Trust and Risk-Based Access Control for Privacy Preserving Threat Detection Systems

Nadia Metoui[1,2], Michele Bezzi[3], and Alessandro Armando[1,4]

[1] Security & Trust Unit, FBK-Irst, Trento, Italy
[2] DISI, University of Trento, Italy
[3] SAP Labs France, Security Research, Sophia-Antipolis, France
[4] DIBRIS, University of Genova, Italy

Abstract. Intrusion and threat detection systems analyze large amount of security-related data logs for detecting potentially harmful patterns. However, log data often contain sensitive and personal information, and their access and processing should be minimized. Anonymization can provide the technical mean to reduce the privacy risk, but it should carefully applied and balanced with utility requirements of the different phases of the process: a first exploration analysis needs less details than an investigation on a suspect set of logs. As a result, a complex access control framework has to be put in place to, simultaneously, address privacy and utility requirements. In this paper we propose a trust- and risk-aware access control framework for Threat Detection Systems, where each access request is evaluated by comparing the privacy-risk and the trustworthiness of the request. When the risk is too large compared to the trust level, the framework can apply adaptive adjustment strategies to decrease the risk (e.g., by selectively obfuscating the data) or to increase the trust level to perform a given task. We show how this model can provide meaningful results, and real-time performance, for an industrial threat detection solution.

Keywords: trust, risk, privacy, utility, privacy-preserving threat detection

1 Introduction

Big Data analytics for security, based on the correlation of security events from several log files, play a key role in state-of-the-art threat detection and prevention techniques [25,33]. Threat detection systems, as intrusion detection systems, are typically characterized by an automatic pattern or anomaly detection phase, which can highlight suspicious events, followed by a detailed investigation performed by an human expert to decide if a real attack is detected or it is a false positive. In this phase, the expert often inspects the raw data (log files) triggering the alert.

However, log data often contain sensitive and personal information (e.g., user ids, IP addresses), and, although the security investigation can constitute a legitimate purpose for their processing, the access and usage should be limited to the relevant and necessary data to accomplish a specific analysis.

Anonymization is often used to pre-process the data, removing sensitive information from log files, and enabling further processing with minimal privacy risk. However, this is achieved by deteriorating the quality or utility of the data. Although some analytics can still be run on anonymized log data [18], in many cases the anonymization can impact the quality of results, and, ultimately, decrease the ability to detect and react to cyber threats.

We propose a trust- and risk-aware access control framework for Threat Detection Systems (TDS), which addresses the concerns described above. Our framework does not require an *a priori*, i.e. off-line, anonymization of the data sources. The automatic pattern detection phase uses the original dataset and anonymization is applied only if further, human based, analysis is needed on the resulting data.

The risk level of each data request is dynamically evaluated by the access control decision point based on several parameters (e.g., context, role and trustworthiness of the requester), and, if needed, anonymization is applied on the specific resulting data set. In this study, we focus on *re-identification* risk and, following common practice, we use k-anonymity as risk metrics. However, our approach is not bound to these choices and can be adapted to use alternative metrics (e.g., l-diversity, t-closeness, and differential privacy). To summarize, the approach has multiple advantages:

- it limits the impact on the utility, since we apply the anonymization only after running the pattern detection on the original data, and we adapt the anonymization strategy to the specific pattern.
- it provides a simple framework to address the, often conflicting, privacy and utility requirements.
- it is based on concepts as trust and risk, which have an intuitive meaning in the business world.
- it offers a flexible configuration, allowing to define a trade-off between security and privacy suitable to the organization's priorities (risk and trust levels can be tuned to set a permissive or a restrictive access, the adjustment strategy can be configured to optimize utility or performance priorities, etc..)

To evaluate the effectiveness of the proposed approach, we have developed a prototype implementation and we experimentally evaluated it by running a number of threat detection patterns based on the SAP Enterprise Threat Detection (ETD) solution. The results obtained (reported in 4.6) show that the model can address the utility and performance requirements of a realistic use-case.

Structure of the paper In the next Section we provide a threat detection system use case, which we use to illustrate the main features of our risk-aware privacy preserving approach. In Sect. 3 we present a trust- and risk-aware approach for privacy enhancing access control model and we describe its application on the proposed use case Sect. 4 is dedicated to an experimental evaluation of the proposed approach in terms or performance, scalability and data utility (after anonymization). Lastly, we discuss the related work in Sect. 5 and we conclude in Sect. 6 with some final remarks.

2 Use Case

Modern intrusion detection systems at application level (called Threat Detection System, TDS, herein)[1], collect security information on the application stack and correlated it with context information, to detect potential threats.

A TDS, typically, first collects application level log files from various systems, it enriches the logs with contextual (e.g., time, location) information, and finally stores the events data in a database table.

The events data are then periodically automatically analyzed against predefined threat patterns to detect potential anomalies and attacks. Any matching with these patterns generates real time alerts. When an alert is raised a human user is informed and actions must be undertaken to evaluate and react to the alerts (e.g., investigate the validity of the alert or locate its cause). Figure 1 illustrates the architecture of the system, as well as the different users involved in the process. For our purpose, two operations of the TDS are important:

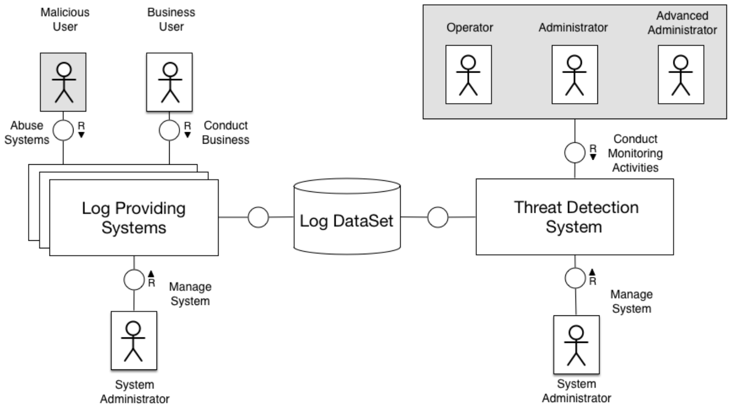

Fig. 1. Business Roles and System Landscape

- *Pattern detection.* A pattern is a representation of a combination of suspicious log events that could indicate a threat. It is often implemented as a set of filters applied to the event database, and compared with some thresholds. If this threshold is exceeded an alert is triggered. For instance the ensemble

