To Warn or Not to Warn: Online Signaling in Audit Games

Chao Yan[1] , **Haifeng Xu**[2] , **Yevgeniy Vorobeychik**[3] , **Bo Li**[4] , **Daniel Fabbri**[1,5] and **Bradley A. Malin**[1,5]

[1]Dept. of EECS, Vanderbilt University
[2]Dept. of Computer Science, University of Virginia
[3]Dept. of Computer Science and Engineering, Washington University at St. Louis
[4]Dept. of Computer Science, University of Illinois at Urbana-Champaign
[5]Dept. of Biomedical Informatics, Vanderbilt University Medical Center
chao.yan@vanderbilt.edu, hx4ad@virginia.edu, yvorobeychik@wustl.edu, lbo@illinois.edu,
{daniel.fabbri, b.malin}@vumc.org

Abstract

In health care organizations, a patient's privacy is threatened by the misuse of their electronic health record (EHR). To monitor privacy intrusions, logging systems are often deployed to trigger alerts whenever a suspicious access is detected. However, such mechanisms are insufficient in the face of small budgets, strategic attackers, and large false positive rates. In an attempt to resolve these problems, EHR systems are increasingly incorporating signaling, so that whenever a suspicious access request occurs, the system can, in real time, warn the user that the access may be audited. This gives rise to an online problem in which one needs to determine 1) whether a warning should be triggered and 2) the likelihood that the data request will be audited later. In this paper, we formalize this auditing problem as a Signaling Audit Game (SAG). A series of experiments with 10 million real access events (containing over 26K alerts) from Vanderbilt University Medical Center (VUMC) demonstrate that a strategic presentation of warnings adds value in that SAGs realize significantly higher utility for the auditor than systems without signaling.

1 Introduction

To provide medical services, healthcare organizations (HCO) collect, store and process personal health data in electronic health records (EHR) systems. Due to the potential value of such data, EHR systems face non-trivial challenges to assuring patient privacy. One would expect that such sensitive information would be provisioned to health care workers on a need to know basis only; however, the complexity of healthcare makes it challenging to know who specifically needs access to which information and when. As a consequence, one the greatest risks to privacy are insiders, that is, authenticated users of EHR systems, who may violate policy and intrude upon the privacy of certain patients by accessing data they were not supposed to use [Fabbri *et al.*, 2013]. Thus, to defend against such attacks, EHR systems are often armed with an alerting capability to detect and notify about potential risks incurred during daily use [Puppala *et al.*, 2016]. This entails the logging of access events, which can be thought of as a collection of rules, each of which defines a semantic type of a potentially malicious situation [Mazzawi *et al.*, 2017]. The notification about potential misuse is provided to administrators who perform retrospective audit investigations [Kuna *et al.*, 2014; Blocki *et al.*, 2012].

However, there are hurdles to instituting robust auditing schemes because 1) the volume of triggered alerts is typically far greater than the auditing capacity of HCOs [Laszka *et al.*, 2017], 2) the majority of triggered alerts correspond to false positives, 3) to mitigate the risk of being caught, attackers prefer to act strategically, and 4) in the retrospective audit setting, attacks are not discovered until they are investigated.

In essence, this is a resource allocation problem in an adversarial environment for which the Stackelberg security game (SSG) is a natural choice to apply for modeling purposes [Do *et al.*, 2017; Sinha *et al.*, 2018]. In particular, the audit game is a variation of the SSG designed to discover an efficient audit strategy [Blocki *et al.*, 2013; Blocki *et al.*, 2015; Yan *et al.*, 2018; Yan *et al.*, 2019]. With respect to strategic auditing, existing research has focused on deriving a defense strategy by solving, or approximating, the Strong Stackelberg Equilibrium (SSE). Unfortunately, it was recently shown that merely applying the SSE strategy may have limited efficacy in some security settings [Xu *et al.*, 2015]. This can be addressed by strategically revealing information to the attacker, a mechanism referred to as *signaling* [Dughmi and Xu, 2016]). In this setting, the goal is to set up a *signaling scheme* to reveal noisy information to the attacker and, by doing so, influence the attacker's decision to favor the defender.

