IT University of Copenhagen

Master Thesis

Software Development and Technology

Role-Mining of Access Control Policies with an Evolutionary Computation Approach

Author:
Theresa Ragna Elisabeth
Brandt von Fackh

Supervisor:
Sebastian
Risi

1. December 2015

Dedicated to my supporting parents
Suzanne and Henrich Brandt von Fackh

1 Abstract

Content:

- Summary: 1 page presenting the research problem, the main results, conclusion and how the thesis advances the field
- one page stating what the thesis is about
- highlight the contributions of the thesis

2 Acknowledgement

Content:

- Acknowledgement to those who have helped you

Example (copied from http://acknowledgementsample.com/acknowledgement-sample-for-master-thesis/):

I would like to express my gratitude to my supervisor Sebastian Risi for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore I would like to thank Omada A/S, especially Theis Nielsson, Hasse Olsson and Dr. Martin Kuhlmann for introducing me to the topic as well for the support on the way.
Also, I like to thank the participants in my survey, who have willingly shared their precious time during the process of interviewing. I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together.

Contents

3 Introduction

The main purpose of Identity Access Management (IAM) is to control access to resources by using security policies. Access Control Lists (ACL's) are maintaining security policies as direct assignments from access rights (permissions) to Identitys (users, computers or other principals). In order to lower the complexity and cost of access control administration, access control models are introduced. The most used access control model in the recent years in enterprise identity management systems is Role-based Access Control (RBAC). [12] RBAC security policies are grouping permissions into roles, which then are assigned to users.

One major task of RBAC is the role engineering, the creation of a Role Model. There are two approaches to build a role model: Top-Down and Bottom-Up. While in the Top-Down approach the role model is created by defining enterprise roles out of business processes, like e.g. the job descriptions, the Bottom-Up approach tries to create reasonable enterprise roles by taking current assignments of users to permissions into account. The Top-Down approach has the disadvantage that the designed roles do not contain all current access right assignments, which could lead to problems, when the model is applied. Furthermore the top-down approach is a long process with high costs. In the Bottom-Up approach data mining techniques are applied on current assignments of users to permissions in order to get results in reasonable time. But often these results do not have any business meaning, so that they do not get adopted by managers and system administrators, which are responsible for the correct assignment of these roles. Several workshops are needed to transform the mined role model into a realistic role model, which is accepted by the business. A hybrid model is combining data mining techniques with business knowledge to exploit the advantages of both approaches. [8] [19] [2]

The desired goal in role mining is finding an RBAC model, which can lower the complexity and cost of access control administration the most. The measuring of the current quality of a role model and selecting criteria for its optimization is still unsolved. There are several quality measures for an RBAC model, which can be dependent on the enterprises policy. The quality measures of an RBAC model can be rely on the overall RBAC state or on single roles within the RBAC model. The most addressed quality measures in existing role mining algorithms are a.o. achieving completeness, minimize the number of roles, decrease role set similarity, increasing role coverage, fulfill role constraints, Minimize Users/Permissions per Role and Minimize/Maximize Roles per User/Permission. [12]

There are several challenges in role mining. When taking business information into account, relevant and usable business information has to be identified and applied within the mining process. Relevant business information could be e.g. the hierarchy of Organizational Units (Org. Units) or location of an user. The quality of the data has also a big influence on the result. This is not only influenced by noise removal of the data set, but also which data set is used.

Additionally the mined role model should take constraints into account, such as Separation of Duty (SoD) and exceptions. [13]

Another challenge in RBAC is the maintaining of the role model. Permissions, Users and business information are evolving over time and the role model might become suboptimal over time. There is a practical need for periodic quality assessment of the resulting role models. [12] A complete re-modelling would not be a feasible solution, since the system administrators and managers would not accept if the role model is regularly restructured. A solution which slowly evolves the role model with the occurring changes has to be designed.

When users with domain knowledge interact within the role mining process, an appropriate visualisation of role mining results has to be developed. Also how to incorporate background domain knowledge to evaluate the results or to guide the search towards a result has to be considered. [9]

4 Background: Domain

In this section a basic introduction to the domain is given. It is important that the motivation and concepts of the domain are known in order to scope the problem definition and for guidance in the implementation of the approach.

4.1 Access Control Models

Access control ensures through policy definitions and enforcement that users[1] can only access resources (e.g. files, applications, networks) they are authorized for. The policy definitions describe which user is authorized for which permission. A permission describes which action (e.g. read, write, execute) can be done on a resource.

There are business, security and regulatory drivers for managing access control policies [15]. The interest of the business is to lowering the costs for managing the permissions for employees and quickly equip employees with necessary permissions such that they can efficiently full-fill their tasks. The security driver is to ensure information security, integrity, and availability to prevent accidental and intentional security breaches. But also to be compliant with regularities, such as the Data Protection Directive in Europe (Directive 95/46/EC)[2], Basel II[3] or company regulations, have an impact on the access control policies.

