Introducing Dynamic Identity and Access Management in Organizations

Michael Kunz[1], Ludwig Fuchs[2], Matthias Hummer [2], and Günther Pernul[1]

[1]Department of Information Systems
University of Regensburg, Regensburg, Germany
`michael.kunz@ur.de,guenther.pernul@ur.de`
[2]Nexis GmbH, Regensburg, Germany
`ludwig.fuchs@nexis-secure.com,matthias.hummer@nexis-secure.com`

Abstract. Efficient and secure management of access to resources is a crucial challenge in today's corporate IT environments. During the last years, introducing company-wide Identity and Access Management (IAM) infrastructures building on the Role-based Access Control (RBAC) paradigm has become the de facto standard for granting and revoking access to resources. Due to its static nature, the management of role-based IAM structures, however, leads to increased administrative efforts and is not able to model dynamic business structures. As a result, introducing dynamic attribute-based access privilege provisioning and revocation is currently seen as the next maturity level of IAM. Nevertheless, up to now no structured process for incorporating Attribute-based Access Control (ABAC) policies into static IAM has been proposed. This paper closes the existing research gap by introducing a novel migration guide for extending static IAM systems with dynamic ABAC policies. By means of conducting structured and tool-supported attribute and policy management activities, the migration guide supports organizations to distribute privilege assignments in an application-independent and flexible manner. In order to show its feasibility, we provide a naturalistic evaluation based on two real-world industry use cases.

Keywords: Identity and Access Management, IAM, ABAC, Policies

1 Motivation

The effective and secure management of employees' access to sensitive applications and data is one of the biggest security challenges for today's organizations [19]. A variety of national and international regulations or certifications like Basel III [3], the Sarbanes-Oxley-Act of 2002 [45], or the ISO 27000 family [23] together with internal guidelines force enterprises to audit and control actions within their systems. At the same time developments like the application of cloud-based services in corporate environments further underline the need for secure user management.

As a result, centralized Identity and Access Management (IAM) relying on the Role-based Access Control (RBAC) [43] paradigm became the core element for increasing user management efficiency and reduce related IT security risks over the last years. However, due to its static nature, the application of RBAC leads to a considerable amount of administrative overhead. Growing numbers of outdated roles stemming from organizational changes together with the need of manually administrating user role assignments as well as role permission assignments result in complex and outdated RBAC structures. Even disregarding the fact that it takes an average of 18 months for its initial implementation, RBAC consumes an average of 2,410,000$ for a firm of 10,000 employees [34]. As a result, researchers and practitioners recently started to point out the need for dynamic access privilege management IAM infrastructures ([42, 27, 14]).

Using Attribute-based Access Control (ABAC) policies [20] for dynamically granting and revoking access based on employees' and privileges' attributes (from hereinafter referred to as dynamic Identity and Access Management (dIAM)) is seen as the next maturity level of company-wide IAM. The ABAC paradigm in general is based on the presumption that using a subject's, object's, and their shared context's attributes an authorization decision can be made. ABAC research traditionally focused on aspects like expressing ABAC rules (e.g. using XACML as standardized language) while only little attention has been paid to its adoption in company-wide IAM environments. This adaptation requires the definition of a potentially high number of policies within the central IAM system, the enforcement of policy decisions within the legacy applications depending on their underlying access control models, as well as the continuous policy maintenance. In order to complete these tasks, companies require a guided approach which is able to manage organizational project complexity as well es the technical heterogeneity of involved applications and protocols. To the best of our knowledge, no such structured approach has been provided up to now.

In this paper we are closing the existing research gap by firstly investigating the main building blocks required for dIAM infrastructures (Section 3). Secondly we propose a migration guide for implementing dIAM which serves as a project guideline dividing the necessary steps into manageable activities (Section 4). We thirdly evaluate our work within two real world use cases in the insurance and research industry. Besides the theoretical structuring of activities we identified the need for automation and thus additionally provided a prototypical software implementation for executing single activities of our migration guide. In order to achieve this we extended an existing IAM-tool proposed in [10] with attribute management and policy generation functionality. This allowed us to facilitate available functionality (e.g. data import or data visualization) and further evaluate our migration guide within real-life projects (see Section 5).

2 Related Work

Traditionally, Identity and Access Management in organizations has been associated with storing user data, maintaining user accounts, and controlling users' access to applications [11]. In today's medium to large-sized companies a centralized management of users following the RBAC paradigm has become the de facto standard approach for handling the challenges imposed by a steadily growing number of digital identities as well as access privileges. Recent surveys underline this growing importance of roles in information security in general and in IAM environments in particular [13]. However, over time and without proper controls such as de-provisioning processes, the number of roles is steadily growing, contradicting the benefits of administrative cost reduction [9]. In order to keep role systems up to date, methodologies and metrics for the ongoing optimization of role-based IAM infrastructures are required [10, 26]. Nonetheless, the static concept of roles in general lacks the ability to adopt to company changes and struggles with situational adaptivity [42]. Both requirements, however, are main challenges of modern IAM infrastructures.

As a result, companies aim at enhancing their existing IAM systems with dynamic ABAC policies in order to increase provisioning capabilities, strategically reduce administrative tasks, and keep IAM infrastructures manageable [21]. While standard ABAC protocols like the eXtensible Access Control Markup Language (XACML) [33] have been around since 2003, Priebe at al. [36] and Yuan et al. [52] were the first to formally define ABAC as an access control model. However, their focus was on formalizing the model and did not consider an application-independent IAM scenario. Jin et al. suggest an attribute-based architecture for IAM focusing on attribute correlation and attribute importance in different IAM-related domains [25]. Their work, however, does not aim at supporting organizations during the set up of a dIAM system. Recently, Hu et al. [20] were amongst the first to provide generalized definitions and best practices while also giving recommendations on deploying ABAC in cross-application settings. They, however, neither provide the structured guidance nor an overview on how to adopt ABAC in an organization-wide IAM system.

Up to now, to the best of our knowledge, no approach constituting the single building blocks of ABAC-based company-wide IAM and aligning them into a structured process model exists. We close this gap in the remainder by firstly gathering the aforementioned building blocks on the basis of a thorough research review (Section 3). Secondly, we structure them in the form of a migration guide which can be employed by organizations that aim at extending their static identity- or role-based IAM towards the integration of ABAC policies (Section 4).

