Bio-inspired Approach to Thwart Against Insider Threats: An Access Control Policy Regulation Framework

Usman Rauf[1](✉), Mohamed Shehab[1], Nafees Qamar[2], and Sheema Sameen[3]

[1] Department of Software and Information Systems,
University of North Carolina at Charlotte, Charlotte, NC, USA
{urauf,mshehab}@uncc.edu, usman.cyberdna@gmail.com

[2] Governors State University, University Park, Chicago, IL, USA
mqamar@govst.edu

[3] IBM T. J. Watson, Yorktown Heights, NY, USA
sheema.sameen@ibm.com

Abstract. With the ever increasing number of insider attacks (data breaches) and security incidents it is evident that the traditional manual and standalone access control models for cyber-security are unable to defend complex and large organizations. The new access control models must focus on auto-resiliency, integration and fast response-time to timely react against insider attacks. To meet these objectives, even after decades of development of cyber-security systems, there still exist inherent limitations (i.e., understanding of behavioral anomalies) in current cyber-security architecture that allow adversaries to not only plan and launch attacks effectively but also learn and evade detection easily. In this research we propose a bio-inspired integrated access control policy regulation framework which not only allows us to understand anomalous behavior of an insider but also provides theoretical background to link behavioral anomalies to the access control regulation. To demonstrate the effectiveness of our proposed framework we use real-life threat dataset for the evaluation purposes.

1 Introduction

The ultimate goal of cyber-security and forensics community is to deal with wide range of cyber threats in (almost) real-time conditions. Whereas in today‘s cyber infrastructures, current Monitoring and Analysis (M&A) technologies (solutions) only address the facet of plethora of problems. The focus of current state of the art is towards developing Security Event Management (SEM) or Security Information and Event Management (SIEM) systems, which can efficiently collect event related information (in the form of logs) from operating systems or network devices (e.g., firewalls). This plethora of information is then used for analytics by a centralized unit for detection of malicious activities via signature-based testing

A. Compagnoni et al. (Eds.): BICT 2019, LNICST 289, pp. 39–57, 2019.
https://doi.org/10.1007/978-3-030-24202-2_4

or correlations between events. Finally, the alarm is generated to update the security personal, who is not only responsible for checking the legitimacy/correctness of alarm, but also have to implement security policies in manual/semi-automatic ways, to cope with real time situations. These technologies (SEM/SIEM) perform well, when it comes to aggregation of information, but due to the inherent limitations, do not provide any hint about ongoing insider attacks and policy synthesis mechanism against a legitimate user turned into a malicious user or insider attacker. These inherent limitations include lack of interaction between detection units and policy synthesis procedures, which results into an inability to synthesize security policies dynamically, via risk assessment or behavioral analysis. Next generation technologies are expected to eradicate these limitations by efficiently integrating modern technologies into a standalone monitoring and detection unit that can be deployed at various nodes in a network. Whereas very little or closer to none efforts are spent in developing technologies that can have the capability to not only detect user/entity abnormal behavior, but also to autonomously react against cyber threats in near real-time conditions using actionable information (threat intelligence).

Biological systems, on the other hand, have intrinsic appealing characteristics as a result of billions of years of evolution, such as adaptivity to varying environmental conditions, inherent resiliency to failures and damages, successful and collaborative operation on the basis of a limited set of rules. Inspired by the nature of cellular regulation mechanism, which helps to maintain an optimal concentration of proteins in a cell via signal transduction mechanism and mitigating the perturbations (insider threats) due to the high/low of productions of certain proteins, we present cellular regulation inspired (integrated) systemic approach through which the security policies can be dynamically altered against an originating threat (via detection and threat analytics). The threats under consideration are behavior anomalies, which make it difficult for the existing technologies (SEM/SIEM) to defend against an insider who is also a legitimate user, as there is no available benchmark or standard, other than guidelines, to differentiate a normal user from an abnormal/anomalous user.

Therefore, our first major contribution is to set forth the criteria to detect behavioral anomalies in near real-time situations. Our second main contribution is to formally model policy regulation problem as state transition system such that the formal tool can be leveraged for policy regulation/synthesis proposes. Finally, our third main contribution involves bio-inspired framework for the integration of threat analytic (for behavioral anomaly) and policy regulation attacks.

2 Related Work

Although the nature of insider threats differs from those of external attacks, the detection techniques can still be characterized in two categories: signature-based, or anomaly-detection methods [26]. Insider threat detection faces unique challenges as compared to challenges faced by detection of external attacks.

This uniqueness is due to many nontechnical factors which contribute towards the change in the behavior of employees, which have legitimate access to the resources with organizational knowledge. Hence, understanding of the change in behavior is of utmost importance to deal with insider threats, given the statistics of insider incidents [9,16].

2.1 Signature-Based Insider Threat Detection Systems

In case of insider attacks the signatures are defined in terms of predefined policies, which trigger alarm once violated, e.g., unauthorized access to a machine or file, for which access policy is predefined. For instance, Agrafiotis et al., develop a tripwire approach to detect actions that are indicators of insider threat based on designed policies on alarming behaviors, and attack-patterns [11]. The authors do not provide any experimental results, or the details about how their proposed approach will reinforce the policy regulation. IBM uses a similar approach as part of the IBM QRadar SIEM solution through the implementation of offences. Offences are designed to detect threats in general and may be used for detecting steps of known insider attacks [1]. Bishop et al., proposed a different approach to detect insider attacks by developing a solution based on the attack trees [13]. The authors build attack graphs to illustrate possible scenarios through which a target can be compromised by an insider. Finally, they determine Minimum Cut Set (MCS) to find out possible countermeasures for an ongoing attack. The proposed approach is highly dependent on the successful design of a process model (attack tree) that identifies the vulnerabilities of the process and possible attack targets. It is also limited to detecting attacks on the proposed targets as only known vulnerabilities can be modeled using attack trees, and the attack tree based approaches do not take into account any change in insider's intention/behavior. The authors also do not discuss how the detection will reinforce policy regulation mechanism.

