Get Your Workload in Order: Game Theoretic Prioritization of Database Auditing

Chao Yan [1], Bo Li [2], Yevgeniy Vorobeychik [1,4], Aron Laszka [3], Daniel Fabbri [1,4], Bradley Malin [1,4]

[1] *Department of Electrical Engineering and Computer Science, Vanderbilt University, TN 37240, USA*
[2] *Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, USA*
[3] *Department of Computer Science, University of Houston, TX 77004, USA*
[4] *Department of Biomedical Informatics, Vanderbilt University, TN 37203, USA*
[1,4] {chao.yan, yevgeniy.vorobeychik, daniel.fabbri, b.malin}@vanderbilt.edu
[2] crystalboli@berkeley.edu [3] alaszka@uh.edu

***Abstract*—A wide variety of mechanisms, such as alert triggers and auditing routines, have been developed to notify administrators about types of suspicious activities in the daily use of large databases of personal and sensitive information. However, such mechanisms are limited in that: 1) the volume of such alerts is often substantially greater than the capabilities of resource-constrained organizations and 2) strategic attackers may disguise their actions or carefully choose which records they touch, thus evading auditing routines. To address these problems, we introduce a novel approach to database auditing that explicitly accounts for adversarial behavior by 1) prioritizing the order in which types of alerts are investigated and 2) providing an upper bound on how much resource to allocate for auditing each alert type. We model the interaction between a database auditor and potential attackers as a Stackelberg game in which the auditor chooses an auditing policy and attackers choose which records in a database to target. We further introduce an efficient approach that combines linear programming, column generation, and heuristic search to derive an auditing policy, in the form of a mixed strategy. We assess the performance of the policy selection method using a publicly available credit card application dataset, the results of which indicate that our method produces high-quality database audit policies, significantly outperforming baselines that are not based in a game theoretic framing.**

I. Introduction

Modern computing and storage technology has made it possible to create *ad hoc* database systems with the ability to collect, store, and process extremely detailed information about the daily activities of individuals [1]. These systems hold great value for society, but accordingly face challenges to security and eventually, personal privacy. Their sensitive property attracts malicious attackers who can gain value through various ways, such as stealing sensitive information, commandeering computational resources, committing financial fraud, and simply shutting the system down [2]. While complex access control systems have been developed for database management, it has been recognized that in practice no database systems will be impervious to attack [3]. As such, prospective technical protections need to be complemented by retrospective auditing mechanisms [4]. Though never preventing attacks in their own right, auditing allows for the discovery of breaches that can be followed up on before they escalate to full blown exploits by adversaries originating from beyond, as well as within, an organization.

In general, auditing relies on the performance of a *threat detection and misuse tracking* (TDMT) module, which raises real-time alerts based on the actions committed to a system for further investigation. Practically, the alert types are specifically predefined by the administrator officials in *ad hoc* applications. Through deploying TDMTs, however, security and privacy have not been sufficiently guaranteed, the main reason of which, lies on their nature of generating a large number of alerts, whereas the number of actual violations tends to be quite small. Therefore, in lieu of an efficient audit functionality in the database systems, TDMTs are not optimized for detecting suspicious behavior.

Given the overwhelming volume of alerts in comparison to available auditing resource and the need to catch attackers, the core query function invoked by an administrator must consider resource constraints. And, given such constraints, we must determine which alerts should be recommended for investigation. To solve this problem, we introduce a game theoretic model, in which an auditor chooses a randomized auditing policy, while potential violators choose their victims or to refrain from malicious behavior after observing the auditing policy. Specifically, our model restricts the space of audit policies to consider two dimensions: 1) how to prioritize alert categories and 2) how much resource to allocate to each category. We propose a series of algorithmic methods for solving it. In addition, we develop a novel search method for computing the amount of investigation resource for each category. We perform an evaluation with a real dataset pertaining to credit card eligibility decisions, the results of which demonstrate the effectiveness of our approach over various alternative techniques.

II. Game Theoretic Model of Alert Prioritization

By defining alert types, each suspicious event can be marked into the corresponding audit bin. A crucial consideration is how to *prioritize* alerts, choosing a subset from a vast pool of possibilities that can be audited given a specified auditing budget. The problem is complicated by the fact that intelligent

adversaries—that is, would-be violators of organizational access policies—would react to an auditing policy by changing their behavior to balance the gains from violations, and the likelihood, and consequences, of detection. We describe a game model between an auditor and multiple attackers.

