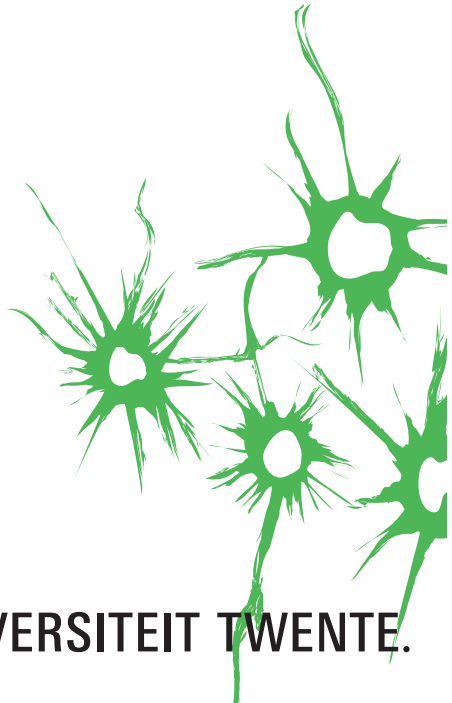
IMPROVING VISUAL ROLE MINING USING METADATA

Jonathan Juursema

Faculty of EEMCS
Master Thesis for Computer Security

Supervisors:
dr. Maarten Everts
ir. Albert Dercksen (External)
drs. Wouter Kuijper (External)



UNIVERSITEIT TWENTE.

AUGUST 27, 2018

Abstract

Many organisations use access control solutions that do not make use of standardised access control models such as the well-studied Role-Based Access Control model (RBAC). Visual role mining is a way for organisations to translate their existing access policies from these solutions into an RBAC policy. We contribute to the existing body of research on visual role mining by extending the framework with the use of metadata in order to enable the elicitation of contextually meaningful roles. We validated these additions by visiting organisations with a proof of concept software application inplementing this framework. These interviews demonstrate that our approach can indeed help with eliciting contextually meaningful roles, and also confirm that in practice visual role mining is a valuable tool.

1 Introduction

Access Control [9] is a concept that describes regulating requests by *subjects* to access *resources* that can be deployed in the digital as well as in the physical domain. In access control, such requests are first evaluated against an *access policy* that contains information on what subjects are allowed to access what resources. An access policy can also include additional contextual information on which the decision to grant or deny access can be based, such as the current time or restrictions on concurrent access to a resource.

Almost every organisation employs some form of access control, if only in the form of password-protected computers and office buildings with physical locks. Many larger organisations have chosen to standardise their access control. There exist many commercially available products that support such standardized access control within a company. Computer networks, for example, can make use of *Microsoft Active Directory* to centralise user credentials, resources and computer administration. Many different commercial solutions also exist for physical access control. These solutions are all based on some form of access control model. However, many of these models are proprietary, not well documented or both. They are also usually incompatible between each other. If an organisation has been using (and thus building an access policy in) such a model, they are effectively vendor-locked; if they wish to switch vendors they are likely to have to build a new access policy from the ground up.

Role-Based Access Control [2, 21] (or *RBAC*) is an access control model that is extensively studied in literature and allows for the assignment of permissions to users indirectly: users can be added to one or more roles and roles can have one or more permissions. *Role Mining* [1] is an area of research that concerns itself with extracting roles (in the context of RBAC) from an access policy that does not (necessarily) have such roles present. Visual role mining finally is a sub-field of role mining that specialises in visualising an existing access policy in such a way that a human can identify possible roles in the visualisation.

There has been relatively little research [20] on visual role mining, iterative role mining and the generation of contextually meaningful roles. Generating contextually meaningful roles however is an important aspect for the people who need to work with these roles [11, 20]. We think that visual role mining can be an excellent starting point in generating contextually meaningful roles. Since the work of Colantonio et al. [16] on visual role mining (in particular their *EXTRACT* and *ADVISER* algorithms, which we will summarise in Section 2) and iterative role mining is very thorough, we choose to build upon their approach to visual role mining. In summary, our work contributes to the advance of the visual role mining framework using metadata and validates its effectiveness in real cases. In particular, this thesis contributes the following:

1. We propose a number of methods to extend the visualisation generated by ADVISER using metadata, to help operators define contextually meaningful roles (Section 3).
2. We propose mADVISER: a variant on ADVISER that also takes into account metadata (Section 1).
3. We build a proof of concept application implementing the EXTRACT and mADVISER algorithms (Section 4.1).
4. We validate our contributions by visiting a number of different organisations and interviewing them in the context of our proof of concept application and (where possible) their own access control policy (Section 4.2).

This thesis is structured as follows. We introduce some necessary background in Section 2. We describe our methods to extend ADVISER as well as mADVISER in Section 3. We show our proof-of-of-concept and outline our validation results in Section 4. We provide an overview of other related work in Section 5 and discuss our findings in Section 6. Section 7 contains the limitations of our work and suggestions for future work.

2 Background

This section introduces the concept of Role-Based Access Control. It also gives a general history of role mining and briefly summarises the work of Colantonio et al. [16] on the EXTRACT and ADVISER algorithm to the extent needed to comprehend this thesis.

2.1 Role-Based Access Control

As mentioned in the introduction the most basic form of access control only considers subjects (or *users*), objects (or *permissions*) and authorisations. In this simple form, authorisations are simply a direct assignment of permissions to users. If such an assignment is present, a user can access a certain permission. If the assignment is not present, the user is barred from accessing the permission. In systems involving a large number of users and permissions this method of assigning permissions directly to users becomes impractical very quickly. Imagine a fictional university. Surely there is a better way to give students access to the buildings and facilities they need, other than manually assigning

every new student to every single thing they need to access individually?

RBAC aims to provide a solution to this problem of complexity. At its core, it introduces a single layer of *indirection* to the otherwise binary world of access control. This layer of indirection is called a *role*. A role can be used to *group* users and permissions. The relationships between users and roles, and roles and permissions are many-to-many: users can be assigned multiple roles, and a role can be assigned to multiple users. The same is true for permissions and roles. Consider again our fictional university. We can now create a role *student*, with the purpose of simplifying our complex situation. We only need to make sure to grant the student role access to all student resources, which is a one-time action. Now, whenever a new student enrolls, we only need to give them access to the *student* role, and they are good to go. Conversely, if a new computer for students is made available we now only once have to give the *student* role access to this computer. If this university teaches 100 students and has 100 computers for students, we have just reduced the number of assignments down from 10,000 (granting each individual student access to each individual computer) to 200 (assigning all students to the one student role, and granting that single role to all computers). This is the power of Role-Based Access Control.

Many extensions (such as two-sorted RBAC [19], time-based RBAC [3, 4] and location aware RBAC [6, 8]) to RBAC have been proposed and documented — these provide more complicated behaviour such as limiting the times between which a user can access permissions.

2.2 Role mining

In order to adopt RBAC, roles need to be defined. Role mining concerns itself with the process of extracting roles from an existing access control policy (or *security policy*, a set of assignments between users and permissions indicating which users can access which permissions). Many different approaches to role mining have been considered. This section will summarise some. For a complete overview, we refer the reader to Section 5 and two excellent surveys of role mining [12, 20] by Molloy et al. and Mitra et al.

Molloy et al. [12] categorise traditional role mining algorithms in two groups. The first group outputs a collection of roles and assigns these roles a priority. Then roles are usually chosen in order to minimise a certain cost or complexity metric. This group includes the *FastMiner* and *CompleteMiner* [7] algorithms. The second group outputs complete (or ready to use) security policies in RBAC, and include the *HierarchicalMiner* [10] algorithm and *ORCA* [5] software. These complete RBAC states perfectly represent the original security policy, meaning that all users have exactly the same permissions as before. This can come at the cost of a large number of roles.

Although briefly considered by Molloy et al., Mitra et al. [20] write in more detail about the challenge of optimising the output of role mining algorithms — figuring out how to define the "best" output. These metrics, used to determine what the "best" output is, usually either try to minimise the total number of roles (at the cost of introducing mismatches between the original security policy and the resulting security policy) or minimising said mismatch at the cost of a higher number of roles. The *MinNoise Role Mining Problem* and δ*-approximate Role Mining Problem* [15] are respective examples of these metrics. The former fixes the maximum number of roles, and aims to minimise the number of mismatches, while the latter sets a required degree of correctness and aims to minimise the number of resulting roles.