[1] We refer to these systems as TDS, to distinguish them from network level intrusion detection systems (often called IDS or SIEM). Moreover, we base our description on the SAP Enterprise Threat Detection, but the analysis could be applied to other solutions, including IDS. For a comparison between application and network level intrusion detection systems, see [16]

Table 1. Roles

Operator	Classify alerts and report patterns anomalies His/Her tasks requires access to pattern detection results (events/log data related to the suspicious pattern) in case of alerts.
Administrator	Has all *Operator* tasks and privileges. They can also Investigate alerts, Create or Reconfigure patterns. He/She should have access the detection results and events data related to the patterns.
Advanced Administrator	Has all *Administrator* tasks and privileges. Can also grant exceptional access to the data by attributing higher trust level to an *Operator* or an *Administrator*

of events indicating a *Failed Login* initiated by the same source (e.g., Terminal) may indicate a *Brute Force Attack* if the number of attempts exceeds, say, 20 attempts in less than 10 minutes.

- *Investigating Alert.* In this phase, an human operator investigates the alert, to decide if this is an actual attack or a false positive. It may require access to the details of the events triggering the alert, or at least of some attributes of these events.

The Investigation phase implies that TDS *Users* access some detailed information from the logs, we will provide some examples in Sect. 4.3. These *Users* have different functions within the process of monitoring potential threats, investigating them and reacting. Table 1 gives an example of how you can divide the user roles in the TDS, and the corresponding access authorizations required to execute their tasks.

Log files contain personal information, such user names, IP addresses, etc, and despite the security investigation can constitute a legitimate purpose for their processing, it should be done according to the data minimization principle, reducing the access to personal data. Therefore, TDS systems often perform some (pseudo-)anonymization before analyzing the event data, such as replacing real user name or IDs with pseudonyms.

However, with the increasing variety and complexity of collected log files, a full anonymization of the log dataset before processing could, on one hand, provide a good privacy protection, but also significantly impact the performance of the system, both in terms of the *utility* (the quality of results of the pattern detection phase, or the information available to the operator for the manual inspection) and processing time (anonymization on large data set could be time consuming, and on data stream re-run regularly)

To address this challenge, a more dynamic approach is needed: instead of anonymizing the complete event data base beforehand, whenever an user performs an operation accessing event tables, we have to apply specific anonymization methods which reduce the privacy risk, but preserving the most relevant information for that operation. In practice, the anonymization process should be customized for each operation (to preserve the information useful for completing the task) and for each type of users, which can have different level of access to the data. In the next section, we will propose a framework that to realize this scenario.

3 Privacy-Enhancing Risk-Based Access Control

In this section we provide a general description of our Trust- and Risk-Based Access Control model, based on previous model we introduced in [1], and we explain how it can be adapted to the use case described in Sect. 2.

3.1 Trust and Risk-Based Access Control

The framework evaluates access decisions using the trust and risk values related to the request. This access evaluation can be represented by the function $Auth(obj, u, p)$ defined as follows. User u is granted permission p on object obj iff the trustworthiness of the incoming request is larger or equal to the risk, i.e.,

$$Auth(obj, u, p) = \begin{cases} \textbf{allow} & \text{if } T(u, C) - R(obj, p, C) \geq 0 \\ \textbf{adjust}_{\Sigma}(T, R) & \text{otherwise} \end{cases} \quad (1)$$

Where $T(u, C)$ is the trustworthiness of the request, which depends on user u and context information C (e.g., security emergency) and $R(obj, p, C)$ is the risk, which depends on the requested object *obj* (e.g. a table a file.) and the permission p (e.g., read or write) [2] and context C.

Access request in evaluated by comparing the risk of the access to the trustworthiness, which plays the role of risk threshold (in practice, the maximum amount of risk that an user can take in a certain context): If $T \geq R$ access is **allowed**, vice-versa if $T < R$, the access cannot be granted *as is*. However, risk-based access control models have been originally devised to increase information accessibility, and they tend to be more permissive (still keeping risk under control) than traditional access control systems. Along this reasoning, in case of $T < R$ instead of denying access, the system can propose an *adjustment strategy* $\sigma \in \Sigma$, to reach the condition $T \geq R$. Clearly, there are two possible methods for adjustment strategies: *(i)* Risk mitigation, σ_R , (decreasing R), or *(ii)* Trust enhancement strategies, σ_T, increase the trustworthiness T. Risk mitigation strategies can include anonymizing the data, or imposing additional obligations on data handling, whereas trust enhancement could be implemented by (temporary) privilege escalation or provision of additional credentials [1] However each of these strategies is expected to have some *negative* side effects: for example, anonymization degrades data quality, impacting utility or privilege escalation can increase the complexity of the security governance; accordingly, the choice of the optimal strategy should balance the access control objectives with the impact of the adjustment strategies.

[2] In most cases the dependency of risk from permission is mediated by roles. For the sake of simplicity, we do not consider here roles, for an extension of this model including roles, we can follow the lines of the models described in [6].

If we focus on data access and privacy risk (as the use case in Sect. 2), and limiting the adjustment strategies to anonymization, we should find an optimal anonymization strategy $\hat{\sigma_R}$ among all the possible anonymization strategies Σ_R, which allows for data access limiting risk (so fulfilling Eq. 1), and, at the same time, maximizing the utility, after the strategy σ_R is applied: $U(\sigma_R)$. This is can be expressed as classical utility-privacy optimization problem:

$$\hat{\sigma_R} = \arg\max_{\sigma_R \in \Sigma_R} U_{\sigma_R(obj)} \tag{2}$$

$$s.t. \quad R_{\sigma_R} \leq T \tag{3}$$

In practical cases (as we will see in Sect. 3.2), the number of mitigation strategies can be very limited, and the optimization problem is reduced to testing a small set of anonymization strategies, and estimating either based on numerical thresholds or expert assessment, if the utility is sufficient for the business task. If this is not the case, trust enhancement mechanism can be triggered or access is denied.

