# BIO-INSPIRED CYBER SECURITY AND THREAT ANALYTICS

by

Usman Rauf

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing & Information Systems

Charlotte

2020

Approved by:

Dr. Mohamed Shehab

Dr. Nafees Qamar

Dr. Wiechao Wang

Dr. Cem Saydam

# ABSTRACT

USMAN RAUF. Bio-Inspired Cyber Security and Threat Analytics. (Under the direction of DR. MOHAMED SHEHAB)

After decades of deploying cyber-security systems, it is a well-known fact that the existing cyber infrastructure has numerous inherent limitations that not only make the maintenance of the current network security devices difficult but also provide the adversary with asymmetric advantages. These limitations include: (1) inability to propagate threat related information due to the lack of mutual interactions among network devices/controllers, (2) absence of self-awareness (for behavioral anomaly and threat analytics) in current architecture of cyber elements, (3) Lack of self-correcting control mechanisms; for instance, error-prone and time-consuming manual configuration methods, which is not effective in real-time threat mitigation, and (4) inability to diagnose misconfiguration (i.e., access control conflicts due to multiparty management). These inherent limitations give rise to such vulnerabilities (i.e., inability to deal with stealthy DDoS attacks, and insider threats) which shift the scale of cyber-warfare in the favor of adversary.

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 with global intelligence. In this thesis, to deal with aforementioned issues, we aim to develop novel bio-inspired auto-resilient and self-correcting security architecture for real time threat deterrence and attack mitigation. The main questions we aim to address are: (1) investigation of the laws governing resilience and robustness in biological systems (at cellular and genetic level) and studying their applicability to cyber infrastructures, (2) design and implementa-

tion of novel nature inspired self-aware, and self-correcting access control and routing architectures. (3) Integration of actionable decision module for threat intelligence for real-time threat deterrence/mitigation and anomalous behavior detection, and (4) verification and evaluation of the real scenarios, to prove the correctness and viability of the proposed approaches.

# ACKNOWLEDGEMENTS

Firstly, I would like to express my utmost gratitude towards my department chair, Prof. Mary Lou Maher, for her continuous moral support, guidance, and patience throughout my stay at UNC Charlotte. I am also greatly indebted to my adviser, Dr. Mohamed Shehab, for allowing me to work under his supervision and his valuable support during this journey. I would also like to offer my sincere gratitude to my co-advisers, Dr. Nafees Qamar and Prof. Wojciech Mazurczyk (Warsaw University of Technology), for their collaboration and critical analysis of my research. Finally, I would like to thank my wife, Dr. Sheema Sameen, for her academic and emotional support throughout all the ups and downs during this phase of my life.

# DEDICATION

***"Dedicated to my parents, my wife, and my daughter"***

TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

With the ever increasing number of data breaches and security incidents, it is evident that the traditional existing cyber-security architectures are unable to defend complex breaches and large scale cyber attacks [55][75][73]. The new models of defense need to focus on auto-resiliency, integration and fast response-time. To meet these objectives, even after decades of development of cyber security systems, there still exist inherent limitations in current cyber-security architecture that allow adversaries to not only plan and launch attacks effectively but also learn and evade detection easily.

The ultimate goal of Cyber security and forensics community is to deal with wide spectrum of cyber threats in (almost) real-time conditions, ranging from infrastructure level DDoS attacks to insider attacks. According to recently published reports of Ponemon institute, Neustar and Verizon Enterprize Solusions, the average value of these attacks have doubled since last few years (2016), and the average losses (of medium level enterprise) from DDoS attacks rises to over $2.5 million in 2017, and average cost from inside breaches is $206,000/incident, $4.3 million for medium scale organization and $7.3 million for large enterprises [49][63][73]. This calls for an immediate need to develop tools and techniques which can efficiently thwart infrastructure level DDoS attacks and deal with insiders anomalous behavior.

If we take a deep look into nature, we will discover that every biological system have intrinsic resilient features which helps it to deal with wide range of external or internal threats in an autonomous way. For instance, (1) as a consequence of evolution of the immune system and the changes that the physiology of living creatures has undergone over the course of centuries, biological systems are resilient and robust in

a true sense. Human physiology (specially, immune system) has been behaving in a reactive and adaptive manner since the beginning, as a result of which life expectancy has increased drastically compared to the previous generations. (2) The information at organism and cellular level is processed in a parallel and distributed manner, without any existence of central control. Consequently, an entire organism is regulated autonomously to maintain relatively stable/normal behavior via major functionality i.e., homeostasis, for the operation of vital functions without any intervention of a central biological controller [27]. (3) In social insect colonies, such as ants and bees, a large number of relatively simple individuals manage to build intricate nests or find the shortest path between the nest and a food source. The task allocation process to achieve such a complex goal in these societies is performed collaboratively, which shows how overall task (under the response of each individual) is optimized towards a goal as a result of global intelligence comprised of simple/small, but yet, important individual responses [25].