2.2 Anomaly-Based Insider Threat Detection Systems

These systems are designed to detect unknown types of attacks and behaviors, and trigger an alarm if encounter any deviation. Some detection-solutions consider non-technical indicators of insider threat. Non-technical indicators, such as the psychological state of the insider are of crucial value to insider-threat detection [10,31]. Therefore, work has been done to incorporate their analysis in insider-threat detection systems. For example, Brdiczka et al., used graph anomaly detection and additional techniques to learn the normal behavior of nodes. They also use psychological profiling to take into consideration an insider's intention, which is a non-technical indicator, with the aim of reducing false positives generated by monitoring technical indicators [15]. The authors use gaming community data, World of Warcrafts (WoW) and social network-based activities, to analyze the behavior of a player in a group. The main limitation of the approach is that the testing data and attributes used for analysis have almost no relevance when it comes to insider threats in an organization, since the settings in which individual react in a character gameplay forum are much different than the malicious activities pattern of a legitimate insider. Finally, they do not present any methodology to map measured threat impact to the policy regulation of an organization.

Chen et al., proposed a belief-based threat detection system which also considers the intention of an insider as an indicator [19]. Their solution is designed to estimate the probability of success of an attack by conducting behavioral analysis using probabilistic model checking. Prediction is done after a potential insider has been identified through intentional analysis using Bayesian networks. The major flaw of the approach is the probability distribution calculation, and modeling of individual threat scenarios as Markove Decision Process (MDPs). The approach becomes highly unrealistic since it requires the modeling of each individual and a certain threat to be modelled and analyzed separately. In an organization of thousands of employees, the analysis becomes highly impractical. Second issue is the assignment of probabilities to the transitions in MPD model, which are unrealistically obtained and follow random Bernoulli distribution. The user based technical attributes are also mentioned which can help understanding user's behavior.

Brdiczka et al., and Chen et al., apply automated analysis of non-technical indicators of insider threat requiring the collection of sensitive data, such as the contents of email communications to be used for sentiment analysis [15,19]. Although their proposed methodology promises to deliver high accuracy but the maximum accuracy of detection system is only 82%. Given the low accuracy, technical challenges related to the deployment and lack of understanding of users behavioral impact on policy regulation, makes this approach impractical and an unviable choice.

Some anomaly-detection methods are developed to detect a certain type of insider threat. For example, Zhang et al. [32], propose a solution to analyze document-access behavior to classify users based on the contents of accessed documents. Each user is identified by the type of documents they usually access. Anomaly detection checks for deviations from historical and current behaviors of the user, and the behavior of the community using the Naive Bayes algorithm and correlation matrices. This approach is limited to monitoring a single indicator (type of accessed files), referring to a specific type of insider threat, for instance information leakage. Whereas according to the CERT guide and SANS survey to insider threats combining multiple indicator can provide better detection efficiency [8,16].

Other detection methods aim at detecting threats to a specific resource in an organization. For example, Senator et al., develop a solution to detect threats to database-access behavior. Their solution is an example approach that is limited to protecting a specific resource (corporate database). Although the authors consider multiple indicators for anomaly-detection algorithms to tackle with the low signal-to-noise ratio challenge in insider threat, they do not provide any information about the impact of detection unit's output on policy regulation, and their approach assumes that the actions will be taken by an analyst [37].

Finally, some detection methods are designed to learn a normal behavior of employees from their online activities. For instance, a more relevant work includes the approach presented by Legg et al., [28]. They developed an automated detection system that uses Principle Component Analysis (PCA) to detect anomalies. They compute hourly feature vectors on the activities of employees and build a 24-hour matrix of activities. Then, PCA is applied to project the multivariate vectors into a 2D space based on the maximum variance exhibited by features. Anomaly detection then measures the distance of points in the projected space from the origin. The variance based anomaly-detection method is difficult to interpret as the classification requires predefined threshold, this limits a security analyst's capability of gaining insight on the decision-making process of the method while investigating the generated alarms. Another issue is that the approach clusters the users together, which makes it harder for policy integration, as policy is defined for a malicious activity not a whole group.

Rashid et al., make use of Hidden Markov Models to learn the normal behavior of employees and analyze deviations from the learned behavior to detect insider threat, and consider normality as a sequence of events [33]. The authors highlight that their model offers the advantages of learning parameters from the dataset that describe an employee's behavior. Their model is also advantageous in learning from data that is sequential in nature. However, the computational cost of training the models increases as the number of states captured increases, while the effectiveness of the method in detecting insider threat is highly impacted by the number of states. Although the proposed approach is able to capture the anomalous behavior of an insider, it does not provide how efficiently it can report to an analyst and how the detection results can be mapped to policy regulation. Moreover, Song et al., use Gaussian Mixture Models for modelling the behavior of users for insider threat and masquerade detection. They compare their results with several other machine learning methods based analysis and find it superior in achieving higher accuracy values. However, their model is applied on system-level events, such as process creation, intended for a biometric identification instead of identifying user behavioral anomalies [36].