In the next subsections we will show how trust and risk can be modeled, with a focus on the application to Threat Detection Systems.

3.2 Privacy Enhancing Approach

Risk Model: Risk in IT security is generally expressed in terms of the likelihood of occurrence of certain (negative) events times the impact [12]. In this paper we will deal with the privacy breach risk. Privacy breaches are often associated with the concept of *individual identifiability*, used in most data protection privacy laws (e.g., EU data protection directive [23], Health Insurance Portability and Accountability Act (HIPAA) [27]). To prevent *individual identifiability* the regulation requires that disclosed information (alone or in combination with reasonably available information from other sources or auxiliary informations [22]) should not allow an intruder: to identify individuals in a dataset (identity disclosure) or to learn private/sensitive information about individuals (attribute disclosure) with a very high probability or confidence (see [32,29]).

To assess the privacy risk (when releasing a given dataset) various privacy metrics have been proposed in the literature (see [4,10] for a review). The most popular metric is k-anonymity [26][3].

In the k-anonymity approach attributes (or columns) in a dataset are classified as follows:

- *Identifiers:* Attributes that can uniquely identify individuals e.g., full name, social security number passport number.
- *Quasi-identifiers (QIs) or key attributes* Attributes that, when combined, can be used to identify an individual, e.g., age, job function, postal code
- *Sensitive attributes:* Attributes that contain intrinsically sensitive information about an individual, e.g., diseases, political or religious views, income.

[3] Other privacy metrics exist (for example, ℓ-diversity, and t-closeness, see [13]), but k-anonymity is still a *de-facto* standard in real applications

In presence of identifiers the re-identification risk is clearly maximum (i.e., probability of re-identification $P = 1$), but even if identifiers are removed, combining QIs individuals can be singled out and this implies a high risk. k-anonymity condition requires that *every* combination of QIs is shared by at least k records in the dataset. A large k value indicates that the dataset has a low re-identification risk, because, at best, an attacker has a probability $P = 1/k$ to re-identify a data entry (i.e., associate the sensitive attribute of a record to the identity of a User). Therefore the (re-identification) risk related to a k-anonymous data-view v is:

$$risk(v) = 1/k_v \times I \tag{4}$$

where I is the impact. In most cases any identity disclosure is considered equally important, and, thus for simplicity sake we will set the impact $I = 1$ this will allow us to normalize the risk and the trust values to $[0, 1]$ (for a discussion on the impact normalization, see [1]).

Trust Model: Several definitions have been proposed in for the concept of Trust in the literature [15]. In this paper, trust plays the role of risk threshold: a very trusted user is allowed to take a large risk (for a discussion on how relating this definition with more classical trust metrics, see [1]. We assign trust level $T_{user}(u)$ to the users according to their competence/roles and the tasks this role is expected to fulfill (see Table 1). Following data minimization policy, a role should have *enough* trust to access the resources (data) needed to fulfill these task and not more. These values are assigned on a scale from 0 to 1, where 0 means that basically no privacy risk can be taken, therefore impacting significantly the quality of accessible data; and 1 means the role should be granted access to maximum amount of data.

Note: the same request can be used to fulfill different tasks in different contexts for instance *"Perform Maintenance and Improvement tasks"* or *"React to a Security Incident"* (if an alert is raised). In the latter the need to react to a security threat overcomes the privacy requirements and the request should receive more permissive results thus have higher level of trust we will define the two context-related trust levels as $T_{context}(Alert) = 1$ and $T_{context}(noAlert) = 0$

To compute the request trustworthiness (total trust value) we can use the approach for multi-dimensions trust computation proposed in [20], where the total trust is computed as weighted sum of trust factor values.

$$T = \sum_{i=1}^{n} W_i \times T_i(\beta_i) \tag{5}$$

with$\{\beta_1 ... \beta_n\}$ a set of trust factors and $T_i()$ and W_i respectively the trust function and weight of the i^{th} trust factors, with $\sum_{i=1}^{n} W_i = 1$ and $T \in [0, 1]$

We are in a 2-dimensions trust case thus we will express our total trust value as the following

$$T(q) = W \times T_{user}(u) + (1 - W) \times T_{context}(c) \tag{6}$$

Adjustment Strategies:

Risk Mitigation: A possible way to decrease the disclosure risk is anonymization. Anonymization is a commonly used practice to reduce privacy risk, consisting in obfuscating, in part or completely, the personal identifiable information in a dataset. Anonymization methods include [9]:

- *Suppression:* Removal of certain records or part of these records (columns, tuples, etc., such UserId column);
- *Generalization:* Recoding data into broader classes (e.g., releasing only a Network prefixes instead of IP addresses etc.) or by rounding/clustering numerical data;

Traditionally, anonymization is run off-line, but more recently risk-based access control models, which use in-the-fly anonymization as mitigation strategy have been proposed [2].

Trust Enhancement: Trust enhancement mechanisms can realized by asking the user to provide additional guarantees (i.e., additional credential) or proofs of obligation enforcement. In our case, we may require trust enhancement for an emergency alert, where there is the need to increase the access to the original data for investigation. This could be implemented as a change in the context, which impacts the trust value according to Eq. 6, or simply increasing temporarily the trust of an user T_u (privilege escalation).

4 Experimental Evaluation

We validate our approach by applying the described model to the scenario described in Sect. 2. The threat detection system is expected to provide real time and a accurate results. In this section of the paper we will investigate the impacts our approach has on the functioning of the threat detection system and whether the expected *Performance* and *Utility* matches the requirement of a real-time.

More in details, as mentioned in Sect. 2 the threat detection system allows to automatically detect potential attack patterns, and then, if an additional investigation is needed, a human operator can browse the log data of the events corresponding to certain pattern for manual inspection.