At the same time current cyber infrastructure and it's management is becoming more complex and challenging [22]. Major limitations in the existing architectures are the absence of self-awareness (against detecting insider threats), and self-correcting features (for auto-resiliency against infrastructure level attacks). In spite of these challenges, the security community is proposing and developing astonishing solutions inspired by self-organizing biological mechanisms which exist inherently in nature.

## 1.1 Contributions

With the above mentioned motivation, the primary objective of this thesis is to design nature inspired solutions to deal with the aforementioned challenges i.e., infrastructure level attacks and insider threats. Towards this direction, our very first contribution is to extensively survey the bio-insoired cyber security domain based on meta analysis (surveying technique). This allows us to understand the underlying potential of a certain natural phenomenon, and discover its relevance and applicability

to the problems in hand. Our second contribution is to formally map, design, implement and evaluate genetic mutation inspired routing mechanism to defend against infrastructure level attacks. Our third contribution in the context of the thesis is to propose the design of an integrated policy regulation framework, inspired by genetic regulation at cellular levels. Our fourth contribution involves the formal modeling and analysis of policy regulation mechanism as state transition system. Rest of the proposal is organized in the following way. In Chapter 2, we provide background about bio-inspired cyber security and discuss the findings of our survey in a brief manner. In Chapter 3, we discuss the problem of *Infrastructure level stealthy DDoS attacks* in details, provide our motivation and objectives, leading to our genetic mutation inspired proposed approach to deal with stealthy DDoS attacks. In the last part of Chapter 3, we provide emulation and simulation based testing and evaluation of the proposed approach. Finally, in Chapter 4, we discuss the problem of *Insider Threats*, existing solutions, and propose a cellular regulation inspired integrated framework to efficiently detect behavioral anomalies and deal with insider threats along with rigorous testing and evaluation using real life CERT threat test dataset.

# CHAPTER 2: Background on Bio-Inspired Cyber Security

There exist three major areas for bio-inspired research (c.f. Figure 2.1): (1) Bio-Inspired Networking is a class of approaches for managing large networks with efficiency and scalability under uncertain conditions (e.g., for autonomous organization with large distributed systems), (2) Bio-Inspired Systems comprise a class of architectures for distributed and collaborative systems (e.g., for distributed sensing and exploration), and (3) Bio-Inspired Computing constitutes a class of algorithm which is solely focused towards efficient computing (e.g., for optimization processes and pattern recognition). Our main focus is to design bio-inspired solutions to the problems

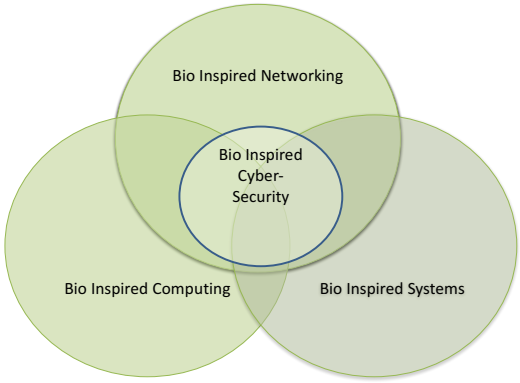



Figure 2.1: Categorization of Bio-Inspired Research and Area of Focus: Bio-Inspired Cyber-Security

of resilience and self-correction in cyber architectures, which is a sub-domain of bio-inspired networking. Although there exist many challenges to overcome the security problems that current cyber infrastructures are facing today, but in the scope of this thesis we aim to only address the (aforementioned) relevant issues.

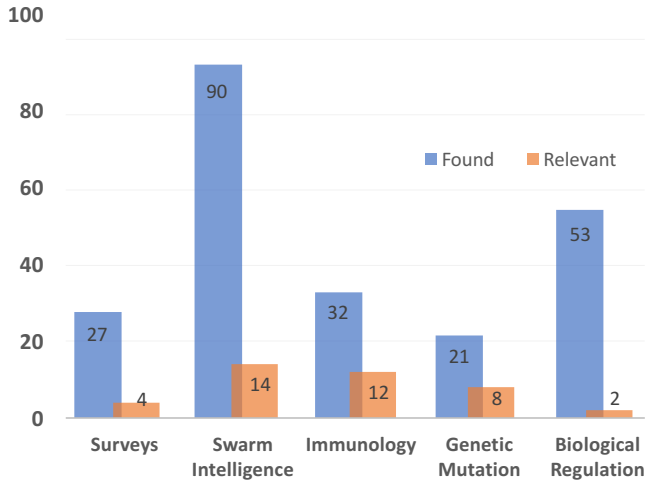



Figure 2.2: Metadata Analysis of Bio-Inspired Cyber Security Literature Review Using Digital Libraries i.e., PubMed, IEEE, Elsevier, Sciencedirect, Springerlink, and Google Scholar