3 Building Blocks of Dynamic Identity and Access Management

In the following we present the core elements of dIAM systems derived from ABAC literature (e.g. building on the findings of [20]) as well as literature from related areas, such as data and information quality management or policy management. Even though most works do not consider their application for company-wide IAM in particular, researchers in general already identified attribute management as well as policy management as the two main aspects of any ABAC implementation. Attribute management [20, 6, 35, 16, 8, 37, 52] in general deals with requirements related to the attributes used within ABAC policies, ranging from the aggregation of attributes up to their ongoing maintenance. Policy management [20, 4, 15, 24, 30, 22, 37] deals with the development and continuous improvement of access policies.

3.1 Policy Management

While policies and their life-cycle in general have been studied in various research areas (e.g. [7]), researchers recently stated the need for a structured approach for policy management in IAM. Building on the generic policy life-cycle model proposed by Buecker et al. ([7], see Figure 1) we outline relevant aspects of policy management in IAM in the following.

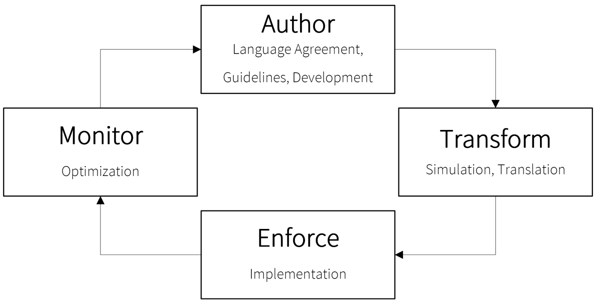


Fig. 1. Policy management based on [7] including corresponding dIAM aspects

Language Agreement
The first challenge prior to defining policies is the agreement upon a common expression language providing the syntax for depicting the semantics of policies interpreted by an IAM infrastructure. Looking at the research area, language requirements have been investigated [44] and comparisons of the suitability of policy languages (e.g. [17]) such as XACML [33] or EPAL [1] have been provided. Other authors like Strembeck [48] rather suggest generating a customized policy language tailored to the specific needs of a certain scenario. Within the area of IAM, however, a standardized approach seems more promising due to the high number of different applications and stakeholders involved.

Guidelines
Besides a common policy language, the establishment of policy guidelines plays an important role during the development as well as maintenance of dIAM systems. Policy guidelines are representing general rules on how policies are to be developed within a specific context. Note that in complex scenarios contradicting policies could potentially be defined. As a result, the establishment of design

guidelines is mandatory in order to avoid semantically correct but inefficiently modeled and contradicting policies. Beckerle and Martucci [4] were the first to formally define security and manageability goals for policies. They exclusively examined general goals for security and authorization rules. However, their results also can be applied in the context of IAM. Examples include the following goals provided in [4]:

- *Rule sets have to grant authorized access*
- *Redundant rules need to be removed.*
- *Contradicting rules need to be removed*
- *Concise rule sets are better than large rule sets*

By means of such exemplary guidelines organizations can increase policy homogeneity and ease policy maintenance.

Development

Policy Development deals with the actual creation of policies. Choosing an appropriate policy development methodology within a given scenario (i.e. an IAM project) is crucial for project success. Available methodologies can be divided into policy engineering and policy mining approaches (see Figure 2). Policy engineering deals with the top-down extraction of policies from business processes or workflows [2, 5], optionally based on security policy templates as shown in [41]. Authors agree that the policy notation used during policy development [47] and the provided tool-support [46] are critical success factors for policy engineering. Policy mining, in contrast, applies data mining technologies for extracting policies from Natural Language Policies [49, 29], currently assigned access privileges [50], or access logs [51, 22]. While providing an increased level of automation, policy mining lacks the integration of business know-how and struggles with low-quality attribute values - above all in the context of company-wide IAM involving numerous stakeholders and policies. Research results from related areas [11] underline that in such scenarios a hybrid approach building on both, an increased level of automation as well as the integration of expert knowledge, is the most promising method for policy modeling.

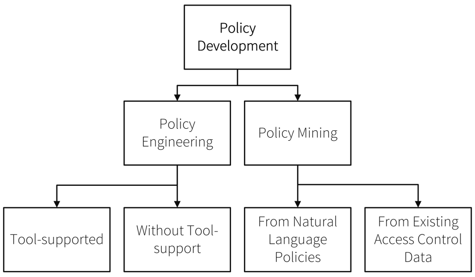


Fig. 2. Policy development methods

Simulation, Translation and Implementation

In company-wide IAM systems a potentially large number of ABAC policies affecting thousands of access privilege assignments might be required. As a result, a tool-supported simulation for anticipating the consequences of newly introduced policies becomes a central step during the setup of a policy base. Simulation tools can support the integration of policy owner feedback prior to policy activation as well as depict the future state of access within systems managed by an IAM infrastructure (e.g. using visual investigations as proposed in [31]). After simulation the policies need to be mapped onto the access control models of the legacy applications connected to an IAM. Those applications commonly are based on static access control models (e.g. SAP based on static roles or the Microsoft Active Directory (AD) based on groups). As a result, the IAM system in place has to carry out the required translations, i.e. the provisioning of dynamically calculated access privileges using static access control concepts (e.g. SAP roles).

Optimization

Once simulated and implemented, policies require the continuous monitoring of their correctness and validity by applying automated analytical methods. Note that due to the high number of expected policies a manual analysis is not feasible in the context of IAM. Lu et al., for instance, provide an approach for discovering inconsistencies and errors within policies at design-time [28]. Recently, Hummer et. al [22] proposed an approach that allows for a structured optimization of policies without interfering with a running IAM system. They apply anomaly detection methods in order to highlight deviations of normal policy patterns and visually present them to human policy engineers.

3.2 Attribute Management

Besides policy-related activities, attributes and their management form the foundation of any ABAC implementation. Attribute management is of great importance for company-wide IAM Despite its importance for company-wide IAM where employees are managed based upon master data attributes and access privileges are handled using attributes. However, attribute management in IAM has not attracted researchers' attention to a great extent up to now.

System & Attribute Selection

The initial selection and definition of application systems as well as related attributes managed within the ABAC policies [20] is the foundation for structured attribute management for dIAM. Note that in case an organization already has a deployed IAM system, basic attribute selection already took place during the initial system setup. Nevertheless, a re-investigation and potential extension of attribute sets commonly needs to be executed. Several master data attributes stored within a personnel management system might, for instance, be unused up to now but needed during later policy definition (e.g. an employee's job position or cost center).