In this paper, we introduce the notion of a Signaling Audit Game (SAG), which applies signaling to alert and auditing. When an alert is triggered by a suspicious access request, the system can, in real time, send a warning to the requestor. At this point, the attacker has an opportunity to re-evaluate his/her utility and make a decision about whether or not to continue with an attack. In contrast to previous models, which are all computed offline, the SAG optimizes both the warning strategy and the audit decision in real time for each alert. To illustrate

This study has been published in 36th IEEE International Conference on Data Engineering (ICDE20). https://ieeexplore.ieee.org/abstract/document/9101660

the performance of the SAG, we evaluate the expected utility of the auditor with a dataset of over 10 million real VUMC EHR accesses and predefined alert types. The results indicate that the SAG consistently outperforms state-of-the-art game theoretic alternatives that lack signaling by achieving higher overall utility while inducing nominal increases in computational burden.

2 Online Signaling in Audit Games

2.1 Motivating Domain

EHR users, such as physicians and nurses, need to access patients' EHRs when providing healthcare services. The routine workflow can be summarized as three steps: 1) a user initiates a search for a patient's EHR by name and date of birth, then the system returns a list of patients (based on a fuzzy matching) along with their demographic information, 2) from the list, this user requests access to a patient's record, and 3) the system returns the requested record. Due to the complex, dynamic and time-sensitive nature of healthcare, HCOs typically grant employees broad access privileges, which creates an opportunity for malicious insiders to exploit patients' EHRs [Gunter *et al.*, 2011]. To deter malicious access, detection tools are commonly deployed to trigger alerts for suspicious events. Alerts are often marked with predefined types of potential violations which help streamline inspection.

2.2 Signaling Audit Games

An SAG is played between an *auditor* and an *attacker* within a predefined audit cycle (e.g., one day). This game is sequential such that alerts arrive one at a time. For each alert, the auditor needs to make two decisions in *real time*: first, which signal to send (e.g., to warn the user/attacker or not), and second, whether to audit the alert. Formally, let X_c^τ denote the event that alert τ will be audited, and X_u^τ denote that it is not audited. We further let ξ_1^τ denote the event that a *warning signal* is sent for alert τ, while ξ_0^τ denotes the event that no warning is sent (i.e. a "silent signal"). The warning ξ_1^τ is delivered privately through a dialog box on the requestor's screen, which might communicate "*Your access may be investigated. Would you like to proceed?*". $X_c^\tau, X_u^\tau, \xi_1^\tau, \xi_0^\tau$ are random variables whose probabilities are to be designated.

We assume that there is a finite set of alert types T and, for each $t \in T$, all alerts are considered equivalent for our purposes (i.e., result in the same damages to the system). The auditor has an auditing budget B that limits the number of alerts that can be audited at the end of the cycle. For each alert type t, let V^t denote the cost (or time needed) to audit an alert of type t. Thus, if θ^t is the probability of auditing alerts of type t and d^t is the number of such alerts, the budget constraint implies that $\sum_t \theta^t \cdot V^t d^t \leq B$.

Since the setting is online, an optimal policy for the auditor must consider all possible histories of alerts and the correlation between alerts. Given that this is impractical, we simplify the scheme so that 1) each alert is viewed independently of alerts that precede it and 2) future alerts are considered with respect to their average relative frequency. Specifically, we assume that each attack effectively selects an alert type t, but do not need to consider the timing of attacks. Rather, we treat each alert as potentially adversarial. This implicitly assumes that an attack triggers a single alert, as we can define alert types that capture all realistic multi-alert combinations.

For convenience, we refer to the alert corresponding to an attack as the *victim alert*. If the auditor fails to audit a victim alert of type t, the auditor and the attacker will receive utility $U_{d,u}^t$ and $U_{a,u}^t$, respectively (subscript d denotes defender, i.e. auditor, and a denotes attacker). On the other hand, if the auditor audits a victim alert of type t, the auditor and the attacker will receive utility $U_{d,c}^t$ and $U_{a,c}^t$, respectively. Naturally, we assume $U_{a,c}^t < 0 < U_{a,u}^t$ and $U_{d,c}^t \geq 0 > U_{d,u}^t$.