### Bio-Inspired Cyber Security Taxonomy

First step towards defining a taxonomic classification, in any domain, is an extensive literature review. Towards this direction, we use meta-analysis tools to search digital libraries of various publishers, and enlist all possible research contributions in cyber security within the last two decades (1995-2017). As a second step, we manually shortlist the most relevant bio-inspired contributions in cyber security for taxonomic classification. For metadeta analysis we use Scopus [5], rOpenSci [4], RgScholar [2], and RISmed [1] to mine relevant articles from major publishers and digital libraries (Springerlink, Elsevier, IEEE, Pubmed, Sciencedirect and Google Scholar). The search attributes that we use for article mining, can be found in section 4.1. Figure 2.2, shows the results of metadata analysis, which forms the basis of our taxonomic classification. The existing approaches in bio-inspired cyber security can be classified into four major areas. Figure 2.3 gives an overview of our proposed classification of bio-inspired cyber security approaches. The first level categorizes the main classes in this domain, and second level in the hierarchy corresponds to the goals affiliated with the prospective classes. Third level, highlights the observed phe-

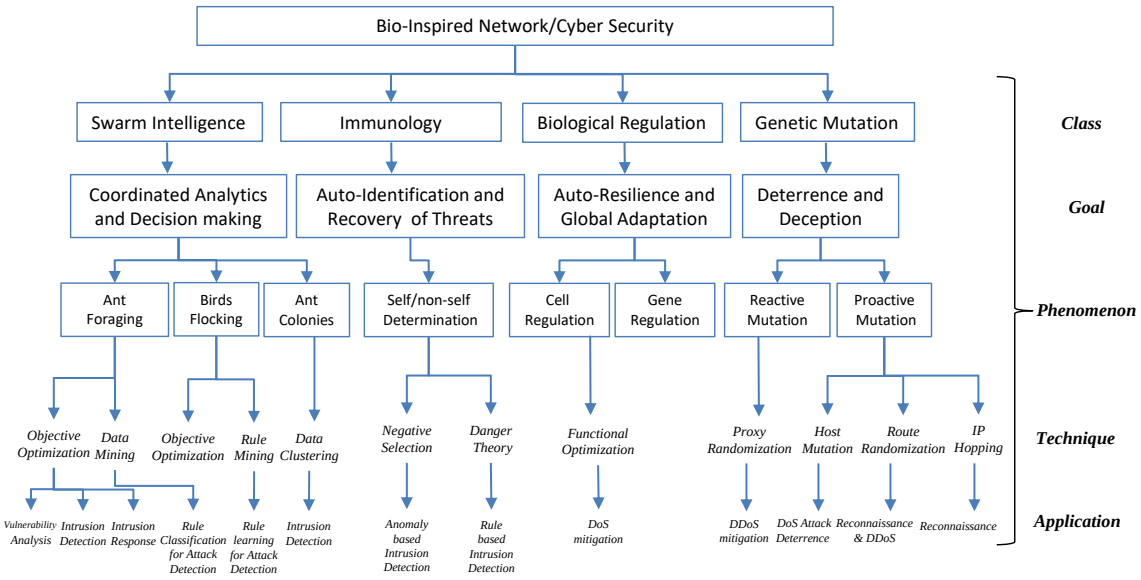



Figure 2.3: Classification of Bio-Inspired research in Cyber-Security

nomenon in the nature, and fourth level in hierarchy enlists all the nature inspired techniques which have been developed so far to fulfill the goals. The final and fifth level in the hierarchy, enumerates all the applications of nature inspired techniques, in the domain of cyber security to accomplish a prospective goal in the hierarchy. As our aim is to understand which bio-inspired class has potential to provide a solution against infrastructure level attacks and insider threats, for this purpose we develop a standard evaluation criteria and discuss it in the next section. The relevant bio-inspired mechanisms and the contribution in the corresponding domains are discussed in Chapter 2, and Chapter 3. For the detailed understanding about each category, we refer our reader to recently publised article [68].

## 2.1 Evaluation

As a next step our aim is to classify bio-inspired network security approaches and to determine the extent of their usefulness, in amplifying and embedding resilience in current cyber infrastructure. The first step, towards evaluating the comparative effectiveness of an underlying mechanism or approach, is to set-up/adopt a criteria/standard which can be followed by security analysts. Towards this direction, the

guidelines and standards proposed by the National Institute of Standards and Technology (NIST), and the International Standards Organization (ISO), form the basis for our evaluation [15] [19].

As the goal of cyber security domain is to design resilient infrastructures, which can reconfigure itself without jeopardizing the mission requirements, therefore, the service accessibility (for a host), usability and the risk associated with these attributes cannot be ignored. On the other hand, the security measures of an institution depends on allocated defense budget, which itself varies from organization to organization. Thus, from an evaluation prospective, an individual should be able to identify the best suited existing techniques and their corresponding capabilities according to their requirements and budget constraints. Hence, end-system usability, efficient self-regulation (adaptivity) and affiliated operational cost forms the remaining basis of our evaluation procedure.