The National Institute of Standards and Technology (NIST) [10] describes various logical access control models, which provide a policy framework that specifies how permissions are managed and who, under what circumstances, is entitled to which permission and enforce these access control decision. Some of these logical access control models are briefly described in the following to enable the reader to differentiate between the concepts.

- **Identity-based Access Control (IBAC)**
  In IBAC a user gets a certain access to a resource by being assigned directly to a permission, which is connected to a resource. On a low-level Access Control Lists (ACLs) are implementations of IBAC. While IBAC may be manageable in companies with a small amount of users and permissions, the maintenance can

[1] For simplicity the term ”User” is used throughout the thesis although the term ”subject” would be more general, since they do not only include a user, but also service accounts or any other subject

[2] `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML; 07-10-2015`

[3] `https://en.wikipedia.org/wiki/Basel_II;07-10-2015`

be quickly overwhelming in consideration of satisfying all drivers of access control policies (see above).

- **Role-based Access Control (RBAC)**
  In RBAC permissions are bundled to roles, which then are assigned to users. A user gets a certain access to a resource by being assigned to a role, which contains the corresponding permission to that access. In other words users inherit all permissions of the roles they are assigned to. The motivation of this extra abstraction layer of a role is to easier maintain the access control of users. Even more abstraction can be introduced by role hierarchies (see next section).The RBAC model, which contains the roles, the user assignments to roles, the permission assignments to roles and the role-hierarchy, needs to be defined before the access control mechanism can enforce the access control. The RBAC model can be an ease of administration in comparison to direct assignments in IBAC. The degree of the benefit is dependent on the RBAC model.

- **Attribute-based Access Control (ABAC)**
  In [10] the authors try to guide to a standard definition of ABAC, since there seems to be no consensus. The basic concept of ABAC is to introduce an additional abstraction layer in form of ABAC policies, which are basically complex boolean rule sets that can evaluate different kinds of attributes. These attributes could be user information (e.g. department, job title), resource information (e.g. threat level) or environment information (e.g. current time, current location). A user gets a certain access to a resource if the rule set is satisfiable. The rule sets need to be defined before the access control mechanism can enforce the access control. Compared to RBAC ABAC seems more flexible, but it also comes with challenges regarding risk and auditing[1].

The scope of this thesis is to research an approach which outputs an RBAC model. Some ideas of the ABAC model are exploited to some extent in the implementation of the thesis approach.

4.2 Functional capabilities of RBAC

The RBAC model is often used in bigger organizations[15] and is leveraged by many Identity- and Access Management (IAM) systems. There are different functional capabilities of RBAC models. The NIST RBAC model [17] distinguished between four levels of RBAC models: Flat, Hierarchical, Constrained and Symmetric RBAC. Each level introduces an additional functional capability to the previous level. Some of the functional capabilities of an RBAC model are introduced in the following.

- **Basic RBAC Capability**
  In the basic RBAC model roles are bundles of permissions and users are assigned to these roles in order to get the according permissions. A user can be assigned to several roles and roles can be assigned to several users. The same many-to-many relation exists between roles and permissions. Therefore a user get can get the same permission several times by different roles.

- **Role Hierarchy**
  In a role hierarchy roles can be assigned to roles and inherit their permissions to the roles they have been assigned to. The role hierarchy can be restricted as a tree or inverted tree, where a role inherits only the permissions of its child-roles, or a partially ordered set, where a role inherits the permissions from any other role, which is assigned to it. It is also possible to limit inheritance in role hierarchies to control the power of impact of roles.

- **Separation-of-Duties (SoD)**
  Separation-of-Duties, also known as Segregation-of-duties, is a security principle for preventing fraud and errors by splitting tasks and associated permissions among multiple users. For example a user who has the permission to request the purchase of goods or services should not have the permission of approving the purchase. Roles in an RBAC model should therefore not have permissions bundled, which violate each other due to SoD. But also user assignments to multiple roles should not violate the SoD requirements.

In this thesis the main focus will be on the basic RBAC Capability and SoD Capability.

4.3 Role Model

A role model describes the roles, user-role assignments and permission-role assignments. The goal is to find a role model, which best leverages the RBAC model by satisfying the business, security and regulatory drivers: Lowering the costs for the administration of access control and equip employees with necessary permissions such that they can efficiently full-fill their tasks while keeping security requirements and being compliant. How this goal is measured in the role model has been described by several quality criteria[12][8]. The criteria concern the role model state, individual roles or both. In the following the different criteria is summarized into three different categories.