Figure 1 demonstrates the key interactions of both players along the timeline. Each yellow block within the audit cycle represents a triggered alert and the corresponding interactions with it. The auditor continues to update the real time probability of auditing any alert (may or may not be triggered) with respect to the alert type and the time point τ. In other words, the auditor commits in real time to the auditing and signaling strategy. In this case, the auditor always moves first.

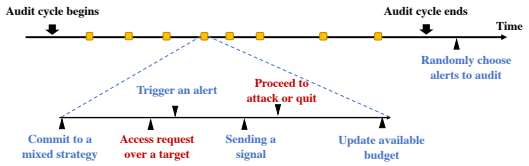


Figure 1: The interactions between auditor (*blue*) and attacker (*red*).

A *warning signaling scheme*, captured by the joint probability distribution of signaling and auditing, can be fully specified through four variables for each τ:

$$\begin{aligned} &\mathbf{P}(\xi_1^\tau, X_c^\tau) = p_1^\tau, \quad \mathbf{P}(\xi_1^\tau, X_u^\tau) = q_1^\tau, \\ &\mathbf{P}(\xi_0^\tau, X_c^\tau) = p_0^\tau, \quad \mathbf{P}(\xi_0^\tau, X_u^\tau) = q_0^\tau. \end{aligned} \tag{1}$$

Upon receiving the signal, the attacker reacts as follows:

- After ξ_1^τ: the system presents two choices to the attacker: *"Proceed"* to access the requested record or quit.
- After ξ_0^τ: the attacker automatically *proceeds* to access the requested record.

For convenience, when possible we omit the superscript τ.

Figure 2 illustrates the temporal sequence of decisions in the SAG. Each edge in the figure is marked with its corresponding joint probability of a sequence of decisions up to and including that edge. Note that the two gray nodes are not extended because they do not lead to any subsequent event.[1] Further,

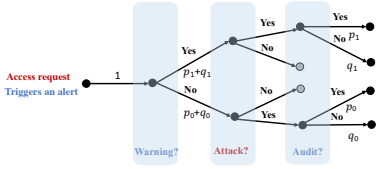


Figure 2: The decision tree of the auditor and an arbitrary user, the actions for which are shown in *blue* and *red*, respectively.

observe that, $p_1 + q_1 + p_0 + q_0 = 1$, and the overall probability of auditing this alert is $\mathbf{P}(X_c) = \mathbf{P}(X_c, \xi_1) + \mathbf{P}(X_c, \xi_0) =$

[1]The upper gray node corresponds to the case when an access request is abandoned. The lower one represents an impossible case because the user automatically gets the requested record.

$p_1 + p_0$. Conditional on the warning signal ξ_1, the probability of auditing this alert is thus $\mathbf{P}(X_c|\xi_1) = p_1/(p_1 + q_1)$.

Since the auditor has a fixed auditing budget, she will need to update the remaining budget after determining the signal-conditional audit probability for the current alert. We use B_τ to denote the remaining budget *before* receiving alert τ. Let t denote the type of alert τ and $\tau + 1$ denote the next alert. After the signaling scheme for τ is executed, the auditor then updates B_τ for the use of the next alert $\tau + 1$ as follows:

- If ξ_1^τ is sampled: $B_{\tau+1} = B_\tau - p_1^\tau/(p_1^\tau + q_1^\tau) \cdot V^t$.
- If ξ_0^τ is sampled: $B_{\tau+1} = B_\tau - p_0^\tau/(p_0^\tau + q_0^\tau) \cdot V^t$.

Additionally, we always ensure that $B_\tau \geq 0$. The key challenge is to compute the optimal $p_1^\tau, q_1^\tau, p_0^\tau, q_0^\tau$ for each alert τ *online* by accounting for the remaining budget and the estimate number of future alerts. This needs to be performed to ensure that the auditor does not spend the budget at a rate that is excessively fast or slow. The SAG can be viewed as a variation on the Stackelberg game, where it includes signaling and makes decisions about auditing *online* upon the arrival of each alert. The premise behind our solution is therefore a Strong Stackelberg equilibrium of the SAG, in which the auditor commits to a randomized joint signaling and auditing decision, and the associated probability distribution is observed by the attacker, who then decides first upon the alert type to use, and subsequently whether to proceed after a warning.