Table 2.1 presents the detailed comparison of existing approaches. The evaluation presented in the table should help a security analyst or domain explorer in understanding about the type of defense that can be implemented using a bio-inspired mechanism. For instance, it helps to understand, whether or not a technique is capable of providing proactive attack prevention (protection), malware detection, attack response features and recovery. We also provide information about the incorporation of end-host usability in a mechanism, its operational mode, and the cost affiliated with it. Finally, we identify, if there is any potential in prospective domain to implement one of the considered defense mechanism, in case it has not been explored already.

By carefully analyzing underlying principles of nature inspired mechanisms, we deduce that *Ant Colony Optimization* and *Immunology* inspired approaches can mainly be used for intrusion detection purposes. Inherently, these mechanisms do not provide any underlying principle which can be mapped to cyber security domain for response

and recovery purposes. As the response and recovery systems usually work on the basis of some predefined policies and constraints, hence, the main limitation of these mechanisms is that they cannot entertain user oriented policies or constraints. Although they can be integrated with a response system [48], to achieve response and recovery purposes, their moderately high operational cost restricts them to operate in an online-mode.

*Genetic Mutation* inspired techniques, on the other hand, have been mainly used for attack deterrence. As the fundamental concepts behind this mechanism allow the representation of cyber parameters as a mutable genetic sequence. Only then, by variation of such sequences, attack vectors can be nullified. Although this mechanism provides the basics for outlier detection, to the best of our knowledge, this aspect has not been explored. The relatively low operational cost (not to be confused with high cost Genetic Optimization Algorithms), makes this mechanism an ideal candidate to operate in an online-mode.

Finally, the fundamental principles in *Biological Regulation* mechanisms seems to have promising characteristics for attack deterrence (by pre-attack regulation of cyber entities), and recovery (post-attack regulation of cyber entities), depending on the response time from neighboring entities (Figure. 4.10). Due to the existence of feedback mechanism in this domain, biological regulation inspired approaches have a tendency to serve as a basic criteria for an outlier detection mechanism, as the misbehaving entity generates an output, which can be observed by the neighboring entities because of their strong intertwined nature. The mechanism itself also provide grounds to develop an integrated system which can incorporate threat detection and deterrence at the same time. In the forthcoming sections we highlight the underlying concepts of *Genetic Mutation* and *Biological Regulation* and their potential to provide support in designing solutions for infrastructure level attacks and insider threats.

Table 2.1: Comparison of Bio-Inspired Approaches for Cyber Security. Symbol "-" refers to the potential capability of corresponding technique which has not be explored yet, and N.A means no corresponding information is available.

| **Inspiration** | **Technique** | **Security Mechanism** | | | | **Usability** | **Adaptivity** | **Cost** |
|---|---|---|---|---|---|---|---|---|
| | | Protection | Detection | Response | Recovery | | | |
| Swarm Intel. | ACO | ✗ | ✓ | ✓ | ✗ | ✗ | offline | Moderate |
| Swarm Intel. | PSO | ✗ | ✓ | ✗ | ✗ | ✗ | offline | High |
| Swarm Intel. | ACC | ✗ | ✓ | ✗ | ✗ | ✗ | offline | Low |
| Immunity | NSA | ✗ | ✓ | - | ✗ | ✗ | offline | High |
| Immunity | DT | ✗ | ✓ | - | ✗ | ✗ | N.A | High |
| Genetic Mutation | Proxy Mutation | ✓ | ✗ | ✓ | ✓ | ✓ | online | Moderate |
| Genetic Mutation | Host Mutation | ✓ | ✗ | ✗ | ✓ | ✓ | online | Moderate |
| Genetic Mutation | Route Mutation | ✓ | - | - | - | ✓ | online | High |
| Genetic Mutation | IP Hoping | ✓ | ✗ | ✗ | ✓ | N.A | online | Moderate |
| Bio. Regulation | Cellular Regulation | - | - | ✓ | ✓ | - | online | N.A |

# CHAPTER 3: Nature-Inspired Approach Against Infrastructure Level DDoS Attacks

## 3.1 Infrastructure Level DDoS Attacks

### 3.1.1 Background & Motivation

The Internet has transformed the world into a global village where even small to moderate enterprises have offices scattered throughout the world. These enterprises often communicate critical information of high business value with each other. The confidentiality, integrity and availability (CIA) of this information are essential for the survival of such businesses.

Over the years, with the advancements in encryption and hashing standards, even a small enterprise can manage and ensure the end-to-end confidentiality and integrity of their critical information. However, they do not have the same level of freedom and control to ensure the availability of this critical information across the Internet. As this information traverses geographical boundaries spanning over multiple autonomous systems (ASes), coordinating a controlled and manageable routing is practically not possible. Therefore, in case of any device or link failure along the network path, due to faults or cyber attacks, leaves such enterprises helpless and they have to resort on the slow Internet recovery process to restore the availability of their critical information [69].