- **Completeness (or Confidentiality/Accessibility Violations)**

  By completeness it is meant to rebuild the current user-permission assignment state by the role model. When the user-role and the permission-role assignments of the role model are resolved, it should cover all current user-permission assignments.

  It is assumed that the current user-permission assignments are in a state, where users get the necessary access to efficiently perform their tasks and no security or compliance regulations are violated. This of course is an ideal situation, but in reality the current state often has quite some noise, especially when no IAM solution has been in place before. Users tend to have more permissions than they actually need, since it is unlikely that someone will claim that he has too many permissions. The measure of completeness of the role model state is therefore in accordance to the initial access control configuration.

  A certain amount of overentitlements (users get too many permissions) or underentitlements (users get too few permissions) in the access control policies can be acceptable, if the least privilege principle is too costly to implement in practice.

  The combinations of permissions within roles or the combination of user-role assignments might violate security regulations such as SoD. This measure is therefore not only taking individual roles but also the role model state into account.

  Constraints could also be ensured in a mechanism posterior to the role model, where individual permissions are detracted from users again, if it violates a constraint. Allowing constraint violations in the role model increases not only the processing and reliance on the posterior mechanism but also the auditing effort.

- **Complexity (or Number of Roles and Assignments)**

The complexity of a role model is measured in its number of roles and the number of user-role and permission-role assignments. It is often connected to the maintenance costs of a role model.
The more roles the role model has, the more maintenance effort is expected. Hence, a minimal set of roles is preferable. Furthermore it should be obvious that the total number of roles should be smaller than the total number of users or total number of permissions. Otherwise there would be no use of the advantage of having RBAC in comparison to IBAC in terms of administration costs. When each user has its individual role with the according individual bundle of permissions, the abstraction layer of the role becomes obsolete.
Also the more user-role and permission-role assignments are needed, the more maintenance effort is expected. Large roles with many permissions may reduce the number of user-role assignments, but may lead to more confidentiality/accessibility violations (conflicting Completeness). The same applies for if each role is used by many users. Small roles with few permissions on the other hand can lead to more administration effort as mentioned above, since many roles are necessary for achieving completeness. The same applies if each role is only used by very few users.
A role model probably consists of very general large roles and specialized small roles. Determining a fix boundary of how many permissions or users can be assigned to a role requires knowledge of the role model or is given by company or security regulations.

- **Comprehension**
  A recently more discussed topic is the "meaningfulness" of roles [19][8]. It is important that the administrators, which are maintaining the role model, can logically understand the role model for maintaining the roles and assignments confidently. Otherwise it might happen that they avoid to work with the role model since they feel not confident to stay in line with security and compliance regulations. Or it will cause them extra effort and costs to work with the role model. The roles should be therefore comprehensive, which can be achieved by giving them a meaning close to business roles, e.g. a role "Employee", which contains all permissions every employee will get and is assigned to every employee-user.
  This criteria can loosen some of the other criteria, which rather concentrate on

the compression of the access control information [7]. For achieving more intuitive meaningful roles it might be necessary to allow more roles than the minimal number of roles resulting from compression. More roles might result in more assignments, which are necessary to keep the role model more comprehensive. Lowering costs by having a more comprehensive role model may contradict lowering costs by having a less complex role model.

4.4 Role Engineering

Role engineering [3] describes the process to create a role model for RBAC. This task is proved to be very difficult in large enterprises.
There are three different approaches to conquer the goal of finding the right role model: Top-Down, Bottom-Up and the Hybrid approach[3][8].

- **Top-Down Approach**
  One approach is to do a top-down analysis, where the roles are built out of business information. Business processes, business roles and security policies are analysed to build a suitable role model. The resulting roles contain high-level permissions, which need to be mapped to technical permissions that are used by IT systems. The roles are easy to understand as they are derived from business concepts. The analysis of the business information by experts to a high-level role model and the mapping into low-level accesses by IT Specialists are very time-consuming and costly. Furthermore the resulting role model could lead to a different access control configuration than the current one. It is likely that users get less permissions than they used to have, which might prevent them of doing their tasks efficiently.

- **Bottom-Up Approach**
  The bottom-up approach exploits the current user-permission assignments and tries to gather a role model out of it. Since this approach often uses Data Mining Techniques, the method is also called Role Mining[11]. This approach on the other hand is often failing in generating a comprehensive role model, which is accepted by the administrators[7].

- **Hybrid approach**
  Since the advantages and disadvantages of the top-down and the bottom-up approach are mirrored, hybrid approaches have been suggested [8][14]. In these ap-

proaches the business information is leveraged to guide the computational generation of a role model out of user-permission assignments.

