A Taxonomic Classification of Insider Threats: Existing Techniques, Future Directions & Recommendations

Usman Rauf[1,*], Fadi Mohsen[2] and Zhiyuan Wei[1]

[1]*Dept. of Math. & Computer Science, Mercy College, NY, USA*
[2]*Information Systems Group, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, 9712 CP Groningen, The Netherlands*
E-mail: urauf@mercy.edu; f.f.m.mohsen@rug.nl; zwei1@mercy.edu
[*]*Corresponding Author*

Received 31 October 2022; Accepted 16 February 2023;
Publication 28 April 2023

Abstract

In the last two decades, the number of rapidly increasing cyber incidents (i.e., data theft and privacy breaches) shows that it is becoming enormously difficult for conventional defense mechanisms and architectures to neutralize modern cyber threats in a real-time situation. Disgruntled and rouge employees/agents and intrusive applications are two notorious classes of such modern threats, referred to as *Insider Threats*, which lead to data theft and privacy breaches. To counter such state-of-the-art threats, modern defense mechanisms require the incorporation of active threat analytics to proactively detect and mitigate any malicious intent at the employee or application level. Existing solutions to these problems intensively rely on co-relation, distance-based risk metrics, and human judgment. Especially when humans are kept in the loop for access-control policy-related decision-making against advanced persistent threats. As a consequence, the situation can escalate and lead to privacy/data breaches in case of insider threats. To confront such challenges,

Journal of Cyber Security and Mobility, Vol. 12_2, 221–252.
doi: 10.13052/jcsm2245-1439.1225

the security community has been striving to identify anomalous intent for advanced behavioral anomaly detection and auto-resiliency (the ability to deter an ongoing threat by policy tuning). Towards this dimension, we aim to review the literature in this domain and evaluate the effectiveness of existing approaches per our proposed criteria. According to our knowledge, this is one of the first endeavors toward developing evaluation-based standards to assess the effectiveness of relevant approaches in this domain while considering insider employees and intrusive applications simultaneously. There have been efforts in literature towards describing and understanding insider threats in general. However, none have addressed the detection and deterrence element in its entirety, hence making our contribution one of a kind. Towards the end of this article, we enlist and discuss the existing data sets. The data sets can help understand the attributes that play crucial roles in insider threat detection. In addition, they can be beneficial for testing the newly designed security solutions in this domain. We also present recommendations for establishing a baseline standard for analyzing insider-threat data sets. This baseline standard could be used in the future to design resilient architectures and provide a road map for organizations to enhance their defense capabilities against insider threats.

Keywords: Insider threats, anomaly detection, attack deterrence, intrusive applications, machine learning.

1 Introduction

To understand the financial implications of such attacks, we studied the most recently released reports on “Cost of Insider Threats” [2, 4, 17]. According to these reports (containing data reporting from over 300 global organizations), the number of organizations experiencing insider incidents (between 21–40 events) annually has increased by 15% over the last four years. The average annual cost affiliated with these threats is just above $15 Million and $648,000 per incident (a 23% increase from the average cost per incident in 2018). These numbers show how drastic the situation is when dealing with insider threats. This calls for an imminent need to design more effective and efficient methods for detecting and mitigating insider threats.

On the other hand, in the mobile application domain, only half of the US mobile consumers are either aware of or have any antivirus-based protection on their mobile devices [27]. Whereas according to a recently published report by McAfee [3], more than half of the smart devices do not contain any security layer to defend against privacy evading intrusive applications. Instead, the users have to rely on application publishing authorities, i.e., Google Play store, or Apple store. These services might be helpful when it comes to the identification of malware or malapps, but they provide no support, user awareness, and defense against intrusive applications. Hence, smart device consumers are vulnerable to all intrusive applications and data theft. The readers must also notice that the financial damage (as mentioned earlier) related to these incidents includes only the data reported by the organizations. It does not incorporate the potential damage that could be done to millions of smartphone users on a day-to-day basis by intrusive applications.

The technology that is closest to a solution for protection against a disgruntled employee is Security Event Management (SEM) or Security Information and Event Management (SIEM) systems [47,48]. The underlying principle of these systems is to efficiently collect logs from firewalls or operating systems, for (positive) correlation-based log/data analysis or predefined policy violation analysis. When either of these events happens, the system triggers an alarm to alert the security administrator or analyst. In the next stage, the security administrator inspects the legitimacy of the alarm, to make sure it is not a false-positive event. Once confirmed, the admin/analyst either reports the incident to the IT facility or takes the necessary actions. In the majority of cases (organizations or institutions), the security admin must report any such incidents to the IT department, as the security admin/analyst mostly does not have privileges to change/tune access control policies. The added delay becomes potentially an ideal situation for an ongoing attack and thus benefits the adversary.

Modern mobile operating systems such as Android and iOS allow users to extend the functionality of their platforms by installing third-party applications. Those apps, in essence, could be considered insider threats to the platform and the users alike. As such, numerous works have been proposed to detect the privacy risks of these applications. The bulk of research in this area has identified two types of risks: malicious apps and intrusive apps. Malicious apps are intentionally built to harm users and their devices. On the other hand, intrusive apps are benign since they are built to provide users with services. However, intrusive apps tend to collect more sensitive information about the users for advertising purposes. For the scope of this paper, we will only focus our discussion on intrusive applications as a source of insider threats. As just like legitimate (disgruntled) employees have access to organizational resources, these applications have a safe passage to the

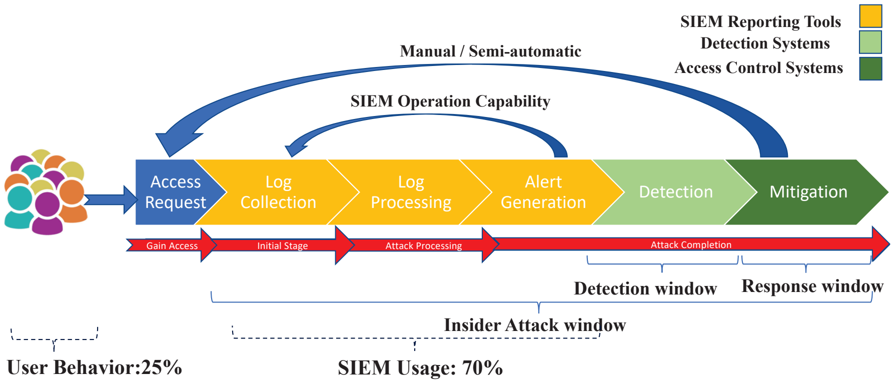

Figure 1 Threat neutralization: illustration of insider attack time-line vs. insider threat deterrence capability.

broad range of on-device resources that can access the private information about the user. This access could potentially be used to collect sensitive information if unnecessary permission is provided to an intrusive application.

One of the most crucial steps, toward thwarting insider threats, is developing an understanding of the threat timeline that emerges as a result of such attacks. Only then could it be possible to evaluate a defense mechanism in a proper context and to understand the degree and depth of protection provided by it. We construct a side-by-side comparison-based analysis of the insider threat and defensive timelines (steps taken in the existing state-of-the-art solutions to neutralize them). The threat timeline of such attacks can be divided into four stages (Figure 1).

The first stage for the attacker does not require any effort since he/she already has access to the resources/device by being a legitimate employee/application. On the other hand, the first stage for the defender is anything, but simple, as it requires the collection of activity-based information stored in system logs; hence, in the first stage, defenders can only gather these logs. During this phase, given the current state-of-the-art technologies, it is nearly impossible for defenders to comprehend an insider's motivation (since they are legitimate employees). The second stage of such attacks involves preparations for deciding what to steal, whether it is an on-device application or a disgruntled employee. As for the defenders, they can only analyze the collected data from the system logs (in the case of an employee) and access permission set (in case of an intrusive application). In the third stage (attack processing), once the insider has decided what to do (e.g., privilege escalation, data theft, or privacy evasion), they could either transfer the data (obtained as a result of the aforementioned type) to a local external drive or a remote server via secure SSH connection. At this stage, SIEMs can start generating correlation-based alerts that are not necessarily true positive.