2.3 EXTRACT and ADVISER

The challenge with more traditional role mining algorithms is that although they mean to efficiently group users and permissions in roles, they have usually no regard for the reason *why* these users and permissions are grouped together. This means that roles are generally without contextual meaning, meaning that it is difficult to indicate what a given role represents. The premise of visual role mining is that organisations are more willing to adopt an access policy if they can understand this *why*. Where traditional role mining algorithms try to generate an optimal set of roles, the purpose of visual role mining is to visualise the security policy in such a way that an operator familiar with the context can find candidate roles that have actual contextual meaning.

Our work is built upon work of Colantonio et al. They propose two algorithms for visual role mining: EXTRACT and ADVISER [16]. The two algorithms take the binary matrix representation (or *user-permission matrix*, a matrix that defines for each user and each permission whether or not that user has that permission) of any security policy. An example of a visualisation of such an (unsorted) user-permission matrix is shown in Figure 1. In this visualisation one axis represents users, the other permissions. A pixel in the visualisation is coloured black if said users is authorised to said permission, otherwise the pixel is coloured white.

The ADVISER algorithm sorts the unsorted matrix, using roles as its input, with the goal of grouping similar user-permission assignments together. This results in a (sorted) user-permission matrix that reveals the structures present in the data. This visualisation then serves as a starting point for the role elicitation: the process of extracting roles and giving

the roles contextual meaning. If roles are not known beforehand, EXTRACT can be used to generate a set of "pseudo" or "good enough" roles that ADVISER can use as its input. An example of a visualisation of a (sorted) user-permission matrix is shown in Figure 2.

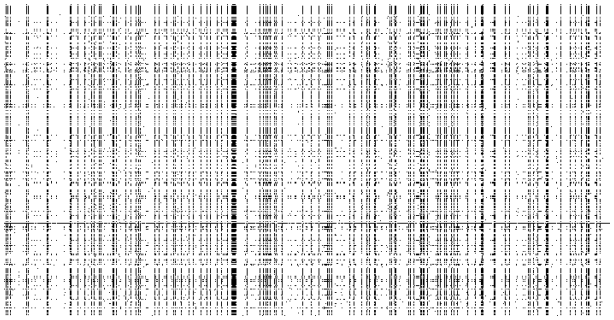


Figure 1: An unsorted user-permission matrix that serves as input for the ADVISER algorithm.

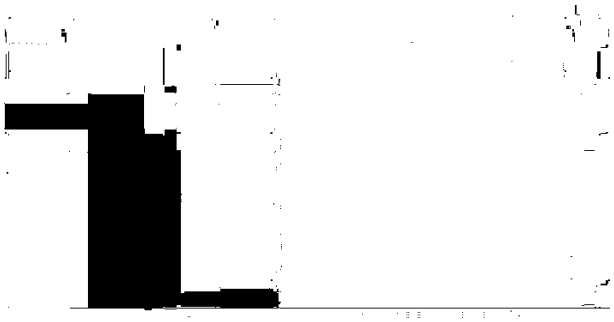


Figure 2: A sorted user-permission matrix generated by the ADVISER algorithm.

The remainder of this section will serve as an intuitive explanation of the EXTRACT and ADVISER algorithms. Please refer to the original paper for a more formal description.

The purpose of the EXTRACT algorithm is to generate pseudo-roles that ADVISER can use as input if such roles are not present in the source policy. The EXTRACT algorithm works by randomly selecting one of the elements in the user-permission matrix that is set to $true$ (in other words, it selects a random existing authorisation). It then takes, for that authorisation (which is a combination of a user and a permission), all users that have that permission and all permissions granted to that user. This set of users and permissions is called a pseudo role. The process of generating such a pseudo role is generated k times, where k can be varied as needed. Usual values are between $k = 10$ for smaller data sets and $k = 1000$ or higher for larger data sets. EXTRACT counts how often it generates the same pseudo role (pseudo-roles are considered the same if they consist of the same users and permissions, irrespective of order) and outputs the pseudo-roles it generated, including the number of times each pseudo role was generated. The count is used by ADVISER as a weight for that pseudo-role.

ADVISER is used to sort the unsorted user-permission matrix. It sorts the (order of the) users and the (order of the) permissions independently. The process for both is identical and in the context of ADVISER users and permissions are usually called *items*. The steps ADVISER goes through are as follows:

1. group all items that are assigned to the same (pseudo) roles in an *item set*;
2. sort the item sets by descending size — simply put, the size is determined by the number of items related to that set;
3. go over each item set;
4. for each item set, insert the item set in a list of sorted item sets so it is next to the item set it is most similar to;
5. the similarity between two item sets is calculated using the Jaccard Coefficient [13] — simply put, by the number of similarities between the two sets;
6. when each item set is placed in the sorted list, the list is expanded into a list of sorted items.

The list of sorted users and the list of sorted permissions can finally be used to construct a new matrix, and this is the (sorted) user-permission matrix that is shown in Figure 2.

3 Improving visualisations

The visualisations produced by ADVISER are a great starting point for the elicitation of roles. We can immedeately spot a number of patterns that would make an excellent starting point for a new role in Figure 2. We propose two methods (visualising and aggregating metadata) to provide more context to such a visualisation. Both methods make use of user-permission metadata. We define user, permission or authorisation metadata (which we will just call *metadata* from now on) as any contextual data that comes with a user, permission or authorisation. We include a list of relevant types of metadata in Appendix B, that serves both as an example and as input for any practical work based on our thesis.

3.1 Data sets used

Because metadata was not available for the data sets used in [16] we used different data sets based on real-world access control settings. A particular data set we will predominantly use for the remainder of this thesis is based on the access policy of a technology company in the Netherlands. This data set contains 1370 users and 321 permissions and will be called *techcompany*. In this data set, permission are

usually representing physical objects such as doors. The unsorted and sorted visualisations of this data set can be found in Figures 3 and 4. A complete list of data sets acquired for, and used in this thesis are laid out in Appendix A.

3.2 Overlaying metadata

Our first proposed method is overlaying metadata onto the sorted authorisation matrix. Selecting the right metadata to overlay over the matrix can present an operator with more contextual information and can give additional visual cues that can help them find contextually meaningful roles in the visualisation.

While it is possible to overlay any kind of metadata over an authorisation matrix, certain types of metadata are particularly useful. Using access logs, for each authorisation we can calculate its usage. Usage can be represented either with a boolean value indicating whether or not a certain authorisation has been used in a certain period of time, or with a numerical value indicating how often an authorisation has been used in a certain period of time. We will focus on the first (whether or not a certain authorisation has been used) since it gives an interesting visualisation using the *techcompany* dataset and is usually readily available. However, other types of metadata can just as well be used such as the number of failed authentications or the number of days since the user last used their permission. An example of the later is shown in Figure 5.

Figure 9 shows the *techcompany* data set overlayed with the authorisation usage metadata. An authorisation is marked as yellow if and only if the authorisation has been used (as indicated by the access logs) in a specific half-year period. From this visualisation one can already draw several conclusions. Many of the authorisations in the visualisation are not actively used. This insight could raise interesting questions within an organisation. *"Why are there so many unused authorisations?"*, *"Does this information change drastically if we visualise a larger period?"* and *"Can we safely revoke these unused authorisations?"* are all relevant questions that can be asked.

Another observation that can be made is that, in the *techcompany* data set, one can already identify continuous blocks of unused permissions. Similar questions to the ones previously described can be asked about these continuous blocks of unused permissions. Any lessons learned from the visualisation can subsequently be translated to the roles that are to be generated.

If continuous blocks of unused permissions are indeed not longer desired, they can be left out completely resulting in less authorisations and possibly fewer roles. In Section 3.6 we propose a variant of ADVISER that further emphasise patterns present in the metadata, by placing — where possible — groups with similar metadata together.

3.3 Aggregating metadata

Our second proposed method is to provide an operator with aggregated metadata that is contextually relevant to possible roles identified by the user. Where the main purpose of our proposal discussed in Section 3.2 was to assist a human with identifying potentially interesting sections in the visualisation, this proposal is meant to assist a human in providing a context for that interesting section. Remember that if identified roles are to be accepted by an organisation it should be understandable where they came from.

For a human to be able to give context to a potential role they should have access to aggregated metadata for that potential role. The relevant metadata here primarily concerns metadata attributes on the users and permissions in that potential role. This provides insight in how that potential role is built up (answering questions like *"what types of users and permissions are in that potential role?"*), without having a close look at every user and permission in that potential role. Appendix B contains a list of types of metadata attributes that are useful in this context. Armed with this new knowledge a human can proceed to commit a potential role and document the relevant context for that role, or conclude that the potential role is not meaningful after all and carry on with other potential roles.