4.5 Role Mining Problem Definitions

- Basic RMP
- Edge RMP
- Interference RMP

4.6 Related Problems

- Boolean Decomposition Problem (BDM)
- Tiling Problem

5 Background: Evolutionary algorithm (EA)

Evolutionary algorithms (EA) are random searches inspired by the concept of the natural evolution. The general idea is to have a given population of individuals, which is evolving to new fitter generations of the population by natural selection (survival of the fittest). There are different variants of EAs: Genetic algorithms (GA), evolution strategies (ES), evolutionary programming (EP) and genetic programming (GP). All variants follow the same common concept described in the following with differences in technical details such as the representation of individuals[5].

The individuals in a population represent candidate solutions for the problem. A fitness function is evaluating each candidate solution in the population. Candidate solutions with a high-rated fitness are more likely to be chosen to seed the next generation than candidate solutions with low-rated fitness. The next generation is generated by applying variation operators like recombination and mutation on the selected candidate solutions. Recombination is applied to two or more candidate solutions (parents) and result in one or more new candidate solutions (children). During the recombination new candidate solutions are generated by merging random parts of the parents. The mutation operator is applied on only one candidate solution. The output is a copy of the candidate solution, where a random part is changed. The new candidate solutions, which are the output of the variation operators, are called the offspring and compete with the candidate solutions in the current population for a spot in the next generation of the population. In a survival selection the candidate solutions for the next generation are chosen. This process is repeated until a sufficient candidate solution is found or a previously set limit (e.g. number of generations) is reached. A flow chart of an evolutionary algorithm process can be seen in figure 1.

The driving forces of evolutionary algorithms are the variation operators, which are generating new diverse candidate solutions (novelty), and the natural selection, which is guiding to better candidate solutions (quality)[5].

By preserving the possibility that candidate solutions with a lower fitness can be selected as seed for the offspring, the chance of getting into a local optimum should be minimized like in other meta-heuristics.

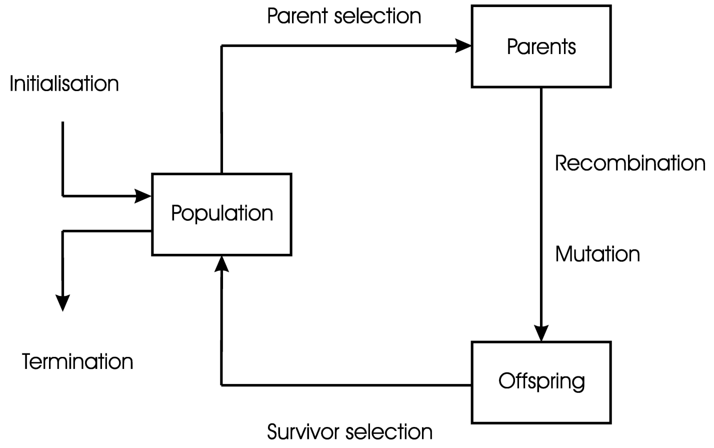

Figure 1: The general scheme of an evolutionary algorithm as a flow-chart [5]

5.1 Components of Evolutionary Algorithms

An EA is defined by its components, procedures and operators, which will be introduced in the following.

- **Individuals**
  The individuals of a population in EA represent candidate solutions. The candidate solutions within the original problem context are called phenotypes, while their representation in the EA are called genotypes or chromosomes. A building block of a chromosome is called a gene, where the possible values for a gene are called alleles. The mapping from the phenotype to the genotype is called encoding and the inverse mapping is called decoding. A representation could be for example a binary string or a string with real values. Both also more complex structures could be the right representation. Finding the right representation of the phenotypes is a difficult task and is crucial for the success of the EA. It also has an impact on the recombination and mutation operators.

- **Population**
  The population is a a list of individuals (genotypes), which can occur several times within the list. The individuals are static objects, which do not change. The population is dynamic, which will change over time due to the exchange of individuals.

The parent selection of the individuals, which will be the seed for the offspring, is mostly carried out on the whole population size. In most EAs the population size remains constant. The diversity of a population describes the measure of how many different individuals are present. The measure can be based on the fitness values, the phenotypes or the genotypes of the individuals. For example two individuals can represent two different phenotypes, but are evaluated with the same fitness value.
The first population consists of randomly generated individuals or of chosen individuals with higher fitness.

- **Evaluation Function (or Fitness function)**
  The evaluation function evaluates the individuals of a population and assigned the fitness of a candidate solution. Since the fitness influences the selection of an individual, the evaluation function encourages improvements. The goal could be to minimize or to maximize the fitness of individuals. Mathematically a minimization function can be easily transformed to a maximization problem and vice versa. The evaluation function in an EA is constructed from the objective function in the phenotype space.