Ideally, the Operator should be in the position to perform the manual analysis, so to decide if the detected pattern is a false or true positive, on data where the personal information are anonymized (or in any case, where the re-identification risk is low). In fact, if the operator has not sufficient information to decide, they needs to access less anonymized (more risky) data, or in other words to get higher access privileges (trust enhancement) getting Administrator rights, or directly involving an Administrator.

Accordingly, we need to check:

- *Utility*. Does the model allow a low trusted operator (i.e., small risk threshold) to perform the investigation in most cases, and relying on trust enhancement for the remaining cases?
- *Performance*. Does the additional anonymization step impact real-time performance?

Before addressing these questions (see Sect. 4.6), we need to describe our prototype implementation (Sect. 4.1), the data set and its classification from a privacy risk perspective (Sect. 4.2), the selection of typical patterns used for the validation (Sect. 4.3), the utility measure (Sect. 4.5) and the trust level setting (Sect. 4.4).

4.1 Prototype Implementation

We developed a prototype of our framework, based on the implementation described in [3]. Our prototype is implemented in Java 8 and uses SAP HANA Database. It is composed from 3 main modules:

- The *Risk Aware Access Control module:* mimics a typical XACML data flow, providing an implementation of the PDP, the PEP and the PIP functionality as well as a set of authorization policies.
- The *Risk Estimation module:* evaluates the privacy risk using pre-configured criteria (privacy metrics, anonymization technique, identifying information). It compares the privacy risk to the request trustworthiness level, then produces an estimation of the minimal anonymization to be applied in order to meet this level.
- The *Trust & Risk Adjustment module:* we implemented the Risk Adjustment Component to perform anonymization. It uses ARX [17] a Java anonymization framework implementing well established privacy anonymization algorithms and privacy metrics such as k-anonymity, ℓ-diversity, t-closeness, etc. (the Trust Adjustment Component was not implemented in this version of the prototype.)

4.2 Data Set and privacy classification

To test the performance of our framework in the TDS use case, we used a data set containing around 1*bn* record of log data collected from real SAP systems deployed in test environment. The logs data set is composed 20 fields (in Table 2 we present a summery of the most important fields)

As described in Sect. 3.2, to anonymize a data set, we first need to formalize our assumptions on the attributes that can be use to re-identify the entry, or, in other words, classify the attributes in terms of identifiers, QIs and sensitive attributes. This classification, typically, depends on the specific domain. QIs should include the attributes a possible attacker is likely to have access to from other sources, whereas sensitive attributes depend on the application the anonymized data are used for. For example, in our experiments we set (obviously) User ID as an identifier, and the IP address as a quasi-identifier. Similarly,

Table 2. An extract of the Log dataset columns, privacy classification of each column and anonymization technique to be applied

Log Events data set		
Attribute	Type	Anonymization
EventID	Non-Sensitive	
Timestamp	Sensitive	
UserId (Origin)	Identifier	Suppression
UserId (Target)	Identifier	Suppression
SystemId (Origin)	QI	Generalization
SystemId (Target)	QI	Generalization
Hostname (Origin)	QI	Generalization
IPAddress (Origin)	QI	Truncation
MACAddress (Origin)	QI	Truncation
TransactionName	Sensitive	
TargetResource	Sensitive	

we assume that the Transaction name (the called function) cannot provide any help for re-identification, therefore we consider it a sensitive attribute (and no anonymization will be applied). Table 2 provides an example of this classification, and, for identifiers and quasi-identifiers, the corresponding anonymization methods applied.

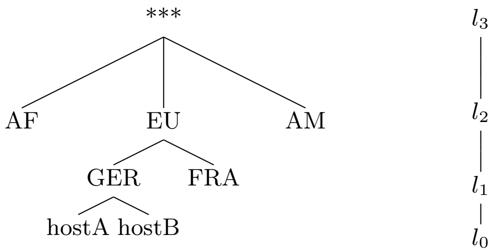

Fig. 2. The generalization hierarchy for host names is organized as following: l_1 and l_2 are a location based generalization by country then by continent. in level l_3 host names are totally obfuscated and entirely revealed at the level l_0.

4.3 Pattern detection and investigation

In our experiments we focus on 5 typically *Patterns* with different and increasing complexity in terms of the size of the returned views and the privacy risk. Two different kind of queries are used during each phase respectively *Detection Queries* and *Investigation Queries.* The selected queries {**Q1** ... **Q5**} described in Table 3 are all *Investigation Queries.* An *Investigation Query* is a "SELECT *" extracting all the details of the events corresponding to certain pattern.

4.4 Roles and Trustworthiness levels

We have 3 roles Operator, Administrator and an Advanced administrator with increasing access requirements (to fulfill their tasks), and we that expect to

Table 3. Queries: Resulting views Size and Risk level

Query	*Corresponding Pattern*	*View Size*	*Risk Level*
Q1	Brute Force Attack	Large (50550)	Very High ($k = 2$)
Q2	Security Configuration Changed	Large (40300)	Medium ($k = 7$)
Q3	Blacklisted Function Called	Medium (14500)	Very High ($k = 1$)
Q4	Table Dropped or Altered	Small (228)	Medium ($k = 6$)
Q5	User Assigned to Admin Group	Very Small (12)	Very High ($k = 1$)

require increasing privacy clearance, or in other words,to be able to accept larger risk. Usually, for k-anonymity, k values in the range $3-10$ are considered medium risk, $k > 10$ low risk, and for $k \leq 2$ the risk is very high (clearly, for $k = 1$ the risk is the maximum, no anonymity) [24] . Therefore we propose the parameter setting described in Table 4, where for sake of simplicity we have considered a single trust factor $T = T_u$ (i.e. we set $W = 1$ in Eq. 6).

Table 4. Users/Roles Privacy clearances and Trustworthiness levels

Role	*Access*	*Privacy*	*Trust Level*
	Requirement	*Clearance*	(Risk Threshold)
Operator	Low	Minimal ($k > 10$)	$T_u \in [0.05, 0.1[$
Admin.	Medium	Medium ($k > 2$)	$T_u \in [0.1, 0.5[$
Adv. Admin.	High	Maximum ($k \leq 2$)	$T_u \in [0.5, 1]$

4.5 Utility Evaluation

The effect of anonymization terms of utility is a widely discussed issue in the literature several generic metrics have been proposed to quantify the *"damage"* caused by anonymization (see [14] for a review). However, these metrics do not make any assumption on the usage of the data (so called *syntactic metrics*), limiting their applicability on realistic use-cases.