A. System Model

Let $\mathbf{E}$ be the set of potential adversaries, some of whom could be violators of privacy policies, and $\mathbf{V}$ be the set of potential victims. We define events, as well as attacks, by a tuple $\langle e, v \rangle$. A subset of these events will trigger alerts. Let T be the set of alert types assigned to different kinds of suspicious behavior. We assume each event $\langle e, v \rangle$ maps to at most one alert type $t \in T$ (with probability P_{ev}^t). Typically, both categorization of alerts and corresponding mapping between events and types is given (e.g. through predefined rules). Let C_t be the cost (e.g., time) of auditing a single alert of type t and let B be the total budget allocated for auditing. Normal events resulting in alerts arrive based on a distribution reflecting a typical workflow of the organization. We assume this distribution is known, represented by $F_t(n)$, which is the probability that at most n alerts of type t are generated. If we make the reasonable assumption that attacks are rare events and that the alert logs are tamper-proof by applying certain technique, then this distribution can be obtained from historical alert logs.[1]

B. Game Model

We model the interaction between the auditor and potential violators as a Stackelberg game, in which the auditor chooses a possibly randomized auditing policy, which is observed by the prospective violators completely, who in response choose the nature of the attack. Both decisions are made *before the alerts produced through normal workflow are generated*.

In general, a specific pure strategy of the defender (auditor) is a mapping from an arbitrary realization of alert counts of all types to a subset of alerts that are to be inspected, abiding by a constraint on the total amount of budget B allocated for auditing alerts. We let pure strategies involve an ordering $\boldsymbol{o} = (o_1, o_2, \ldots, o_{|T|})$ $\quad (\forall i, j \in \mathbb{Z}^+$ and $i, j \in [1, |T|]$, if $i \neq j$, then $o_i \neq o_j)$ over alert types, where the subscript indicates the position in the ordering, and a vector of thresholds $\mathbf{b} = (b_1, \ldots, b_{|T|})$, with b_t being the maximum budget available for auditing alerts in category t. Let $\mathbf{O}$ be the set of feasible orderings. We interpret a threshold b_t as the maximum budget allocated to t; thus, the most alerts of type t that can be inspected is $\lfloor b_t / C_t \rfloor$. The auditor is allowed to choose a randomized policy over alert orderings, with $p_{\boldsymbol{o}}$ being the probability that ordering $\boldsymbol{o}$ over alert types is chosen, whereas the thresholds $\mathbf{b}$ are deterministic and independent of the chosen alert priorities.

Each adversary $e \in \mathbf{E}$ may target any potential victim $v \in \mathbf{V}$. The adversary is assumed to target at most one victim. In addition, we assume that any given potential adversary is actually unlikely to consider attacking. We formalize it by introducing a probability p_e that an attack by e is considered at all (i.e., e does not even consider attacking with probability $1 - p_e$).

Suppose we fix a prioritization $\boldsymbol{o}$ and thresholds $\mathbf{b}$. Let $o(t)$ be the position of alert type t in $\boldsymbol{o}$ and o_i be the alert type in position i in the order. Let $B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})$ be the budget remaining to inspect alerts of type t if the order is $\boldsymbol{o}$, the defender uses alert type thresholds $\mathbf{b}$, and the vector of realizations of benign alert type counts is $\mathbf{Z} = \{Z_1, \ldots, Z_{|T|}\}$. Then we have

$$B_t(\boldsymbol{o},\mathbf{b},\mathbf{Z}) = \max\left\{\left\lfloor\left(B - \sum_{i=1}^{o(t)-1} \min\left\{b_{o_i}, Z_{o_i} C_{o_i}\right\}\right)/C_t\right\rfloor, 0\right\}.$$

If the total budget that is eaten by inspecting alerts prior to t is larger than B, $B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})$ returns 0, and no alerts of type t will be inspected. Next, we can compute the number of alerts of type t that are audited as

$$n_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z}) = \min\left\{B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z}), \lfloor b_t / C_t \rfloor, Z_t\right\}.$$

Then, the probability that an alert of type t generated through an attack is detected is approximately

$$P_{al}(\boldsymbol{o}, \mathbf{b}, t) \approx \mathbb{E}_{\mathbf{Z}}\left[\frac{n_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})}{Z_t}\right]. \tag{1}$$

We can further approximate this probability by sampling from the joint distribution over alert type counts $\mathbf{Z}$.