On the other hand, in the case of intrusive applications, there is no standard way, nor are there industrial tools to generate privacy alarms. Once an application has been installed on the device with the requested permissions, there is no way for an end-user to rectify the damage unless he/she decides to uninstall the application. In the employee's case, these alerts are investigated by human security (Security Operation Center (SOC)) analyst, which takes an average of 15 minutes according to the SPLUNK guide and recently published reports [2, 17].

The last stage for the attacker is relatively brief, as it only involves data transmission. This could take a few to several minutes, depending on the complexity and nature of the data. This stage could overlap with any defender activities, as the defender technology may only be in the alert generation (threat analysis) phase instead of detection and mitigation. The threat analysis stage for defenders enormously requires human assistance (SOC analysts), as the access control policies are not tuned automatically and need to be tuned either by a security analyst or (primarily) by the IT department.

In the case of intrusive applications, the defender has no capability and time to mitigate the threat. Therefore, there is a need to have a mitigation mechanism that can detect any intruding actions (requiring abnormal permission requests). However, there is no existing mechanism that provides this feature. Most of the research efforts in this domain are geared toward designing recommendation systems rather than mitigation systems.

The recent surveys also show that 70% of reported organizations use some SIEM technology (not for insider threats) for general security monitoring [2, 17]. Among these organizations, only 25% can monitor employee behavior. Therefore, it leaves most organizations at the mercy of SOC analysts and their ability to write detection rules since SIEMS can only identify events according to pre-defined rules and cannot generate alerts against unseen events/behaviors or ongoing attacks. These statistics and comparisons of the aforementioned two threat timelines show that the security gaps in the existing technology enormously benefit the adversaries. These gaps must be filled with new and more agile methods.

The same is the case with intrusive applications. The lack of existing standards and architectures is the main reason for the repeated attacks on users' privacy [6, 49]. There is an urgent need for a robust solution or combination of solutions to mitigate intrusive apps at run-time. The solution would synthesize and assign the optimal (low-risk) set of permissions to an application. Existing recommend systems cannot contain these threats, forming this survey's basis. Thus, our goal is to identify existing solutions in this domain and evaluate them. The evaluation determines if a solution can potentially be used to defend against insider threats.

Toward this objective, we propose a five-point evaluation criterion. For each potential solution, we first evaluate if this solution focuses on detection or deterrence. For detection, we further identify whether the proposed solution uses any existing (i) signature-identification-based detection approaches or an (ii) anomaly-identification-based detection approaches. (iii) Our third dimension in the evaluation criterion is a data-based check. It verifies whether a behavioral attribute is present in the detection data set. Although employee-based insider threats and intrusive applications are two different aspects of the same problem, understanding how an insider behaves is crucial in designing a mitigation solution. (iv) The fourth dimension in the evaluation criterion is to confirm whether the proposed solution provides or relies on any risk/reputation-based metric. (v) Our final dimension in the evaluation is to confirm whether the proposed solution can provide autonomous deterrence by autonomous permission/policy regulation.

Threat detection and deterrence in domains such as ***insider employees*** or ***intrusive applications*** is still emerging. To the best of our knowledge, there is no single contribution toward encapsulating and summarizing existing defense approaches in these domains. This becomes the motivation of our work, and towards this direction, we contribute the following: we first propose a taxonomic classification of existing approaches to assist a potential reader and then evaluate each one of them according to our proposed criteria. We believe that this is essential not only for understanding behavioral anomalies but also for deterring an insider promptly and effectively. Second, we propose our future recommendations and conclusion, which can be substantially beneficial to deal with the existing challenges in this domain.

The rest of the paper is organized as follows in section. 2, we hierarchically describe our taxonomic classification. In Section 3, we discuss the *Detection* part of the literature. Section 4, presents the approaches focused on the *Deterrence* aspects and further classifies them into manual/semi-autonomous and fully autonomous solutions. In Section 5, we evaluate classified approaches according to our proposed criteria, and finally, in Section 6, we discuss existing data sets to test newly developed approaches in this domain and propose our recommendations/guidelines to develop a baseline standard towards dealing with insider threats.

2 Taxonomy of Insider Threat Detection & Deterrence Mechanisms

Figure 2 provides an overview of our proposed classification. The very first level in the hierarchy of classification contains objectives. The primary two objectives in the insider threat domain are to either detect an insider or to deter an ongoing insider attack. Different methods/processes have been adopted to achieve these objectives, bringing us to the second classification level. For instance, *Insider Threat Detection* can be either signature-based or anomaly-based. Whereas on the other hand, *Deterrence* can be achieved either via autonomous methods or semi-automatic methods. The third level in the taxonomy contains techniques implemented in the methods to achieve desired objectives. These techniques can be as simple as calculating correlation based on log analysis, as in the case of SIEM, or representation of the access control mechanism on the bio-inspired principles. To better inform our readers about the potential of an approach, at the final level of the hierarchy, we highlight what type of capabilities can be attributed to a particular approach. Eventually, in the last level, we identify the goals and objectives attributed to a specific approach. In the next section, we discuss each existing method in detail under the umbrella of the main goals and highlight their limitations.

3 Threat Detection in Insiders & Intrusive Application

Due to the nature of contributions in the insider threat detection domain, we classify the related work in this domain into two categories: (1) anomaly-based threat detection techniques and signature-based threat detection techniques [21]. Due to the involvement of non-technical factors for understanding the behavior of a legitimate individual/employee, this domain faces rare and unique challenges, as compared to the challenges faced by the other detection domains [47]. For instance, involvement of unrelated activities, unusual variation or shift in a user's behavior, lack of verifiability of privilege escalation, and inter-dependency of activity attributes on each other. All these parameters make *insider threat detection* domain much more

