.

Han, D.j., Zhuo, H.k., Xia, L.t., Li, L., 2012. Permission and role automatic assigning of user in role-based access control. Journal of Central South University 19, 1049–1056. URL: https://doi.org/10.1007/s11771-012-1108-0, doi:10.1007/s11771-012-1108-0.

Hein, P., Biswas, D., Martucci, L.A., Muhlhauser, M., 2011. Conflict detection and lifecycle management for access control in publish/subscribe systems, in: 2011 IEEE 13th International Symposium on High-Assurance Systems Engineering, pp. 104–111. doi:10.1109/HASE.2011.50.

Heinrich, B., Klier, M., 2009. A novel data quality metric for timeliness considering supplemental data, in: Newell, S., Whitley, E.A., Pouloudi, N., Wareham, J., Mathiassen, L. (Eds.), 17th European Conference on Information Systems, ECIS 2009, Verona, Italy, 2009, pp. 2651–2662. URL: http://aisel.aisnet.org/ecis2009/14.

Herzberg, A., Mass, Y., Mihaeli, J., Naor, D., Ravid, Y., 2000. Access control meets public key infrastructure, or: assigning roles to strangers, in: Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000, pp. 2–14. doi:10.1109/SECPRI.2000.848442.

Hill, L., 2006. How automated access verification can help organizations demonstrate HIPAA compliance: A case study. J Healthc Inf Manag 20, 116–122.

Horne, D., 2011. Permissions. Springer US, Boston, MA. chapter Permissions. pp. 924–927. URL: https://doi.org/10.1007/978-1-4419-5906-5_786, doi:10.1007/978-1-4419-5906-5_786.

Hornsteiner, M., Groll, S., Puchta, A., 2020. Towards a user-centric iam entitlement shop - learnings from the e-commerce, in: 13th International Conference on Security of Information and Networks, Association for Computing Machinery, New York, NY, USA. pp. 1–4. URL: https://doi.org/10.1145/3433174.3433585, doi:10.1145/3433174.3433585.

Hounder, F., 2010. Conflict Detection and Resolution of XACML Policies. Master's thesis. University of Applied Sciences Rapperswil.

Hu, H., Ahn, G.J., Kulkarni, K., 2011. Anomaly discovery and resolution in web access control policies, in: Proceedings of the 16th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 165–174. URL: https://doi.org/10.1145/1998441.1998472, doi:10.1145/1998441.1998472.

Hu, H., Ahn, G.J., Kulkarni, K., 2012. Detecting and resolving firewall policy anomalies. IEEE Transactions on Dependable and Secure Computing 9, 318–331. URL: https://doi.org/10.1109/tdsc.2012.20, doi:10.1109/tdsc.2012.20.

Hu, H., Ahn, G.J., Kulkarni, K., 2013. Discovery and resolution of anomalies in web access control policies. IEEE Transactions on Dependable and Secure Computing 10, 341–354. URL: https://doi.org/10.1109/tdsc.2013.18, doi:10.1109/tdsc.2013.18.

Hu, J., Khan, K.M., Zhang, Y., Bai, Y., Li, R., 2016. Role updating in information systems using model checking. Knowledge and Information Systems 51, 187–234. URL: https://doi.org/10.1007/s10115-016-0974-4, doi:10.1007/s10115-016-0974-4.

Hu, J., Zhang, Y., Li, R., 2010a. Towards automatic update of access control policy, in: Proceedings of the 24th International Conference on Large Installation System Administration, USENIX Association, USA. p. 1–7.

Hu, J., Zhang, Y., Li, R., Lu, Z., 2010b. Role updating for assignments, in: Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 89–98. URL: https://doi.org/10.1145/1809842.1809859, doi:10.1145/1809842.1809859.

Hu, V.C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., 2014. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Technical Report. U.S. Department of Commerce. URL: https://doi.org/10.6028/nist.sp.800-162, doi:10.6028/nist.sp.800-162.