Other approaches propose to assess the accuracy loss (Utility loss) of a system (i.e., IDS in [19], Classifier in [5]) by comparing the results of certain operations run on original then anonymized dataset using use case related criteria (i.e., in the context of a TDS the comparison criteria can be the number *False positives*)

Although interesting for our context, this approach can not be applied in our use case, since it assumes that the analysis is run directly on anonymized data, whereas, in our use case, the pattern detection is performed on *clear* data, and the anonymization is applied only on the results (data-view).

We propose a method combining both approaches and that would include an evaluation:

– *From Syntactic standpoint:* The information loss caused by the anonymization, we use the Precision Metric that allows us to estimate the precision degradation of QIs based on the level of generalization with respect to the generalization tree depth (e.g., for th generalization tree 2 if we allow access to continent instead of host-names we used the 3^{rd} level generalization out of 4 possible levels so $d_p(hostnames) = 3/4 = 75\%$ precision degradation for host-names).

- *From Functional standpoint:* The effect of this loss on our use case. During the investigation phase, the operator, mostly, bases their analysis on a subset of attributes, which are different for each attack pattern. Thus we will assign a utility coefficient uc to different attributes based on the relevance of the attribute to the pattern/query.

Combining the to approaches we compute the the utility degradation of a data-view v as

$$U_d(v) = \sum_{a_i \in A} uc_{a_i} \times d_p(a_i) \tag{7}$$

with $A = \{a_1..a_i\}$ the set of attributes in the data set. We also set the precision degradation of the identifiers to $d_p(identifiers) = 1$ as they will be totally suppressed after the anonymization.

4.6 Results and Analysis

For our experiments, we want to investigate: *(i)* Performance: the impact of on-the-fly anonymization (as risk mitigation strategy) on the performance (response time). *(ii)* Utility: we would like to investigate if the quality of resulting data is generally enough to fulfill the expected tasks for every user/role for various pattern investigation.

In order to evaluate these aspects we run several experiments considering 5 patterns and 7 users/role with different trustworthiness level, $t = \{0.055, 0.083\}$ Operators, $t = \{0.12, 0.15, 0.45\}$ Administrators, and $t = \{0.9, 1\}$ Advanced Administrators. The corresponding size and anonymity level of the views returned by the queries (corresponding to the selected patterns) are reported in Table 3. In the rest of this section we will indicate both the queries and the corresponding views as **Q1**, **Q2**, **Q3**, **Q4** and **Q5**.

Performance and scalability To evaluate the the performance of our tool, including the computational overhead caused by the anonymization, we run queries **Q1**, **Q2**, **Q4**, and **Q5** (described in Table 3) using our access control prototype experiment, 100 times for each query to average out the variance of the response time. In Figure 3 we report the results of the experiments for the four queries for the 6 trustworthiness levels.

For **Q1**, we observe that the anonymization process increases significantly the response time. In fact when the query is carried out by the most trusted user ($t = 0.9$), with no anonymization needed, the response time on average is less then 15ms (see Figure 3.Q1, diagonally striped bar corresponding to $t = 0.9$). By decreasing the trustworthiness of the requester the view must be anonymized and the average response time increases to 150ms in the worst case (cf. Figure 3.Q1,

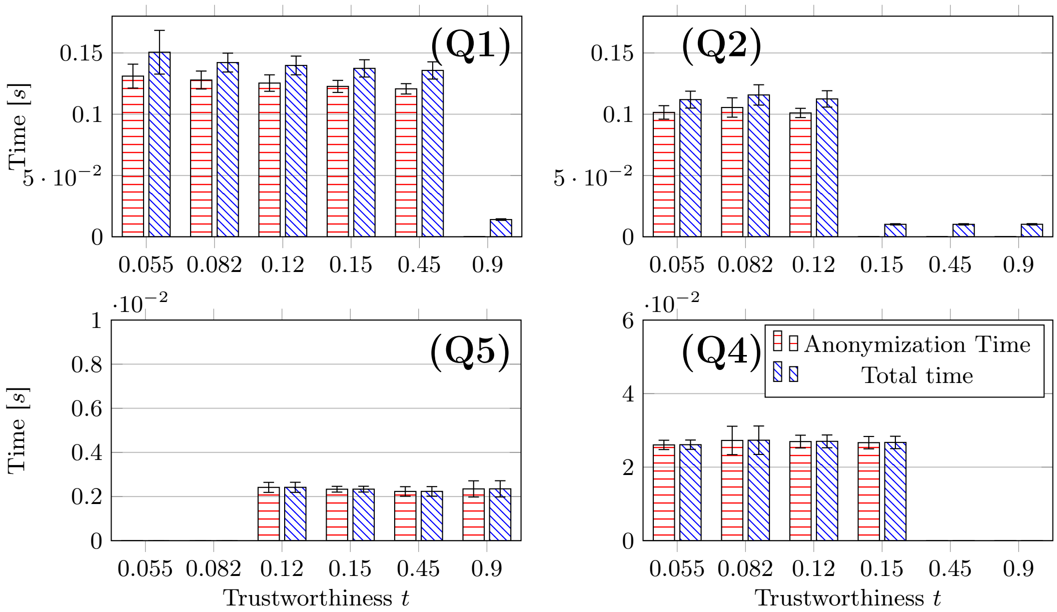

Fig. 3. Average anonymisation time (horizontal striped bars) and average total response time (diagonally striped bars) for **Q1**, **Q2**, **Q4**, and **Q5** (data-views) and 6 different users (trust levels).

diagonally striped bar corresponding to $t = 0.055$). This time difference is entirely due to the anonymization time (130 ms, as shown in Figure 3, **Q1**, horizontal striped bars corresponding to $t = 0.055$). Increasing the trust level decreases the needed anonymization, but it slightly affects anonymization time. We can observe a similar behavior in the other queries (see Figure 3, **Q2**, **Q4**, and **Q5**), with an increase of response time when anonymization takes place and no significant variations in performance for different levels of anonymization. For instance, for **Q2** and **Q4** we have two views with an already medium level of anonymity (respectively $k = 7$ and $k = 6$),the anonymization (when needed) still impacts the performance in the same scale then **Q1** and **Q5** with very low anonymity level (respectively $k = 2$ and $k = 1$).