3 Optimizing SAGs

Now, we design an algorithm for solving SAGs. Here we fix the alert τ to a particular type t and, thus, the superscript will, at times, be omitted for notational convenience.

From the perspective of the attacker, whether to *proceed* or *quit* after receiving a warning signal depends on his conditional expected utility:

$$\mathbb{E}_a^t(util|\xi_1) = \frac{p_1^t}{p_1^t + q_1^t} \cdot U_{a,c}^t + \frac{q_1^t}{p_1^t + q_1^t} \cdot U_{a,u}^t.$$

We impose the constraint $\mathbb{E}_a^t(util|\xi_1) \leq 0$ such that the attacker's best response to ξ_1 is to quit, in which case both players will receive 0 utility. We do not enforce constraints for ξ_0 because the potential attacker does not have any option but to proceed. In this case, the expected utility of the auditor is

$$\mathbb{E}_d^t(util|\xi_0) = \frac{p_0^t}{p_0^t + q_0^t} \cdot U_{d,c}^t + \frac{q_0^t}{p_0^t + q_0^t} \cdot U_{d,u}^t.$$

Overall, the expected utility for the attacker is

$$\mathbb{E}_a^t(util) = (p_0^t + q_0^t) \cdot \mathbb{E}_a^t(util|\xi_0) = p_0^t \cdot U_{a,c}^t + q_0^t \cdot U_{a,u}^t.$$

Accordingly, the auditor's expected utility is

$$\mathbb{E}_d^t(util) = (p_0^t + q_0^t) \cdot \mathbb{E}_d^t(util|\xi_0) = p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t.$$

However, a side effect is that, the warnings sent by the auditor may pose an additional utility loss to the auditor in practice, which we call *usability cost*. This is because when normal users request access to sensitive data and receive a warning message, they may walk away by choosing quit instead of "Proceed", which induces a loss in operational efficiency for the organization. For each type t', we set this loss to be proportional to the product of the probability of sending warnings $p_1^{t'} + q_1^{t'}$, the probability of being deterred $P^{t'}$ and the expectation of the number of future false positive alerts to the end of the current audit cycle $E_\tau^{t'}$. The loss incurred for each quit by a normal user is set to be $C_{t'}(< 0)$. Then, the expected utility of the auditor can be updated as $p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t + \sum_{t'=1}^{|T|}(p_1^{t'} + q_1^{t'}) \cdot P^{t'} \cdot E_\tau^{t'} \cdot C_{t'}$.

The optimal signaling scheme (or, more concretely, joint signaling and audit probabilities) can be computed through the following set of LPs:

$$\begin{aligned}
\max_{\boldsymbol{p_0},\boldsymbol{p_1},\boldsymbol{q_0},\boldsymbol{q_1},\boldsymbol{B}_\tau} \quad & p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t + \sum_{t'=1}^{|T|}(p_1^{t'} + q_1^{t'}) \cdot P^{t'} \cdot E_\tau^{t'} \cdot C_{t'} \\
s.t. \quad & \forall t', \quad p_0^t \cdot U_{a,c}^t + q_0^t \cdot U_{a,u}^t \geq p_0^{t'} \cdot U_{a,c}^{t'} + q_0^{t'} \cdot U_{a,u}^{t'}, \\
& \forall t', \quad p_1^{t'} \cdot U_{a,c}^{t'} + q_1^{t'} \cdot U_{a,u}^{t'} \leq 0, \\
& \forall t', \quad p_1^{t'} + p_0^{t'} = \mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}} \left(\frac{B_\tau^{t'}}{V^{t'} d_\tau^{t'}} \right), \\
& \forall t', \quad p_1^{t'} + p_0^{t'} + q_1^{t'} + q_0^{t'} = 1, \\
& \sum_{t' \in \{1,\ldots,|T|\}} B_\tau^{t'} \leq B_\tau, \\
& \forall t', \quad B_\tau^{t'} \in [0, B_\tau], \quad p_0^{t'}, q_0^{t'}, p_1^{t'}, q_1^{t'} \in [0, 1],
\end{aligned} \tag{2}$$