Huang, J., Nicol, D.M., Bobba, R., Huh, J.H., 2012. A framework integrating attribute-based policies into role-based access control, in: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 187–196. URL: https://doi.org/10.1145/2295136.2295170, doi:10.1145/2295136.2295170.

Hummer, M., Groll, S., Kunz, M., Fuchs, L., Pernul, G., 2018. Measuring identity and access management performance - an expert survey on possible performance indicators, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications. pp. 233–240. URL: https://doi.org/10.5220/0006557702330240, doi:10.5220/0006557702330240.

Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G., 2015. Advanced identity and access policy management using contextual data, in: 2015 10th International Conference on Availability, Reliability and Security, pp. 40–49. doi:10.1109/ARES.2015.40.

Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G., 2016. Adaptive identity and access management - contextual data based policies. EURASIP Journal on Information Security 2016. URL: https://doi.org/10.1186/s13635-016-0043-2, doi:10.1186/s13635-016-0043-2.

Jabal, A.A., Davari, M., Bertino, E., Makaya, C., Calo, S., Verma, D., Russo, A., Williams, C., 2019. Methods and tools for policy analysis. ACM Computing Surveys 51, 1–35. URL: https://doi.org/10.1145/3295749, doi:10.1145/3295749.

Jaferian, P., Rashtian, H., Beznosov, K., 2014. To authorize or not authorize: Helping users review access policies in organizations, in: Proceedings of the Tenth USENIX Conference on Usable Privacy and Security, USENIX Association, USA. p. 301–320.

Jin, C., Shen, A., Yu, W., 2016. The rbac system based on role risk and user trust. Int. J. Comput. Commun. Eng 5, 374–380.

Kern, A., Walhorn, C., 2005. Rule support for role-based access control, in: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 130–138. URL: https://doi.org/10.1145/1063979.1064002, doi:10.1145/1063979.1064002.

Kunz, M., Fuchs, L., Hummer, M., Pernul, G., 2015a. Introducing dynamic identity and access management in organizations, in: Information Systems Security. Springer International Publishing, pp. 139–158. URL: https://doi.org/10.1007/978-3-319-26961-0_9, doi:10.1007/978-3-319-26961-0_9.

Kunz, M., Fuchs, L., Netter, M., Pernul, G., 2015b. How to discover high-quality roles? a survey and dependency analysis of quality criteria in role mining, in: Communications in Computer and Information Science. Springer International Publishing, pp. 49–67. URL: https://doi.org/10.1007/978-3-319-27668-7_4, doi:10.1007/978-3-319-27668-7_4.

Kunz, M., Puchta, A., Groll, S., Fuchs, L., Pernul, G., 2019. Attribute quality management for dynamic identity and access management. Journal of information security and applications 44, 64–79.

Levy, Y., Ellis, T.J., 2006. A systems approach to conduct an effective literature review in support of information systems research. Informing Sci. Int. J. an Emerg. Transdiscipl. 9, 181–212.

Lu, J., Xin, Y., Peng, H., Han, J., Lin, F., 2017. Supporting user authorization queries in RBAC systems by role-permission reassignment, in: Cyberspace Safety and Security. Springer International Publishing, pp. 468–476. URL: https://doi.org/10.1007/978-3-319-69471-9_35, doi:10.1007/978-3-319-69471-9_35.

Lu, J., Xu, D., Jin, L., Han, J., Peng, H., 2014. On the complexity of role updating feasibility problem in RBAC. Information Processing Letters 114, 597–602. URL: https://doi.org/10.1016/j.ipl.2014.06.003, doi:10.1016/j.ipl.2014.06.003.

Marouf, S., Shehab, M., Squicciarini, A., Sundareswaran, S., 2011. Adaptive reordering and clustering-based framework for efficient xacml policy evaluation. IEEE Transactions on Services Computing 4, 300–313. doi:10.1109/TSC.2010.28.