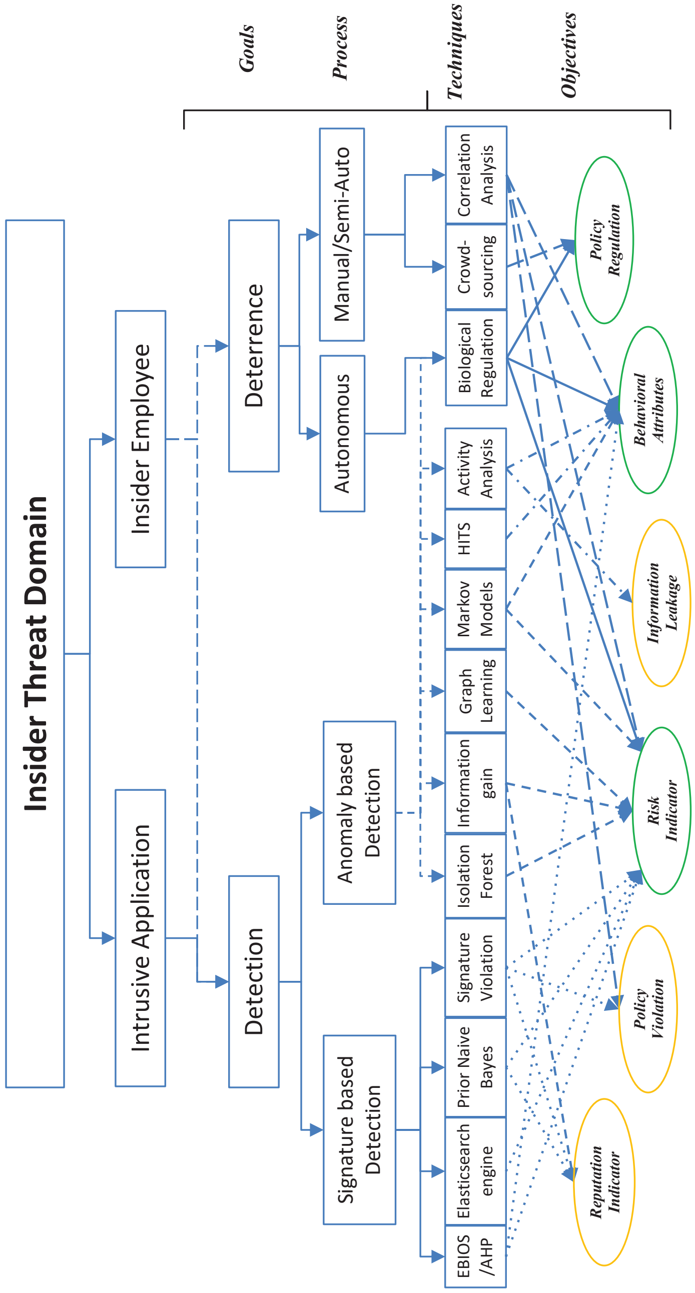

Figure 2 Classification of insider threat detection & deterrence research.

complex, making an *insider threat* one of the biggest existing challenges in the cyber-security domain [2, 14].

3.1 Signature-based Threat Detection Systems

Signature-based detection methods are designed to detect known real-world threats based on the signatures affiliated with an attack or threat. Signatures, in terms of insider threat, can be described as a pre-existing policy, i.e., illegitimate access to a key resource, file, or system. In literature, Agrafiotis et al. [8] proposed a tripwire solution based on the policies defined over alarming behaviors, and attack patterns, to predict/detect actions that are indicators of insider threat [47]. The proposed approach neither provides any understanding of how the result of detection will reinforce the policy regulation, nor any real-life threat test data-set-based experimental evaluation. IBM SIEM solution, QRadar, works on the same principles, via incorporation of offenses (signatures) implementation. This signatures-based offense repository is then used to detect threats, in general, [25].

Attack trees based approaches, to detect insider threats, have also been proposed in literature [11]. The authors leverage the attack-graph-based concept to model all possible scenarios through which an insider can compromise a certain asset/target. The main advantage of constructing attack graphs/trees is the possibility of computing a minimum cut set. A minimum cut set, once computed against each scenario, can then be used to help design countermeasures. The approach has a significantly high dependency on the accuracy of the modeled process (attack graph), whereas, such models can only incorporate known vulnerabilities and are not useful when it comes to unknown threats. The proposed approach also does not provide any basis to account for behavioral anomalies and access regulation.

Gates et al. purposed to communicate risk-based information to users along with the permissions required by an application. The authors assign a risk score to each application based on its permissions relative to the rest of the applications in the data set. The authors claim that the inclusion of risk score information significantly affects the users' decision in application selection [20]. Although it provides an added layer of awareness to the users, the authors do not show how significant/effective this inclusion can be in avoiding the installation of intrusive applications.

The paper of Mohsen et al. proposed a security-centric ranking algorithm to assist users in avoiding installing intrusive applications. It is built on top of the Elasticsearch engine. It calculates an intrusiveness score for an app based on user preferences, the broadcast receivers of an app, and its requested permission stacks. A high privacy score of an app shows that it uses permissions and configurations common to all its peers. In their demo system, users can search for applications, and the results would then be displayed and ranked based on their relevance to the query and intrusiveness score. The authors tested the efficiency and accuracy of the approach via bench-marking and a user study, respectively. The results suggested that the presented approach protects mobile users' privacy and enforces the least privilege among developers to satisfy users' demands for privacy [33]. In an extended version of this work, Mohsen et al. conducted further bench-marking after tripling the size of the dataset [34]. In addition, the authors conducted a statistical analysis of the intrusiveness scores of more than a million applications. The results showed that the system scales well with the increase in the data set size. The distribution of the intrusiveness scores over the different genres proves its applicability in a real-life scenario.

Bisgin et al. [10] proposed a machine learning model for the prediction of malicious Android applications based on their permissions requests and system broadcast receivers. Before this work, many researchers have only used permissions. In their work, they showed the contribution of broadcast receivers in increasing the detection accuracy of the models.

3.2 Anomaly-based Threat Detection Systems

Anomaly-based detection systems do not rely on the signatures of an event, rather they develop an approximate understanding of malicious activity by constructing a profile about the normal behavior. Therefore, any deviation from normal behavior is categorized as an anomaly.

The focus of these systems is on identifying unknown attacks and behaviors for which no known fingerprints/signatures are available. In literature, for learning behavior or normal pattern the use of non-technical psychological indicators has also been proposed, as using such indicators can enhance the chances of understanding the psychological state of an insider for detection purposes [7, 39].

One such approach which utilizes non-technical indicators, to predict an individual's behavior is proposed by Brdiczka et al. [13]. The authors use a game data set (World of Warcrafts), along with activity-related information from social networks to evaluate their approach. Although they provide grounds for developing a detection mechanism to incorporate non-technical indicators, in reality, the relevance between a game player and an insider is closer to none due to the differences in the nature of attributes, motivation, and intent in which an individual behaves [47]. The proposed research also does not present any hint about how to utilize the measured threat impact of an individual for policy regulation purposes.

Belief-based approaches have also been used in the literature to detect behavioral anomalies. Chen et al. [16], are the first to propose a formal framework based on probabilistic theory to deal with insider threats. The authors use a belief-based *Bayesian* modeling approach to first construct belief about the intention of an insider to attack. Then, it uses a probabilistic model checking to calculate the probability of an attack by a potential insider. The proposed approach suffers from fundamental limitations when it comes to calculating the probability values and Markov Decision Process (MDP) based modeling for all threat scenarios.

The resultant model becomes highly unrealistic and complex as MDP requires modeling of each threat and an individual along with state transitions (actions that an individual can perform to trigger some threats) among them, while only allowing the analysis of an individual at a time. Whereas it becomes highly impractical when analyzing an organization with hundreds (if not thousands) of employees. The second limitation of the proposed approach is the utilization of Bernoulli's distribution for assigning the values of probabilities to the transitions in the MDP model. In reality, the actions of an insider do not follow Bernoulli's distribution, as this only allows the outcome to have two possibilities (yes/no), and hence cannot predict complex unknown behavior (as insider attacks are a series of unknown events). The approach also does not use technical attributes, which could enhance the chances of detection [47].

A similar approach proposed by Brdiczka et al. [13] also leverages an automated technique and incorporates non-technical attributes (by collecting sensitive data, i.e., social network surfing) for the analysis. The authors aimed to increase detection accuracy by incorporating non-technical indicators, whereas the evaluation could only deliver 82% accuracy. However, the approach performs better than existing approaches. However, it is highly impractical for real-life usage. This is due to some technical challenges, i.e., low accuracy and the unavailability of a mapping mechanism from detection to policy regulation [47].

In literature, some methods focus on specific threat detection rather than a generic one. For instance, Zhang et al. [42] analyze document access patterns to understand users' intentions based on the document's contents. The authors proposed constructing all users' profiles and defining anomaly as a deviation of the current access pattern from the history profile. The proposed method is focused on monitoring only a single indicator (access file type), resulting in the capability only to analyze the specific type of insider threat, i.e., information leakage. According to recently published research and technical reports, combining multiple attributes for detection purposes can enhance the detection accuracy [14, 17, 47].