Constraints & Data Types

After selecting required attributes, a definition of their data types, values and constraints needs to be carried out. Data types commonly range from boolean to single-valued and multi-valued attributes [6]. Researchers recently analyzed the effects of policy evaluation performance and highlighted its relation to the used attributes and attribute values [32]. Regarding attribute constraints, Bijon et al., for instance, introduce constraints on attribute assignments and values [6]. As further examples, Jin et al. provide a methodology for the classification of attributes according to their criticality and importance for access [25], while there also exists an overview of data and systems that are typically involved in an IAM environment [22].

Data Integration

As aforementioned, company-wide IAM commonly handles large amounts of data stemming from numerous applications, databases, or directory services. Organizations already operating an IAM hence need to review and extend existing integration processes to reflect the needs of future dynamic ABAC policies. IAM systems in general differentiate between source and target systems whereas a source system for certain attributes can act as target system for other attributes at the same time. An example could be an HR system providing master data of employees while at the same time receiving employees' email addresses from a mail application. Note that the definition of master sources for attributes has implications on attribute ownership. It is e.g. likely that human resources representatives are responsible for reviewing and validating attributes stemming from the personnel system.

Cleansing & Quality Controls

Policies created on the basis of erroneous attribute values essentially lead to security vulnerabilities, compliance violations, and administrative overhead. As a result, a structured review and cleansing of incorporated attribute values is a mandatory building block of dIAM prior to policy development. For an overview of potential data quality problems, cf. [39]. Hummer et al. recently argued that for optimizing policies, a centralized view on available and utilizable attributes spanning all involved systems is necessary in order to detect data errors and inconsistencies [22]. Data cleansing additionally builds on available attribute quality controls (e.g. rules for valid attribute values). Such quality controls, e.g., support the automated monitoring of attribute value changes and the advent of new attribute values and attribute types. We suggest to apply measures and metrics (for an overview cf. [18]) as well as best practices [40] from the field of data and information quality management to address these challenges.

4 Migration Guide

After describing the building blocks of an ABAC-based IAM, this Section of the paper introduces our tool-supported migration guide supporting a step-by-step

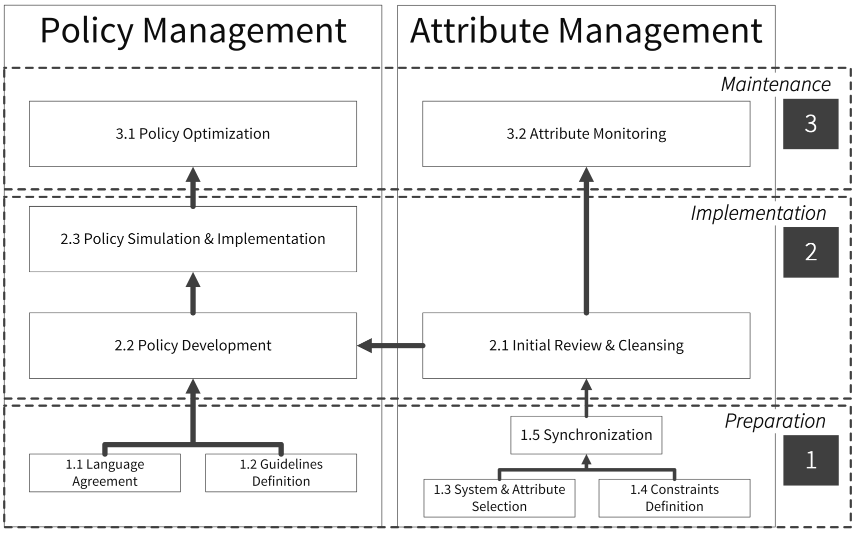


Fig. 3. Process model for migrating towards dIAM

migration from an existing static towards a dynamic IAM solution. It consists of three phases, namely a preparatory phase, an implementation phase and a maintenance phase (see Figure 3). The goal of the preparatory phase is to achieve a common understanding of policies and provide an attribute base used during later process phases. The subsequent implementation phase covers the cleansing of attributes and actual development of policies while the maintenance phase provides measures for continuous monitoring and improvement of the policy system. Note that due to space restrictions we cannot provide a detailed presentation of all involved sub-activities but rather aim at giving an overview of required tasks. In order to increase automation, we implemented a prototypical software for supporting the execution of attribute and policy management activities (Phase 2 of our migration guide). It is able to exchange data with an existing IAM system supporting the respective ABAC implementation process.

4.1 Preparation Phase

Due to the complexity and heterogeneity of static IAM environments, several preparatory activities have to be completed before ABAC policies can be defined. Relevant systems, attributes, responsibilities, and guidelines have to be reviewed and defined in order to foster a common understanding on a technical as well as organizational level among involved stakeholders.

Attribute Management

During system and attribute selection source systems for attribute data (e.g. personnel management systems) need to be investigated for attributes required during policy definition (Activity 1.3). Additional sources like IAM systems themselves or other applications providing information about user accounts or access privileges (e.g. ownership, criticality) might be identified. Note that organizations having basic attribute synchronization processes in place commonly have not dealt with the facilitation of extended attributes for complex access control decisions. By investigating system documentation or conducting expert interviews they hence need to review and extend the currently used attribute types in order to reflect ABAC requirements.

At the same time, data types need to be defined and constraint definitions for the attributes need to be established (Activity 1.4, cf. Section 3, *Constraints & Data Types*). This, amongst others, includes the definition of data types, master data sources, data ownerships, valid attribute values, or attribute ranges, i.e. intervals (if the data type is a numeric type) of validity. This way, erroneous attribute values can be identified during the subsequent cleansing activities.

After successfully completing the system and attribute selection and definition of constraints and data types, the attribute synchronization (Activity 1.5) takes place. Attribute values are imported into the IAM during this phase. At the same time conflicts like different encodings or granularity issues (e.g. *address* vs. *street* and *zip code*) can be detected.