The adversary does not directly choose alert types, but rather the victim. Thus, the probability of detecting an attack $\langle e, v \rangle$ under audit order $\boldsymbol{o}$ and audit thresholds $\mathbf{b}$ is then

$$P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle) = \sum_t P_{ev}^t P_{al}(\boldsymbol{o}, \mathbf{b}, t). \tag{2}$$

Let $M(\langle e, v \rangle)$ denote the penalty of the adversary when captured by the auditor, $R(\langle e, v \rangle)$ denote the benefit if the adversary is not audited, and $K(\langle e, v \rangle)$ the cost of an attack. The utility of the adversary e is then

$$\begin{aligned} U_a(\boldsymbol{o},\mathbf{b}, \langle e, v \rangle) &= P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle) \cdot M(\langle e, v \rangle) \\ &+ (1 - P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle)) \cdot R(\langle e, v \rangle) - K(\langle e, v \rangle). \end{aligned} \tag{3}$$

By assuming the game is zero-sum, the auditor's goal is to find a randomized strategy $p_{\boldsymbol{o}}$ and type-specific thresholds $\mathbf{b}$ to minimize the expected utility of the adversary:

$$\min_{p_{\boldsymbol{o}}, \mathbf{b}} \sum_{e \in \mathbf{E}} \sum_{\boldsymbol{o} \in \boldsymbol{O}} p_{\boldsymbol{o}} \max_v U_a(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle). \tag{4}$$

We call this optimization challenge the *optimal auditing problem (OAP)*. The optimal auditing policy can be computed using the following mathematical program, which directly extends the standard linear programming formulation for computing mixed-strategy Nash equilibria:

$$\begin{aligned} &\min_{\mathbf{b}, p_{\boldsymbol{o}}, \mathbf{u}} \ \textstyle\sum_{e \in \mathbf{E}} p_e u_e \\ &s.t. \quad \forall \langle e, v \rangle, \ \ u_e \geq \textstyle\sum_{\boldsymbol{o} \in \boldsymbol{O}} p_{\boldsymbol{o}} U_a(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle) \\ &\quad \textstyle\sum_{\boldsymbol{o} \in \boldsymbol{O}} p_{\boldsymbol{o}} = 1, \\ &\quad \forall \boldsymbol{o} \in \boldsymbol{O}, \ \ 0 \leq p_{\boldsymbol{o}} \leq 1. \end{aligned} \tag{5}$$

[1] The probability that adversaries manipulate the distribution in the sensitive practices to fool the audit model is almost zero.

Indeed, if we fix the decision variables $\mathbf{b}$, the formulation becomes a linear program. Nevertheless, since the set of all possible alert prioritizations is exponential, even this linear program has exponentially many variables. Furthermore, introducing decision variables $\mathbf{b}$ makes it non-linear and non-convex.

III. Solving the Alert Prioritization Game

The two main challenges for solving the OAP are: 1) an exponential set of possible orderings of alert types, and 2) a combinatorial space of possible choices for the threshold vectors $\mathbf{b}$. In this section, we develop a column generation approach for the linear program induced when we fix a threshold vector $\mathbf{b}$. We then propose a search algorithm to compute the auditing thresholds.[2]

A. Column Generation Greedy Search

Since the number of constraints is small compared with the exponential number of variables, only a limited number of variables will be non-zero. Borrowing the basic idea of column generation, we propose a method we refer to as Column Generation Greedy Search (CGGS), in which we iteratively solve a linear program with a small subset of variables, and then add new variables with a negative reduced cost.

Specifically, we begin with a small subset of alert prioritizations $\boldsymbol{Q} \subseteq \boldsymbol{O}$ and solve the linear program induced after fixing $\mathbf{b}$ in Equation 5, restricted to columns in $\boldsymbol{Q}$. Next, we check if there exists a column (ordering over types) that improves upon the current best solution. By minimizing the reduced costs, we generate one new column in each iteration and add it to the subset of columns $\boldsymbol{Q}$ in the master problem. This process is repeated until we can prove that the minimum reduced cost is non-negative. At this point, we have solved the original (unrestricted) linear program in a suboptimal manner.

B. Iterative Shrink Heuristic Method

We now develop a heuristic procedure, which we call the Iterative Shrink Heuristic Method (ISHM), to find suboptimal alert type thresholds. First, it should be recognized that $\sum_t b_t \geq B$; otherwise, it would clearly waste auditing resources. Though no explicit upper bound on the thresholds, given the distribution of the number of alerts Z_t for an alert type t, we can obtain an approximate upper bound on b_t, where $F_t(b_t/C_t) \approx 1$. Consequently, searching for a good solution can begin with a vector of audit thresholds, such that for each b_t, $F_t(b_t/C_t) \approx 1$. Leveraging this intuition, we iteratively shrink the values of a good subset of audit thresholds according to a certain step size ϵ[3].