One way to aggregate data is to provide a summary of the users and permissions contained in that role. Consider again our fictional university from Section 2. In this system, all permissions may be doors that can have the *building* they belong to as an attribute. All staff, on the other hand, may have the *faculty* they work for as an attribute. Consider a hypothetical possible role that is in need of some context. The aggregated data may show that almost all staff in that possible role work for the behavioural sciences faculty and that all doors in that possible role belong to the building of that faculty. A conclusion would be that this possible role *means* to give access to the behavioural sciences faculty to its staff. However, more conclusions are possible. Perhaps the permissions in the possible role represent only a subset of all doors in the building, and the possible role is actually meant to give a specific subset of the staff access to a specific group of rooms on the faculty building (such as HR staff to the HR floor of the building).

Another way to aggregate data, which would complement the aggregation method described in the previous paragraph, could be to aggregate the entire

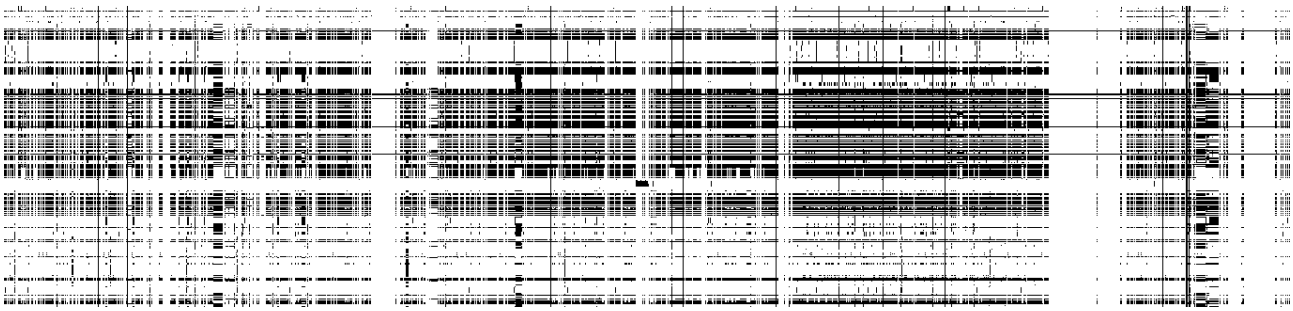


Figure 3: An unsorted user-permission matrix representation of the *techcompany* data set.

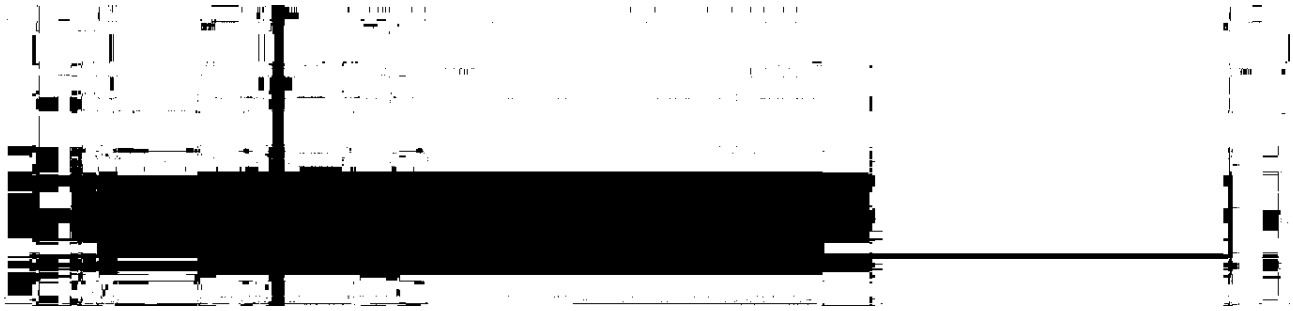


Figure 4: A sorted user-permission matrix representation of the *techcompany* data set generated by the ADVISER algorithm.

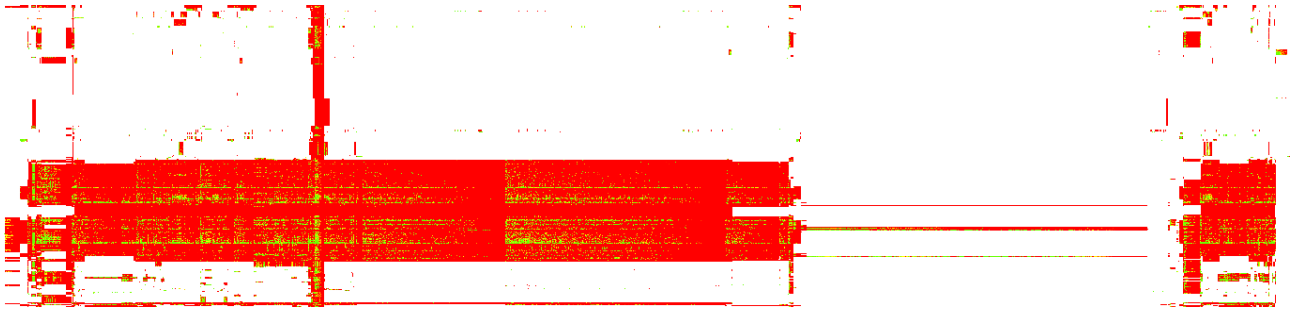


Figure 5: A variant on Figure 4 overlayed with authorisation usage metadata. Red authorisations have been used more than 200 days ago, and the greener the authorisation, the more recent an ahorisation has been used. Sorted with mADVISER.

data set and use it as a context for the aggregated data as described previously. Consider again our fictional university. If after aggregating the entire data set it turns out that almost all staff of the behavioural sciences faculty is included in the possible role, and the same is found for the doors in the building of that faculty, it can be concluded that the possible role is indeed meant to provide access to the behavioural sciences faculty building to its staff.

By showing data aggregations as context to a human, they can make more informed decisions over possible roles. In particular, they are better equipped to identify whether or not a possible role is contextually relevant and are able to document this contextual meaning.

3.4 Iteration

An iterative process of role mining has already been briefly described by Colantonio et al. in their paper on EXTRACT and ADVISER. They describe three steps of iterative visual role mining:

1. Identify the most relevant roles with a visual inspection. The most relevant roles are likely the roles corresponding to the biggest sections of the visualisation. These should be set aside.
2. Assign meaning to these roles together with other people within the organisation (such as managers of users and administrators of the permissions in the roles) and verify if these roles are accepted.
3. After accepting the identified roles, the user-permission assignments corresponding to the roles can be removed from the data. A new round of analysis can then be done on the remaining data.

We go beyond the work of Colantonio et al. by implementing the iterative process using EXTRACT and ADVISER and discussing the effectiveness in Section 4. An example of such an iterative process is shown in Figures 6 through 8. Figure 6 shows the *techcompany* dataset after applying EXTRACT and ADVISER. Figure 7 shows a typical role selection. The selection includes twelve roles elicited out of Figure 6 and highlight a number of large structures. These roles include a small number of "false positives" or authorisations that were not present in the original dataset. Introducing this inaccuracy allows us to select more freely and reduce the number of roles needed in the end (Section 2 goes a little deeper into this trade-off). Figure 8 finally shows a new visualisation of the *techcompany* dataset. In this visualisation, every authorisation included in any of the twelve selected is left out. The EXTRACT and ADVISER algorithm are run again over the resulting authorisations, resulting in Figure 8.

3.5 Limitations of EXTRACT and ADVISER

Colantonio et al. write in their paper that the visualisations generated by the EXTRACT and ADVISER algorithms are not necessarily a globally optimal one (in terms of the metrics they used) but instead is locally optimal one. When working with these visualisations, it should be possible to construct roles from parts of the visualisation that the algorithms may have failed to put together due to this behaviour.

The proof of concept we discuss in Section 4.1 addresses this concern.

3.6 Improving ADVISER

As outlined in Section 3.2, we propose to overlay visualisations generated by the ADVISER algorithm with contextual information to aid humans in making conclusions about the access policy as a whole, as well as in identifying relevant parts of the visualisation as possible roles. Figure 9 shows an example of such an overlay using the *techcompany* data set.