Recently, not only the incidents challenging the network availability are on the rise, the scale and sophistication of such Distributed Denial of Service (DDoS) attacks has also been rapidly increasing. According to recently published reports by **Incapsula**, the average cost of an unmitigated (6-24 hrs long) DoS attack in 2017 was estimated to be around ≈$500,000 (USD). Whereas the attacker can cause this much damage

($40,000/hr) by merely spending $35/hr (by generating 220 Gbps traffic targeting a link/router) [10].

Hence, the ossified nature and inherent limitations of internet infrastructure have shifted the scales in the favor of the adversary. This forms the main motivation of the thesis and inspires us to design methods to shift scale back in the favor of defenders and eliminate the inherent limitations to an extent that the cost affiliated to an attack can be raised significantly to deter an adversary obfuscate his targets/plans.

### 3.1.2 Problem Statement

The main reasons, which substantially increase the impact of DDoS attack, are attributed to the ossified nature of internet infrastructure, that has resulted into following inherent vulnerabilities. First, network paths are mostly static which makes network reconnaissance for identifying links-to-destination association feasible with low-cost [55]. Second, the power-law distribution of traffic flows over network links results into emergence of critical links, i.e., to any destination only a few links carry majority of its traffic [31]. Third, a destination shares its critical links with its geographical neighbors [55]. Finally, the inability of an enterprise to avoid critical links, if under attack, as the routing infrastructure belongs to Internet Service Providers (ISPs) [69].

These fundamental design level vulnerabilities enable an adversary to easily learn, compute a set of critical links and plan devastating attacks to isolate the traffic between entire regions/states [55][75]. Hence, such inherent vulnerabilities impose severe threats to critical infrastructures, i.e., energy grids and financial institutions, and demands novel approaches for resilient cyber agility to defend against infrastructure level attacks.

Therefore, the focus of this thesis is to design nature inspired resilient routing approach and algorithms which not only enables an enterprise to evade critical links (in a proactive/reactive manner), but can also obfuscate the information gain about

critical link for an attacker.

### 3.1.3 Goals & Objectives

According to the literature, if the cyber attributes can somehow be represented as sequences of elements (e.g., routes are sequence of links in a network), genetic mutation phenomenon can be mimicked by efficiently changing the elements of sequence to achieve constructive reconfiguration (resilience) and mitigate against ongoing attacks [68]. In the scope of this thesis we aim to represent routing problem as sequence of peers by considering end-points (peers) and design resilient routing functionality that can employ end-hosts based virtual routers to increase the routing diversity and avoid critical links over the internet.

We assume that there exists geographically distributed end-hosts based infrastructure over the internet, which we aim to leverage. Practical examples of such architectures where end hosts (peers) can be selected and programmed are R2Lab and Planetlab [6] [3]. Planetlab/R2Lab are publicly available prominent virtualized infrastructure and testbeds, in which the physical network resources are geographically distributed across hundreds of physical domains (mostly universities) on the Internet. Our final main objective is to leverage such architecture for the testing and evaluation of our proposed approach. Towards these objectives, the main tasks to be achieved can be listed as follows:

- Mathematical Formalization of End-to-End (E2E) resilient reachability problem as a constraint satisfaction problem [35, 71].
- Development of a genetic mutation inspired synthesis based planning algorithmto find the satisfiable set of peers for the sake of avoiding critical link between a pair of source and destination.
- Minimize the overlap/intersection of physical links and peers between multiple flows, to maximize the agility in the system.

- Testing, evaluation and feasibility of deployment for the proposed method.

## 3.2 Related Work

To the best of our knowledge, although, there has not been any research which directly maps principles of the Genetic Mutation to the Cyber-Security domain, there are some approaches in the domain of *Moving Target Defense* (MTD), which work on the similar randomization principles to improve the resiliency against active cyber threats e.g., Denial of Service (DoS), and scanning attacks.

We classify the relevant literature on MTD into two categories: reactive and proactive mutation based approaches. For reactive MTD, the defenders make adaptations in a lazy manner, and only when they are under attacks. While for proactive MTD, the defenders also make adaptations before they are attacked, aiming to thwart attacks proactively and to reduce the negative impact of attacks.

### 3.2.1 Reactive Mutation based Security Approaches

Since the first appearance of DDoS attacks in 2000 [42], several defense mechanisms based on filtering attack traffic or limiting a client's bandwidth share have been proposed to mitigate attacks at the Internet scale [44]. Unfortunately, in order to be effective, these solutions rely on large-scale adoption and coordination among different network elements. These limitations promoted overlay-based architectures that can hide the location of target servers behind a well-provisioned, distributed overlay network.

For instance, Angelos et al. [74], proposed an approach (named as MOVE) to protect services against attackers who control and disrupt only a subset of network elements. In MOVE, target services accept traffic only from a subset of overlay nodes (the secret servlets) and when a DoS attack is detected, the target service is migrated to a new host to mitigate the impact of the attack.