Another similar approach that works on the same concept has been proposed by Senator et al., which observes an insider based on his/her database access behavior [53]. The authors incorporate multiple attributes to deal with the low signal-to-noise ratio challenge. However, they do not provide any information about how their detection results can be leveraged to tune policy against an insider. Additionally, their approach relies on a security manager/analyst for necessary (access control reporting/tuning) actions.

Legg et al. [29] proposed a Principle Component Analysis (PCA) based solution to analyze behavioral anomalies by observing the employees' online activities. The authors proposed to construct user activity vectors hourly and build a 24-hour activity matrix for employees. This activity matrix is passed onto the PCA module to calculate the distance between activity vectors and classify users into groups identifying the insiders with high variance. These methods require a user-defined threshold/criteria to classify the data into a certain number of groups. Hence, it limits the ability of a security manager/analyst to interpret information and trigger necessary actions promptly to deal with an ongoing threat. Another limitation of the approach is that it clusters identical users in the same group. Integrating it with policy regulation mechanisms makes it challenging, as traditional policies are defined hierarchically, and all identified malicious individuals may not belong to one hierarchy, highlighting the need for an automated policy optimization/synthesis process.

Finally, Rashid et al. [43] proposed a detection method based on Hidden Markov Models (HMM). The proposed method learns the normal (behavioral) profiles of employees and analyzes deviations from the normal profiles to detect insider threats. Normal behavior is considered as the sequence of events in the context of the paper. The proposed approach performs well while learning from data that is sequential. However, due to the complexity of the problem and the increasing computation cost of training large models, the efficiency of the proposed problem in real-life scenarios can be very limited.

In the mobile application domain, Quattrone et al. proposed *Privacy-Palisade* [41] to identify intrusive applications. In this work, the set of applications is divided into similarly-function groups. Each app is then assigned a privacy score based on its permission requests and relative to other apps in the same group. An intrusive app requests permissions that are considered unusual to the group. The authors reported that their approach was able to label 5%–10% of google store applications as outliers. However, they did not provide any distinction between outliers and anomalous applications. They didn't also share the list of outlier applications for further evaluation.

Hu et al. proposed a completely different approach to tackle intrusive apps [24]. The authors modeled the behavior of an application based on its static call graphs. Instead of ranking the permission stack, they used a page ranking algorithm HITS to simulate a method call graph, therefore, determining a static call graph of an android application. This approach allows security analysts to detect flaws and calls to sensitive methods. Their approach is mostly focused on malware. Nonetheless, it can potentially be used to detect intrusive applications if it is applied to permission-related information. One major flaw of the proposed approach is that the detection is completely dependent on social ranking rather than using it as one of the broader attributes for detection.

Kartik et al. proposed a similar approach to malware detection (IPDroid). The authors proposed an algorithm to find the best set of permissions and intents to detect exploits in android applications. The approach uses Information Gain to rank the permissions and intents, then uses machine learning to find the best-set combination of permissions and intent. The experimental results demonstrate that the best set consists of 37 features, i.e., 20 intents and 17 permissions, achieving an accuracy of 94.73% [28]. The proposed approach intents to find the best set of permissions and intent. However, the authors neither provide any calibration of underlying parameters for prediction, nor any mathematical criteria for labeling prediction.

4 Insider Threat Deterrence: Access Policy & Application Permission Regulation

In this section, we discuss and enlist contributions, which either provide a way of autonomously tuning access control policy or propose to deal with an insider in a semi-automatic way. In either case, the fundamental ingredients to actively deter an insider involves the integration of *threat analytics* (via considering behavioral attributes) and *policy tuning/synthesizing* mechanisms.

4.1 Manual/Semi-Autonomous Approaches

RBAC is the most intensively used architecture in the access control domain. Though it has certain benefits, this control's architecture does not allow automated synthesis of security policies at run time. Due to these inherent limitations, several extensions to this architecture have been proposed over the recent years [15, 18, 19] to eliminate this problem. These extensions into RBAC integrate the risk/trust notion in the access control mechanism. These approaches proposed tuning access policy against risk or trust level changes. However, they do not provide comprehensive details about how the risk or trust values are calculated. In addition, they do not specify how the access control policy can be synthesized to satisfy all constraints fully automatedly.

Sudip et al. [15] proposed associating a trust interval to each role, and then trust intervals are assigned to each employee. Hence, the roles of users are changed/switched according to their trust levels. This approach violates the fundamental property of RBAC, in which users are assigned roles according to their function in the organization's hierarchy instead of trust levels.

Feng et al. [19] proposed that the users must be assigned to the roles according to their context information and trustworthiness. A similar approach is proposed by Gimmock et al. [18], in which permissions are labeled with risk and trust thresholds. If the trust of the user requesting authorization balances the risk of the action, permission is granted. These approaches incorporate the notions of trust and risk in access control mechanisms. However, none of these approaches provide any clear answers to the most fundamental questions in the insider threat deterrence domain, such as *how the trust and risks are computed*, and *how the risk can be minimized on run-time*.

Ma et al. [31] proposed to assign roles with confidence levels, whereas employees/users are assigned clearance levels. Actions and assets are assigned values according to the nature of their significance. Based on these assignments, risk against a user (with a specific role) trying to access an asset is calculated. However, the proposed approach does not provide any way to mitigate insider threats, as the parameters representing trust/risk are not dynamic and are not linked with the user's behavior. In addition, the authors also do not provide any experimental evaluation of the proposed mechanism.

In [50], the authors proposed assigning users a specific budget depending on their organizational roles. They also set a cost factor for each access permission. When an authorized user requests access to an asset, the value of the budget is consumed (as there is a cost affiliated with permission), and approval is granted only if the cost is under the available budget of the user. The authors claim that these approaches can force individuals to spend their budget more cautiously. However, the proposed method may provide significant advantages to a disgruntled employee. For instance, a legitimate employee turned into a malicious insider may not care about spending his all budget to steal critical files/assets, leaving the organizational assets vulnerable [47].

Similar approaches which aim at minimizing risk exposure have also been proposed in the literature [9, 38]. For instance, Nissanke et al. [38] proposed an approach for risk analysis in which the permission set is labeled with the risk values, and role hierarchy is organized based on the risk. The author aims to assist a security analyst such that he/she may only assign permissions to the roles after considering the risk affiliated with these permissions. The proposed approach does not allow the maintenance of a role hierarchy, which is unrealistic, as the roles in an organization are required to maintain a hierarchy, as RBAC hierarchically allows inheritance.

Aziz et al. [9] proposed a method to optimize risk exposure against the system's evolution. The risk is defined in an interval over reals $[t, t') \in \mathbb{R}$. A subset of obligations is associated with the users based on the assessed risk. The number of obligations can increase or decrease against a user at any time. However, the authors incorporate risk in the system but do not provide information about risk calculation, behavioral analysis, and fulfillment checks against imposed obligations.

Malicious applications have been studied far more than intrusive applications. This is because the former is relatively older than the latter. In addition, the threat of malicious applications is more severe than that of invasive applications. Therefore, more research is needed to counter intrusiveness among mobile apps. For example, designing automated permission/policy enforcement approaches.

Rashidi et al. presented an automated permission enforcement approach for Android called DroidNet [44]. DroidNet assists normal users in making safe decisions concerning permission requests. DroidNet uses crowdsourcing to collect suggestions from expert users and use this information to provide recommendations to users about whether to accept the application's permission request or not. The authors evaluated the working of their application via simulation instead of real-life user-based experience. The authors neither used any risk-based metric nor permission/broadcast receiver-related information to assess the intrusiveness of an application.

Instead, the framework heavily relies on the suggestions of expert users via crowd-sourcing.