To give more structure to this combined visualisation, we propose a new variant of the ADVISER algorithm: mADVISER (*Metadata and Access Data VISualizER*). The aim of mADVISER is to sort the users and permissions in such a way that in the final visualisation the overlay metadata is sorted as much as possible, with only minimal changes to the structures identified by the ADVISER algorithm. mADVISER is shown as Algorithm 1.

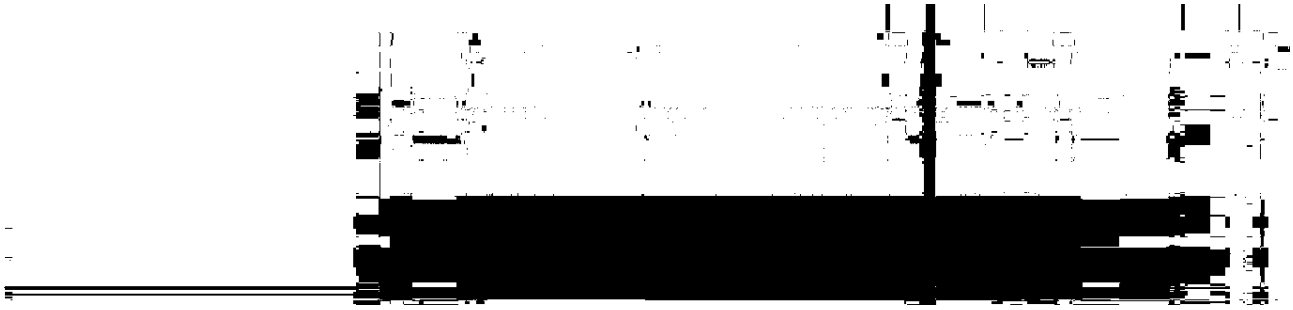


Figure 6: A visualisation of the *techcompany* dataset before starting with an iterative process.

Figure 7: Figure 6 after selecting 12 roles. The authorisations included in any of these roles are marked in blue. Marked in red are a number of "new" authorisations, these are explained in Section 4.1

Figure 8: A new visualisation of the *techcompany* dataset with only the authorisations not included in any of the roles indicated in Figure 7.

Algorithm 1 The mADVISER algorithm.

1: **procedure** ADVISER(*USERS, PERMS, ROLES, UA, PA*)
2: $\sigma_U \leftarrow$ SortSet(*USERS, UA, ROLES*)
3: $\sigma_P \leftarrow$ SortSet(*PERMS, PA, ROLES*)
4: **return** σ_U, σ_P

5: **procedure** SORTSET(*ITEMS, IA, ROLES*)
6: $\overline{ITEMS} \leftarrow \{I \subseteq ITEMS$ **sorted by descending item** $weight(I) \mid \forall i, i' \in I, roles(i) = roles(i')\}$
7: $\sigma \leftarrow \emptyset$
8: **for all** $I \in \overline{ITEMS}$ sorted by descending areas of $roles(I)$ **do**
9: **if** $|\sigma| < 2$ **then** σ.append(I)
10: **else**
11: **if** Jacc(I, σ.first) $>$ Jacc(I, σ.last) **then**
12: $p \leftarrow 1$
13: $j \leftarrow$ Jacc(I, σ.first)
14: **else**
15: $p \leftarrow |\sigma| + 1$
16: $j \leftarrow$ Jacc(I, σ.last)
17: **for** $i = 2 \ldots |\sigma|$ **do**
18: $j_{prec} \leftarrow$ Jacc($I, \sigma[i-1]$)
19: $j_{succ} \leftarrow$ Jacc($I, \sigma[i]$)
20: $j_{curr} \leftarrow$ Jacc($\sigma[i-1], \sigma[i]$)
21: **if** max{$j_{prec}, j_{succ} > j \wedge min(j_{prec}, j_{succ}) \geq j_{curr}$} **then**
22: $p \leftarrow i$
23: $j \leftarrow$ max{j_{prec}, j_{succ}}
24: σ.insert(p, I) ▷ between the $(p-1)^{th}$ and the p^{th} elements
25: **return** σ.expand

The difference between the two algorithms is printed in bold on line 6. Instead of just taking the items (either users or permissions, depending on the stage in which the algorithm is) in the order in which they are present in the original access matrix, we instead first sort the items based on a function $weight(ITEM)$. This function returns the weight of an item. The weight of the item is calculated by averaging all values in the overlay dataset for that item. If the overlay dataset contains boolean values (which should then be interpreted as 1 for $true$ and 0 for $false$) this will result in a decimal value between 0 and 1, representing the fraction of values that equal $true$ for that item. If the overlay dataset is numeric, this will result in a decimal value that represents the average overlay value for that item. Defining $weight(ITEM)$ like this makes sure the sorting works both when boolean values are used as well as when numerical values are used. Following this approach the $\overline{ITEMS}$, or item sets, are sorted internally. Even when the item sets are re-ordered later on in the algorithm, the order of the items within the item sets remains the same.

The order of items within an itemset is not defined in ADVISER. This gives us room to freely change that order without changing the intended behaviour of ADVISER.

Applying mADVISER over a number of data sets results in the visualisation shown in Figures 9 through 14. In these figures, the first figure of a data set visualise the data using ADVISER, while the second visualises the data using mADVISER. The most notable difference between the visualisations with ADVISER and mADVISER is that we can now clearly identify a few areas without any overlay data. Remember that the the overlays represent whether or not a user had used the permission over a given period.

The areas without any overlay data (in other words, without yellow marks) are, in this example, representing groups of users that have not used a group of permission in a given period. The constitution of these groups can be examined using the tools we proposed in Sections 3.2 and 3.3. Based on such an examination it can be decided to not include certain users or permissions in a new role, or investigate why permissions are not being used (perhaps a door is broken and opens automatically without employees having to present their credentials). We can also clearly see groups of users who have not used any permissions at all over that period. Using the proposed tools this group of users can be examined and an appropriate course of action established, such as removing these users from the dataset altogether (effectively revoking all their authorisations) and generating a new visualisation using a process similar to the process described in Section 3.4.

The effect of mADVISER can be subtle when used on an entire data set, especially if many smaller structures are present (the difference between Fig-

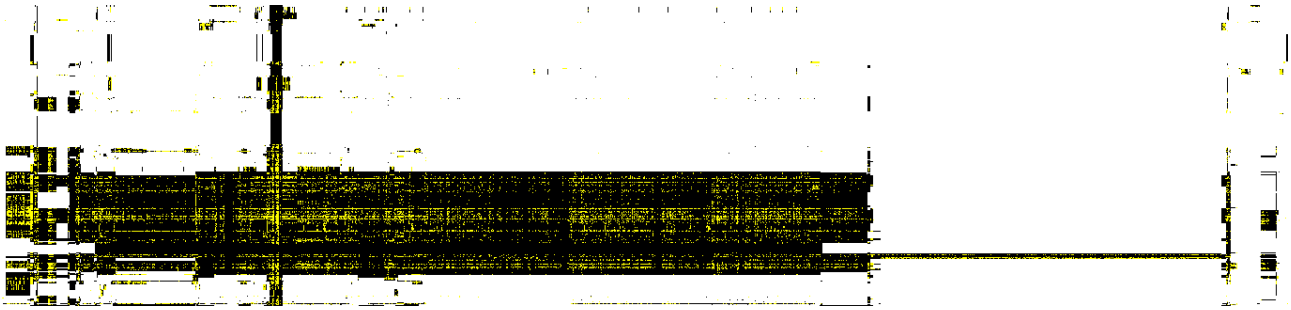


Figure 9: Figure 4 overlayed with authorisation usage metadata. Each yellow authorisation has been used at least once in a one month period.

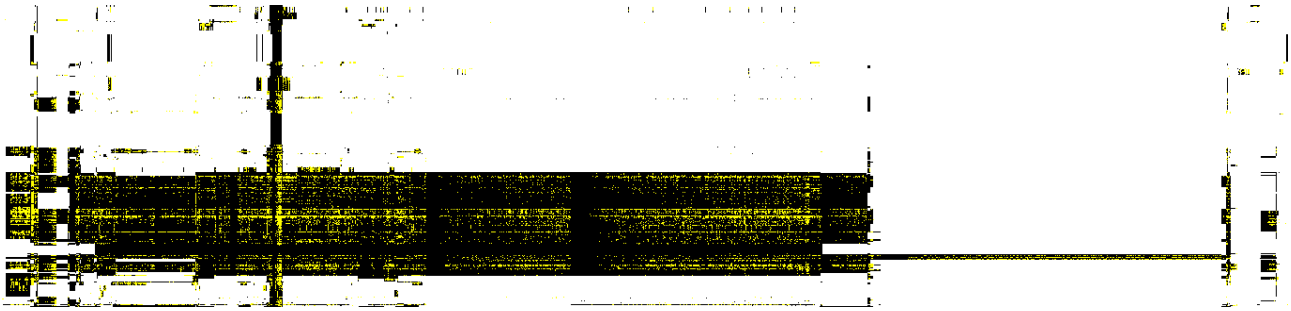


Figure 10: Figure 4, sorted by mADVISER.