2.3 Policy Regulation in RBAC

Although RBAC is the most widely used access control architecture which has several benefits, it cannot automatically revoke users' access if they are on the verge of behaving maliciously or not behaving properly. For this reason, several approaches have incorporated the notion of trust in RBAC [17,23,25]. However, existing approaches neither present a comprehensive analysis of the way in which trust thresholds should be assigned, nor specify how to enforce such policies or reduce the risk exposure automatically. In [17], roles are associated with trust intervals, and trust intervals are assigned to users. Users are assigned to roles according to their trust levels. This model does not capture the intuitive nature of RBAC systems in which users are assigned to roles according to their organization's functions, not trust levels. In [25], users are assigned to roles based on trustworthiness and context information. A similar approach was proposed in [18], where role thresholds are a function of the risk of the operations. If the trust of the user offsets the risk of the action, the access is granted. However, none of the existing works provide a clear understanding about trust computation and do not provide any method to reduce the risk that an organization faces at runtime by selecting roles with minimum risk exposure.

In the research proposed by Ma et al. [29], each role is assigned a minimum level of confidence and each user is assigned a clearance level. Based on these values, the risk associated with a user activating a role is calculated. Objects and actions are assigned a value according to their importance and criticality. However, this work does not mitigate insider threats as the trustworthiness of users is defined as a static parameter that does not depend on users' behavior. In addition, the authors do not consider role hierarchy in their work and do not present experimental results [29].

In [12,18,30], the main focus is also to reduce the risk exposure. In [30], a risk based analysis is proposed to ensure that system administrators assign permissions to the roles considering the risk inherent to those permissions. Each permission is assigned a risk value, and the role hierarchy is organized based on these risk values. This may not be appropriate, as it is more intuitive to organize the role hierarchy according to the employee's structure. We argue that maintaining a role hierarchy that matches the organization's hierarchy is more intuitive for security administrators. Additionally, this work does not reduce the risk exposure of the organization during the role activation process.

In [12], a model that modifies the policy to minimize the risk exposure as systems evolve is proposed. The risk is considered as a parameter which varies in an interval over $[t, t') \in \mathbb{R}$. User is assigned obligations on the basis of assessed risk. The approach do not provide any understanding about risk calculation, behavioral anomaly, and obligation fulfillment check. Since, the proposed system can assume any arbitrary state given the values of risk and obligation attributes, it makes management (adding, removing or updating policies) much more harder for an administrator, as there is no way to comprehend underlying state of the system, making it cumbersome to modify it and prone to errors.

Chen et al., propose a model in which the risk associated with a role is calculated using the trustworthiness of the user, the degree of competence that a user has to activate a role, and the degree of appropriateness of the permission-role assignments. Each permission is assigned a mitigation strategy, which is a list of risk thresholds and an associated obligation pair [18]. When the user wants to obtain a set of permissions, the role with minimum risk is selected. Then, the system consults the mitigation strategy to see which action is more appropriate: to deny the access or to allow the access imposing an obligation. The authors, do not account for the context as an important component to define the risk threshold that should be enforced.

Salim et al., propose to assign costs of access to permissions depending on the risk of their operations, and to assign to each user a budget [35]. Users are assigned roles, but being assigned or not does not necessarily determine whether or not a user should be allowed to activate a role. If the user accesses permissions that he/she can obtain through an authorized role, the cost is reduced. In case the user is not authorized to a role, the cost of activating the role is taxed. Nonetheless, if the user has enough budget to make the operation, he/she can access the permissions. The authors claim that this mechanism incentivizes users to spend their budget cautiously, activating low cost (low risk) roles. However, this scheme exacerbates the risk of insider threats. Users can use their budget to perform unauthorized accesses without being detected; e.g., if a disgruntled employee wants to quit the organization, he/she would not mind expending all his budget performing a malicious action.

Many commercial products also incorporate risk in their solutions; e.g., SAP [3], Oracle [4], IBM [6] 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 [14], when the user is about to infer some unauthorized information, the system prevents it by either denying access or providing scrambled data. This is not adequate for all types of organizations. We believe that real organizations may need to provide access to multiple pieces of information to a single employee even if they result in undesirable inference. Existing RBAC extensions do not consider the risk of inferred information. New ways to mitigate the inference risk in RBAC-based systems are needed and this forms our motivation as in this article we intend to develop a integrated detection and response systems which can make use of threat intelligence to autonomously regulate access control policies. Towards this direction, in the next section we discuss our (bio) inspiration system in details.

3 Cellular Regulation via Signal Transduction

Every functionality in the human body and evolution of the morphological features is highly influenced or controlled at (intracellular) molecular level [20,34,38]. Genes and proteins are the main ingredient of this controlling mechanism, which play together in a programmed manner to perform multiple tasks in an organism. Genes are the informative subunits of DNA and they decode instructions in the form of proteins. When a gene is switched on, information flows from genetic to proteomic level as complex processes of transcription and translation. Some proteins have the function of regulating the expression of genes by turning them on or off. This complex interactions of genes and proteins to regulate the cell against any external or internal threat/perturbation after receiving signals from cellular receptors is referred as Cellular Regulation via Signal Transduction [34]. Regardless of where the control comes from, whether its hard coded in DNA or nucleolus of the cell, in cellular regulation, the key principle is the regulation of different parts of DNA (resulting in controlled synthesis of different proteins) against any uprising threat at cellular level.

As a first step towards creating bio-inspired resilient architecture, we intend to understand how this feedback notion works and can help us in accomplishing our objectives. In the next section, we present a real life biological phenomenon for the better understanding of readers.

3.1 Blood Pressure Regulation System

Renin angiotensin system (RAS) plays a crucial role in physiological functioning of human body by regulating blood pressure. This hormone control system is triggered to avoid the drop of blood pressure towards some critical life threatening level in different stress conditions e.g., dehydration and hemorrhage.

In human body the decrease in blood pressure is primarily sensed by specialized cells in kidneys which increase the production of Renin enzyme as a consequence. Renin catalyzes a protein called Angiotensinogen, which is produced by liver, into another protein angiotensin I. Angiotensin I is further converted in to Angiotensin II by angiotensin converting enzyme (ACE). Angiotensin II is the main product of RAS system which increases blood pressure by a triple action plan: (1) it constricts blood vessels in kidneys by contraction of smooth muscle, cells (2) it enhance the production of aldosterone hormone which helps in Na+ retention in kidneys, and (3) triggers the production of vasopressin hormone in the brain. All these three actions performed by angiotensin II are essential for blood regulation in body.