Policy Management

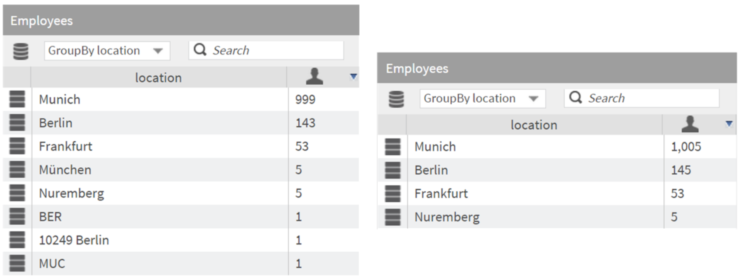
Regarding policy management, a general language agreement (Activity 1.1) for policy expression as well as the definition of policy guidelines need to be established prior to policy creation. Most of the currently available IAM implementations, for instance, are able to foster XACML as standardized policy language. Additionally, a shared understanding among project stakeholders on an organizational level needs to be established in the form of a company-wide glossary with definitions for important terminology. Available policy types like grant or denial policies should, for instance, be described. Furthermore, guidelines for policies (Activity 1.2, cf. Section 3, *Policy Guidelines*) can act as sources on how the human policy engineers are requested to model policies. Imagine a scenario in which only grant policies are allowed. Policy engineers should hence not have the option to design denial policies throughout a tool-supported policy creation process at all. Additionally, guidelines for the strategic maintenance of policies (Phase 3 of our migration guide) need to be defined. By introducing policy and attribute ownerships and requiring a periodic certification process, companies can essentially increase long-term policy quality.

4.2 Implementation Phase

After the preparatory activities have been completed, organizations enter the implementation phase (Phase 2) of our migration guide, i.e. the initial development and setup of a dIAM based on ABAC policies. Concerning attribute management, a systematic initial review and cleansing (Activity 2.1) of attribute data is required before the initial creation of policies as well as their subsequent simulation and implementation (Activities 2.2 and 2.3) can be carried out (cf. Section 3, *Policy Development; Simulation, Translation and Implementation*).

Attribute Management

Medium and large-sized organizations commonly struggle with data quality issues regarding their digital identities and access privileges. As a result, a dedicated cleansing process for improving attribute data quality is a crucial success factor for implementing dIAM. Following an initial assessment of attribute data (e.g. the identification of empty or invalid attribute values) the manual or automated cleansing of attributes needs to take place. We argue that a tool-based detection and cleansing process fosters user adoption by reducing the overall project complexity. Automated error identification can, for instance, be carried out by means of data mining or data quality metrics. Data mining, for instance, can be applied to detect outliers and unusual attribute values (see [12]). Based on predefined quality metrics (e.g. general rules like the currency [18] of an attribute value or a list of valid location attribute values) it leads to an overall higher quality of defined policies. Figure 4 (left side) gives a simple attribute cleansing example by grouping current location attribute values from a personnel system within our prototype after the attribute synchronization took place. Existing data errors such as typos, different language codings, or misspellings can be identified easily. The right side of Figure 4 displays the attribute values after cleansing by human experts in collaboration with attribute owners.



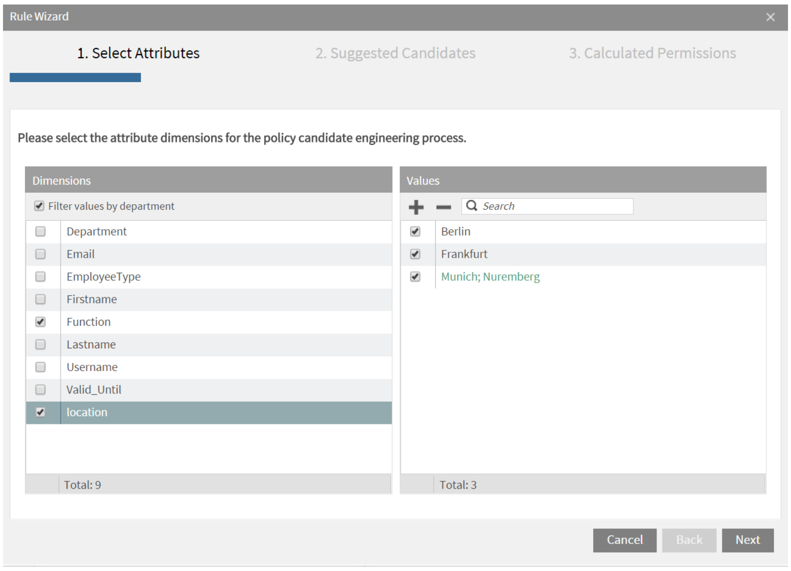
location	👤
Munich	999
Berlin	143
Frankfurt	53
München	5
Nuremberg	5
BER	1
10249 Berlin	1
MUC	1

location	👤
Munich	1,005
Berlin	145
Frankfurt	53
Nuremberg	5

Fig. 4. Before and after manual cleansing by grouping of attribute *location* and its various occurrences

Policy Management

As aforementioned, a potentially high number of policies bundling a wide range of access privileges or responsibilities are managed in corporate IAM environments. As a result, a manual policy generation by human policy engineers is not feasible. Organizations thus aim at employing automation techniques for creating policies and reviewing them in a hybrid manner (e.g. by experts who provide business knowledge and semantics, see Section 3). As one example of a potential role development approach in large IAM environments, we thus implemented policy mining algorithms that are able to automatically generate candidates for grant policies based on given attribute information. In order to support human review processes we additionally developed a simple representation of policies using a wizard-based graphical interface within our prototype (see Figure 5). Using this approach, a human policy engineer can select combinations of available attributes (left side of Figure 5, e.g. *function* and *location*) and optionally merge semantically or syntactically equivalent attribute values (right side of Figure 5, bundling the attribute values *Munich* and *Nuremberg*). In a second review step, suggested policy candidates are then displayed to the policy engineer. Continuing our example above, access is granted on the basis of the combination of employees' *location* and *function*. As a result, three policies for each function attribute value are generated, e.g. one policy for *sales representatives* in *Berlin*, *Frankfurt*, and *Munich/Nuremberg* each. During review, a human policy engineer can alter or remove unneeded policies (e.g. in case no *sales representatives* are located in *Frankfurt*). During a third step our prototype calculates the access rights shared by policy members based on customizable data mining algorithms. This way, a policy engineer could, for instance, enforce that only access rights that are not yet included in other policies are considered during the access privilege calculation or that critical access privileges are in general excluded from policy generation.



Please select the attribute dimensions for the policy candidate engineering process.

☑ Filter values by department

☐ Department
☐ Email
☐ EmployeeType
☐ Firstname
☑ Function
☐ Lastname
☐ Username
☐ Valid_Until
☑ location

Total: 9

☑ Berlin
☑ Frankfurt
☑ Munich; Nuremberg

Total: 3

Fig. 5. Automated tool-based policy mining and review

Completing the third step of our policy development wizard, policy owners are assigned and the policy candidates can be saved and exported to an IAM system. Ownership assignment can take place either based on rules (e.g. line managers are responsible for policies that affect their department) or manually.