From these experiments, we observe that when anonymization is applied the response time increases, but, even in the worst cases, the increase is far less than one order of magnitude, and, basically, it has no impact on the real-time response of the system. Moreover, the application of different levels of anonymization (different k in our case) have a small impact. We will investigate in the next paragraph the effect of the data-view size on the Anonymization and Response time.

Let us analyze the behavior of the anonymization time increasing the size of the data set. Typically patterns run in limited time window (e.g., 10 to 30 minutes) producing small-sized data-views (i.e., in the range of $10 - 10^3$). To

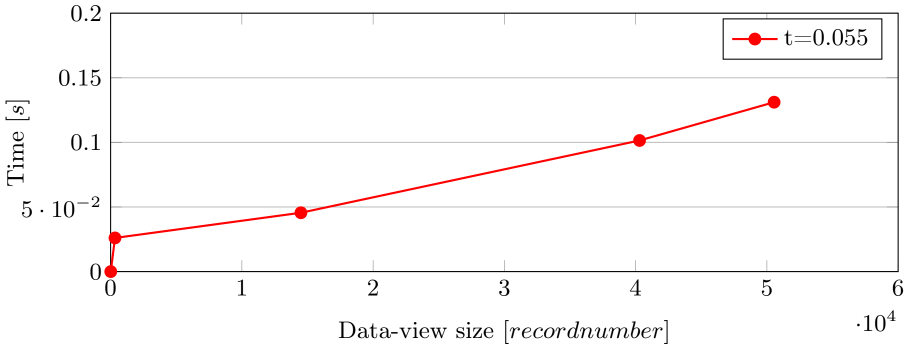

Fig. 4. Average anonymization time variation according to data-view sizes (for trustworthiness $t = 0.055$).

investigate the scalability of our approach, in Figure 4, we report the average anonymization time variation for 5 different data-view {**Q1** to **Q5**} (with 5 different sizes see Table 3) and a low trustworthiness level ($t = 0.055$, so anonymization is always applied). As mentioned above, the worst case (around 510^4 records) takes less than 150ms, and a linear extrapolation of the data allows as to estimate the anonymization time for a 10^5 data view (so, 100 times the typical size) around 200ms, which it can be safely considered as a real-time response for our use case.

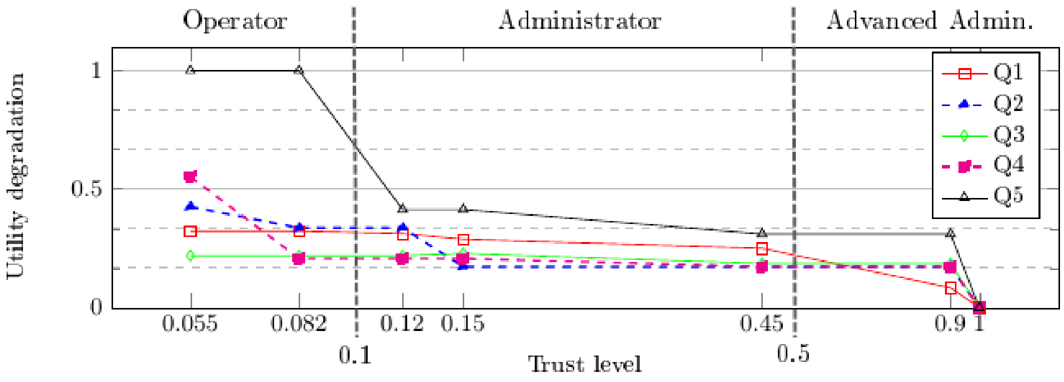

Fig. 5. Utility degradation by trust level for different queries

Utility: Trustworthiness levels (i.e., risk threshold) should be set to allow the best a trade off between data exploitation and privacy protection. In our use case we set our trustworthiness levels respecting a conventional distribution of privacy risk levels presented in Table 4, and we would like to investigate the convenience of this repartition by answering the following question: Do these trustworthiness levels provide enough data (or data with enough utility) to allow each user/role to fulfill their tasks described in Table 1. In Figure 5, we report the the utility degradation according the six selected trustworthiness levels, representing the 3 roles (reported on the top of the figure). We can observe that the utility degradation (obviously) decreases as we increase the trust level, with the limiting case of $t = 1$ with no utility loss (and no anonymization) for the Advanced Administrator. For most of the patterns (4 over 5, so except **Q5**), the Operator role has a maximum utility loss of 30%, showing that the specific anonymization transformations applied are strongly decreasing the risk, and limiting the impact on the utility. That should allow to perform the analysis on the anonymized data, without the need to enhance the trust level (so no need to get Admin rights).

In the case of **Q5**, the anonymization is not able to significantly decreases the risk, without largely impacting the utility. In fact, the Operator is left with no information (utility degradation = 1), and to analyze the result an increase of the acceptable risk threshold (trust level) is needed. Enhancing trust (i.e. assigning Admin rights to the Operator) could reduce the utility degradation in the $30\%-40\%$ range, likely allowing the assessment of the pattern result. We should note, that **Q5** is particularly hard to anonymize, because it has a small amount of events (around 10), and, since k-anonymity is measure of indistinguishability, it needs strong anonymization.