Meier, S., Fuchs, L., Pernul, G., 2013. Managing the access grid - a process view to minimize insider misuse risks, in: 11th International Conference on Wirtschaftsinformatik (WI2013), pp. 1051–1065. URL:

https://epub.uni-regensburg.de/27930/.

Menges, F., Latzo, T., Vielberth, M., Sobola, S., Pöhls, H.C., Taubmann, B., Köstler, J., Puchta, A., Freiling, F., Reiser, H.P., et al., 2021. Towards gdpr-compliant data processing in modern siem systems. Computers & Security 103, 102165.

Miseldine, P.L., 2008. Automated xacml policy reconfiguration for evaluation optimisation, in: Proceedings of the fourth international workshop on Software engineering for secure systems, pp. 1–8.

Mitra, B., Sural, S., Vaidya, J., Atluri, V., 2016. A survey of role mining. ACM Computing Surveys 48, 1–37. URL: https://doi.org/10.1145/2871148, doi:10.1145/2871148.

Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J., 2008. Mining roles with semantic meanings, in: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 21–30. URL: https://doi.org/10.1145/1377836.1377840, doi:10.1145/1377836.1377840.

Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J., 2010. Mining roles with multiple objectives. ACM Transactions on Information and System Security 13, 1–35. URL: https://doi.org/10.1145/1880022.1880030, doi:10.1145/1880022.1880030.

Moses, T., 2005. extensible access control markup language (xacml) version 2.0. URL: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

Narouei, M., Takabi, H., 2019. A nature-inspired framework for optimal mining of attribute-based access control policies, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, pp. 489–506. URL: https://doi.org/10.1007/978-3-030-37231-6_29, doi:10.1007/978-3-030-37231-6_29.

Nazerian, F., Motameni, H., Nematzadeh, H., 2019. Emergency role-based access control (e-rbac) and analysis of model specifications with alloy. Journal of Information Security and Applications 45, 131–142. URL: https://www.sciencedirect.com/science/article/pii/S2214212618303843, doi:https://doi.org/10.1016/j.jisa.2019.01.008.

Ni, Q., Lobo, J., Calo, S., Rohatgi, P., Bertino, E., 2009. Automating role-based provisioning by learning from examples, in: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 75–84. URL: https://doi.org/10.1145/1542207.1542222, doi:10.1145/1542207.1542222.

Oberholzer, S., 2011. Optimizing xacml policies. University of Applied Sciences Rapperswil.

Pan, N., Sun, L., He, L.S., Zhu, Z.Q., 2018. An approach for hierarchical RBAC reconfiguration with minimal perturbation. IEEE Access 6, 40389–40399. URL: https://doi.org/10.1109/access.2017.2782838, doi:10.1109/access.2017.2782838.

Pang, C., Hansen, D., Maeder, A., 2007. Managing rbac states with transitive relations, in: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 139–148. URL: https://doi.org/10.1145/1229285.1229306, doi:10.1145/1229285.1229306.

Pang, C., Zhang, X., Zhang, Y., Ramamohanarao, K., 2008. The efficient maintenance of access roles with role hiding, in: Das, G., Sarda, N.L., Reddy, P.K. (Eds.), Proceedings of the 14th International Conference on Management of Data, December 17-19, 2008, IIT Bombay, Mumbai, India, Computer Society of India / Allied Publishers. pp. 139–149. URL: http://www.cse.iitb.ac.in/%7Ecomad/2008/PDFs/13.pdf.

Parkinson, S., Khan, S., Chrpa, L., 2020. Automated planning for administrating role-based access control. AAAI .

Puchta, A., Böhm, F., Pernul, G., 2019. Contributing to current challenges in identity and access management with visual analytics, in: Data and Applications Security and Privacy XXXIII. Springer International Publishing, pp. 221–239. URL: https://doi.org/10.1007/978-3-030-22479-0_12, doi:10.1007/978-3-030-22479-0_12.