After agreeing upon policy definitions, their simulation and implementation within an IAM test environment takes place. Due to the high number of organizational changes (e.g. restructuring organizational hierarchies, ownerships, and responsibilities) such policy simulation is a cornerstone of every policy modeling initiative. After final approval, the implementation of policies in the productive system occurs.

4.3 Maintenance Phase

The last phase of our migration guide (Phase 3) is dedicated to the continuous improvement of the previously implemented ABAC policies. In order to ensure long-term applicability of the defined rule set and minimize system complexity over time, a structured process for a periodic assessment and re-design of existing and new policies needs to be established. As a result, the maintenance phase deals with ensuring both, the correctness of policies and a high level of attribute quality (Activities 3.1 and 3.2, cf. Section 3, *Policy Optimization*).

Attribute Management

Regarding attribute management (Activity 3.2), we recommend the introduction of a structured monitoring process comprising two main activities, namely the periodic identification and review of quality metric violations as well as the definition of organizational agreements.

Quality measures defined during the previous phases of the migration guide form the basis for continuous attribute quality assurance. Throughout automated and periodic checks the correctness of attribute values can be investigated based on given quality measures and outlier detection methodologies. Examples for such checks can be periodic certifications of attributes by attribute owners or the detection of wrong attribute values using valid value lists. Besides such technical measures organizational agreements have to be made, e.g. in order to handle scenarios when new applications are connected to an IAM. In such cases, the IAM team has to decide whether the provided attributes fulfill the initially established constraints and attribute quality levels.

Policy Management

Besides the strategic management of attribute types and their values, the long-term maintenance of ABAC policies together with the potentially automated proposal of newly required but not yet defined policies need to be ensured. Note that both maintenance activities are highly dependent on each other. In contrast to attribute monitoring, discovering erroneous and outdated policies requires an increased level of automation. While single-valued attribute errors might be easily identified, a misconfiguration of policies granting critical access privileges can hardly be identified without tool-support. For addressing this challenge, Hummer et al. recently suggested measures and processes for strategic policy maintenance [22]. They, for instance, introduce tool-supported outlier and anomaly detection for identifying unused or outdated policies into the field of IAM.

5 Evaluation

After proposing our migration guide we now execute a naturalistic ex post evaluation covering two industry use cases based on the evaluation framework by Pries-Heje et al. [38]. The used real-life data-sets originate from companies operating in the health insurance in Switzerland (from hereinafter refereed to as 'Insucomp') as well as the research sector in Germany (from hereinafter refereed to as 'Rescomp'). All attribute values have been anonymized accordingly. While Rescomp already had a working IAM system in place, Insucomp conducted a policy development project as part of their initiative to initially implement an IAM system. The project duration was six months (Rescomp) and nine months (Insucomp) respectively, with both projects sharing the same overall goals:

1. Automatically providing new employees with correct basic access.
2. Increasing the amount of automatically distributed privileges by using dynamic provisioning policies.

In order to achieve these goals, both companies executed Phases 1 and 2 of our migration guide and facilitated our prototypical tool implementation during policy development. Insucomp additionally implemented basic measures for policy and attribute maintenance (Phase 3) while Rescomp plans to do so in future. Note that even though both use cases only aimed at policy definition based on subject attributes, our model can also be applied during the general development of policies comprising subject, object, and environmental attributes.

5.1 Insucomp

Insucomp is employing 349 external and 866 internal employees which in total own 7,777 accounts in 13 different application systems, including one AD and one SAP instance. In total, 2,297 different access rights are directly assigned to the user accounts resulting in 54,059 access rights assignments. Insucomp's variety of applications using static access privilege assignments in combination with manual provisioning processes resulted in large administrative efforts over the last years. As a result, a new IAM system based on dynamic access control policies had to be introduced between 2014 and 2015.

Preparation Phase

Throughout a kick-off workshop, Insucomp initially taught policy engineers guidelines on how to semi-automatically construct policies (Activity 1.2) while the IAM software implemented during the overall IAM project pre-defined the applied policy language (Activity 1.1). In the specific case the proprietary modeling capabilities of the Dell One Identity Manager were employed due to the reduced expected technical implementation efforts required. The system and attribute selection (Activity 1.3) took place in an iterative manner. Firstly, the HR system was defined as source for employee master data. The available attributes together with access privileges from all 13 applications were imported into our prototype. Consecutively, policy engineers and the responsible line officers agreed upon the exclusion of certain access rights from the Microsoft AD, the SAP, and the Customer Relationship Management system from further consideration. This decision was based on several reasons: Firstly, granting certain access rights in an automated manner would have resulted in an increase of license costs. Secondly, selected access privileges from the Customer Relationship Management system were classified as critical from an IT security perspective and hence excluded from automated provisioning processes. Regarding the attributes for policy development, the domain experts and IAM team selected an employee's *position* as the main HR attribute for the policy construction. Constraints and data types were defined accordingly:

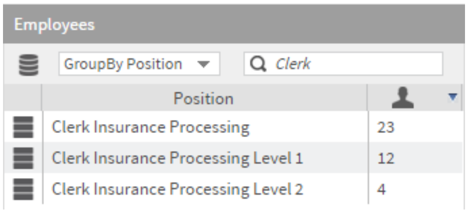
- C1: The German value for the *position* is used in policies.
- C2: A *code* is introduced for each value, referring to exactly one *position*.
- C3: A policy definition needs to contain both, a human-readable *position* as well as its respective machine-readable 4-digit *code*.
- C4: The *position* is a string value.

During attribute synchronization (Activity 1.5), violations of those constraints were identified. As an example, several languages were originally used to express an employee's position. In coordination with the HR department, the German *position* attribute value (C1) was selected as the defining attribute for later policy evaluations. Other languages were excluded from the data import and from now on are represented as translation of the main value (i.e. the German value) within a new attribute field in the HR system.