The angiotensin II protein performs all three tasks by first binding to the receptors of target cells. The binding of angiotensin II with receptors trigger cascade of biochemical reactions which result in aforementioned tasks responsible for elevation of blood pressure. The reaction stops eventually when all the receptors are bound by the protein. The rennin secretion also stops due to increase in blood pressure up to normal levels and hence it also blocks the conversion of angiotensinogen to angiotensin II.

The cause for hypertension or high blood pressure is hidden somewhere in the RAS system. As the angiotension II is the key functional element of this system so most of the therapies are designed to control this protein by blocking its activity which is done by blocking of ACE enzyme, responsible for the conversion of angiotensin II from angiotensin I. The drugs targeting ACE, also known as ACE inhibitors, has shown promising results in therapy of high blood pressure disease. For detailed information and discussion about RAS, we divert our readers to the article presented by Dressler [24]. In the next section we summarize the working procedure of cellular regulation in the form of a framework, and propose similar framework for regulation of insider threats, inspired by the phenomenon of cellular regulation.

4 Cellular Regulation Inspired Mapping and Proposed Framework

To proceed further with the idea of integrating auto-resiliency characteristics of biological systems in current cyber architecture, there must exist some analogy and mapping of actions among fundamental entities of both domains. Table 1 illustrates mapping of cellular regulation principles towards cyber-security concepts. This also forms the basis of our proposed framework (c.f. Fig. 1).

Table 1. Analogy between cellular regulation and cyber-security

Biological characteristics	Cyber characteristics
Fundamentals	
DNA: is a conjunction of regions, which are individually represented as genes. These regions, under any perturbation, are regulated (turned on/off) via complex processes to neutralize any threat that cells may face	Cyber Policies, on the other hand, are conjunction of rules, which can be added or removed (activated or turned off), to deal with any potential threat to an organization
Sensing Mechanism	
At cellular level, a cell's receptors are responsible for receiving signals about ongoing activities and traverse them within a cell. Nature has fine tuned sensing mechanism through which elements within a cell are made aware about the changing environment around them	Log collection mechanisms in cyber domain, on the other hand, can be tuned to collect information about users and their activities at system/network level in real time. These mechanisms (if implemented in real time) can perfectly mimic and correspond to sensing mechanism in cellular regulation
Actuation/Regulation	
At cellular level, concentration of a protein is changed via turning on/off its regulating genes (regions of DNA), against any perturbation	If we consider policy as a disjunctive conjunction of all possible rules, then dynamic selection of a suitable subset of rules can be referred as turning on/off regions of a policy, to make it more appropriate according to the current scenario and perfectly corresponds to the regulation at cellular level

In the next step, we propose cellular regulation inspired framework to deal with insider threats. As we established earlier, although the cellular regulation process encompasses and neutralizes both internal and external threats, but our focus in this article is toward developing an architecture which can deal with insider threats. The motivation of choice is due to the limitation in availability of the testing datasets. Although with minor modification the proposed framework can also be extended to deal with external threats, but we limit our focus to

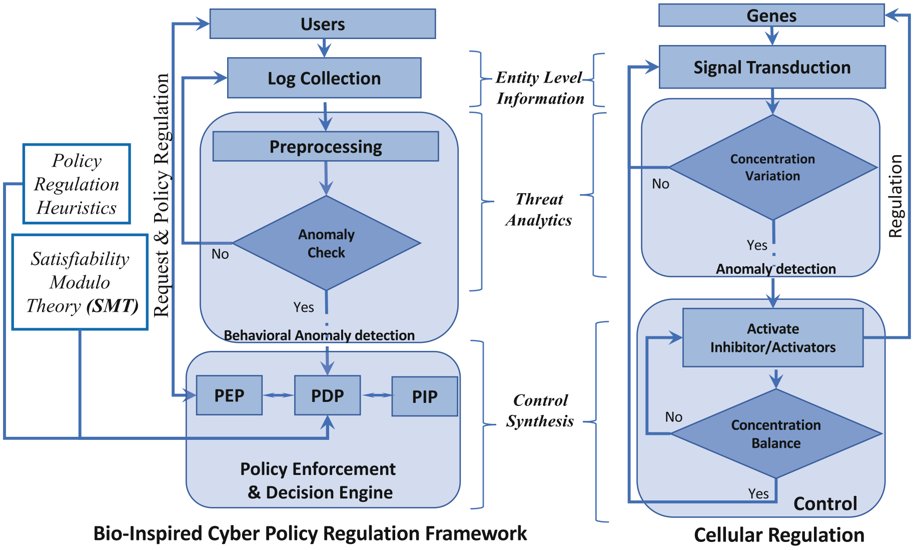


Fig. 1. Mapping of cellular regulation mechanism to the proposed cyber policy regulation framework

the internal/insider threats only, as we can observe the variations in insider's behavior via activity log collection. Figure 1 presents a detailed description of our proposed framework along with side by side comparison to the working principles of Cellular regulation process. We discuss each component in details in the forthcoming discussion.

4.1 Sensing Module

As we established earlier that in cellular regulation, sensing is performed by cell's receptors, which monitor continuously the concentration levels of proteins, and disperse that information internally (within a cell), so that various genes can be made aware of the perturbations in their environment. In our case we propose to collect event logs of user's activities, and pass them to a central entity (Threat Analytics Module) for threat analysis. Rather than dispersing information to every entity in the system (as in case of cellular regulation), we propose to collect logs in centralized manner, as our control (policy regulation) originates from central authority (Policy Enforcement & Decision Engine).