where we assume type t is the best one for the attacker to potentially exploit, and $\boldsymbol{B}_\tau = \{B_\tau^t\}$ for all t. Note that, in the objective function, the incurred additional loss is an accumulated value that considers the amount of time remaining in the period for the current audit cycle. The likelihood of sending warning signal in the current time point is a real time estimation of future warnings. Due to the fact that attacks are extremely rare in practice in comparison to the magnitude of alerts, in solving LP (2) we use the expected number of future alerts $\mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}}(d_\tau^{t'})$ to approximate $E_\tau^{t'}$. As a result, $\mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}}(d_\tau^{t'})$ can then be estimated from historical data collected in previous audit cycles. Our goal is thus to find the optimal signaling scheme for all types, and simultaneously, the best budget allocation strategy. We use $\boldsymbol{p_0}$, $\boldsymbol{p_1}$, $\boldsymbol{q_0}$ and $\boldsymbol{q_1}$ to denote the warning signaling scheme for all types, namely, the set $\{p_0^{t'}|\forall t'\}$, $\{p_1^{t'}|\forall t'\}$, $\{q_0^{t'}|\forall t'\}$ and $\{q_1^{t'}|\forall t'\}$, respectively.

The first constraint in LP (2) ensures that attacking type t is the best response strategy for the attacker. The second constraint indicates that the attacker, when receiving a warning signal, will quit attacking any type. A difference on the constraint of budget allocation between SAG and SSG is that we leave out a constant $\mathcal{B}$ from the available budgets for purpose of auditing, at the end of the audit cycle, a special attacking strategy of an attacker, who keeps requesting sensitive data and quitting until receiving no warning signal. We refer to the optimal solution among the $|T|$ instances of LP (2) as the *Online Stackelberg Signaling Policy (OSSP)*. In particular, we use $\boldsymbol{\theta}_{ossp}$ to denote the vector of coverage probability at OSSP.

A set of important theoretic features of OSSP are presented in the full version of this study [Yan *et al.*, 2020].

4 Model Evaluation

4.1 Dataset

To perform a meaningful evaluation, we assessed the approach with a dataset of EHR access logs from the Vanderbilt University Medical Center (VUMC). The data covers 56 continuous

Table 1: The advantages of OSSP over online SSE in terms of the mean (and the standard deviation) of the differences in the auditor's expected utility (15 testing days).

B	$C_t = -1$		$C_t = -5$		$C_t = -10$	
	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$
30	60.87 ± 28.31 15.99%	47.01 ± 32.17 12.45%	40.43 ± 23.95 10.59%	29.89 ± 28.77 7.92%	26.91 ± 25.77 7.06%	10.94 ± 24.93 2.90%
50	165.83 ± 24.49 47.26%	147.51 ± 27.74 42.65%	143.19 ± 33.98 40.87%	117.52 ± 34.56 34.20%	127.31 ± 37.55 36.23%	106.21 ± 38.85 31.21%
70	252.57 ± 20.44 77.31%	235.14 ± 23.57 72.87%	227.59 ± 33.10 69.31%	204.33 ± 36.77 63.63%	225.35 ± 37.58 68.73%	198.69 ± 40.93 61.89%

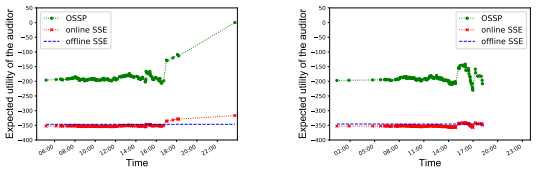
normal working days in 2017. The total number of unique accesses ⟨Date, Employee, Patient⟩ is on the order of $10.75M$. The mean and standard deviation of daily unique accesses are approximately $192K$ and $8.97K$, respectively. We focus on the following alerts types: employee and patient: 1) share the same last name, 2) work in the same department, 3) share the same residential address, and 4) are neighbors within a distance less than 0.5 miles. When an access triggers multiple distinct types of alerts, their combination is regarded as a new type. We refer readers to [Yan *et al.*, 2020] for the statistics of alert types, as well as the experts' estimates of payoff structure for players.