- **Parent Selection**
  The parent selection is the selection of individuals, which will be the seed for the next generation. Typically individuals with a better fitness value get more often selected to further improve the individuals. But also individuals with a low quality fitness get a chance to pass on their genes into the next generation. This ensures that the search is not too greedy and get into a local optimum. The balance between parents with a high-quality fitness and parents with a low-quality fitness is often probabilistic.

- **Variation Operators (Recombination and Mutation)**
  The variation operators are responsible for discovering the search space by creating new individuals on the bases of existing individuals in the current population. All newly created individuals of a generation is called the offspring.
  There are different types of variation operators. While the mutation oparator only takes one individual as input, the recombination (or crossover) operator takes at least two individuals as input. The mutation operator creates a new individual

(mutant), which slightly differs from the input-individual (original). The change from the original to the mutant is chosen randomly. If the change is not random but rather guided, it is defined as an heuristic operator. The recombination operation is creating one or more new individuals (children) from its parent individuals by mixing randomly genes of these. A child might have a lower, equal or higher fitness value than its parents depending on the combination of genes.
Variation operators are depending on the representation of the phenotypes.

- **Survivor Selection (Replacement)**
  The survivor selection is the replacement strategy of the population and happens after the creation of the offspring. The current population is replaced by a new population - the new generation - which is mostly the same size as the population, which is being replaced. Like in the parent selection, the selection is based on the fitness values of the individuals and is often deterministic. In a fitnesss-biased selection for example the individuals of the population and the offspring are sorted by their fitness values and then the top segment is selected for the next population generation. In an age-biased selection only the individuals from the offspring are considered for the new population generation. If the current fittest individual of a population is kept in the next population, the concept of elitism is used.

- **Termination Condition**
  The EA is either terminated when one individual is reaching a known optimal fitness value or when predefined computation conditions are met. These conditions can be for example the number of generations, number of fitness evaluations, maximum allowed CPU time or a threshold under which the population diversity has to fall.

5.2 Multi-Objective Genetic Algorithms

A problem might have not only one objective but several objectives. It is unlikely that there is one optimal solution, which is considered optimal for each single objective. It is rather likely to have a set of optimal solutions, where the solutions are optimal in respect to all objectives combined. This set of optimal solutions is also known as Pareto-optimal solutions.

- NSGA2

Why? The higher the role number (1 Role for each user), the more likely it is to have no violations. The lower the role number, the more violations

- Improved NSGA2 (Fortin)
  Why? Different Individuals have same fitness

- Weighted NSGA2
  Why? 2nd objective is less important
  Issue? Skipped fronts, no symmetry in domination matrix

5.3 Co-Evolution

5.3.1 Symbiotic, Adaptive Neuro-Evolution (SANE)

5.3.2 Enforced Sub-Populations (ESP)

5.4 Human interaction

6 Related Work

Contents:

- A survey of the literature (journals, conferences, book chapters) on the areas that are relevant to your research question. One section per area. The chapter should conclude with a summary of the previous research results that you want to develop further or challenge. The summary could be presented in a model, a list of issues, etc. Each issue could be a chapter in the presentation of results. They should definitely be discussed in the discussion / conclusion of the thesis.

- The Literature Review provides the necessary background information to familiarize the reader with prior research and relevant theory. Three general types of literature reviews exist: the broad scan, the focused review, and the comprehensive critique.

- More than a literature review

- Organize related work - impose structure

- Be clear as to how previous work being described relates to your own.

- The reader should not be left wondering why you've described something!!

- Critique the existing work - Where is it strong where is it weak? What are the unreasonable/undesirable assumptions?

- Identify opportunities for more research (i.e., your thesis) Are there unaddressed, or more important related topics?

- After reading this chapter, one should understand the motivation for and importance of your thesis

- You should clearly and precisely define all of the key concepts dealt with in the rest of the thesis, and teach the reader what s/he needs to know to understand the rest of the thesis.

6.1 Role Mining with Data Mining

Role Mining has been first coined in [11]. In the following years several researchers have analyzed Role Mining further and defined several Role Mining Problems, Quality Measures, Cleaning Techniques and Algorithms.

6.2 Role Mining of "Meaningful Roles"

In the recent years several researchers are investigating the problem finding "meaningful" roles, since the classic Role Mining approaches outputs RBAC models, which are often not accepted in practice.

6.3 Role Mining with Bio-inspired Techniques

In [16] the basic RMP is tackled with an evolutionary algorithm. In a first version of the approach the authors use a specific representation of the phenotypes. They change this representation in an improved approach. In this thesis the suggested improved approach is used as starting point in my approach to deal with the RMP. In a recent paper [**?**] the authors concentrate on their first approach again.
In [4] the authors use two different Artificial Intelligence (AI) approaches to do Role Mining. In one approach they use a genetic algorithm. The second approach uses an Ant Colony approach. The result shows that...
In [**?**] an evolutionary approach for solving the policy generation problem is introduced.