Although the log collection process for threat analytic is not a novel concept, but it exists in individuality and in this research we propose to integrate it with access control mechanism by following the principles of cellular regulation. Most of the recent recommendations propose active log collection mechanisms but none of them provide any guidelines about how to make activity logs information meaningful and usable for threat analytic using machine learning methods.

4.2 Threat Analytic Module

Feature Engineering: As a first step towards threat analytics, one of the main contributions of this article is to setup a criteria for pre-processing/feature-engineering of the data (logs) collected from the sensing module. Event related data is stored in the logs with time stamps, identifying activity performed by an individual at certain time. Activities under consideration can vary depending on organizational needs, but most generic activities include: login/logoff details, web surfing, use of plugin devices, and duration of specific activities.

The first challenge is to convert these event based time-labeled data in to a meaning full dataset. Which can be utilized by machine learning methods. As a first step we convert time-stamps into a cyclic temporal variable which varies between 0–24 h. This makes it easier to assign a numeric value to any event and allows machine learning classifiers to learn and tie an activity with a number rather than an uninterpretable string (date and time). The second main issue we address during feature engineering phase is the separation of each individuals data, so that we can train classifiers over an individual's behavior. The sample of the updated dataset with temporal encoded information can be found in [2].

Possibility of being able to learn an individual's behavior (from activity logs) allows us to measure the variation in it as well. Once the deviation of an individual from its own (and others) is predictable, we can easily compare predictions with ongoing activities. For instance, if an employee logs in during a certain time window over a course of time, then using machine learning methods, classifiers can be trained to learn about the login time window slot and predict in which time slot an individual mostly/normally login and starts working. We use OneHotEncoding method to encode the employees data and separate each individual's data from the log files. Finally we combine all employees data to construct a dataset which is finally usable by machine learning methods [2].

Anomaly Check: Having an abnormal work routine do not refer to an anomaly. Therefore, we do not consider abnormality as an anomaly. An individual may have different work patterns, and may be abnormal but as far as the high risk activities and checks are not triggered, we consider a user to be just abnormal and not anomalous. The best way to find out anomaly in this situation is to compare an individual's profile with its own working routine and observe significant and abrupt variation. For instance, an employee has not used SSH connection to a secure data repository at midnight, in the recent month, but suddenly has established SSH connection and is trying to upload data to a remote server around mid-night. We consider this type of variation in behavior as anomalies. In our proposed framework we use machine learning to train models against a user's profile (long term or short term behavior), and then predict there activities every time they try to access resources. If our predictions do not match the ongoing activities, threat analytic module considers it as user behavior anomaly and reports it to the Policy Enforcement & Decision Engine. We discuss the details of accuracy of our behavioral anomaly detection unit in the forthcoming Sect. 5.

4.3 Policy Regulation Module (PRM)

The third and most important component of our proposed *Bio-inspired Policy Regulation Framework* is *Policy Enforcement & Decision Engine*. It consists of three sub-modules: Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Information Point (PIP). PIP contains information about organizational policies (e.g. given user attributes what level of access can be granted). PEP receives user's attribute and passes it onto PDP. PDP utilizes PIP knowledge-base as a reference point and makes decision about granting/revoking access against a certain user/insider. We propose to integrate PDP with behavioral anomaly detection unit, so that they can operate autonomously, and regulate access control without human intervention.

In order to work autonomously, PDP requires threat information regarding an insider/user which generates access request and on this basis it decides whether or not the access should be regulated. There are multiple ways to implement this integration. (1) PDP can inquire about a user's threat level from anomaly detection module, or (2) anomoly detection unit can update threat levels and push these details into the PDP module. Since, detection and regulation modules are working independently (other than the dependency of PDP for threat levels), we propose the later option, to avoid any excess overhead. Although, PDP has the tendency to make decision about access control, but the current architecture of PDP lack the notion of understanding about threat and synthesizing access control against it. Towards this direction we formalize access control problem as constraint satisfaction problem, and use Satisfiability Modulo Theory (SMT) solver to solve it [21]. Use SMT allows us to enhance the capability of PDP to understand notion of threat and solve the problem of access regulation using constraint satisfaction concepts.

Before allowing access to a user's request, PRM checks if the threat level has altered or not. If threat level changes the decision engine either revokes or limits the access of a user/insider, according to the organizational requirements. Decision engine can then be integrated with the policy enforcement module to enforce the policy.

We formally define Policy Regulation Module as a transition system. A *Policy Regulation Transition System (PRTS)* can be defined as a *8-tuple*:
$\mathcal{M} = (S, s_0, u_i^r, A_j^c, P_i, \mathcal{T}_{ij}^k, \hookrightarrow, \delta)$:

- S is a finite set of states of a policy (possible configurations) with cardinality in $\mathbb{N}$;
- $s_0 \in$ S is the initial/current state/configuration of policy;
- u_i^r is the rank of user i with values in $\mathbb{N}$;
- A_j^c is the confidentiality level of asset j with values in $\mathbb{N}$;
- $\mathcal{T}_{ij}^k$ is the threat level/impact of a given request by user i to access asset j, which can be calculated as:

$$\mathcal{T}_{ij}^k = L_i \times Imp_j$$

  whereas, L_i represents the likelihood of behavioral anomaly for a give user (i), and Imp_j is the impact if the asset (j) is being compromised.