Asma Hamed and Hella Kaffel Ben Ayed proposed a slightly different approach but targeting similar goals [23]. The authors developed a privacy scoring algorithm based on the relative severity of permissions and interactions and their importance rating. They found that most requested permissions are the ones that are related to storage, device, location, account, and other personal information. The data set that was used in this study has only 64 applications. The authors did not provide any details about contamination within the data samples.

4.1.1 Commercial Tools

Many commercial products also incorporate risk in their solutions, e.g., SAP [51], Oracle [40], IBM [26], and Beta Systems [5]. These products mitigate risk by closely monitoring and auditing the usage of risky permissions. The risk values, however, are not used to make access control decisions, missing the opportunity to incorporate the overall known behavior of the users to prevent insider threats. The threat of inference of unauthorized information is particularly relevant in the insider threat context. This threat occurs when through what seems to be innocuous information, a user is capable of inferring information that should not be accessible. In existing approaches to deal with inference threat, [12], when the user is about to conclude some unauthorized information, the system prevents it by denying access or providing scrambled data. This is only adequate for some types of organizations. Organizations may need to provide access to multiple pieces of information to a single employee, even if they result in undesirable inferences. Existing RBAC extensions do not consider the risk of inferred information. Hence, limiting the existing tools to mitigate the inference risk in RBAC-based systems.

4.2 Autonomous Approaches

Recently bio-inspired concepts have also been used to design auto-resilient architecture and protocols to address the autonomous policy regulation challenges. In this section, we discuss these approaches in detail. Although Rauf et al. [45] recently published an extensive literature review of bio-inspired cybersecurity approaches, the article does not include insider threat detection or deterrence-based methods in their scope.

4.2.1 Bio-inspired approaches to access control regulation

As a result of evolution, biological systems depict promising features and intrinsic appealing characteristics. These characteristics include inherent resiliency to failures and perturbations, adaptability to varying environmental conditions, and collaborative behavior based on a limited set of rules. A cellular regulation mechanism is an example of previously mentioned attributes, which mitigates the perturbations (unusual protein concentration rate) at the cellular level via signal transduction mechanism by maintaining the optimal amount of protein concentration due to abnormal behavior of specific proteins. For a detailed discussion on cellular regulation, we refer our readers to the recently published article [45].

Rauf et al. [46] proposed a genetic regulation-inspired formal framework to regulate network-level policy against an attack autonomously. The framework includes an automated reconciliation protocol (among access control security devices, e.g., firewall) which each autonomous device can use to optimize the risk against the assets it is protecting. The authors integrate qualitative risk levels in access control and represent each access control device as an individual process/entity. The authors do not provide any detection method against behavioral anomalies (nor consider behavioral attributes while calculating risk). Instead, their approach assumes that risk is mapped to the permission set. The higher the asset's value, the higher the risk associated with it if access is granted to a request from a low-tier system/user. Therefore, their proposed approach implements a reconciliation method among devices, where devices can synthesize and reconcile policy configuration, for which the risk is minimum. Although this is the first effort towards autonomous reconciliation among access control devices, the approach is limited in its application to only network-level access, as it does not monitor or address systems-level access control.

Another approach that works on the principles of *Cellular Regulation* has also been recently proposed [47]. The authors present an integrated framework for a systemic approach to autonomously synthesize the access control policy in real-time against an originating insider threat via integration of *detection* and *threat analytic* with policy decision procedure. The authors use behavioral indicators (e.g., login time, surfed websites, file access pattern) to construct a behavioral vector against users and apply supervised machine learning algorithms to assess behavioral deviation. These behavioral deviation scores are then used to calculate risk against access permission. The trouble is eventually used in policy regulation mechanisms to synthesize a correct policy configuration. As for an access control policy, the authors define it as follows:

$$\boxed{\mathcal{P} : \bigvee_{k} \big(\bigwedge_{ij} (r_{ij})\big) \text{ whereas, } i, j, k \in \mathbb{N}}$$

Where r_{ij} represents a rule in a configuration when user i is allowed to access asset j, therefore, a policy becomes a conjunction of disjunctions, forming a sequence of regions as in a DNA sequence.

The authors proposed to model the problem of policy regulation as a state transition system. The possible policy configurations and risk vectors against users at a given time are taken as input. The system then provides a satisfiable configuration as an output. Although the authors present a detailed description of the policy regulation system and theoretical semantics, a thorough evaluation of the system is still to be seen.

In the next section, we briefly discuss our evaluation criteria and evaluate all the approaches mentioned above, describing which goals they achieve while addressing the problem of insider threats.

5 Evaluation

Key elements which we consider for evaluation are the integration of ***behavioral attributes*** for intent prediction/detection, ***Risk/Trust/Reputation*** score for understanding deviation from the expected behavior, and automated ***Policy Regulation*** mechanism. We believe these three elements are most significant when dealing with insider threats. As proposed by recently published technical reports, behavioral and risk indicators can play a crucial role in understanding malicious intent [2, 17]. Hence, these elements form the basis of our evaluation.

In Table 1, we summarize the previous research efforts in the context of insider threat detection and deterrence. We evaluate the literature based on the criteria that we mentioned above. First, we check if a contribution belongs to the anomaly or signature-based detection domain. Second, we evaluate whether a proposed method/contribution considers behavioral attributes in the detection mechanism. The third most important element in our literature review is whether a proposed method incorporates risk, trust, or reputation-based metrics in the access control problem. Finally, we evaluate a contribution against whether the proposed method links risk-based analysis to

Table 1 Existing techniques and limitations

			Behavioral	Risk	Policy
Approach (Year)	Signature	Anomaly	Attributes	Indicator	Regulation
Agraotis et al. [16]	✓	✗	✗	✗	✗
Bishop et al. [16]	✓	✗	✗	✓	✗
Oliver et al. [12]	✗	✓	✗	✗	✗
Chen et al. [15]	✗	✓	✗	✓	✗
Ted et al. [13]	✗	✓	✓ (single indicator)	✗	✗
Zhang et al. [14]	✗	✓	✓ (single indicator)	✗	✗
Rashid et al. [16]	✗	✓	✓	✗	✗
Legg et al. [17]	✗	✓	✓	✗	✗
Nissanke et al. [04]	✗	✗	✗	✓	✗
Aziz et al. [06]	✗	✗	✗	✓	✗
Sudip et al. [06]	✗	✗	✗	✓	✗
Ma et al. [10]	✗	✗	✗	✓	✗
Liang et al. [12]	✗	✗	✗	✓	✗
Feng et al. [17]	✗	✗	✗	✓	✗
Rauf et al. [19]	✗	✗	✗	✓	✓
Rauf et al. [19]	✗	✓	✓	✓	✓
Rauf et al. [21]	✗	✓	✓	✓	✓
Gates et al. [14]	✓	✗	✗	✓	✗
Quattron et al. [15]	✗	✓	✗	✓	✗
Hu et al. [18]	✗	✓	✓	✗	✗
Rashidi et al. [18]	✗	✗	✗	✓	✗
Khariwal et al. [20]	✗	✓	✗	✓	✗
Mohsen et al. [18]	✓	✗	✗	✓	✗
Mohsen et al. [22]	✓	✗	✗	✓	✗
Hamed et al. [16]	✗	✗	✗	✓	✗

the policy regulation process since we aim to determine if a proposed method can effectively deter known/unknown insider attacks.

6 Existing Data-sets & Recommendations

Although insider employees and intrusive applications are two categories of *insider threat* domain, The nature of the data and approach towards designing countermeasures based on the data sets available in the literature are completely different.