7 Problem Analysis

Content:

- continuing from Chapter 2 explain the issues
- outline your solution / extension / refutation

8 Approach

Content:

- The research method by which you will investigate the world.

  A short summary of the available methods

  Your choice

  Detailed report of how you actually carried out your research. Presenting how you selected the people taking part is of special importance.

- The third chapter of the Master's Thesis is the Materials and Methods, sometimes also referred to as the Study Design and Methodology. It is in this section that the writer describes the sample population and procedures used in enough detail that others could replicate the study and verify its validity. This section should begin with specific data on the number of study participants, how they were chosen, and relevant demographic information. The writer may also present the rationale for the specific sample size used. Next, all tools and instruments used in the study should be described. If such tools are described in detail elsewhere in the literature, the writer can indicate this along with a relevant reference. Actual surveys or questionnaires will not be presented here. Instead, the writer provides an overview and then inserts copies of the tools into the Appendix at the end of the paper.

Limitation of Data Mining Techniques
Why EA?

8.1 Evolutionary algorithm (EA)

Basis of evolutionary Systems:[5]

- Variation operators for novelty
- Selection for improving quality
  - Parent selection and Survivor selection

Components, Procedures, Operators to be specified in order to define a particular EA:[5]

8.1.1 Representation (Definition of individuals)

- Mapping from phenotypes to genotypes (Encoding)

  Finding the genotype [16]

  ACL = UxP

  Y = UxR

  X = RxP

  RM = XxY

  Problem: Size of R is unknown

  Solution from [16]: 3rd chromosome Z to mark R's as "active" or "passive"

  Other solutions:

  Gray coding??
- Variable = Locus; Value = Allele
- Mapping from genotypes to phenotypes (Decoding)

8.1.2 Evaluation function (or Fitness function)

- represent the requirements to adapt to
- defines what improvement means
- Quality measure to genotypes
- composed from a quality measure in the phenotype space
- Turn minimization problem to maximization problem first

8.1.3 Population

- Multiset of genotypes
- Additional spatial structure: Distance measure or neighbourhood relation
- Measures:
  - Diversity = Number of different solutions present
  - Entropy

8.1.4 Parent selection mechanism

- Responsible for pushing quality improvements
- typically probabilistic
- low quality individuals also get a small chance in order to avoid local optimums

8.1.5 Variation operators, recombination and mutation

- Create new individuals from old ones
- Mutation
  - Unary variation operator
  - Stochastic: Output (child) depends on random choices
  - Guarantees that the space is connected
- Recombination / Crossover
  - Binary variation operator
  - Merge information from 2 parent genotypes
  - Stochastic: Choice of parent parts and their merging depend on random drawings
  - Recombination operators with higher arities possible

8.1.6 Survivor selection mechanism (Replacement)

- Called after creating offsprings
- deterministic
- Replacement strategy

8.1.7 Initialisation procedure

- First population is randomly selected or chosen with higher fitness

8.1.8 Termination condition

- Known optimal fitness level reached
- Other conditions which certainly stop algorithm
  - Max. Allowed CPU time
  - Total number of fitness evaluations
  - Period of time (number of generations or fitness evaluations)
  - Population diversity drops under a given threshold

8.2 Genetic Programming (GP)

8.3 Multi-Objective Genetic Algorithms

- NSGA2
  Why? The higher the role number (1 Role for each user), the more likely it is to have no violations. The lower the role number, the more violations
- Improved NSGA2 (Fortin)
  Why? Different Individuals have same fitness
- Weighted NSGA2
  Why? 2nd objective is less important
  Issue? Skipped fronts, no symmetry in domination matrix

8.4 Co-Evolution

8.4.1 Symbiotic, Adaptive Neuro-Evolution (SANE)

8.4.2 Enforced Sub-Populations (ESP)

8.5 Human interaction

9 Experiments

This chapter describes the setup of the experiments executed. First the data sets used in the experiments are described. Then the measures and the setup of the experiments are introduced.

9.1 Data Sets

9.1.1 Synthetic Datasets

Some papers are providing data generators for synthetic datasets for the performance evaluation of the various role mining algorithms. The data generator in [18] takes as input the number of users, permissions and roles to generate a pair of User-Role-Assignment and Role-Permission-Assignment matrices. Combining these two matrices, the corresponding User-Permission-Assignment matrix is obtained. In the data generator introduced in [Lu et al. 2014]...
"In [Lu et al. 2014], several sets of synthetic data of varying parameters are used. The data gen- erator takes as input the number of users and permissions, density of 1s in the PA, density of 1s in the UA and the noise level to produce the UA and PA matrices."