- dec_k^{ij} is decision variable in $\mathbb{B}$, which helps SMT finding a new state for transition;
- $\mathrm{C}(\mathcal{T}_{ij}^k)$ is a set of constraints over the threat vector/values;
- $\hookrightarrow$ is a finite set of transitions such that: $\hookrightarrow \subseteq (S \times \mathbb{N} \times \mathbb{N} \times \mathbb{B})^2 \times \mathrm{C}(\mathcal{T}_{ij}^k)$;
- δ is a finite set of transition rules which maps $\mathrm{C}(\mathcal{T}_{ij}^k)$ to set of transition $\hookrightarrow$;

We define security policy as disjunctive conjunction of rules, and rules (r_{ij}) contain information about user (u_i), and requested asset (A_j).

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

The above mentioned expression shows the assumption that there exist all possible combination of the rules (we call configurations of a policy) in the knowledgebase, and given this assumption PRTS can switch configuration under the effect of information provided by Threat Analytics Module as per the following semantics.

$$(s, u_i^r, A_j^c, \mathcal{T}_{ij}^k, dec_k^{ij}) \xrightarrow{\mathcal{T}_{ij}^k >= \theta; dec_k^{ij} = 0} (s', u_i^r, A_j^c, \mathcal{T}_{ij}^k, dec_k^{ij'})$$

$$(s, u_i^r, A_j^c, \mathcal{T}_{ij}^k, dec_k^{ij}) \xrightarrow{\mathcal{T}_{ij}^k < \theta; dec_k^{ij} = 1} (s'', u_i^r, A_j^c, \mathcal{T}_{ij}^k, dec_k^{ij''})$$

The above mentioned formalism defines the semantics of our proposed PRTS. Constraints over $(\mathcal{T})$ work as guards over transitions $\hookrightarrow \subseteq (S \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{B})^2$. The transition from one configuration to another configuration is only fired once the guards evaluate to true and invariants are violated (threat level of a user increases). Whereas, s, s' and s'' are distinct states such that s and $s' \in S$ and $s \cap s' : \emptyset$. If the threat is below a transition triggering threshold s and s'' may or may not be the same. Once the threat impact is evaluated by detection unit, the new configuration is selected by SMT solver by solving the following constraint:

$$\exists_{i,j,k} \left(\bigvee_{k:1}^{n}((\bigwedge_{i,j:1}^{m}(r_{ij})) \bigwedge (dec_k^{ij})) \right) whereas, i, j, k \in \mathbb{N} \qquad (1)$$

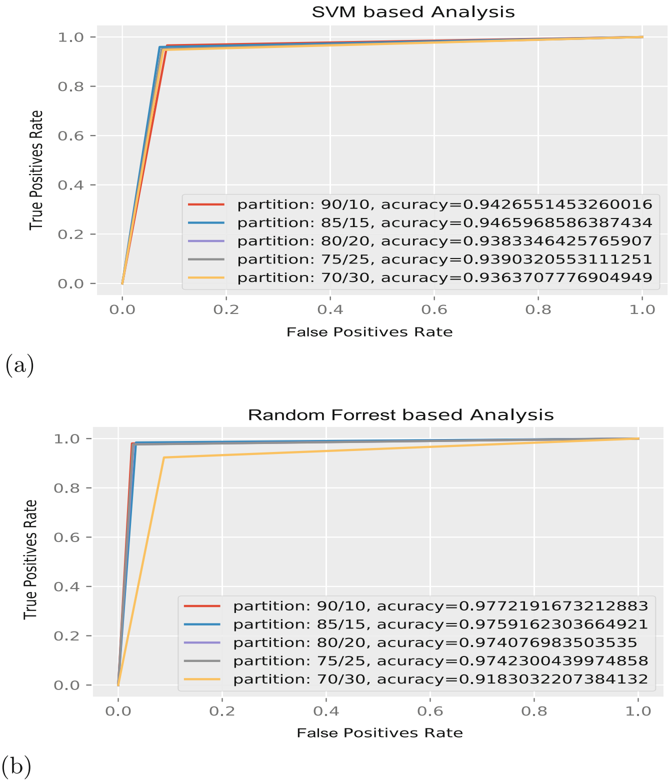
The above expression only finds the configuration of the policy in which dec_k^{ij} is true or 1 (as it is a binary decision variable), which means allowing only the configurations in which the threat is under acceptable threshold and values of i,j,k does not necessarily have to be equal. For instance given a current state s if the value of $\mathcal{T}_{ij}^k$ becomes higher than the acceptable threshold (θ), as per the analysis provided by detection module, transition will occur as per the above mentioned transition semantics, and new state will be selected by solving the above mentioned constraint 1. In the following section we discuss the evaluation and effectiveness of our approach.

5 Evaluation of Bio Inspired Policy Regulation Framework

5.1 Effectiveness of Behavioral Anomaly Detection Unit

To measure the accuracy of our behavioral anomaly detection unit, we use threat test data set released by CERT in 2016 [7,27]. The dataset contains the log activities for one thousand employees and contains five different types of insider threat scenarios, for detailed description of the scenarios we refer our reader to the dataset details (*file: scenario.txt*) in [7] . In the context of this paper, we only focus on the first scenario, and use information of the dataset which is relevant to this scenario. Although our approach can be used to deal with other scenarios, we only use scenario one for testing and evaluation purposes.