4.2 Experimental Setup

The audit cycle is defined as one day from *0:00:00* to *23:59:59*. From the dataset, we construct 15 groups, each of which contains the alert logs of 41 continuous normal working days as the historical data (for estimating the distributions of future alerts in all types), and the alert logs of the 1 subsequent day as the day for testing purpose. We set up a real time environment for evaluating the performance in terms of the auditor's expected utility. We set the audit cost per alert to $V^t = 1, \forall t \in \{1, ..., |T|\}$ and the frequency at which users quit when they receive the warning messages to $P^t = 0.186$ for all types based on observations.

We compare the real time auditor's expected utility for each triggered alert between the OSSP and both the *offline* (which determines the auditing strategy at the end of the auditing cycle) and *online SSE* (which determines the auditing strategy for each alert in real time without signaling).

To investigate the robustness of the results over different game conditions, we evaluate the performance by varying three factors. First, we vary the loss value for the auditor with respect to each quit of a normal user when receiving a warning message. We set $C_t = \{-1, -5, -10\}$. Second, to deter the attacker who quits until they receive no warning in the safe period for an SAG, we assess a series of constant budgets, which we set to $\alpha = \{1\%, 5\%\}$ of the total available budget B. We do not consider this situation in the baseline strategies because such loss does not apply. Third, we vary the total auditing budget. Specifically, we consider $B = \{30, 50, 70\}$.



(a) Day 1 (b) Day 4

Figure 3: The auditor's expected utility in the OSSP and alternative equilibria with $B = 50$, $\alpha = 1\%$ and $C_t = -1$ for the OSSP.

4.3 Results

Due to space limitations, we only show the results of two testing days along the timeline in Figure 3, as the results in other testing days and game conditions show similar patterns.

There are two notable findings and implications. First, in terms of the expected utility of the auditor, OSSP significantly outperforms the offline SSE and the online SSE. This suggests that the SAG increases auditing effectiveness. We believe that this advantage is due to the optimized signaling mechanism, which ensures the loss of the auditor is zero when sending warning messages. Second, the sequences of online SSE are close to the corresponding offline SSE sequences. This indicates that the auditing procedure does not benefit from determining only the coverage probability for each of the alert types in real time. In other words, the signaling mechanism in the SAG can assist the auditing tasks in various environments.

We then expanded the investigation to consider various conditions of auditing. We computed the mean (and standard deviation of) differences between the OSSP and the corresponding online SSE for each triggered alert across 15 testing days. The results are shown in Table 1, where we also indicate the percentage of the averaged improvement in each setting. From the results, we have the following significant observations. First, it is notable that OSSP consistently outperforms the online SSE with respect to the auditor's expected utility. For example, in the setting that $C_t = -1$ for all t and $\alpha = 1\%$, as B grows from 30 to 70, the auditor's expected utility improvement grows from 16% to 77%. Second, by fixing B and C_t for all t, the auditor's expected utility decreases when we reserve more budget to investigate the repeated requests by single user. Yet, this is not unexpected because this approach reduces the amount of consumable auditing resources. Third, by increasing the cost of deterring a single normal data request, we also weaken the advantages of OSSP over the online SSE.

In addition, we tested the average running time for optimizing the SAG on a single alert across all the testing days. Using a laptop running Mac OS, an Intel i7 @ 3.1GHz, and 16GB of memory, we observed that the SAG could be solved in 0.06 seconds on average. As a consequence, it is unlikely that system users would unlikely perceive the extra processing time associated with optimizing the SAG in practice.

5 Conclusion

Alert-based auditing is often deployed in EHR systems to address attacks to patients' privacy. However, the volume of alerts is often beyond the capability of administrators, thus limits the effectiveness of auditing. Our research illustrates that strategically incorporating signaling mechanisms into the data request workflow can significantly improve the auditing work. We investigated the features, as well as, the value of a game theoretic auditing, along with an Online Stackelberg Signaling Policy to solve the game. While we demonstrated the feasibility of this approach with the audit logs of an electronic medical record system, the approach is sufficiently generalized to support auditing in a wide range of environments.