Figure 5 also shows that in most cases increasing the trust level for Administrator or even Advanced Administrator (except of course for $t = 1$, where we have no anonymization) the impact on utility degradation is moderate: for example **Q1** and **Q4** are almost flat in the Administrator zone, similarly **Q2** has a first drop, and stays flat in the Administrator and Advanced Administrator parts. In other words, increasing the risk thresholds, we could take more risk, but we do not gain much in terms of the utility. This counter-intuitive effect is mostly due to the difficulty to find an anonymization strategy able to equalize the risk threshold. As mentioned in Sect. 3.1, in practical cases the number of possible anonymization strategies is limited, and to fulfill the condition of Eq. 1 the final risk may be quite below the risk thresholds (trust values). In practice, in many cases, even increasing the risk thresholds (trust values), it is not possible to find a more optimal (from the utility point of view) anonymization strategy. In Fig. 5 we show the utility loss for four patterns both showing the risk thresholds (dotted lines) and the *actual* risk achieved after the anonymization. In the ideal case, the two curves should be the same, meaning that we could always find a transformation that equalize actual risk and risk thresholds (trust), but in practice we see that we are often far from this optimal condition. For example, for pattern **Q2**, with risk thresholds $t = 0.15$, $t = 0.45$ (Administrator role) and $t = 0.9$ (Advanced Administrator), indicated with red circles, we have the same value of utility degradation. In fact, the anonymization strategy found for $t = 0.15$ case, corresponds to an actual risk of 0.14 (square dots with a circle in Fig. 5, upper-right panel), so quite close to the threshold. Increasing the thresholds to $t = 0.45$ and $t = 0.9$ (round dots with a circle in the figure), no better strategies were found, so the same anonymization strategy is applied, and clearly the final risk is still 0.14 (and utility is the same), well below the thresholds. Similar effects are also present for the other patterns.

The experimental analysis shows that adapting the anonymization to the specific patterns, we can mostly preserve enough information for the investigation, keeping the privacy risk low. In cases where this is not sufficient, typically characterized by small data set, the trust enhancement strategy can support the access to less-anonymized data.

5 Related Work

Privacy issues in intrusion detection: Privacy issues related to shearing and/or using network and log data in IDS and TDSs has received a growing interest in the last few years. Several analysis were proposed in the literature to describe privacy breaches related to sharing and using log data and privacy preserving approaches have been proposed to address these issues.

A strict enforcement of the *need-to-know* principle has been proposed for reducing the likelihood of privacy violations. For example, Ulltveit-Moe et *al.* in [31] propose to set two profiles of users according to the expertise level: the first profile allows monitoring tasks using anonymized data the second consists of security experts, with clearance to perform necessary privacy-sensitive operations to investigate attacks. This model clearly increases the privacy protection, but it is hard to apply in realistic cases, since it relies on anonymizing the entire (source) data set beforehand, resulting in either low privacy or low utility. In our approach, we use a similar approach, strictly adopting the need-to-know principle, but, as described in Sect. 3, the anonymization is dynamically only on the data set resulting from a pattern, and according to the *trust* level of the users/roles. As a result, we can use the *better* anonymization transformation depending on the specific utility of each pattern, assuring an increase of both privacy and utility.

Other works focus on specific anonymization techniques for logs (see [21] for review), and on measuring the privacy risk. For example, in [30], the authors use entropy to measure privacy leakage in IDS alerts. We implemented several of the proposed anonymization techniques in our prototype, and, although based on k-anonymity, our framework can include other privacy measures by changing the risk function. More specifically, entropy based privacy metrics can be easily integrated with k-anonymity approach, as shown in [18].

Risk Based Access Control systems: Several risk and trust based access control models have been introduced in the last years. (e.g. [6,7,8,11,28]), where for each access request or permission activation, the corresponding risk is estimated and if the risk is less than a threshold (often related to trust) then the operation is permitted, otherwise it is denied. Cheng et al. [8] estimate risk and trust thresholds from the sensitivity labels of the resource and clearance level of the users in a multi-level-security system. They also consider a trust enhancement mechanism (the authors call it risk mitigation strategy in their paper) that allow users to spend *tokens* to access resources with risk higher than their trust level. The details on how this mechanism can be applied in real cases are not provided.

Chen et al. [6] introduced an abstract model which allows role activation based on a risk evaluation compared to predefined risk thresholds. Trust values are considered, and they impact (decreasing) risk calculation. If risk is too high, the model includes mitigation strategies, indicated as system obligations. The paper does not specify how to compute the risk thresholds, trust, and the structure of obligations. In a derived model [7], mitigation strategies have been explicitly defined in terms of user obligations (actions that have to be fulfilled by the user). The model also introduces the concept of *diligence score*, which measured the diligence of the user to fulfill the obligations (as a behavioral trust model), and impact the risk estimation. Another extension has been proposed [2,3], focusing on re-identification risk and anonymization is used as mitigation strategy (as in our paper).

Following the original Chen et al. [6] model, these papers consider trust as part of the risk value. We can essentially map our model to the Chen et al. [6] approach; in fact renaming the difference $R - T$ as risk in Eq. 1, and explicitly defining as a threshold the impact of risk mitigation, we obtain mostly the same model as described in [6]. However, as we discussed in [1], explicitly introducing the risk/trust comparison allows for: *i)* trust enhancement and risk mitigation strategies are clearly separated, making easier to find an optimal set of strategies to increase access, keeping risk under control, *ii)* trust thresholds are not dependent on the risk scenario, and, if we consider multiple risk factors, we can compare the overall risk with the trust. Our model addresses these issues, clearly separating trust aspects from risk.

6 Conclusions and Future work

Motivated by a strong need to improve privacy protection in security monitoring products, such as Threat Detection Systems, we proposed an access control model able to address their, complex, privacy and utility requirements. We adapted a Risk-based Access Control approach (described in [2,1]) for a threat detection solution, where anonymization is dynamically applied to reduce the privacy risk. Automatically applying specific anonymization strategies, in real-time, for each pattern, we showed how this model is able to provide a simple solution for investigating potentially harmful patterns, with a minimal privacy risk. In the cases where significantly reducing risk results in an excessive degradation of the quality of data, the model supports mechanisms of trust enhancement to access less-anonymized data. We also showed that the anonymization step does not impact the real-time performance of the systems for typical data set.