Scenario: *User who did not previously use removable drives or has variable routine begins logging in different hours, using a removable drive, and uploading data to a listed malicious website.* After prepossessing of the provided user activity logs



(a)

(b)

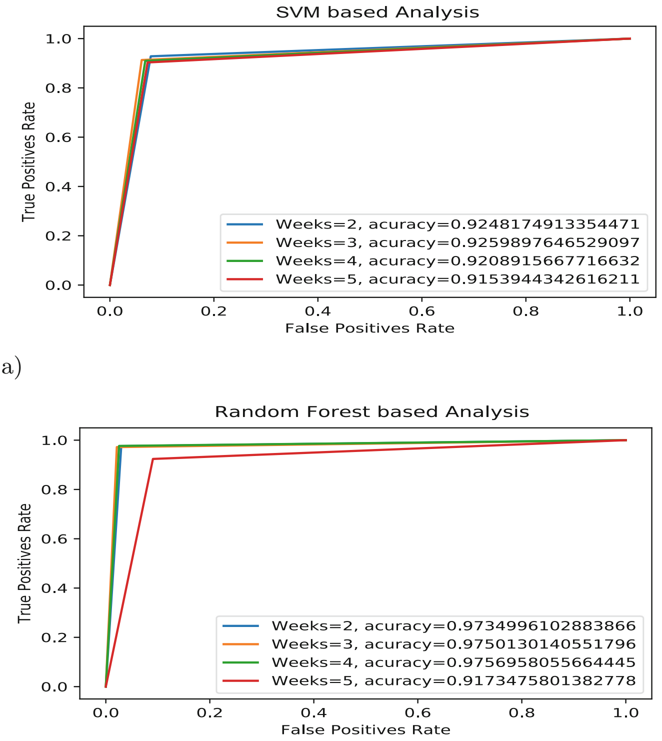
Fig. 2. Evaluation of train-test sample size partitioning to find optimized partitioning size

[7], we construct our own threat test dataset which can be accessed and used for machine learning purposes [2]. As we mentioned earlier we use OneHotEncoding so that the information regarding employees which are to be considered for analysis should be in separate columns as depicted in our processed dataset. Each attribute, for example login time of a day, and activity performed (use of external hard drive, or visit to malicious website) are represented in separate columns. Values of time attribute are mapped between 0–24 h range, whereas the values of activities are binary, representing "1" if an activity is triggered by a certain employee at a certain time, and "0" otherwise.

We use state of the art Random Forrest and SVM methods for predictions about insider's behavior. Following RoC curves and accuracy labels show that we were able to achieve almost 98% accuracy. Which means we were able to predict the behavior of an insider with high accuracy. In data science and machine learning the accuracy of analysis is highly dependent on two factor, (1) partitioning size of dataset while training the prediction classifier and (2) temporal variation in data sample size (e.g. variation in number of weeks). We vary both of these factors to observe the impact and find the optimized values for number of weeks to be considered for effective predictions and optimized size of partitions for training-and-test purposes.

Effect of Variations in Train-Test Partitioning Size. RoC curves in Fig. 2 show how the effectiveness of our behavioral detection unit varies with the variation in partitioning. We deduce from our analysis that Random forest based predictions were more accurate than SVM based predictions, and we were able to achieve ≈98% accuracy. We also observe ideal cutoff point for the train-test split to be 75%/25% by variation of train-test split ratios, since we do not observe any significant improvements by increasing the training set size.

Effect of Temporal Variations. Figure 3 shows how the effectiveness of behavioral detection unit varies with Temporal data size variation. For instance, given a general perception that having large data size or training a model over data dispersed over larger period of time helps to increase the accuracy of the results. We find it contradicting in case of behavioral anomaly problem. Our results show that the effectiveness of behavioral anomaly detection decreases rapidly if we train our model over the logs of larger period of time. Which means if we consider two week's logs (of an employee) for training and prediction, the accuracy will be higher than the scenarios in which we consider the logs of five weeks. We believe the degradation in the effectiveness is due to the over approximation of the models leading to higher false positive values. This type on analysis, helps us setting up a benchmark over the estimation and prediction parameters. Hence, we deduce that 2-to-3 weeks logs are ideal for training and prediction purposes.



(a)

(b)

Fig. 3. Temporal evaluation to find optimal number of weeks for prediction analysis

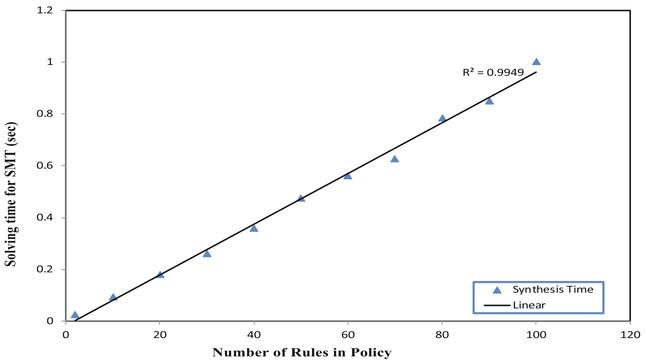


Fig. 4. SMT time to calculate satisfiable configuration of security policy

5.2 Efficiency of Policy Regulation Module

We also conduct a preliminary evaluation of the proposed *Policy Regulation Module* to assess its time complexity against number of rules in a policy, given the information from *Threat Analytic Module*. To compute satisfiable configuration of access control policy, we use Z3 SMT solver [22]. All the experiments are conducted on Core i5 machine with 2.4 GHz processor and 16Gb of memory. The result presented in the Fig. 4 shows how the time require to compute satisfiable configuration varies with the increase in the number of rules (in a policy). The result shows that the complexity is near to linear. In future, we aim to conduct extensive analysis about deployment of *Policy Regulation Engine* on larger scale.

6 Conclusion and Future Work

In this paper, we present a novel *Cellular Regulation inspired Access Control Policy Regulation Framework*, which not only observes the variation in the environment (behavioral anomalies of insiders), but also triggers necessary responses (policy regulation) to secure the system. Our first main contribution is to setup a feature (activity) based temporal criteria for understanding behavioral anomalies. The proposed integrated framework then utilizes the state of the art machine learning methods for the detection of behavioral anomalies. Our third main contribution is to model policy regulation problem as state transition system, which can utilize the results from behavioral anomaly module to refine the access control policy. We evaluate the efficiency and effectiveness of our proposed framework on real-life threat dataset provided by CERT. Our evaluation illustrates that we were able to achieve the accuracy of 98% (92% in worst case scenario) in case of behavioral anomaly detection. The evaluation also shows that we were able to synthesize the regulated policy in linear time for small set of rules. To the best of our knowledge, this is the first effort towards dealing with insider threats by unifying detection and deterrence systems. In future we aim to test our proposed system on medium to large scale examples for rigorous evaluation and incorporate more parameters for behavioral prediction to achieve high accuracy.