In each atomic searching action, ISHM first makes a subset of thresholds b_t strategically shrink. Next, it checks if this results in an improved solution. We introduce a variable l_h, which indicates the level (or the size) of the given subset of $\mathbf{b}$, and $\epsilon \in (0, 1)$, which controls the step size.

[2]The pseudocode for the two algorithms can be find at XXXXX.

[3]"Good" in this context means that shrinking thresholds within the subset improves the value of the objective function.

TABLE I: Description of the defined alert types.

ID	Alert type Description	Mean	Std
1	No checking account, *Any purpose*	370.04	15.81
2	Checking < 0, *New car, Education*	82.42	7.87
3	Checking > 0, Unskilled, *Education*	5.13	2.08
4	Checking > 0, Unskilled, *Appliance*	28.21	5.25
5	Checking > 0, Critical account, *Business*	8.31	2.96

By assigning $l_h = 1$, we begin with shrinking each of the audit thresholds. If the best value for the objective function in the candidate subsets at $l_h = 1$ after shrinking shows an improvement, then the shrink is accepted and the shrinking coefficient is made smaller. When no coefficient leads to improvement, we increase l_h by one, which induces tests of threshold combinations at the same shrinking ratio. Once an improvement occurs, the search course resets based on the current $\mathbf{b}$. The search terminates once $l_h > |T|$.

IV. Model Evaluation

A. Data Overview

The adopted dataset for model evaluation is the Statlog (German Credit Data) dataset available from the UCI Machine Learning Repository. It contains 1000 credit card applications with 20 attributes describing the status of the applicants pertaining to their credit risk. Before issuing a credit card, banks would determine if it could be fraudulent. In face of a large number of applications, it requires retrospective audits to determine whether specific ones should be canceled. Leveraging the provided features, we define five alert types, which are triggered by the specific combinations of attribute values and the purpose of application, as depicted by Table I. The eight selected purposes of application are the "victims" in our audit model. In the description field, italicized words represent the purpose of the application, while the other words represent feature values.

We used the five alert types to label applications, excluding any that fail to receive a label. Among these, we randomly selected 100 applicants who may choose to "attack" one of the eight purposes of applications, for a total of 800 possible events.

B. Comparison with Baseline Alternatives

The performance of the proposed audit model was investigated by comparing with several natural alternative audit strategies as baselines. The first alternative, *Audit with random orders of alert types*, is to randomize the audit order over alert types. Though random, this strategy mimics the reality of random reporting. In this case, we adopt the thresholds out of the proposed model with $\epsilon = 0.1$ to investigate the performance. The second alternative, *Audit with random thresholds*, is to randomize the audit thresholds. For this policy, we assume that 1) the auditor's choice satisfies $\sum_i b_i \geq B$ and 2) the auditor has the ability to find the optimal audit order after deciding upon the thresholds. The third alternative, *Audit based on benefit*, is a naive greedy audit strategy, where the auditor prioritizes alert types according to their utility loss. In

this case, the auditor investigates as many alerts of a certain type as possible before moving on to the next type in the order.

The following performance comparisons are assessed over a broad range of auditing budgets. For our model, we present the values of the objective function with three different instances of the step size ϵ in ISHM: $[0.1, 0.2, 0.3]$. Figures 1 summarize the performance of the proposed audit model and three alternative audit strategies for the dataset. As expected, as the budget increases, the auditor sustains a decreasing average loss. It can be seen that the proposed audit model significantly outperforms the alternative baselines. Specifically, as the auditing budget increases, the auditor's loss trends towards, and becomes, 0 in our approach. This means that the attackers are completely deterred. For the alternatives *Audit with random thresholds* outperforms other strategies. And, the strategy that greedily audits alert types (in order of loss) tends to perform quite poorly.

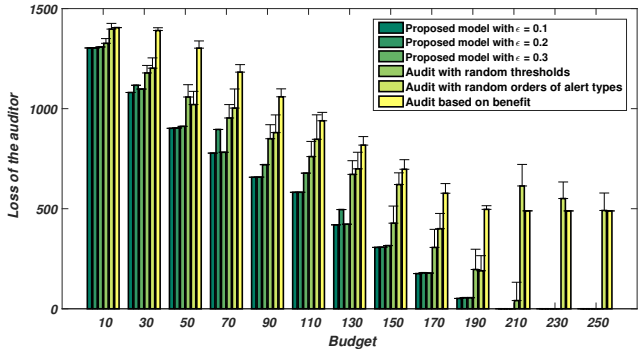

Fig. 1: Loss of the auditor in the proposed and alternatives audit model in the dataset.