Qi, H., Di, X., Li, J., 2018. Formal definition and analysis of access control model based on role and attribute. Journal of Information Security and Applications 43, 53–60. URL: https://www.sciencedirect.com/science/article/pii/S221421261730368X, doi:https://doi.org/10.1016/j.jisa.2018.09.001.

Rao, K.R., Nayak, A., Ray, I.G., Rahulamathavan, Y., Rajarajan, M., 2021. Role recommender-RBAC: Optimizing user-role assignments in RBAC. Computer Communications 166, 140–153. URL: https://doi.org/10.1016/j.comcom.2020.12.006, doi:10.1016/j.comcom.2020.12.006.

Royer, D., 2008. Enterprise identity management, in: Fischer-Hübner, S., Duquenoy, P., Zuccato, A., Martucci, L. (Eds.), The Future of Identity in the Information Society, Springer US, Boston, MA. pp. 433–446.

Saffarian, M., Tang, Q., Jonker, W., Hartel, P., 2009. Dynamic User Role Assignment in Remote Access Control. Number TR-CTIT-09-14 in CTIT Technical Report Series, Centre for Telematics and Information Technology (CTIT), Netherlands. Eemcs-eprint-15311.

Samarati, P., de Vimercati, S.C., 2001. Access control: Policies, models, and mechanisms, in: Focardi, R., Gorrieri, R. (Eds.), Foundations of Security Analysis and Design, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 137–196.

Sandhu, R.S., 1998. Role-based access control, in: Advances in Computers. Elsevier, pp. 237–286. URL: https://doi.org/10.1016/s0065-2458(08)60206-5, doi:10.1016/s0065-2458(08)60206-5.

Sandhu, R.S., Samarati, P., 1994. Access control: principle and practice. IEEE communications magazine 32, 40–48.

Schrimpf, A., Drechsler, A., Dagianis, K., 2021. Assessing identity and access management process maturity: First insights from the german financial sector. Information Systems Management 38, 94–115. URL: https://doi.org/10.1080/10580530.2020.1738601, doi:10.1080/10580530.2020.1738601, arXiv:https://doi.org/10.1080/10580530.2020.1738601.

Seifermann, S., Heinrich, R., Werle, D., Reussner, R., 2022. Detecting violations of access control and information flow policies in data flow diagrams. J. Syst. Softw. 184. URL: https://doi.org/10.1016/j.jss.2021.111138, doi:10.1016/j.jss.2021.111138.

Servos, D., Osborn, S.L., 2017. Current research and open problems in attribute-based access control. ACM Comput. Surv. 49. URL: https://doi.org/10.1145/3007204, doi:10.1145/3007204.

Shafiq, B., Vaidya, J.S., Ghafoor, A., Bertino, E., 2012. A framework for verification and optimal reconfiguration of event-driven role based access control policies, in: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 197–208. URL: https://doi.org/10.1145/2295136.2295172, doi:10.1145/2295136.2295172.

Shamoon, I., Rajpoot, Q., Shibli, A., 2012. Policy conflict management using xacml, in: 2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC), pp. 287–291.

Sheng, S., Osborn, S.L., 2004. A classifier-based approach to user-role assignment for web applications, in: Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 163–171. URL: https://doi.org/10.1007/978-3-540-30073-1_12, doi:10.1007/978-3-540-30073-1_12.

Stepien, B., Matwin, S., Felty, A., 2012. An algorithm for compression of xacml access control policy sets by recursive subsumption, in: 2012 Seventh International Conference on Availability, Reliability and Security, pp. 161–167. doi:10.1109/ARES.2012.38.

Strembeck, M., 2005. A role engineering tool for role-based access control, in: Proceedings of the Third Symposium on Requirements Engineering for Information Security SREIS, pp. 1–8.

Strembeck, M., 2010. Scenario-driven role engineering. IEEE Security & Privacy Magazine 8, 28–35. URL: https://doi.org/10.1109/msp.2010.46, doi:10.1109/msp.2010.46.