Recently, Wang et. al. [80] proposed a similar approach named as: MOTAG

(MOving Target defense mechanism AGainst internet DDoS attacks), to protect web services by hiding the application server's location behind a large pool of proxy servers. Originally designed to support services that require client authentication and later extended to support anonymous users [53], MOTAG leverages the on demand availability of resources in a cloud environment to spawn new proxy servers when attacked. To limit the impact of an attack, MOTAG migrates clients to new proxies and shuffles the client-to-proxy assignment to isolate insiders who divulge the location of the secret proxy servers. However, the shuffling process employed by MOTAG to isolate insiders does not consider the overhead associated with instantiating and maintaining new proxies.

To address this issue, Wood et al. [82] proposed DoSE, a cloud-based architecture that provides a cost-effective mechanism to isolate insiders and confine an attack to a few proxies. In DoSE, each client is associated with a risk value which captures the likelihood that the client will indulge in a DoS attack. Additionally, each proxy server has an upper bound on the risk that it can tolerate. During an attack, DoSE assigns clients to proxies based on the corresponding risk parameters and updates the risk value of clients associated with attacked proxies. Similar to MOTAG, DoSE instantiates new proxies to reduce the impact of the attack and migrates victims to new proxies based on their updated risk values. By maintaining a state for each client (through risk), DoSE limits the number of proxies needed to identify insiders, thereby reducing the cost to maintain such an architecture.

### 3.2.2 Proactive Mutation based Security Approaches

This category involves research which proposes to change different parameters of network (e.g., Routes or IP addresses) proactively to avoid links under congestion or scanning attacks. The first effort, which involves the idea for mutating routes for wired networks is called Random Route Mutation (RRM) [39]. RRM leverages the fact that there exist multiple paths/routes between two nodes in a network topol-

ogy, forming a set of disjoint/overlapped routes. The multiple routes can then be used to transmit multiple flows iteratively (in a random fashion) between source and destination. The proposed approach shows promising results if there exist multiple disjoint routes between source and destination, and requires ISP level coordination for changing routes, which is not practical in real life. In addition, no end-hopping is involved in the RRM method, which enables attackers to recover communication data between hosts by sniffing multiple switches.

Existing DDoS mitigation techniques rely on flattering and analyzing the attack traffic. However, as the nature of attacks is becoming more stealthy, it is almost impossible to distinguish between benign and attack situations [55]. Nonetheless, a very common prerequisite is required for such attacks to be successful (e.g., identifying critical links through reconnaissance). If the critical nature in such scenarios is changed to non-critical before the attack is launched, the attack reconnaissance can be rendered useless for the adversary. Virtual Networks (VN) deployed on physical nodes, provide such flexibility via migration process, in which resources can be migrated to a geographically distant machine. Gillani et al. [46] proposed such an approach which proactively evades DDoS attack by changing the foot prints of critical resources; therefore link DDoS attacks are resisted. The authors emulate the state of the art crossfire attack scenario [55], to show the usefulness of their proposed approach.

Similar approaches, which involves randomly changing IP addresses of the prospective hosts (Random Host Mutation (RHM)), has also been proposed for the wired network security in the literature. For instance, OpenFlow(OF)-RHM [51] shows that the effectiveness of sniffer attack and worm propagation can be reduced by proactively assigning different IP addresses to the end-hosts, but virtual IP should stay unchanged during one continuous communication, which can enable attackers to obtain complete data of one communication from a switch. Improvements to these approaches were


Figure 3.1: Concept of Genetic Mutation

later proposed in the recent years [50][62].

Although the researchers have been using the concept of parameter mutation, it was not until recently that one-to-one correspondence between *Genetic Mutation* and *Moving Target Defense* was established [68]. In the next section we discuss how the concept of *Genetic Mutation* can be used for creating resilience and deterrence against DDoS.

## 3.3 Genetic Mutation

Since genes are nothing more than regions on nucleotide bases of DNA, any alteration in them (nucleotide bases) is refereed as mutation, regardless of the number of possible ways they can be mutated. Mutation can be of multiple types e.g., (a) it can be substitution of a single base, (b) insertion of a sub-region, and (c) deletion of one or more nucleotide basis [7]. Figure 3.1 illustrates these types in details.

Genetic mutations have always been associated with different diseases but in reality, not all mutations are harmful. Mutation is the alteration of sequence of a gene, which could be detrimental, beneficial or neutral. Some of the beneficial mutations observed in the nature are as follows. (1) increase in survival period of colorectal cancer: research shows that the production of mutation due to the regular use of *Aspirin* triggers the possibility of long lasting survival for the patients of colorectal cancer [58]. (2) Malaria resistance: genetic mutation causing sickle cell is also observed to produce resistance against malaria. This mutation is favorable in the areas which are prevalent for malaria [81]. (3) Lactose tolerance: person can switch from lactose intolerant state to lactose tolerant state after the appearance of a lactose tolerant mutation.

Inspired by the concept of *Mutation,* if cyber attributes can somehow be represented as sequences of elements (e.g., routes are sequence of links in a network), the functionality of genetic mutation operator can be mimicked by efficiently changing the elements of sequence to achieve constructive reconfiguration (resilience) and mitigate against ongoing attacks.