6.1 Data Set for the Analysis of Employee-based Insider Threats

The most detailed and widely used data sets to test an approach against the behavior of an insider threat were released by CERT and CMU in 2016 and 2020, respectively [1,22,30]. The data sets contain the daily log activities for more than one thousand users. The data is classified into five threat scenarios (file: scenario.txt). Following are the brief details about all the scenarios for which it could be used:

- ***Data Theft*:** An employee who is in the last part of his/her employment period (tenure/contract) attempts to transfer sensitive (or any) data by any means (via remote connection or local transfer)
- ***Intellectual Property Theft to benefit a Competitor*:** An employee starts browsing/exploring the employment website of a competitor company and attempts to steal sensitive information by any means (via remote connection or local transfer)
- ***Disgruntled System Administrator*:** System admin steals the password of a superior authority, and later (days, weeks, or months after securing access to the machine) uses the account to send out (an alarming) email to the staff or all the employees in the company, causing panic.
- ***Privacy Invasion*:** An employee/admin uses another employee's machine to scrap for important/personal information and files and tries to export it. This conduct becomes more and more frequent over the period.
- ***Data Theft by Former Employee*:** A former employee, who still maintains access privileges, attempts to steal data for personal gains after being fired from the position.

The data-set **4.2** in [30] is a "dense needle" data set containing instances from most of the scenarios. Any or all of these scenarios could be used to train and test newly proposed detection/deterrence mechanisms against employee-based insider threats. These scenarios will work as a policy adopted by an organization to keep a check on incidents if anything with similar signatures occurs. Contrarily, the data can also be used to train machine learning models to learn about the expected behavior of individuals and test the models against abnormal activities (scenarios). Finally, we recommend removing maligned employees' data before training the models because the data are not labeled. Moreover, if malicious samples exist in the data set, the training ability to learn normal behavior will significantly reduce. For a detailed description of the scenarios related to sub-data-sets, we refer our reader to the corresponding link [30].

6.2 Data Set for the Analysis of Intrusive Applications

The data set used to test the proposed intrusiveness-detection models was mainly collected from the Google Play store by Mohsen et al. [35]. It contains both the metadata of the applications and their binaries-APK (Android Package Kit) files. The metadata comprises twenty-four attributes (features) representing all the information on the Play store pages, such as the description, size, downloads, and ratings. All manifest files were extracted from the binaries and then parsed to retrieve the permissions and broadcast receivers. This resulted in the generation of 312 binary attributes containing 137 permissions and 175 receivers.

Additionally, the collected applications' status was recorded on three different occasions. The status was then used to generate a new binary attribute indicating whether the app was removed from the store. It is important to note that the data set collection took months and was executed in different stages. Though the data set was used in various publications along the way [33, 34, 36, 37], the main instances of the data can be accessed under these references: [32], and [35].

6.3 General Guidelines for the Analysis

In this section, we conduct a preliminary investigation of the data mentioned above sets and provide our recommendations to establish standard guidelines for analyzing any such data sets. Our recommendations are based on an extensive literature review, past contributions, and the experiments we conducted during our investigation in this paper. Although the employee and application-based data sets differ in nature and involve different features, the proposed guidelines are equally applicable.

1. ***Noise Removal based on A Prospective Criteria*:** During this step, the data must be pre-processed and cleaned of any outliers. To establish the outlier threshold, the security analyst must use either statistical criteria involving the six-sigma rule or an unsupervised machine learning method.
   Regardless of which approach is used, the analyst must set up a criterion and keep it consistent throughout the various phases of the analysis.

2. ***Divide & Conquer Or Unity is Strength: Big Data Vs. Small Data Decision*:** This is the most critical phase of insider threat analysis research, where an analyst must decide if he/she wants to divide the data into small segments or combine the data into a more significant chunk to get better accuracy of labeling.

   One approach is not suitable for all types of data sets. For instance, analyzing combined behavioral data sets of employees might lead to more false positives, eventually affecting the overall accuracy of the predictions [48]. Therefore, we recommend dealing with employee-based cases on an individual basis. Whereas, for intrusive application analysis, we recommend categorical segregation of the data.
3. ***Feature Elimination*:** After the big/small data decision, the next step is to perform feature elimination. This step should help an analyst to narrow down the number of features, especially if they plan to use a big data approach. This step could also reduce computational overhead, given the limited processing power of mobile devices. Regardless of the method chosen by an analyst, this step could exponentially affect the analysis accuracy and heavily relies on small/big data decisions.

   We conduct some preliminary evaluation by performing two different experiments to establish evidence. First, on the combined genre data set of intrusive apps, and second, on the single genre-based data set (extracted from the primary data set) [35]. An analyst may choose any arbitrary method or design a novel metric. Once the data profile is selected, we use Inverse Information Gain (IIG) process to measure the importance of the feature. *IIG* is a metric we define to reverse the measurement effect of *Information Gain* using *Shannon's Entropy* [52]. Information gain is usually defined as the quality of information a feature carries within the context of a data set. Whereas in this scenario, we are interested in the features which are of the most minor importance. We formally define our metric as follows:

   $$\boxed{\mathcal{IIG}_i : (IG_i)^{-1} \mapsto \{0,1\}}$$

   Whereas IIG_i represents the value of an attribute. The IG_i is usually measured in range {0,1}. Hence, IIG values could result in the range {1,10}. To keep it consistent with IG, we scale/map the results back to the {0,1} range. Therefore, the higher the value, the less quality of information a feature contains. An analyst may choose not to scale the values back, but in our experiments, we choose to scale them back for easy interpretation of results. Once a score is assigned to each

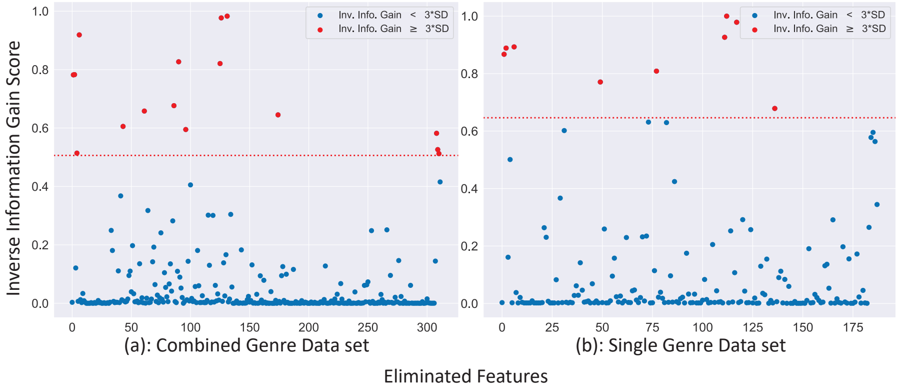

Figure 3 Feature elimination test on similar threshold: combined genre vs. single genre.

feature, we establish a criterion for deciding a threshold for dropping the features. In both experiments, we measure the standard deviation (SD) of IIG and define the values above 3*SD as extreme values of low importance. Figure 3 shows the results for both experiments.

To elaborate on the results, we compare the number of features dropped in both experiments and present the results in Table 2. The experiments resulted in seven excessive drainage features for the case of collective data analysis compared to single category analysis. It shows that when analyzing collective categories, an analyst could drop some of the features that may not seem important in the context of the joint data set but are critical features for a specific category. Therefore, it is essential to understand and compare different perspectives before finalizing the set of eliminated features.

4. ***Data Labeling: Defining Intrusiveness Criteria*****:** In this phase, an analyst depends on a deep understanding of an insider threat data set acquired from the previous steps. By establishing baseline criteria, either by using any existing metric or by designing a novel metric, the analyst should be able to define normality. Eventually, any significant variation from normality will be construed as an anomaly. We recommend that the analyst confirms by adding known malicious samples to the data set whether they are tagged/labeled malicious by the proposed criteria.
5. ***Learning What the Normality Looks Like*****:** In the final stage, we recommend that machine learning models be trained offline to avoid any latency for run-time decision-making and communications between client and server (in the case of Android application-based defense).