Takabi, H., Amini, M., Jalili, R., 2007. Trust-based user-role assignment in role-based access control, in: 2007 IEEE/ACS International Conference on Computer Systems and Applications, pp. 807–814. doi:10.1109/AICCSA.2007.370725.

Takabi, H., Joshi, J.B., 2010. Stateminer: An efficient similarity-based approach for optimal mining of role hierarchy, in: Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 55–64. URL: https://doi.org/10.1145/1809842.1809853, doi:10.1145/1809842.1809853.

Tayi, G.K., Ballou, D.P., 1998. Examining data quality. Communications of the ACM 41, 54–57.

Tsiostas, D., Kittes, G., Chouliaras, N., Kantzavelou, I., Maglaras, L., Douligeris, C., Vlachos, V., 2020. The insider threat: Reasons, effects and mitigation techniques, in: 24th Pan-Hellenic Conference on Informatics, Association for Computing Machinery, New York, NY, USA. p. 340–345. URL: https://doi.org/10.1145/3437120.3437336, doi:10.1145/3437120.3437336.

Turkmen, F., Crispo, B., 2008. Performance evaluation of xacml pdp implementations, in: Proceedings of the 2008 ACM workshop on Secure web services, pp. 37–44.

Vaidya, J., Atluri, V., Guo, Q., Adam, N., 2008. Migrating to optimal rbac with minimal perturbation, in: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 11–20. URL: https://doi.org/10.1145/1377836.1377839, doi:10.1145/1377836.1377839.

Verma, D., Calo, S., Witherspoon, S.A., Manotas, I., Bertino, E., Jabal, A.A., de Mel, G.R., Swami, A., Cirincione, G., Pearson, G., 2019. Managing training data from untrusted partners using self-generating policies, in: Pham, T. (Ed.), Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, SPIE. pp. 1–15. URL: https://doi.org/10.1117/12.2519682, doi:10.1117/12.2519682.

Wand, Y., Wang, R.Y., 1996. Anchoring data quality dimensions in ontological foundations. Communications of the ACM 39, 86–95.

Xia, H., Dawande, M., Mookerjee, V., 2014. Role refinement in access control: Model and analysis. INFORMS Journal on Computing 26, 866–884. URL: https://doi.org/10.1287/ijoc.2014.0603, doi:10.1287/ijoc.2014.0603.

Xiang, C., Wu, Y., Shen, B., Shen, M., Huang, H., Xu, T., Zhou, Y., Moore, C., Jin, X., Sheng, T., 2019. Towards continuous access control validation and forensics, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 113–129. URL: https://doi.org/10.1145/3319535.3363191, doi:10.1145/3319535.3363191.

Xu, T., Naing, H.M., Lu, L., Zhou, Y., 2017. How do system administrators resolve access-denied issues in the real world?, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. p. 348–361. URL: https://doi.org/10.1145/3025453.3025999.

Xu, Z., 2014. Mining Meaningful Role-Based and Attribute-Based Access Control Policies. Ph.D. thesis. Stony Brook University.

Yi-qun, Z., Jian-hua, L., Quan-hai, Z., 2007. A general attribute based rbac model for web service, in: IEEE International Conference on Services Computing (SCC 2007), pp. 236–239. doi:10.1109/SCC.2007.8.

Zhang, D., Ramamohanarao, K., Ebringer, T., 2007. Role engineering using graph optimisation, in: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 139–144. URL: https://doi.org/10.1145/1266840.1266862, doi:10.1145/1266840.1266862.

Zhang, W., Chen, Y., Gunter, C., Liebovitz, D., Malin, B., 2013. Evolving role definitions through permission invocation patterns, in: Proceedings of the 18th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 37–48. URL: https://doi.org/10.1145/2462410.2462422, doi:10.1145/2462410.2462422.

Zong, Y., Bhargava, B., Mahoui, M., Zhong, Y., 2011. Trustworthiness based authorization on www, in: IEEE Workshop on Security in Distributed Data Warehousing, pp. 1–6.