References

1. IBM QRadar, SIEM
2. www.dropbox.com/s/rerwekvuji12icm/logon_hotencoded_cleaned_data.csv?dl=0
3. Access risk management. Technical report (2012)
4. Application access controls Governor. Technical report (2012)
5. Identity and access Governance. Technical report (2012)
6. Resource access control facility (RACF). Technical report (2012)
7. CERT threat test dataset. CERT (2016)
8. Defending against the wrong enemy. Technical report, SANS Insider Threat Survey (2017)
9. Insider threat report. Technical report, CA Technologies (2018)

10. McCormac, A., Parsons, K., Butavicius, M.: Preventing and profiling malicious insider attacks. Technical report, Defense Science and Technology Organization, April 2012
11. Agrafiotis, I., Erola, A., Goldsmith, M., Creese, S.: A tripwire grammar for insider threat detection. In: Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, MIST 2016, pp. 105–108. ACM (2016)
12. Aziz, B., Foley, S.N., Herbert, J., Swart, G.: Reconfiguring role based access control policies using risk semantics. J. High Speed Netw. **15**(3), 261–273 (2006)
13. Bishop, M., et al.: Insider threat identification by process analysis. In: 2014 IEEE Security and Privacy Workshops, pp. 251–264, May 2014
14. Biskup, J.: History-dependent inference control of queries by dynamic policy adaption. In: Li, Y. (ed.) DBSec 2011. LNCS, vol. 6818, pp. 106–121. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22348-8_10
15. Brdiczka, O., et al.: Proactive insider threat detection through graph learning and psychological context. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW), pp. 142–149 (2012)
16. Cappelli, D.M., Moore, A.P., Trzeciak, R.F.: The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud). Addison-Wesley, Boston (2012)
17. Chakraborty, S., Ray, I.: 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, SACMAT 2006, New York, NY, USA, pp. 49–58. ACM (2006)
18. Chen, L., Crampton, J.: Risk-aware role-based access control. In: Meadows, C., Fernandez-Gago, C. (eds.) STM 2011. LNCS, vol. 7170, pp. 140–156. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29963-6_11
19. Chen, T., Kammüller, F., Nemli, I., Probst, C.W.: A probabilistic analysis framework for malicious insider threats. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2015. LNCS, vol. 9190, pp. 178–189. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20376-8_16
20. Davidson, E.H., Erwin, D.H.: Gene regulatory networks and the evolution of animal body plans. Science **311**(5762), 796–800 (2006)
21. Davis, M., Putnam, H.: A computing procedure for quantification theory. J. ACM **7**(3), 201–215 (1960)
22. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78800-3_24
23. Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., Moody, K.: Using trust and risk in role-based access control policies. In: Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, SACMAT 2004, New York, NY, USA, pp. 156–162. ACM (2004)
24. Dressler, F.: Self-organized network security facilities based on bio-inspired promoters and inhibitors. In: Dressler, F., Carreras, I. (eds.) Advances in Biologically Inspired Information Systems, pp. 81–98. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72693-7_5
25. Feng, F., Lin, C., Peng, D., Li, J.: 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, September 2008
26. Gheyas, I.A., Abdallah, A.E.: Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. Big Data Anal. **1**(1), 6 (2016)

27. Glasser, J., Lindauer, B. : Bridging the gap: a pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops, pp. 98–104, May 2013
28. Legg, P.A., Buckley, O., Goldsmith, M., Creese, S.: Automated insider threat detection system using user and role-based profile assessment. IEEE Syst. J. **11**(2), 503–512 (2017)
29. Ma, J., Adi, K., Mejri, M., Logrippo, L.: Risk analysis in access control systems. In: 2010 Eighth International Conference on Privacy, Security and Trust, pp. 160–166, Aug 2010
30. Nissanke, N., Khayat, E.J.: Risk based security analysis of permissions in RBAC. In: WOSIS (2004)
31. Nurse, J.R.C., et al.: Understanding insider threat: a framework for characterising attacks. In: 2014 IEEE Security and Privacy Workshops, pp. 214–228, May 2014
32. Zhang, R., Chen, X., Shi, J., Xu, F., Pu, Y.: Detecting insider threat based on document access behavior analysis. In: Han, W., Huang, Z., Hu, C., Zhang, H., Guo, L. (eds.) APWeb 2014. LNCS, vol. 8710, pp. 376–387. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11119-3_35
33. Rashid, T., Agrafiotis, I., Nurse, J.R.C.: 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, MIST 2016, New York, NY, USA, pp. 47–56. ACM (2016)
34. Rauf, U.: A taxonomy of bio-inspired cyber security approaches: existing techniques and future directions. Arab. J. Sci. Eng. **43**, 6693–6708 (2018)
35. Salim, F., Reid, J., Dawson, E., Dulleck, U.: An approach to access control under uncertainty. In: 2011 Sixth International Conference on Availability, Reliability and Security, pp. 1–8, August 2011
36. Song, Y., Salem, M.B., Hershkop, S., Stolfo, S.J.: System level user behavior biometrics using Fisher features and Gaussian mixture models. In: 2013 IEEE Security and Privacy Workshops, pp. 52–59, May 2013
37. Ted, E., et al. 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 (2013)
38. Thomas, L.C., d'Ari, R.: Biological feedback. CRC Press, Boca Raton (1990)