In this thesis we propose to redefine the routing and reachability problem by involving end-points and use the concept of genetic mutation operator to find an alternative path avoiding the critical links. We refer our proposed method as End-Point Route Mutation (EPRM), and discuss the technical details in the following section.

## 3.4 Problem Definition

Traditional networks are modeled as a graph *G(V, E)*, where $V$ is a set for vertices (in this case end-hosts) and $E$ is a set of edges (in this case intermediate peers and links among them). Therefore, corresponding to traditional networks a placement of EPRM problem can be considered as a five tuple $T$=($S$, $D$, $P$, $C_i$, $Lab$), where;

- $S \in V$ is source for a certain flow
- $D \in V$ is destination for a certain flow
- $P \subset V$ is a set of potential peers, such that: $S \cap P : \emptyset$ and $D \cap P : \emptyset$ to avoid the cycles
- $C_i$ is a set of connectivity constraints for a pair $(S, D)$
- $\mu$ is a Genetic Mutation inspired operator, such that it take a set of input containing source, destination and a set of constraints (i.e., reachability and link capacity) and provides a mutated route containing set of peers, $\mu(S, D, C_i, P) : (S, D) \longrightarrow P_i$ where $P_i \subseteq P$

Figure 3.2 provides an overview of our proposed approach for routing agility. A virtual path is comprised of suitable peers ($P_i$) with a specific order and changing the order

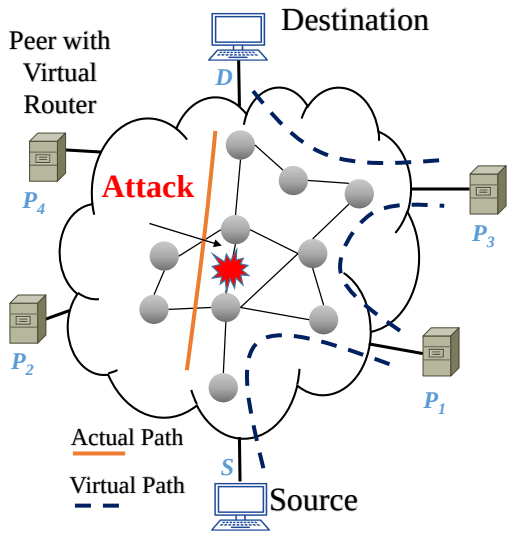



Figure 3.2: Concept of End Point Route Agility

results into a different virtual path. The goal of EPRM is to efficiently compute the set virtual paths for critical communications and then decide a sequence to use them for a source-destination pair, while satisfying following constraints.

- ***QoS Constraints***: The selected mutation set (MS) is the set of all valid virtual paths for a source-destination pair. The MS should meet QoS requirements such as; available bandwidth of all virtual paths must be greater than the traffic load generated by the source, number of hops should be limited and the E2E delay must be comparable to the delay of the actual physical path of the source-destination pair (Fig. 3.2).
- ***Resilience Constraint***: To increase unpredictability or chaos for the attacker, all virtual paths in a sequence must have minimum (link) overlap with the immediate neighboring virtual paths.

### 3.4.1 Hardness of Problem

E2E resilient reachability of a single flow/transmission requires the upper bound on the number of peers to be used to reach the destination, which according to the theory [43] is a NP-complete problem and cannot be solved in polynomial time, but

the efficiency is highly dependent on the value of *Upper Bound (UB)*. In current architecture, we use *UB=3* as an upper bound, which means a transmission/flow can pass through at-most three intermediate peers or in other words, a virtual path cannot have length greater than 3 hops. Our evaluation shows that only 3 hops can add enough variance to evade and deter persistent attackers.

### 3.4.2 Attack Model

For DDoS attacks, we assume that an attacker can disrupt/compromise limited number of links$\subset$ *critical_set*, which she believes to be critical, for a specific period of time. For selection of target our attack model follows uniform probability distribution to select the the target link. There can be two types of attackers. *Naive Attacker*: An attacker who does not have any information about the critical links and nodes, and attacks the network in a random fashion for a random amount of time without any prior information about the flows. *Sophisticated Attacker*: An attacker who can gain information about the critical links and nodes in the network, and can launch the attack accordingly to maximize her devastation. She can also attack randomly on different critical links and move the attack surface over these critical network links to avoid detection and to harden the recovery.

## 3.5 Proposed Approach

### 3.5.1 Challenges

The main challenge in EPRM protocol is to determine a correct-by-construction mutation set (set of virtual paths), given a pair of source-destination, and then efficiently mutate the virtual path, on the fly during multiple transmissions, to disable the capabilities of an attacker for launching DDoS attack on specific links or nodes. We do not rely on configuring infrastructure level devices for monitoring or forwarding packets through a desired path.