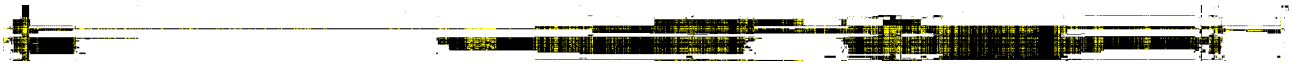


Figure 11: The *museum* data set, sorted by ADVISER.

Figure 12: The *museum* data set, sorted by mADVISER.

Figure 13: The *fincompany* data set, sorted by ADVISER.

Figure 14: The *fincompany* data set, sorted by mADVISER.

ures 13 and 14 is much more profound than that between Figures 9 and 10). The effect also becomes more profound when applied to a subset of data. An ideal example of this can be found in Figures 15 and 16. In this visualisation, we effectively visualise only one candidate role of the *techcompany* data set. In the sorted version, we can more clearly distinguish a large group of inactive users and unused permissions. We can also identify a number of near-universally used permissions and a number of more-than-average active users. This visualisation can prompt further questions, such as why some permissions in this candidate role are used more often than others. Perhaps this is an indication that it might be more meaningful to split the candidate role in two candidate roles. Note, however, that this is an ideal example, and depending on the amount of noise and subsection of the data set visualised, results will be more or less profound.

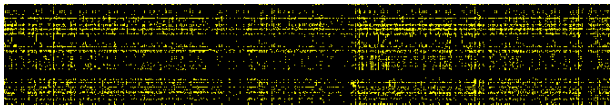


Figure 15: A visualisation of a subset of users and permissions from the *techcompany* data set.

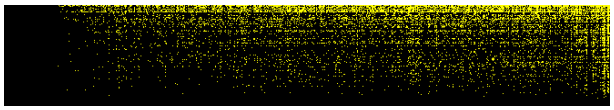


Figure 16: The data from Figure 15, sorted by mADVISER.

mADVISER further optimises visualisations with overlays, making it easier for humans to digest the information provided by the overlay and the visualisation itself. Because mADVISER addresses an undefined state in ADVISER it does not change its documented behaviour (except for a neglegible increase in execution time).

4 Validation

In Section 3 we propose a number of methods that we suppose contribute to the visual role mining framework. To validate whether or not these methods are actually beneficial, we validate these methods together with a number organisations. For this validation we built a proof of concept (PoC) that implements EXTRACT, mADVISER and our proposed methods.

[1]https://github.com/jonathanjuursema/vrm-app

4.1 Software prototype

We developed a proof-of-concept role mining application to aid in validating our approach with external organisations. Our PoC is a web-based application built in Python on the Tornado web framework. The application is open source and can be found on GitHub.[1] Our application accepts formatted CSV (*comma seperated values*) files as an input. This makes sure that we can easily re-format data from any source system and ingest it in our application; a necessity given that we want to work with various organisations for our validation.

As mentioned earlier, our application implements both the EXTRACT and mADVISER algorithms as well as our other suggestions. For EXTRACT, we use $k = 1000$ – this value was determined empirically using the datasets used in this thesis and had acceptable performance in terms of execution time. The basic functionality of our application, the implementation of the algorithms and our method of overlaying metadata is shown in Figure 17. The main point of interaction with the application is the interactive version of the visualisation shown in Figure 10 that takes up most of the screen. There are several keyboard and mouse controls available to interact with the visualisation that enable exploring the visualisation and the selection of possible roles.

Additionally, various visual cues are present. Label 1 marks a part of the regular visualisation. The yellow dots represent, as they do in Figures 9 and 10, the overlayed metadata. A possible role selected by an operator is marked by label 2. Green marked areas represent "correct" authorisations (authorisations that would be granted if the role was to be committed, and that are also present in the source data) whereas red marked areas represent "new" authorisations (authorisations that would be granted if the role was to be committed, but that was not present in the source data). Label 3 marks a previously committed role. In committed roles blue areas represent "correct" authorisations in already committed roles, whereas brown authorisations represent "new" authorisations in already committed roles. Label 4 marks a number of buttons that makes further interactivity available to the operator besides the keyboard and mouse commands. The top button allows the operator to commit the current selection (remember label 2) as a role. The lower button presents an overlay with in-depth information about the selection, shown in Figure 18. Additional information about the state of the application is marked with label 5. It shows the currently selected user and permission (the one that the mouse is currently hovering over) as well as the number of users and permissions included in the current selection (if available). It also includes a warning if the operator made a selection that includes "new" authorisations (here

Figure 17: A screenshot of our application loaded with the *techcompany* data set. The labels indicate a part of the regular visualisation (1), a selected role (2), a previously committed role (3), application controls (4) and basic metadata of the selection (5).

called "superfluous"). One last button is available for the operator that allows the operator to export the raw visualisation, as rendered in the tool, as an image. This button is labelled 6. This functionality is also the source of the various visualisations present in this thesis.

Our implementation of the metadata aggregation method is shown in Figure 18. This window shows a complete list of all users and permissions included in the current selection (labelled 1) as well as which "new" authorisations are included in the possible role, to facilitate closer examination (labelled 2). The main dialog window contains aggregated details about the selection made by the operator. It contains of both users (labelled 3) and permissions (labelled 4) a number of attributes that are present in the metadata and their aggregated totals. The fractions shown are to be interpreted as follows: *this selection includes 3 users with carriertype 1, that is 0.37% of all 812 users that have carriertype 1* (taking the first metadata attribute of the users as an example).

The PoC also takes into account the limitations to EXTRACT and ADVISER put forth in Section 3.5. In particular, the application allows for the flexible selection of structures for exploration or commitment as roles by providing the possibility to make a flexible "extended" selection in addition to the more traditional primary selection. The latter remains simple and is always a continuous rectangle. This makes it possible to correct for optimisation oversights by the EXTRACT and ADVISER algorithms.

The iterative functionality discussed in Section 3.4 is also implemented in the PoC. After committing roles, they can be removed from the visualisation at will. At any time either both algorithms or only the ADVISER algorithm can be re-run to generate a visualisation of what becomes effectively a customisable subset of the authorisation matrix. If removing a committed role leaves a certain user without further permissions, or a permission without further users, the item is removed from the visualisation entirely. This further reduces the complexity of the visualisation by reducing the size.

4.2 Interviews

To verify the effectiveness of our proposed additions, we visit a number of organisations to conduct interviews in the context of various access control solutions. The interviews are summarised in Table 1. We aim for a combination of organisations and access control solutions that are both securing physical and digital assets to see if our approach works for various use cases. In addition to the interviews, we have additional contact with some of the organisations to get a better understanding of how they employ access control in their organisation.