We based our analysis on real TDS, using a small sample of typical patterns. A more extensive analysis is needed to be able to implement a robust solution. In particular, the parameter setting (risk thresholds) can be complex in presence of a large number of patterns. In addition, although widely used k-anonymity has its own limitation, for example, in presence of multiple overlapping data sets, it is well known that the k-anonymity condition cannot be fulfilled (lack of composability). Other privacy models exist, such differential privacy, which could be integrated in our framework.

Acknowledgments The research leading to these results has received funding from the FP7 EU-funded project SECENTIS (FP7-PEOPLE-2012-ITN, grant no. 317387).

References

1. A. Armando, M. Bezzi, F. Di Cerbo, and N. Metoui. Balancing trust and risk in access control. In *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*, pages 660–676. Springer International Publishing, 2015.
2. A. Armando, M. Bezzi, N. Metoui, and A. Sabetta. Risk-aware information disclosure. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, pages 266–276. Springer, 2015.
3. A. Armando, M. Bezzi, N. Metoui, and A. Sabetta. Risk-based privacy-aware information disclosure. *Int. J. Secur. Softw. Eng.*, 6(2):70–89, Apr. 2015.
4. M. Bezzi. An information theoretic approach for privacy metrics. *Transactions on Data Privacy*, 3(3):199–215, 2010.
5. J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 70–78, New York, NY, USA, 2008. ACM.
6. L. Chen and J. Crampton. Risk-aware role-based access control. In C. Meadows and C. Fernandez-Gago, editors, *Security and Trust Management*, volume 7170 of *Lecture Notes in Computer Science*, pages 140–156. Springer Berlin Heidelberg, 2012.
7. L. Chen, J. Crampton, M. J. Kollingbaum, and T. J. Norman. Obligations in risk-aware access control. In N. Cuppens-Boulahia, P. Fong, J. García-Alfaro, S. Marsh, and J. Steghöfer, editors, *PST*, pages 145–152. IEEE, 2012.
8. P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *IEEE Symposium on Security and Privacy*, pages 222–230. IEEE Computer Society, 2007.
9. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Theory of privacy and anonymity. In M. Atallah and M. Blanton, editors, *Algorithms and Theory of Computation Handbook (2nd edition)*. CRC Press, 2009.
10. C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. *Trans. Data Privacy*, 6(2):161–183, Aug. 2013.
11. L. Dickens, A. Russo, P.-C. Cheng, and J. Lobo. Towards learning risk estimation functions for access control. In *In Snowbird Learning Workshop*, 2010.
12. M. Friedewald and R. J. Pohoryles. *Privacy and Security in the Digital Age: Privacy in the Age of Super-Technologies.* Routledge, 2016.
13. B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, June 2010.
14. G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *Proceedings of the 33rd international conference on Very large data bases*, pages 758–769. VLDB Endowment, 2007.

15. A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618 – 644, 2007. Emerging Issues in Collaborative Commerce.
16. M. Kaempfer. scn.sap.com/community/security/blog/2015/03/04/sap-enterprise-threat-detection-and-siem-is-this-not-the-same, 2015.
17. F. Kohlmayer, F. Prasser, C. Eckert, and K. A. Kuhn. A flexible approach to distributed data anonymization. *Journal of Biomedical Informatics*, 50:62 – 76, 2014. Special Issue on Informatics Methods in Medical Privacy.
18. A. Kounine and M. Bezzi. Assessing disclosure risk in anonymized datasets. In *Proceedings of the FloCon Workshop*, January 2009.
19. K. Lakkaraju and A. Slagell. Evaluating the utility of anonymized network traces for intrusion detection. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, SecureComm '08, pages 17:1–17:8, New York, NY, USA, 2008. ACM.
20. X. Li, F. Zhou, and X. Yang. A multi-dimensional trust evaluation model for large-scale p2p computing. *Journal of Parallel and Distributed Computing*, 71(6):837–847, 2011.
21. K. Mivule and B. Anderson. A study of usability-aware network trace anonymization. In *Science and Information Conference (SAI), 2015*, pages 1293–1304. IEEE, 2015.
22. A. Narayanan, J. Huey, and E. W. Felten. A precautionary approach to big data privacy. In *Data Protection on the Move*, pages 357–385. Springer, 2016.
23. C. of Europe. Handbook on european data protection law. Technical report, 2014.
24. C. on Strategies for Responsible Sharing of Clinical Trial Data;. *Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk.* National Academies Press (US), Washington (DC), 2015.
25. A. Oprea, Z. Li, T.-F. Yen, S. H. Chin, and S. Alrwais. Detection of early-stage enterprise infection by mining large-scale log data. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, pages 45–56. IEEE, 2015.
26. P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.
27. M. A. Scholl, K. M. Stine, J. Hash, P. Bowen, L. A. Johnson, C. D. Smith, and D. I. Steinberg. Sp 800-66 rev. 1. an introductory resource guide for implementing the health insurance portability and accountability act (hipaa) security rule. Technical report, 2008.
28. R. A. Shaikh, K. Adi, and L. Logrippo. Dynamic risk-based decision methods for access control systems. *Computers & Security*, 31(4):447–464, 2012.
29. M. Templ, B. Meindl, and A. Kowarik. Introduction to statistical disclosure control (sdc). *Project: Relative to the testing of SDC algorithms and provision of practical SDC, data analysis OG*, 2013.
30. N. Ulltveit-Moe and V. A. Oleshchuk. Measuring privacy leakage for IDS rules. *CoRR*, abs/1308.5421, 2013.
31. N. Ulltveit-Moe, V. A. Oleshchuk, and G. M. Køien. Location-aware mobile intrusion detection with enhanced privacy in a 5g context. *Wireless Personal Communications*, 57(3):317–338, 2011.
32. J. Vaidya, C. W. Clifton, and Y. M. Zhu. *Privacy preserving data mining*, volume 19. Springer Science & Business Media, 2006.
33. R. Zuech, T. M. Khoshgoftaar, and R. Wald. Intrusion detection and big heterogeneous data: A survey. *Journal of Big Data*, 2(1):1–41, 2015.