Table 2 Feature elimination comparison. multiple genres vs. one genre

Dropped Features (All Genres)	Dropped Features (Single Genre)
ACCESS_COARSE_LOCATION	ACCESS_COARSE_LOCATION
ACCESS_FINE_LOCATION	ACCESS_FINE_LOCATION
ACCESS_WIFI_STATE	ACCESS_WIFI_STATE
GET_ACCOUNTS	GET_ACCOUNTS
READ_PHONE_STATE	READ_PHONE_STATE
VIBRATE	VIBRATE
WAKE_LOCK	WAKE_LOCK
WRITE_EXTERNAL_STORAGE	WRITE_EXTERNAL_STORAGE
android.intent.action.BOOT_COMPLETED	android.intent.action.BOOT_COMPLETED
ACCESS_NETWORK_STATE	
READ_EXTERNAL_STORAGE	
CAMERA	
RECEIVE_BOOT_COMPLETED	
CAMERA.1	
CONTACTS	
LOCATION	

7 Conclusion & Future Directions

This paper considers the threats that could arise from an insider employee or an intrusive application by clearly defining and classifying insider threats at a granular level. This research also presents a detailed review of existing methods and approaches to addressing the above-mentioned threats. The third main contribution of this research is to develop qualitative criteria for evaluating the existing techniques and their future potential. As a final contribution, the paper presents detailed guidelines and information about existing data sets that can be used to test and evaluate any novel approach in the future. Based on our analysis, only the methods designed on the bio-inspired principles show some potential to be used for effective deterrence since they provide a way of integrating *Threat Analytics* with *Policy Regulation* mechanism. Although an integrated framework has been proposed recently [47], the rigorous evaluation of an actual data set is still to be seen. Another technical limitation is the usage of traditional supervised machine-learning methods for behavioral prediction. Although they provide reasonable accuracy, they are limited in encapsulating the inter-dependencies of behavioral attributes, as they capture the behavior of independent features (as opposed to non-linear behavior, using deep learning). Since an intrusive attack is a sequence of activities, and activities are not separate, it gives rise to non-linear dynamics with complex inter-dependencies of attributes, hence the need for more efficient deep-learning-based approaches which can account for this. Similarly, a very limited amount of work has been done in the literature to counter intrusive applications. Deterring intrusive applications brings additional challenges into the picture, as only a few resources are available on mobile devices to conduct run-time anomaly detection. Although lightweight applications can be designed by performing computationally complex ML training using cloud-based servers, this raises concerns over the privacy of the end users. Users must provide/share their application-related data to leverage such run-time detection algorithms. How well future developments can balance these trade-offs is yet to be seen.

Acknowledgement

This research was funded by a National Centers of Academic Excellence in Cybersecurity grant (H98230-22-1-0256), which is part of the National Security Agency (NSA). Any findings, conclusions, or recommendations expressed in this research are those of author(s) and do not necessarily reflect the views of the sponsor.

References

[1] (2016) CERT Threat Test Dataset. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

[2] (2018) Insider Threat Report. URL http://crowdresearchpartners.com

[3] (2022a) As mobile usage skyrockets, nearly half of consumers do not protect personal data. www.mcafee.com/cs-cz/consumer-corporate/newsroompress-releases/press-release.html?news_id=9042347b-54f5-4149-bd16-f72357b35f13

[4] (2022a) Cost of insider threats: Global. https://static.poder360.com.br/2022/01/pfpt-us-tr-the-cost-of-insider-threats-ponemon-report.pdf

[5] (2022b) Identity and Access Management Suite, Beta Systems. URL https://www.betasystems-iam.com/en/products/garancy-iam-suite/

[6] (2022b) Malware hits millions of android users. https://techstory.in/malware-hits-millions-of-android-users-the-apps-you-need-to-delete/

[7] A M, K P, M B (2012) Preventing and Profiling Malicious Insider Attacks. Tech. rep., Defense Science and Technology Organization

[8] Agrafiotis I, Erola A, Goldsmith M, Creese S (2016) A tripwire grammar for insider threat detection. In: Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, ACM, MIST '16, pp 105–108

[9] Aziz B, Foley SN, Herbert J, Swart G (2006) Reconfiguring role based access control policies using risk semantics. J High Speed Netw 15(3):261–273, URL http://dl.acm.org/citation.cfm?id=2692141.2692146

[10] Bisgin H, Mohsen F, Nwobodo V, Havens R (2021) Enhancing malware detection in android application by incorporating broadcast receivers. International Journal of Information Privacy, Security and Integrity 5(1):36–68, DOI: 10.1504/IJIPSI.2021.119168, URL https://www.inderscienceonline.com/doi/abs/10.1504/IJIPSI.2021.119168, https://www.inderscienceonline.com/doi/pdf/10.1504/IJIPSI.2021.119168

[11] Bishop M, Conboy HM, Phan H, Simidchieva BI, Avrunin GS, Clarke LA, Osterweil LJ, Peisert S (2014) Insider threat identification by process analysis. In: 2014 IEEE Security and Privacy Workshops, pp 251–264, DOI: 10.1109/SPW.2014.40

[12] Biskup J (2011) History-dependent inference control of queries by dynamic policy adaption. In: Li Y (ed) Data and Applications Security and Privacy XXV, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 106–121

[13] Brdiczka O, Liu J, Price B, Shen J, Patil A, Chow R, Bart E, Ducheneaut N (2012) Proactive insider threat detection through graph learning and psychological context. In: Security and Privacy Workshops (SPW), 2012 IEEE Symposium on, pp 142–149

[14] Cappelli DM, Moore AP, Trzeciak RF (2012) The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud). Addison-Wesley

[15] Chakraborty S, Ray I (2006) Trustbac: Integrating trust relationships into the rbac model for access control in open systems. In: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, ACM, New York, NY, USA, SACMAT '06, pp 49–58

[16] Chen T, Kammüller F, Nemli I, Probst CW (2015) A probabilistic analysis framework for malicious insider threats. In: Human Aspects of Information Security, Privacy, and Trust, Springer International Publishing, pp 178–189

[17] Cole E (2017) Defending Against the Wrong Enemy. Tech. rep., SANS Insider Threat Survey

[18] Dimmock N, Belokosztolszki A, Eyers D, Bacon J, Moody K (2004) Using trust and risk in role-based access control policies. In: Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, ACM, New York, NY, USA, SACMAT '04, pp 156–162

[19] Feng F, Lin C, Peng D, Li J (2008) A trust and context based access control model for distributed systems. In: 2008 10th IEEE International Conference on High Performance Computing and Communications, pp 629–634, DOI: 10.1109/HPCC.2008.37

[20] Gates CS, Chen J, Li N, Proctor RW (2014) Effective risk communication for android apps. IEEE Transactions on Dependable and Secure Computing 11(3):252–265, DOI: 10.1109/TDSC.2013.58

[21] Gheyas IA, Abdallah AE (2016) Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. Big Data Analytics 1(1):6

[22] Glasser J, Lindauer B (2013) Bridging the gap: A pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops, pp 98–104, DOI: 10.1109/SPW.2013.37

[23] Hamed A, Ben Ayed HK (2016) Privacy risk assessment and users' awareness for mobile apps permissions. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp 1–8, DOI: 10.1109/AICCSA.2016.7945694

[24] Hu Y, Kong W, Ding D, Yan J (2018) Method-level permission analysis based on static call graph of android apps. In: 2018 5th International Conference on Dependable Systems and Their Applications (DSA), pp 8–14, DOI: 10.1109/DSA.2018.00014