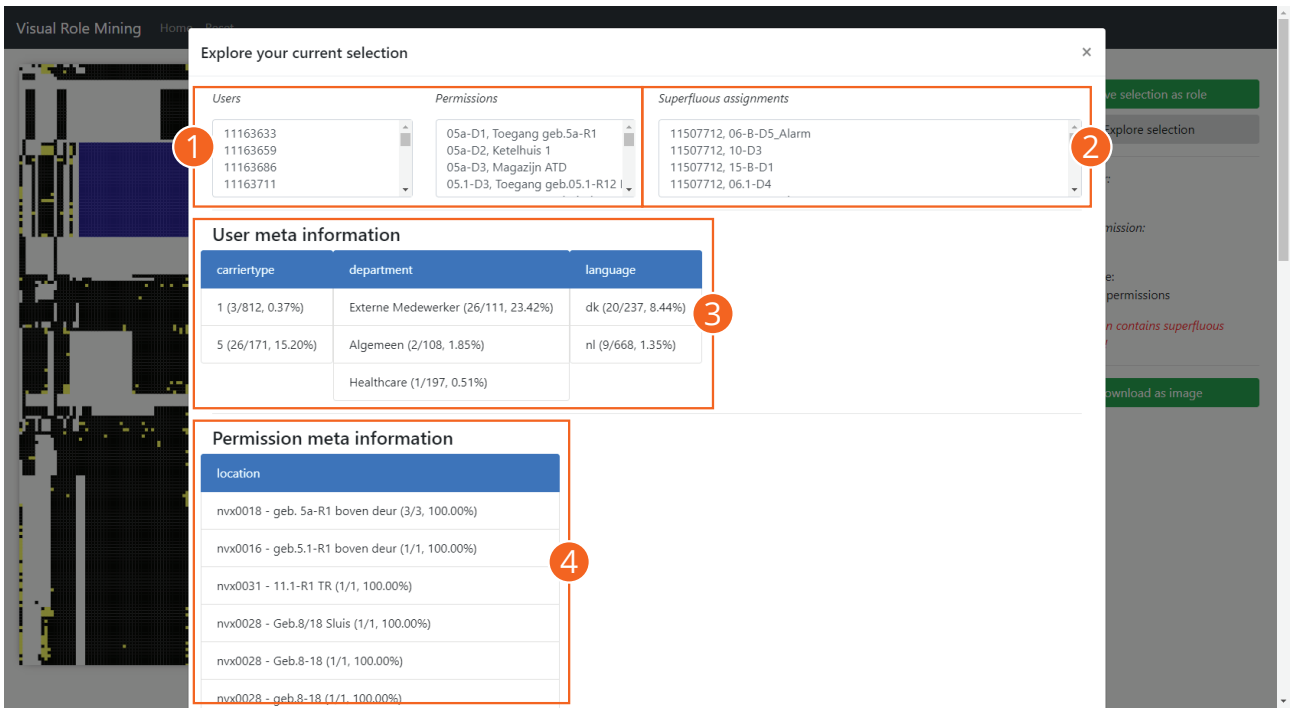


Figure 18: A screenshot of our application loaded with the *techcompany* data set, while exploring a possible role using the aggregation method. The labels indicate the contents of the current selection (1), "new" authorisations (2) and aggregated metadata attributes of the users (3) and permissions (4) in the selection.

Interview	Organisation Type	Data used
1	Technology company	techcompany
2	Museum	museum
3	Financial company	fincompany
4	University	demo

Table 1: An overview of the conducted interviews.

During these interviews we would first ask the interviewee a number of open-ended questions regarding their access control solution in place at their organisation, if that information is not already known to us beforehand. We then provide, and we consider this the primary component of the interview, a demonstration of our PoC and invite the interviewee to get hands-on with the application themselves. Please note that this hands-on exercise is not intended to be a user test since we are not interested in our particular implementation of the proposed additions. Instead, we guide the interviewee when they have questions and, where applicable, ask the interviewee to reflect on the information they were able to get out of the application. We explicitly did not ask for their opinion on the application itself.

We have to address two caveats planning these interviews. First, since we had no access to organisations that were considering or executing a migration between access control solutions, we needed to simulate such a migration. Second, because none of the access control solution contexts in which we interviewed used a simple or extended form of Role-Based Access Control, there were no roles to compare the mined roles against. Therefore we chose to view the hands-on exercises as successful if the interviewee was able to extract, in their own opinion, contextually meaningful roles during the exercise as this is also the desired end result during an actual migration.

4.2.1 Interview data sets

During the hands-on exercise we try for the interviewee to work with a data set they are comfortable with. For some organisations this turned out to be possible. We prepare the data sets by requesting from each organisation the following four pieces of data: a binary access control matrix (which would serve as our UP for the EXTRACT algorithm), (non-personal) metadata regarding the users (such as the department they work in), metadata regarding the permissions (such as the name of the permission or a building/room such a permission is associated with) and the access log (which user access which permission and when).

For some organisations we were unable to work with data from that organisation during the interview. For this purpose, we created a synthetic demonstration data set, *demo*. This data set is further explained in Appendix A.

4.2.2 Interview takeaways

We find that in general, our approach towards visual role mining is regarded as positive; the hands-

on exercise is well received by all but one interviewee. Two of the organisations are very proactive during the hands-on exercise, making suggestions and actively leading the discussion. These organisations also express interest in a follow-up evaluation of their access policy. During the hands-on exercises, some of the organisations also make personal notes containing employee or permission identifiers that they think they should inspect in the actual access control system at a later point, for example because a group of users seems to have more permissions than necessary.

During hands-on exercises with organisations working with their own data, all organisations are able to make assumptions about sections of their access policy (such as *"A candidate role containing a small number of permissions shared by a large number of employees is probably some basic access to the main entrances."*) based on their interaction with the application. These assumptions are mostly made based on an inspection of the visualisation and the (number of) permission included in such a section, and could be confirmed by providing the interviewee with aggregated metadata of that section. In some cases the aggregated metadata does not confirm the assumptions, however. In these cases, often the aggregated metadata allows the interviewee to formulate a new assumption. For example, a group of people having access to rooms where network switches are located turn out to be from facility services, not necessarily from IT.

Although not explicitly assumed, we expected that during role mining employees would not be thought about individually (due to the sheer number of employees in a larger organisation). We however find that organisations working with their own data often refer to individuals or small groups of individuals. This may be relevant for future work. Remember that, due to privacy considerations, we decided not to include personally identifiable information in the data sets. Therefore we are unable to verify assumptions based on individuals (such as *"These broadly authorised individuals are probably Amy, Mark and John from facility services."*).

We also find that for organisation exploring their own data the hands-on exercise becomes more of a validation exercise rather than a role mining activity. While we intend for the hands-on exercise to loosely mimic a role mining scenario, this shift of focus indicates that visual role mining — together with our proposed additions — can also be used as a tool organisations can use to verify or audit their current security policy.

Interviewees working with the *demo* data set find it more difficult to make assumptions about the data. This could be explained in a number of ways. It is possible that this follows from the fact that they are working with data they are not familiar with, or that the *demo* data set is not a good representation of a real security policy. It is however equally likely that our approach is simply not applicable to all role mining use cases. One interviewee indicates that they appreciate the approach of aggregating metadata in order to onderstand the contextual meaning of a role, but do not see the added value of visualising the access policy. They prefer a more automatated system that would propose candidate roles, which they can then give contextual meaning (and approve or deny) using aggregated metadata over that candidate role.

Comparison with the original work of Colantonio et al. [16] is tested informally. Our interviews start with a discussion of only the visualisation produced by mADVISER. Only after discussing the mADVISER visualisation (without any metadata to aid the discussion other than the overlay discussed in Section 3.2) did the hands-on exercise with our PoC take place. We notice that although the mADVISER visualisation allowed for some basic conclusions to be made, these conclusions where mostly educated guesses (*These people are probably from department* X.) or limited in contextual relevance (*There are a few people who have only minimal access, but who are they?*). Only after starting the hands-on exercise did interviewees arrive at more concrete conclusions and got more actionable insight.

We did not need the iterative approach discussed in Section 3.4 during our interviews. For the interviewees there was already a lot to learn from the "first" visualisation iteration, and we could easily fill the available interview time discussing it. We did however not exhaustively discuss the visualisation during these interviews, often leaving many of the smaller structures undiscussed.

Practical feedback (such as feature proposals, user experience tweaks etc.) was also proposed during the interviews. Although these are not relevant in the context of our interviews (remember that we intend to only explicitly evaluate concepts, not our implementation of them), we include them for use in follow-up work. The practical feedback can be found in Appendix C.

5 Related work

We mostly work with the EXTRACT and ADVISER visual role mining algorithms proposed by Colantonio et al. [16] due to the effectiveness, performance and documentation of their algorithms. There is, however, another visual role mining effort. Eucharista et al. [17] implemented a varient to ADVISER which they call VISRODE. The VISRODE algorithm uses the Sørensen–Dice coefficient instead of the Jaccard distance used by ADVISER to sort the user-permission matrix based on pseudo-roles generated by EXTRACT — beyond that, their approach is

roughly the same. It would therefore be trivial to apply our contributions to VISRODE instead of ADVISER and rewrite our software prototype to support VISRODE. To the best of our knowledge these are the only recent developments regarding visual role mining algorithms.

Regarding practical implementations of role mining with a visual component, Schlegelmilch et al. [5] developed a role mining application that identifies clusters in existing user-permission assignments. Instead of our approach of visualising the entire data set, they visualise the hierarchy of these (already identified) clusters in the application. Their application is similar to ours in that is also allows for humans to contribute their own contextual insight to guide the algorithm.