These data generators only consider users, permissions and their assignment to each other. They do not consider user information. Due to this a new synthetic data generator was created for this thesis, which is described in the following. The data generator

9.1.2 Real Datasets

In many research papers the same datasets are used for performance evaluation of role mining Algorithms. Table 1 lists some of these data sets and their components.

The authors of [6] obtained these datasets from Cisco firewalls and the Lotus Domino server of the Hewlett Packard (HP) networks. The healthcare dataset was collected from the US Veteran's Administration.

Also see [19]

Algorithm 1 Algorithm for creating an synthetic dataset for testing of role mining algorithms

```
procedure CreateRoles(roleCnt,permissionCnt,maxPermissionForRole,maxPermissionUsage)
    roles = [ ] for each roles[r] with r = 0..roleCnt-1
    permissionBucket = maxPermissionUsage for each permissionBucket[p] with p = 0..permissionCnt-1
    for role in roles do
        permissionForRoleCnt = pick randomly between 1..maxPermissionForRole
        tempPermissionBucket = permissionBucket - role
        if length of tempPermissionBucket >0 then
            permission = draw randomly number of permissionForRoleCnt out of tempPermissionBucket
            role.add(permission)
        else
            Error
        end if
    end for
    return roles
end procedure

procedure CreateUsers(userCnt,attribute)
    users = [ ] for each users[u] with u = 0..userCnt-1
    for user in users do
        for a,attrValues in attributes do
            user[a] = pick randomly a value of attrValues
        end for
    end for
    return users
end procedure

procedure CreateRules(roleCnt,attributes,maxRuleConditionCnt)
    rules = [] for each rules[r] with r = 0..roleCnt-1
    for rule in rules do
        selectedAttributes = pick maxRuleConditionCnt random attributes
        for a,attrValues in selectedAttributes do
            rule[a] = pick randomly values of attrValues
        end for
    end for
    return rules
end procedure
```

Dataset	Users	Permissions	User Permission Assignments
healthcare	46	46	1486
domino	79	231	730
emea	35	3046	7220
apj	2044	1146	6841
firewall-1	365	709	31951
firewall-2	325	590	36428
americas-small	3477	1587	105205
americas-large			

Table 1: Real Datasets

9.2 Visualisation of the RBAC Model

9.3 Experiments with Single-Objective GAs

9.3.1 Setup

- Number of turns
- Random Start-population
- Fixed Start-population

9.3.2 Measures

- Fitness (Min, Max, Avg)
- Time complexity
- Space complexity

9.4 Experiments with Multi-Objective GAs

9.4.1 Setup

- Number of turns
- Random Start-population
- Fixed Start-population

9.4.2 Measures

- Fitness (Min, Max, Avg)
- Time complexity
- Space complexity

9.5 Experiments with Co-Evolution

9.5.1 Setup

- Number of turns
- Random Start-population
- Fixed Start-population

9.5.2 Measures

- Fitness (Min, Max, Avg)
- Time complexity
- Space complexity

9.6 Experiments with Human Interaction

9.6.1 Setup

- Number of turns
- Random Start-population
- Fixed Start-population

9.6.2 Measures

- Fitness (Min, Max, Avg)
- Time complexity
- Space complexity

10 Results and Evaluation

adequacy, efficiency, productiveness, effectiveness (choose your criteria, state them clearly and justify them) be careful that you are using a fair measure, and that you are actually measuring what you claim to be measuring if comparing with previous techniques those techniques must be described in Chapter 2 be honest in evaluation admit weaknesses

The Results and Discussion portion of the thesis. These two components should remain separated with the appropriate headings within this chapter, as they both serve a different function. The results portion only presents the hard data without any accompanying analysis or interpretation. This section should include, where possible, a visual representation of the data, such as in charts, graphs, or tables. Each figure should have a brief description associated with it and clearly marked labels. The results of all statistical analyses should be presented, such that the reader has enough information to determine reliability, validity, and the statistical significance of the relationships among variables. This section should also be clearly organized by subheadings.

Which could be the system you made and the reasons for various design decisions, what your interview objects said, observations of people using a computer system, stories of a development process, numeral data from a questionnaire, etc. The discussion of the findings can be included in these chapters, or the discussion can be put in a separate chapter. The issues from the theory chapter (chapter 2) should be discussed here.

11 Discussion and Future Work

State what you've done and what you've found Summarize contributions (achievements and impact) Outline open issues/directions for future work

The final chapter of the Master's thesis is the Conclusions chapter. Here is where the writer sums up the entire project in one to two brief paragraphs. This chapter should remind the reader of the initial problem statement or hypothesis and then relate that to the results from the study. The writer should then present any conclusions reached or any new insights that arose from this work. Finally, the writer should present the research in terms of the overall impact in the field. For example, how will the results of this study change the way a person or organization behaves or makes decisions? One caution when writing this chapter is not to merely reiterate the other portions of the thesis. Instead, the writer should strive to leave a lasting impression upon the reader, conveying with the same passion that drove the research project the importance of the work completed.