Implementation Phase

Following our migration guide a subsequent data cleansing process was conducted. Inspecting all attribute values within our prototype (Activity 2.1), Insucomp was, amongst others, able to discover ten erroneously defined *positions*. Additionally, *positions* with an inappropriate semantic granularity level were detected. For instance, initially one *position* for *Clerk Insurance Processing* existed within the HR system. However, for representing two semantically distinct insurance levels, Insucomp had to model two additional types of clerks with different access rights. As a result, the IAM team enforced the creation of more detailed *positions* and *codes* within the HR system. In the given example, two new *positions* were created in the HR system and employees were assigned accordingly (see Figure 6). Finishing the data cleansing activities a total of 253 *positions* have been available in the final attribute base.

After successful attribute cleansing, the actual detection of policy candidates within our prototype and the respective review together with domain experts



Position	
Clerk Insurance Processing	23
Clerk Insurance Processing Level 1	12
Clerk Insurance Processing Level 2	4

Fig. 6. Example for refactoring of employee *positions*

took place (Activity 2.2). As a side effect, Insucomp was able to discard 3,600 excessive assignments (i.e. 6.7% of all access privilege assignments) during the policy review process as our prototype highlighted additional (potentially excessive) privileges of employees assigned to a certain policy. This had a large impact on the overall project, further underlining the importance of secure provisioning and de-provisioning processes based on dynamic policies.

Finally, Insucomp exported the defined policies from our prototype and imported them within their newly set-up IAM system (Activity 2.3). They randomly selected sample policies in order to simulate correct functionality throughout various identity lifecycle processes (i.e. onboarding, change, and offboarding of employees). As a result, a total of 253 policies were put into operation. This led to the dynamic provisioning of 32% of all access rights among Insucomp's 13 connected application systems, essentially reducing the manual administrative workload while at the same time increasing the level of IT security.

Maintenance Phase

At the end of the migration project, Insucomp defined measures and quality controls in order to ensure the correctness of policies and attributes (Phase 3 of our migration guide). For conducting structured attribute management (Activity 3.2) newly introduced or changed attributes or attribute values have to be reported by the HR department to the IAM team in the future in order to adapt policies accordingly. Policy optimization has not been carried out up to now but is one element of the Insucomp IAM roadmap within the next year.

5.2 Rescomp

Rescomp already employed a working IAM system prior to the beginning of their policy definition project. Nonetheless, user management still was executed manually to a large extent for the 473 employees and the 761 different access rights (5,774 user privilege assignments in total). Rescomp's dynamic research environment requires automated and flexible access privilege provisioning in the future (e.g. for external employees like students needing temporary access to critical company data while undergoing regular organizational changes at the same time). As a result, a dIAM migration project was initiated in 2014. Similar to Insucomp, Rescomp executed the first two phases of our migration guide. Even though they have not executed maintenance activities up to now, they recently defined policy optimization as one element of their future IAM roadmap.

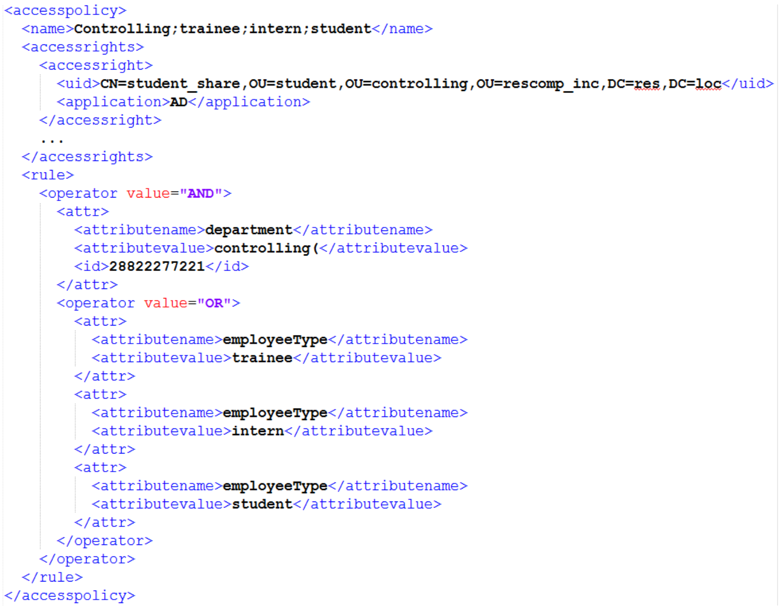
Preparation Phase

As a preparatory activity, Rescomp defined general guidelines for policy modeling (Activity 1.2). They introduced three types of valid policies, namely location-based policies, department and type-based policies, as well as function-based policies. Location-based policies represent the *physical location* of employees e.g. for granting physical access to buildings. *Department*- and *type*-based polices, in contrast, are defined based on the departmental assignment of employees in combination with their type, essentially granting access to departmental file shares for *internals*, *trainees*, *students*, or *externals*. In addition, function-based policies were defined to further refine employee's access rights according to their *job function*. Besides the three policy types, the IAM team defined a guideline regarding the definition of empty policies, i.e. policies that currently no employee is matching. In accordance with their project goals they decided to prepare such policies prior to an initial match of an employee (Activity 1.2). They, for instance, created a policy for all members of the technical service *department* whose *type of contract* is student. Students might only work within the department during their term holidays and thus the according policy might be unused for certain periods of the year but still is required during other months.

Following the migration guide, they selected two installations of their Microsoft AD for inclusion of access rights and provided the employee attributes from the HR system in place. During Activity 1.4 *department*, *type of contract*, *function*, *project* and *location* were selected as attribute base for policy definition. Similarly to Insucomp, Rescomp defined constraints and data types for these attributes. They, for instance, decided that regarding the *types of contract internals*, *apprentices*, and *students* should be treated equally in terms of their access rights.