Another (visual) role mining application is the Role Modeling Assistant (RMoA) [14] by Giblin et al. Their application features a number of functions. RMoA allows for the visualisation of metadata (as shown in Figure 19), whereas our PoC only summarises metadata in textual form. It would, however, be trivial to adapt our tool to also visualise the metadata. It is interesting to note that although RMoA also works with metadata, it does not use metadata in the process of giving contextual meaning to roles the way we do — RMoA helps humans to understand which metadata attributes tell something unique about a user (or in their words, are *descriminatory*). Indeed, RMoA leaves giving roles a contextually meaningful name as a topic for future work. Finally, it is also possible to manually define roles in RMoA, in addition to making use of role mining algorithms.

RMiner [18] is a third (visual) role mining application that combines multiple role mining algorithms into one application. RMiner supports multiple role mining algorithm and features some rudimentary visualisation tools. However, they use visualisation merely as a means to validate the output of the role mining algorithms, whereas we use visualisation as the actual role mining process. RMiner also only supports "autonomous" role mining algorithms — algorithms that output a complete RBAC policy — and leaves no room for direction by people, unlike our method. It should be noted that it is possible to add support for other role mining algorithms to RMiner.

To the best of our knowledge, this is the only work directly related to visual role mining. In Section 2 we discuss more general related work focusing on role mining in general and RBAC.

6 Discussion

We explored several ways of using metadata to improve the process of role mining: by creating an overlay for the visualisation produced by ADVISER (Section 3.2) and by providing aggregated metadata for subsections of the visualisation (Section 3.3). These serve to help a human in eliciting contextually meaningful roles using visual role mining. To increase the added value of overlaying metadata we also proposed mADVISER: a variant of ADVISER (Section 3.6) that allows for new patterns to emerge from the visualisation when using such a metadata overlay. We created a proof of concept that implements all of the above (Section 4.1) and asked several organisations for their opinion of the result in an interview (Section 4.2).

From the interviews we learned that our approach seems to work well. Most of the organisations appreciated the hands-on exercise and were interested in a follow-up; we received various post-interview questions from some of the organisations regarding conclusions drawn during the interview. We believe that our work has helped these organisations gain insight in and critically reflecting on their security policy.

In particular, we validated that the two main contributions of our work related to improving visual role mining clearly assisted during the interviews. The metadata overlay in combination with mADVISER contributed to guiding the hands-on exercise: interviewees generally used both the mADVISER visualisation and the metadata overlay to identify interesting areas in the visualisation. These areas could also be correlated with well-defined groups of users (such as curators in the *museum* data set) most of the time, as determined by the aggregated metadata available for these areas. Several questions along the lines of *"why do these people never use their authorisation"* confirmed that mADVISER is an improvement over ADVISER, and the aggregated metadata again helped answer most of these questions.

Although the intention for our approach was for it to be used during a migration to a system implementing Role-Based Access Control, based on observations from our interviews it can also successfully be applied for periodic analysis of an existing access control policy. This shows that visual role mining has more applications than has up to this point been discussed on literature.

In conclusion, we have proposed a number of additions to the process of visual role mining. We subsequently validated these additions as useful during hands-on exercises with real-world datasets and show that visual role mining has a place in the context of (periodic) analysis of existing access control policies.

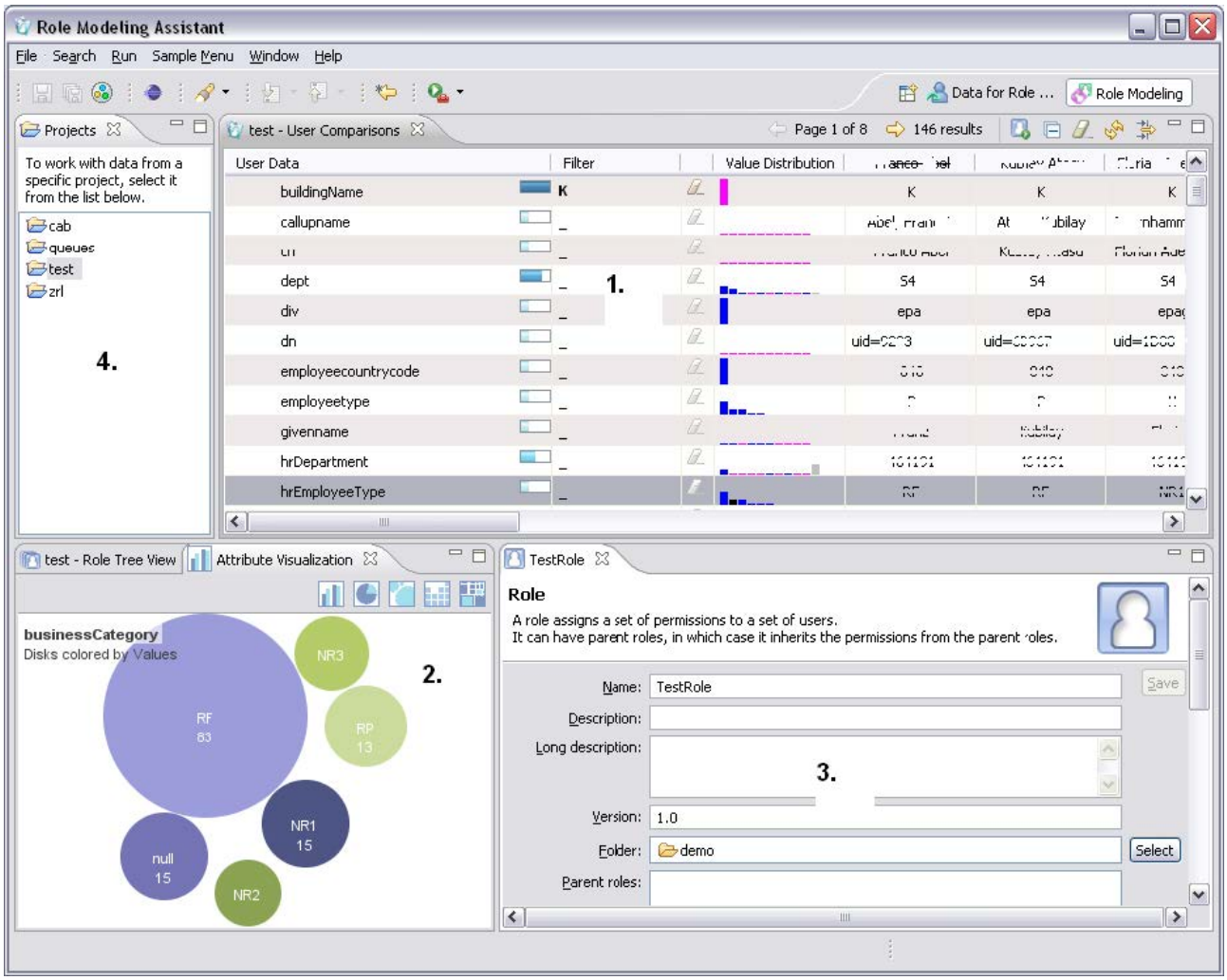


Figure 19: An example of how RMoA visualises metadata. The area marked 1 is meant to help identify possible interesting metadata attributes (mostly via the value distribution), and the area marked 2 gives a breakdown of the values of a particular metadata attribute. Picture from [14].

7 Limitations & suggestions for future work

Although we believe our work to contribute new insights as discussed in Section 6, we were not able to fully explore some of avenues we set out for this thesis. In particular, in Section 4.2 we note that we were not able to find organisations that are considering or executing a migration between access control solutions. This means that our validation only partially proves the effectiveness of our methods for real role mining scenarios. Full validation can be achieved by performing a study following one or more organisations in an actual migration project, using the methods (and perhaps an improved version of our proof of concept). Due to time constraints we are also limited to a relatively small number of interviews. Therefore we only had limited opportunity to compare the effectiveness of the work of Colantonio et al. to our own. Although our validation is positive in general and the performance comparison with the work of Colantonio is indirectly addressed, these results are therefore only qualitative. This can be addressed in a replication study. Another shortcoming of the interviews is that the iterative angle was not validated during the interviews only because the allotted interview time was already filled just discussing the larger structures. However, since this also means that we did not get to the smaller structures it cannot be said that the iterative approach is not necessary either. A follow-up study could perform longer, more exhaustive interviews in which the entire security policy is considered.