Summary of the problem, the main findings and the discussion. Structured according to the issues in chapter 2. Comparison with the literature presented in chapter 2: how do your results fill in, advance or contradict previously reported research? What are the implications of your research for people working in the field that you have studies? In which direction should further research go?

12 Conclusion

13 References

https://cs.uwaterloo.ca/ brecht/thesis-hints.html

http://www.mastersproposal.com/format_of_thesis.html

14 Bibliography

[1] Ed Coyne and Timothy R. Weil. Abac and rbac: Scalable, flexible, and auditable access management. *IT Professional*, 15(3):14–16, 2013.

[2] Edward J. Coyne, Timothy R. Weil, and Rick Kuhn. Role engineering: Methods and standards. *IT Professional*, 13(6):54–57, November 2011.

[3] Edward J Coyne, Timothy R Weil, and Rick Kuhn. Role engineering: Methods and standards. *IT Professional*, (6):54–57, 2011.

[4] Xuanni Du and Xiaolin Chang. Performance of AI algorithms for mining meaningful roles. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, pages 2070–2076, 2014.

[5] Agoston E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing.* SpringerVerlag, 2003.

[6] Alina Ene, William Horne, Nikola Milosavljevic, Prasad Rao, Robert Schreiber, and Robert E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, SACMAT '08, pages 1–10, New York, NY, USA, 2008. ACM.

[7] Mario Frank, Joachim M. Buhman, and David Basin. Role mining with probabilistic models. *ACM Trans. Inf. Syst. Secur.*, 15(4):15:1–15:28, April 2013.

[8] Mario Frank, Andreas P. Streich, David Basin, and Joachim M. Buhmann. A probabilistic approach to hybrid role mining. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 101–111, New York, NY, USA, 2009. ACM.

[9] Jiawei Han. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[10] Vincent C. Hu, David Ferraiolo, Rick Kuhn, Arthur R. Friedman, Alan J. Lang, Margaret M. Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, and Scarfone Cybersecurity. Guide to attribute based access control (abac) definition and considerations (draft), 2013.

[11] Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. Role mining - revealing business roles for security administration using data mining technology. In *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, SACMAT '03, pages 179–186, New York, NY, USA, 2003. ACM.

[12] Michael Kunz, Ludwig Fuchs, Michael Netter, and Günther Pernul. Analyzing quality criteria in role-based identity and access management. In *Proceedings of the 1st International Conference on Information Systems Security and Privacy*, pages 64–72, 2015.

[13] Haibing Lu, Jaideep Vaidya, Vijayalakshmi Atluri, and Yuan Hong. Constraint-aware role mining via extended boolean matrix decomposition. *IEEE Trans. Dependable Sec. Comput.*, 9(5):655–669, 2012.

[14] S. Mandala, M. Vukovic, J. Laredo, Yaoping Ruan, and M. Hernandez. Hybrid role mining for security service solution. In *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pages 210–217, June 2012.

[15] Alan C O'Connor and Ross J Loomis. 2010 economic analysis of role-based access control. *NIST, Gaithersburg, MD*, 20899, 2010.

[16] Igor Saenko and Igor Kotenko. Genetic algorithms for role mining problem. In *Proceedings of the 2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing*, PDP '11, pages 646–650, Washington, DC, USA, 2011. IEEE Computer Society.

[17] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The nist model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, volume 2000, 2000.

[18] Jaideep Vaidya, Vijayalakshmi Atluri, and Janice Warner. Roleminer: Mining roles using subset enumeration. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 144–153, New York, NY, USA, 2006. ACM.

[19] Zhongyuan Xu and Scott D. Stoller. Algorithms for mining meaningful roles. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, SACMAT '12, pages 57–66, New York, NY, USA, 2012. ACM.

15 Appendix

Acronyms

ACL Access Control List. 8, 34, *Glossary:* Access Control List (ACL)

IAM Identity Access Management. 8, 34, *Glossary:* Identity Access Management (IAM)

Org. Unit Organizational Unit. 9, 34, *Glossary:* Organizational Unit (Org. Unit)

RBAC Role-based Access Control. 8, 9, 34, *Glossary:* Role-based Access Control (RBAC)

SoD Separation of Duty. 9, 34, *Glossary:* Separation of Duty (SoD)

Glossary

Access Control List (ACL) Access Control List describes 8

Identity Access Management (IAM) Identity Access Management describes 8

Organizational Unit (Org. Unit) An organizational unit is 9

Role-based Access Control (RBAC) Role-based Access Control describes 8

Separation of Duty (SoD) Separation of Duty describes 9

Identity An identity is a ... plural. 8