[25] IBM (2021) IBM QRadar, SIEM. URL https://www.ibm.com/downloads/cas/OP62GKAR

[26] IBM (2022) Resource Access Control Facility (RACF). URL https://www.ibm.com/products/resource-access-control-facility

[27] Jovanovic B (2022) Virus alert: 31 antivirus statistics and trends. https://dataprot.net/statistics/antivirus-statistics/

[28] Khariwal K, Singh J, Arora A (2020) Ipdroid: Android malware detection using intents and permissions. In: 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp 197–202, DOI: 10.1109/WorldS450073.2020.9210414

[29] Legg PA, Buckley O, Goldsmith M, Creese S (2017) Automated insider threat detection system using user and role-based profile assessment. IEEE Systems Journal 11(2):503–512

[30] Lindauer B (2020) Insider threat test dataset. Carnegie Mellon University, DOI: https://doi.org/10.1184/R1/12841247.v1

[31] Ma J, Adi K, Mejri M, Logrippo L (2010) Risk analysis in access control systems. In: 2010 Eighth International Conference on Privacy, Security and Trust, pp 160–166

[32] Mohsen F (2021) More than a million Android Apps with Two Privacy Scores. DOI: 10.34894/CW7PAH, URL https://doi.org/10.34894/CW7PAH

[33] Mohsen F, Abdelhaq H, Bisgin H, Jolly A, Szczepanski M (2018) Countering intrusiveness using new security-centric ranking algorithm built on top of elasticsearch. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp 1048–1057, DOI: 10.1109/TrustCom/BigDataSE.2018.00147

[34] Mohsen F, Abdelhaq H, Bisgin H (2022a) Security-centric ranking algorithm and two privacy scores to mitigate intrusive apps. Concurrency and Computation: Practice and Experience 34(14):e6571, DOI: https://doi.org/10.1002/cpe.6571, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6571

[35] Mohsen F, Karastoyanova D, Azzopardi G (2022b) The manifest and store data of 870,515 Android mobile applications. DOI: 10.34894/H0YJFT, URL https://doi.org/10.34894/H0YJFT

[36] Mohsen F, Karastoyanova D, Azzopardi G (2022c) To remove or not remove mobile apps? a data-driven predictive model approach. DOI: 10.48550/ARXIV.2206.03905, URL https://arxiv.org/abs/2206.03905

[37] Montenegro F, Bisgin H, Mohsen F, Sobers NM (2021) Predicting intrusiveness of android apps by applying lstm networks on their descriptions. In: Arai K, Kapoor S, Bhatia R (eds) Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1, Springer International Publishing, Cham, pp 1–15

[38] Nissanke N, Khayat EJ (2004) Risk based security analysis of permissions in rbac. In: WOSIS

[39] Nurse JRC, Buckley O, Legg PA, Goldsmith M, Creese S, Wright GRT, Whitty M (2014) Understanding insider threat: A framework for

characterising attacks. In: 2014 IEEE Security and Privacy Workshops, pp 214–228

[40] Oracle (2012) Application Access Controls Governor. URL https://docs.oracle.com/cd/E37379_01/doc.8643/e36194.pdf

[41] Quattrone A, Kulik L, Tanin E, Ramamohanarao K, Gu T (2015) Privacypalisade: Evaluating app permissions and building privacy into smartphones. In: 2015 10th International Conference on Information, Communications and Signal Processing (ICICS), pp 1–5, DOI: 10.1109/ICICS.2015.7459926

[42] R Z, X C, J S, F X, Y P (2014) Detecting insider threat based on document access behavior analysis. In: Web Technologies and Applications, Lecture Notes in Computer Science, Springer, vol 8710, pp 98–104

[43] Rashid T, Agrafiotis I, Nurse JR (2016) A new take on detecting insider threats: Exploring the use of hidden markov models. In: Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, ACM, New York, NY, USA, MIST '16, pp 47–56

[44] Rashidi B, Fung C, Nguyen A, Vu T, Bertino E (2018) Android user privacy preserving through crowdsourcing. IEEE Transactions on Information Forensics and Security 13(3):773–787, DOI: 10.1109/TIFS.2017.2767019

[45] Rauf U (2018) A taxonomy of bio-inspired cyber security approaches: Existing techniques and future directions. Arabian Journal for Science and Engineering DOI: https://doi.org/10.1007/s13369-018-3117-2

[46] Rauf U, Mohsin M, Mazurczyk W (2019a) Cyber regulatory networks: Towards a bio-inspired auto-resilient framework for cyber-defense. In: Bio-inspired Information and Communication Technologies, Springer International Publishing, Cham, pp 156–174, DOI: http://dx.doi.org/10.1007/978-3-030-24202-2_12

[47] Rauf U, Shehab M, Qamar N, Sameen S (2019b) Bio-inspired approach to thwart against insider threats: An access control policy regulation framework. In: Bio-inspired Information and Communication Technologies, Springer International Publishing, pp 39–57, DOI: https://doi.org/10.1007/978-3-030-24202-2_4

[48] Rauf U, Shehab M, Qamar N, Sameen S (2021) Formal approach to thwart against insider attacks: A bio-inspired auto-resilient policy regulation framework. Future Generation Computer Systems 117:412–425, DOI: https://doi.org/10.1016/j.future.2020.11.009, URL https://www.sciencedirect.com/science/article/pii/S0167739X20330338

[49] S Anthony (2022) Malware Hits Millions of Android Users. https://www.tomsguide.com/news/malware-hits-10-million-android-users-delete-these-apps-right-now, online; accessed 10 September 2022

[50] Salim F, Reid J, Dawson E, Dulleck U (2011) An approach to access control under uncertainty. In: 2011 Sixth International Conference on Availability, Reliability and Security, pp 1–8, DOI: 10.1109/ARES.2011.11

[51] SAP (2022) Sap access control 12.0. https://help.sap.com/docs/SAP_ACCESS_CONTROL.

[52] Shannon CE (1948) A mathematical theory of communication. The Bell System Technical Journal 27:379–423, URL http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

[53] Ted E, Goldberg HG, Memory A, Young WT, Rees B, Pierce R, Huang D, Reardon M, Bader DA, Chow E, et al (2013) Detecting insider threats in a real corporate database of computer usage activity. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1393–1401

Biographies

Usman Rauf received his B.S. degree in Computational Physics, in 2008, from University of the Punjab, Pakistan. He was awarded Scholarship for Service award (2009–2011) to pursue his M.S. degree in Computational Sciences & Engineering, from Research Center for Modeling & Simulation at National University of Sciences and Technology, Pakistan. He graduated in 2020, with his Doctorate, from University of North Carolina at Charlotte, USA, on a fully funded Ph.D. scholarship. Since 2020, he is serving as an Assistant Professor of Cybersecurity at Mercy College, NY, USA. During his academic journey he has participated and lead several research & educational grants by U.S. agencies.

Fadi Mohsen obtained his BSc degree in Computer Information Systems from the University of Jordan, Jordan. He was awarded the Fulbright Scholarship in 2008 to pursue his MSc in Computer Science at the University of Colorado at Colorado Springs, USA. In 2016, he received his Ph.D. in Computing and Informatics from the University of North Carolina at Charlotte, USA. He is currently an assistant professor at the University of Groningen. His research interests lie in usable security, mobile, and web security, moving target defense, and security analytics.

Zhiyuan Wei received his B.S. degree in computer science and technology from Shanghai JianQiao University, Shanghai, China, in 2021 and the M.S. degree in Cyber security from Mercy College, NY, USA, in 2023. He is currently working as a software engineer at Rocky Mountain Robotech LLC, CO, USA. Since 2022, he has been working as researchers assistant on Insider threat analytics. His research interests include threat analytics, data Science and machine learning.