We also limited the complexity of the data considered in this thesis. In Section 4.2 we describe how we pre-processed our data for the interviews. Due to the absence of existing roles as described in Role-Based Access Control, we opt to simply use EXTRACT to generate pseudo-roles. Instead, we could use other existing data structures in the access control solution and try to convert these to roles. A follow-up study could address the question of whether chosing this approach over using EXTRACT would yield other results, although this does not necessarily fall within the scope of this thesis. Additionally, during the data pre-processing we explicitly chose to strip additional attributes from the data sets. These are mostly temporal, such as restrictions on the hours during which a permission can be used. This is done because the EXTRACT and ADVISER algorithms do not support temporal constraints and neither do any of the other visual role mining algorithms. There are, however, a number of role mining algorithms[3, 4] that do support time-based constraints. A follow-up study can investigate whether it is possible to find a hybrid algorithm — combining visual role mining and these temporal role mining algorithms — and if our suggested methods remain useful in the context of such a hybrid approach.

Finally, research into the practical functionality of a role mining application may be warranted before beginning development of a production ready role mining solution. Aside from focussing more on how functionality is presented to operators of the software (in other words, real user-tests as opposed to the more conceptual approach we have taken), follow-up research could also focus on new concepts such as suggestion of candidate roles based on a selection (*This area is selected, automatically extend this selection with other relevant areas.*), adding features to identify individuals (as discussed in Section 4.2.2) and effectively visualising metadata attribute breakdown (effectively combining this work with the work on RMoA discussed in Section 5).

8 Acknowledgements

I would like to thank a number of people for their contributions to this thesis. Their contribution made it possible to make this thesis what it is. Maarten Everts provided valuable feedback, critique and encouragement in his role as thesis supervisor. Nedap in Groenlo was a wonderful host to me as a graduation intern. They allowed me to take the project in any direction I deemed interesting, supporting me all the way. Albert Dercksen and Wouter Kuijper provided valuable substantive feedback as well as help in finding the right people within the organisation whenever needed. Loes van Hove helped me make the most out of the interviews. I finally want to convey a big thanks to the people of the organisations who agreed to an interview. They went out of their way to help me with information, resources and interview time.

9 References

[1] Edward J Coyne. “Role engineering”. In: *Proceedings of the first ACM Workshop on Role-based access control*. ACM. 1996, p. 4.

[2] Ravi S Sandhu et al. “Role-based access control models”. In: *Computer* 29.2 (1996), pp. 38–47.

[3] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. “TRBAC: A temporal role-based access control model”. In: *ACM Transactions on Information and System Security (TISSEC)* 4.3 (2001), pp. 191–233.

[4] James BD Joshi et al. “A generalized temporal role-based access control model”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.1 (2005), pp. 4–23.

[5] Jürgen Schlegelmilch and Ulrike Steffens. “Role mining with ORCA”. In: *Proceedings of the tenth ACM symposium on Access control models and technologies*. ACM. 2005, pp. 168–176.

[6] Indrakshi Ray, Mahendra Kumar, and Lijun Yu. “LRBAC: a location-aware role-based access control model”. In: *International Conference on Information Systems Security*. Springer. 2006, pp. 147–161.

[7] 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*. ACM. 2006, pp. 144–153.

[8] Maria Luisa Damiani et al. “GEO-RBAC: a spatially aware RBAC”. In: *ACM Transactions on Information and System Security (TISSEC)* 10.1 (2007), p. 2.

[9] Sabrina De Capitani Di Vimercati et al. “Access control policies and languages”. In: *International Journal of Computational Science and Engineering* 3.2 (2007), pp. 94–102.

[10] Ian Molloy et al. “Mining roles with semantic meanings”. In: *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM. 2008, pp. 21–30.

[11] Alessandro Colantonio et al. “A formal framework to elicit roles with business meaning in RBAC systems”. In: *Proceedings of the 14th ACM symposium on Access control models and technologies*. ACM. 2009, pp. 85–94.

[12] Ian Molloy et al. “Evaluating role mining algorithms”. In: *Proceedings of the 14th ACM symposium on Access control models and technologies*. ACM. 2009, pp. 95–104.

[13] Flavio Chierichetti et al. “Finding the jaccard median”. In: *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2010, pp. 293–311.

[14] Chris Giblin et al. “Towards an integrated approach to role engineering”. In: *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*. ACM. 2010, pp. 63–70.

[15] Jaideep Vaidya, Vijayalakshmi Atluri, and Qi Guo. “The role mining problem: A formal perspective”. In: *ACM Transactions on Information and System Security (TISSEC)* 13.3 (2010), p. 27.

[16] Alessandro Colantonio et al. “Visual role mining: A picture is worth a thousand roles”. In: *IEEE Transactions on Knowledge and Data Engineering* 24.6 (2012), pp. 1120–1133.

[17] A Eucharista and K Haribaskar. “Visual elicitation of roles: using a hybrid approach”. In: *Orient. J. Comput. Sci. Technol* 6.1 (2013), pp. 103–110.

[18] Ruixuan Li et al. “RMiner: a tool set for role mining”. In: *Proceedings of the 18th ACM symposium on Access control models and technologies*. ACM. 2013, pp. 193–196.

[19] Wouter Kuijper and Victor Ermolaev. “Sorting out role based access control”. In: *Proceedings of the 19th ACM symposium on Access control models and technologies*. ACM. 2014, pp. 63–74.

[20] Barsha Mitra et al. “A survey of role mining”. In: *ACM Computing Surveys (CSUR)* 48.4 (2016), p. 50.

[21] *Role Based Access Control | CSRC*. Nov. 2016. URL: `https://csrc.nist.gov/Projects/Role-Based-Access-Control`.

A List of data sets

For this thesis we have had the opportunity to work with a number of (real world) access control data sets. These will be listed in this appendix.

A.1 Access control at a technology company

The *techcompany* data set represents the access control policy from a technology company in the Netherlands. It consists of 1370 users and 321 permissions which represent employees and (for the most part) physical doors and barriers. This data set also contains access logs and metadata and is visualised in Figure 16.

A.2 Access control at a museum

The *museum* data set represents the access control policy from one building of a large museum in the Netherlands. It consists of 2792 users and 106 permissions which represent employees, constractors, external users and physical doors and barriers. This data set also contains access logs and metadata and is visualised in Figure 12.

A.3 Access control at a financial company

The *fincompany* data set represents the access control policy of the building complex of an international financial company located in the Netherlands. It

consists of 901 users and 27 perimssions which represent employees, contractors and physical doors. This data set also contains access logs and metadata and is visualised in Figure 14.

A.4 Demo access control file

The *demo* data set is a fictional data set created based on a technical university in the Netherlands. It consists of 100 users and 25 permissions that are supposed to represent students, staff and building sections related to a number of academic departments. This data set also contains metadata and is visualised in Figure 20.

Figure 20: The *demo* data set, sorted by ADVISER.

B List of metadata attributes

In Section 3 we propose methods to improve upon existing visual role mining techniques using metadata. This list is a collection of metadata attributes that can be useful when implementing these methods into practice. The list is mostly a collection of metadata attributes encountered often in organisations' systems during the course of this research.

In this context only attributes are considered that can be identified on an item but that are not exclusive to that item (in other words, they are shared with other items of the same type). This excludes attributes like a unique permission identifier or a user's real name. The implicit idea behind these attributes is that they can categorise a group which the item is part of.

B.1 Attributes for users

- Department
- Faculty
- Group
- Study, chair
- Job title
- Office location
- Start of affiliation
- Type of affiliation (such as part-time, freelance, contractor)
- Rank (such as, in more formal organisations, commander, lieutenant, brigadier)
- Certification

B.2 Attributes for permissions

- The owner of the permission or resource
- Building or building section (for doors)
- Subnet, VLAN (for computer systems)
- Security classification (such as, in more formal organisations, classified, top-secret)
- Certification required to use

B.3 Attributes for authorisations

- Any user attribute for the user that created the authorisation
- Start of the authorisation
- Usage frequency (if access logs are available)
- Last use (if access logs are available)

C Suggested features for role mining applications

This appendix is an overview of features and/or functionality suggested for the proof of concept suggested to us during interviews. These features and/or functionality are not necessarily interesting from an academic point of view, but are documented here for use in further development of the proof of concept. The list is in no particular order.

- Include credentials (such as ID cards) that are not directly assigned to an employee, such as general purpose cards that can be given to visitors.
- For any given user or permission, see what permissions they have access to or what users have access to them right from the visualisation.
- Manually rearrange (blocks of) rows and columns.
- Visualise permissions between certain times, and indicate the differences between them. (*Who has access during the night, but not during the day?*)