References

[Blocki *et al.*, 2012] Jeremiah Blocki, Nicolas Christin, Anupam Datta, and Arunesh Sinha. Audit mechanisms for provable risk management and accountable data governance. In *Proceedings of the 2012 International Conference on Decision and Game Theory for Security*, pages 38–59, 2012.

[Blocki *et al.*, 2013] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel D Procaccia, and Arunesh Sinha. Audit games. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 41–47, 2013.

[Blocki *et al.*, 2015] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel D Procaccia, and Arunesh Sinha. Audit games with multiple defender resources. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, volume 15, pages 791–797, 2015.

[Do *et al.*, 2017] Cuong T Do, Nguyen H Tran, Choongseon Hong, Charles A Kamhoua, Kevin A Kwiat, Erik Blasch, Shaolei Ren, Niki Pissinou, and Sundaraja Sitharama Iyengar. Game theory for cyber security and privacy. *ACM Computing Surveys*, 50(2):30, 2017.

[Dughmi and Xu, 2016] Shaddin Dughmi and Haifeng Xu. Algorithmic bayesian persuasion. In *Proceedings of the 48th annual ACM symposium on Theory of Computing*, pages 412–425, 2016.

[Fabbri *et al.*, 2013] Daniel Fabbri, Ravi Ramamurthy, and Raghav Kaushik. Select triggers for data auditing. In *Proceedings of the 29th IEEE International Conference on Data Engineering*, pages 1141–1152, 2013.

[Gunter *et al.*, 2011] Carl A Gunter, David Liebovitz, and Bradley Malin. Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE Security & Privacy*, 9(5):48, 2011.

[Kuna *et al.*, 2014] Horacio D Kuna, Ramón García-Martinez, and Francisco R Villatoro. Outlier detection in audit logs for application systems. *Information Systems*, 44:22–33, 2014.

[Laszka *et al.*, 2017] Aron Laszka, Yevgeniy Vorobeychik, Daniel Fabbri, Chao Yan, and Bradley Malin. A game-theoretic approach for alert prioritization. In *Proceedings of the 31st AAAI Workshop on Artificial Intelligence for Cyber Security*, 2017.

[Mazzawi *et al.*, 2017] Hanna Mazzawi, Gal Dalal, David Rozenblatz, Liat Ein-Dorx, Matan Niniox, and Ofer Lavi. Anomaly detection in large databases using behavioral patterning. In *Proceedings of the 33rd IEEE International Conference on Data Engineering*, pages 1140–1149, 2017.

[Puppala *et al.*, 2016] Mamta Puppala, Tiancheng He, Xiaohui Yu, Shenyi Chen, Richard Ogunti, and Stephen TC Wong. Data security and privacy management in healthcare applications and clinical data warehouse environment. In *Proceedings of the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pages 5–8, 2016.

[Sinha *et al.*, 2018] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: looking beyond a decade of success. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5494–5501, 2018.

[Xu *et al.*, 2015] Haifeng Xu, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe. Exploring information asymmetry in two-stage security games. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1057–1063, 2015.

[Yan *et al.*, 2018] Chao Yan, Bo Li, Yevgeniy Vorobeychik, Aron Laszka, Daniel Fabbri, and Bradley Malin. Get your workload in order: Game theoretic prioritization of database auditing. In *Proceedings of the 34th IEEE International Conference on Data Engineering*, pages 1304–1307, 2018.

[Yan *et al.*, 2019] Chao Yan, Bo Li, Yevgeniy Vorobeychik, Aron Laszka, Daniel Fabbri, and Bradley Malin. Database audit workload prioritization via game theory. *ACM Transactions on Privacy and Security*, 22(3):17, 2019.

[Yan *et al.*, 2020] Chao Yan, Haifeng Xu, Yevgeniy Vorobeychik, Bo Li, Daniel Fabbri, and Bradley A Malin. To warn or not to warn: Online signaling in audit games. In *Proceedings of the 36th IEEE International Conference on Data Engineering*, pages 481–492, 2020.