Implementation Phase

Due to an already high attribute quality provided by the HR system, attribute cleansing was not required as no errors were identified during the attribute investigation. As a result, the IAM team subsequently conducted the policy development (Activity 2.2) in cooperation with business representatives. They started with the definition of basic *location* policies and continued with the creation of *department* and *employee type*-based policies as well as policies for employees' *function* attributes. Business representatives were asked to review the policy candidates using our prototype. In total, this process lead to the definition of 449 policies for automatic access privilege assignments, covering a total of 34.8% of all managed access privileges. Regarding the access rights, 45.9% of all initially existing privileges can now be assigned in an automatic way, i.e. they are included in at least one policy. All policies were exported from our prototype using the XML-notation and consecutively transferred into the existing LDAP-based IAM system of Rescomp using custom Python scripts (Activity 2.3). Figure 7 presents



```xml
<accesspolicy>
  <name>Controlling;trainee;intern;student</name>
  <accessrights>
    <accessright>
      <uid>CN=student_share,OU=student,OU=controlling,OU=rescomp_inc,DC=res,DC=loc</uid>
      <application>AD</application>
    </accessright>
    ...
  </accessrights>
  <rule>
    <operator value="AND">
      <attr>
        <attributename>department</attributename>
        <attributevalue>controlling(</attributevalue>
        <id>28822277221</id>
      </attr>
      <operator value="OR">
        <attr>
          <attributename>employeeType</attributename>
          <attributevalue>trainee</attributevalue>
        </attr>
        <attr>
          <attributename>employeeType</attributename>
          <attributevalue>intern</attributevalue>
        </attr>
        <attr>
          <attributename>employeeType</attributename>
          <attributevalue>student</attributevalue>
        </attr>
      </operator>
    </operator>
  </rule>
</accesspolicy>
```

Fig. 7. Example policy export using XML notation

a short XML export example of one department and employee type-based policy bundling students, trainees, and internships within a controlling department.

6 Conclusion

Dynamically assigning and revoking access privileges in company-wide IAM infrastructures has gained significant importance when it comes to automated and secure user management. Migrating to a dynamic IAM infrastructure based on ABAC policies can decrease manual administrative efforts while at the same time increasing the overall IT security level within companies. In order to support organizations during their required migration efforts, we proposed a novel three-step migration guide for implementing dynamic IAM based on ABAC policies in a structured manner. Up to now, no such structured process model highlighting and coordinating the respective migration tasks has been proposed. Our migration guide covers the required preparation, setup, as well as maintenance tasks and additionally offers tool-support in order to automate attribute and policy management activities. By doing so it increases the flexibility of policy engineers, reduces errors during policy modeling, and speeds-up the overall process of policy creation. Evaluating our migration guide throughout two real-life use cases we have further underlined its practical applicability.

In the future, we plan to extend our software prototype by implementing automated identity attribute monitoring activities that support companies during long-term attribute maintenance. In contrast to organizational guidelines this would support the enforcement of quality rules for attribute management. Additionally, we plan to expand policy development and policy maintenance capabilities in order to allow for a better cooperation between the responsible domain experts and the policy engineers.

Acknowledgment. The research leading to these results was supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association.

References

1. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise Privacy Authorization Language (EPAL 1.2). Submission to W3C (2003)
2. Aubert, J., Gateau, B., Incoul, C., Feltus, C.: SIM: An Innovative Business-Oriented Approach for a Distributed Access Management. IN: Proc. of the 3rd Int. Conf. on Information and Communication Technologies: From Theory to Applications (ICTTA). pp. 1–6 (2008)
3. Basel Committee on Banking Supervision: Basel III - A Global Regulatory Framework for More Resilient Banks and Banking Systems (2011)
4. Beckerle, M., Martucci, L.A.: Formal Definitions for Usable Access Control Rule Sets From Goals to Metrics. IN: Proc. of the 9th Symp. on Usable Privacy and Security (SOUPS). p. 2 (2013)
5. Bhatti, R., Bertino, E., Ghafoor, A.: X-FEDERATE: a Policy Engineering Framework for Federated Access Management. IEEE Transactions on Software Engineering 32(5), 330–346 (2006)
6. Bijon, K.Z., Krishman, R., Sandhu, R.: Constraints Specification in Attribute Based Access Control. Science 2(3), pp–131 (2013)
7. Buecker, A., Andrews, S., Forster, C., Harlow, N., Lu, M., Muppidi, S., Norvill, T., Nye, P., Waller, G., White, E.T.: IT Security Policy Management Usage Patterns Using IBM Tivoli Security Policy Manager. IBM Redbooks (2011)
8. Chadwick, D.W., Inman, G.: Attribute Aggregation in Federated Identity Management. IEEE Computer 42(5), 33–40 (2009)
9. Elliott, A., Knight, S.: Role Explosion: Acknowledging the Problem. IN: Proc. of the Int. Conf. on Software Engineering Research and Practice (SERP). pp. 349–355 (2010)
10. Fuchs, L., Kunz, M., Pernul, G.: Role Model Optimization for Secure Role-Based Identity Management. IN: Proc of the 22st Eur. Conf. on Information Systems (ECIS) (2014)
11. Fuchs, L., Pernul, G.: HyDRo – Hybrid Development of Roles. IN: Proc. of the 4th Int. Conf. on Information Systems Security (ICISS). pp. 287–302 (2008)
12. Fuchs, L., Pernul, G.: Qualitätssicherung im Identity- und Access Management. HMD Praxis der Wirtschaftsinformatik 50(1), 88–97 (2013)
13. Fuchs, L., Pernul, G., Sandhu, R.: Roles in Information Security - a Survey and Classification of the Research Area. Computers & Security 30(8), 748–769 (2011)
14. Gartner: Gartner IAM 2020 Predictions, `http://www.avatier.com/products/identity-management/resources/gartner-iam-2020-predictions/`
15. Gupta, P., Stoller, S.D., Xu, Z.: Abductive Analysis of Administrative Policies in Rule-based Access Control. IEEE Transactions on Dependable and Secure Computing 11(5), 412–424 (2014)