V. Related Work

The development of computational methods for raising and subsequently managing alerts in database systems is an active area of research. Generally speaking, there are two main categories by which alerts are generated in a TDMT: 1) machine learning methods [5]–[7], and 2) rule-based approaches [8]–[11]. While these methods trigger alerts for investigators, they result in a significant number of false positives and they fail to consider the situation that smart attackers may circumvent the prioritization and aggregation mechanisms. Naturally, game-theoretical approaches provide some novel points of view [12], [13]. Laszka *et al.* proposed a framework for alert prioritization, which adopted an exhaustive auditing strategy across alert types of a given order [14], which is limited in practice. Recently, the problem of assigning alerts to security analysts has been introduced [15], with a follow-up effort casting it within a game theoretic framework [16]; however, it is assumed that the number of alerts is fixed, which is not the case in practice.

VI. Discussion and Conclusions

TDMTs are usually deployed in database systems to address a variety of attacks; however, an overwhelming alert volume is far beyond the capability of auditors with limited resources. Our research illustrates that policy compliance auditing can be improved by prioritizing which alerts to focus on via a game theoretic framework, allowing auditing policies to make best use of limited auditing resources while simultaneously accounting for strategic behavior of potential policy violators. As such, the proposed model and the effective heuristics we offer in this study fill a major gap in the field.

VII. Acknowledgement

This work was supported, in part by grant R01LM10207 from the National Institutes of Health, grant CNS-1526014, CNS-1640624, IIS-1649972 and IIS-1526860 from the National Science Foundation, grant N00014-15-1-2621 from the Office of Naval Research and grant W911NF-16-1-0069 from the Army Research Office.

References

[1] A. McAfee, E. Brynjolfsson, T. H. Davenport *et al.*, "Big data: the management revolution," *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012.

[2] L. Ablon, M. C. Libicki, and A. A. Golay, *Markets for cybercrime tools and stolen data: Hackers' bazaar*. Rand Corporation, 2014.

[3] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: attack and defense modeling," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 4, pp. 853–865, 2010.

[4] H. D. Kuna, R. García-Martinez, and F. R. Villatoro, "Outlier detection in audit logs for application systems," *Information Systems*, vol. 44, pp. 22–33, 2014.

[5] A. A. Boxwala, J. Kim, J. M. Grillo, and L. Ohno-Machado, "Using statistical and machine learning to help institutions detect suspicious access to electronic health records," *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 498–505, 2011.

[6] Y. Chen, S. Nyemba, and B. Malin, "Detecting anomalous insiders in collaborative information systems," *IEEE Transactions on Dependable and Secure Computing*, no. 99, pp. 1–1, 2012.

[7] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.

[8] C. Gunter, D. Liebovitz, and B. Malin, "Experience-based access management," *IEEE Security and Privacy Magazine*, vol. 9, pp. 48–55, 2011.

[9] D. Fabbri and K. LeFevre, "Explaining accesses to electronic medical records using diagnosis information," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 52–60, 2013.

[10] D. Fabbri, R. Ramamurthy, and R. Kaushik, "Select triggers for data auditing," in *Proceedings of the IEEE ICDE*, 2013, pp. 1141–1152.

[11] D. Fabbri and K. LeFevre, "Explanation-based auditing," *Proceedings of the VLDB Endowment*, vol. 5, no. 1, pp. 1–12, 2011.

[12] J. Blocki, N. Christin, A. Datta, A. D. Procaccia, and A. Sinha, "Audit games," *arXiv preprint arXiv:1303.0356*, 2013.

[13] J. Blocki, N. Christin, A. Datta, A. Procaccia, and A. Sinha, "Audit games with multiple defender resources," *arXiv preprint arXiv:1409.4503*, 2014.

[14] A. Laszka, Y. Vorobeychik, D. Fabbri, C. Yan, and B. Malin., "A game-theoretic approach for alert prioritization," in *AAAI Workshop on Artificial Intelligence for Cyber Security*, 2017.

[15] R. Ganesan, S. Jajodia, A. Shah, and H. Cam, "Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 1, p. 4, 2016.

[16] A. Schlenker, M. Tambe, C. Kiekintveld, H. Xu, M. Guirguis, A. Sinha, S. Sonya, N. Dunstatter, and D. Balderas, "Don't bury your head in warnings: a game-theoretic approach for intelligent allocation of cybersecurity alerts," in *Proceedings of the 26th IJCAI*, 2017.