16. Hamlen, K., Liu, P., Kantarcioglu, M., Thuraisingham, B., Yu, T.: Identity Management for Cloud Computing: Developments and Directions. IN: Proc. of the 7th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW). p. 32 (2011)
17. Han, W., Lei, C.: A Survey on Policy Languages in Network and Security Management. Computer Networks 56(1), 477–489 (2012)
18. Heinrich, B., Kaiser, M., Klier, M.: How to Measure Data Quality? A Metric-based Approach. IN: Proc. of the 6th Int. Conf. on Computer and Information Science (ICIS) (2007)
19. Hovav, A., Berger, R.: Tutorial: Identity Management Systems and Secured Access Control. Communications of the Association for Information Systems 25(1), 42 (2009)
20. Hu, V.C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K.: Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Tech. Rep. NIST SP 800-162 (2014)
21. Huang, J., Nicol, D.M., Bobba, R., Huh, J.H.: A Framework Integrating Attribute-based Policies Into Role-based Access Control. IN: Proc. of the 17th ACM Symp. on Access Control Models and Technologies (SACMAT). pp. 187–196 (2012)
22. Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G.: Advanced Identity and Access Policy Management Using Contextual Data. IN: Proc. of the 11th Int. Conf. on Availability, Reliability and Security (ARES) (2015)
23. Iso: ISO/IEC 27000 Information Technology — Security Techniques — Information Security Management Systems — Overview and Vocabulary (2009)
24. Jin, X., Krishnan, R., Sandhu, R.: A Unified Attribute-based Access Control Mmodel Covering DAC, MAC and RBAC. IN: Proc. of the 12th Annual Conf. on Data and Applications Security and Privacy (DBSec). pp. 41–55 (2012)
25. Jin, Z., Xu, J., Xu, M., Zheng, N.: An Attribute-Oriented Model for Identity Management. IN: Proc. of the Int. Conf. on E-Education, E-Business, E-Management and E-Learning (IC4E). pp. 440–444 (2010)
26. Kunz, M., Fuchs, L., Netter, M., Pernul, G.: Analyzing Quality Criteria in Role-based Identity and Access Management. IN: Proc. of the 1st Int. Conf. on Information Systems Security and Privacy (ICISSP) (2015)
27. Kunz, M., Hummer, M., Fuchs, L., Netter, M., Pernul, G.: Analyzing Recent Trends in Enterprise Identity Management. IN: In Proc. of the 25th Int. Workshop on Database and Expert Systems Applications (DEXA). pp. 273–277 (2014)
28. Lu, J., Li, R., Hu, J., Xu, D.: Inconsistency Resolving of safety and utility in access control. Journal on Wireless Communications and Networking (1), 1–12 (2011)
29. Marfia, F.: Using Abductive and Inductive Inference to Generate Policy Explanations. IN: Proc. of the Int. Conf. on Security and Cryptography (SECRYPT) (2014)
30. Medvet, E., Bartoli, A., Carminati, B., Ferrari, E.: Evolutionary Inference of Attribute-Based Access Control Policies. IN: Proc. of the 8th Int. Conf. on Evolutionary Multi-Criterion Optimization (EMO). pp. 351–365 (2015)
31. Meier, S., Fuchs, L., Pernul, G.: Managing the Access Grid-A Process View to Minimize Insider Misuse Risks. IN: Proc. of the 11th Int. Tagung Wirtschaftsinformatik (WI) (2013)
32. Ngo, C., Makkes, M.X., Demchenko, Y., De Laat, C.: Multi-data-types Interval Decision Diagrams for XACML Evaluation Engine. IN: Proc. of the 11th Annual International Conference on Privacy, Security and Trust (PST). pp. 257–266 (2013)
33. OASIS: eXtensible Access Control Markup Language (XACML) Version 3.0 (2013)

34. O'Connor, A.C., Loomis, R.J.: 2010 Economic Analysis of Role-Based Access Control. Tech. rep. (2010)
35. Park, J., Zhang, X., Sandhu, R.: Attribute Mutability in Usage Control. IN: Research Directions in Data and Applications Security XVIII, pp. 15–29 (2004)
36. Priebe, T., Dobmeier, W., Muschall, B., Pernul, G., others: ABAC-Ein Referenzmodell für Attributbasierte Zugriffskontrolle. IN: Sicherheit. vol. 62, pp. 285–296 (2005)
37. Priebe, T., Dobmeier, W., Schläger, C., Kamprath, N.: Supporting Attribute-based Access Control in Authorization and Authentication Infrastructures with Ontologies. Journal of Software 2(1), 27–38 (2007)
38. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for Design Science Research Evaluation. IN: Proc. of the 16th Eur. Conf. on Information Systems (ECIS). pp. 1–12 (2008)
39. Rahm, E., Do, H.H.: Data Cleaning: Problems and Current Approaches. IEEE Database Engineering Bulletin 23(4), 3–13 (2000)
40. Redman, T.C.: Data Quality for the Information Age. Artech House, Inc., Norwood, MA, USA, 1st edn. (1997)
41. Rudolph, M., Schwarz, R., Jung, C.: Security Policy Specification Templates for Critical Infrastructure Services in the Cloud. IN: Proc. of the 9th Int. Conf. for Internet Technology and Secured Transactions (ICITST). pp. 61–66 (2014)
42. Sandhu, R.: The Authorization Leap from Rights to Attributes: Maturation or Chaos? IN: Proc. of the 17th ACM Symp. on Access Control Models and Technologies (SACMAT). pp. 69–70 (2012)
43. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based Access Control Models. Computer (2), 38–47 (1996)
44. Seamons, K., Winslett, M., Yu, T., Smith, B., Child, E., Jacobson, J., Mills, H., Yu, L.: Requirements for Policy Languages for Trust Negotiation. IN: Proc. of the 3rd Int. Workshop on Policies for Distributed Systems and Networks (POLICY). pp. 68–79 (2002)
45. SOX: Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745 (2002)
46. Stepien, B., Felty, A., Matwin, S.: A Non-technical XACML Target Editor for Dynamic Access Control Systems. IN: Proc. of the Int. Conf. on Collaboration Technologies and Systems (CTS). pp. 150–157 (2014)
47. Stepien, B., Matwin, S., Felty, A.: An Algorithm for Compression of XACML Access Control Policy Sets by Recursive Subsumption. IN: Proc. of the 7th Int. Conf. on Availability, Reliability and Security (ARES). pp. 161–167 (2012)
48. Strembeck, M.: Engineering of Dynamic Policy-Based Systems: A Policy Engineering of Dynamic Policy-Based Systems: Language Based Approach. Hab. Th. (2008)
49. Xiao, X., Paradkar, A., Thummalapenta, S., Xie, T.: Automated Extraction of Security Policies from Natural-language Software Documents. IN: Proc of the 20th Int. Symp. on the Foundations of Software Engineering (SIGSOFT). p. 12 (2012)
50. Xu, Z., Stoller, S.D.: Mining Attribute-based Access Control Policies from RBAC Policies. IN: Proc. of the 10th Int. Conf. and Expo on Emerging Technologies for a Smarter World (CEWIT). pp. 1–6 (2013)
51. Xu, Z., Stoller, S.D.: Mining Attribute-Based Access Control Policies from Logs. IN: Proc. of the 28th Annual Conf. on Data and Applications Security and Privacy (DBSec). pp. 276–291 (2014)
52. Yuan, E., Tong, J.: Attributed Based Access Control (ABAC) for Web services. IN: Proc. of the Int. Conf. on Web Services (ICWS). p. 569 (2005)