Mostly, all end-hosts use TCP based communication channel and the physical path

between them relatively stays stable for almost 80% of the time [54]. In TCP sessions, both source and destination use TCP congestion window to decide how much data should be transmitted and using this transmission history between any two peers, we can easily estimate the safe available bandwidth of the Internet path connecting these peers, corresponding to different periods of time. This is achieved by deploying an agent (virtual router) on every peer, which shares this information to a source upon a request.

As a result, the source can alter the sequence or narrow down its choices of virtual paths to be used based on QoS constraints. To reduce the overhead, mutation sets are pre-computed according to reachability constraints and their sequences are evaluated on run time. We use SMT to formalize the problem and the constraints as uninterpreted coupled functions in first order logic. Evaluation of these function as true means that there exist an interpretation for the first order logical formula which satisfies the QoS and resilience constraints. In order to allow smooth transition between virtual paths (mutations), configuration changes are carefully observed and managed to avoid disruption between multiple transmissions.

### 3.5.2 SMT Formalization of EPRM

#### 3.5.2.1 Formalization of E2E Resilient Reachability

The formalization of E2E resilient reachability is fundamental to EPRM protocol for calculating the mutation set for multiple flows. The problem of computing set of routes with all possible length, is computationally expensive. Therefore, to make this approach practical for usage we put an upper bound on the number of intermediate peers to be used, for calculation of reachability, such that the problem can be solve in polynomial time. The most important ingredient of virtual reachability is logical path connectivity, which we define as a boolean function that accepts two arguments

and maps it to true/false if two nodes are logically connected/not connected.

$$L(x, y) : \mathbb{N}^2 \mapsto \mathbb{B}$$

Where x and y are source and destination in set of peers. Using the same notion we can define virtual reachability between two nodes as a conjunction of multiple constraints:

$$R(x, y) : \mathbb{N}^2 \mapsto \mathbb{B}$$
$$R(x, y) \rightarrow \left( \bigwedge_{z=1}^{4} \Psi_z \right)$$

The above relation states that; there exists a virtual path between x and y if following constraints ($\Psi_i$) hold true.

$$\Psi_1 = \forall_{x,y} \Bigg( \exists_{i,j,k} \Big( L(x,i) \wedge L(i,j) \wedge L(j,k) \wedge L(k,y) \Big) \vee$$
$$\exists_{i,j} \Big( L(x,i) \wedge L(i,j) \wedge L(j,y) \Big) \vee$$
$$\exists_{i} \Big( L(x,i) \wedge L(i,y) \Big) \Bigg)$$
$$\Psi_2 = \forall_x \Big( \neg L(x,x) \Big)$$
$$\Psi_3 = \forall_{x,y} \Big( L(x,y) \rightarrow L(y,x) \Big)$$
$$\Psi_4 = \forall_u \Big( \neg L(u,x) \Big) \wedge \forall_v \Big( \neg L(y,v) \Big) \; s.t. \; x = source \; and$$
$$y = destination$$

Note that virtual path is bidirectional: $R(x, y) \leftrightarrow R(y, x)$. $\Psi_1$ is the most fundamental property, which defines the characteristic of bounded reachability. We put an upper bound on the length through $\Psi_1$, by restricting the maximum number of

intermediate peers to be '3'. Therefore, using our reachability formalization a peer *'x'* (source) can use at most '3' intermediate peers (*i,j,k*) to reach the destination *'y'*, under critical/normal circumstances. Although the upper bound value can be set to any arbitrary number ($\geq 1$), where increasing the number of intermediate peer will certainly increase the diversity and resilience in the system (mutation space), but for the simplicity and for the batter understanding of the readers we make above mentioned limit on upper bound.

The value of upper bound is variable and can be changed as per user requirements. This does not effect the protocol structure, and only increases the length of formalization. Where *i,j* and *k* can assume any value in peers set $P_i$, as far as connectivity constraints/requirements are satisfied. $\Psi_2$ implies that no node should be reachable to itself directly, to avoid infinite length self-cycles. $\Psi_3$ implies that logical path and reachability is a bidirectional property. If *'y'* is reachable by *'x'*, then *'x'* is also reachable by *'y'* in terms of connectivity. Finally, $\Psi_4$ implies that while considering *'x'* as a source and *'y'* as destination, avoid all irrelevant and useless interpretations (to reduce computational complexity) of the function *R(x,y)* in which *'x'* is not source and *'y'* is not destination. In simpler words this property is responsible for fixing the source and destination to reduce time and space complexity. Note that this is a generalized definition for computing reachable set from multiple source to multiple destinations as *'x'* and *'y'* are variable.

#### 3.5.2.2 Formalization of QoS Constraints

As we discussed earlier, QoS constraint is strongly subjected to bounded delays (nodal processing time), maximum number of hops to be used for a transmission and link available bandwidth for quality transmission. This inherently restricts reachability formalization to only find a path which may only have at most certain number of hops ($\Theta_{hops}$), certain available band-width (which can accommodate desired transfer rate: $\Theta_{trans\_rate}$) and certain amount of load ($\Theta_{load}$) on intermediate peers. Where

$\Theta_{hops}$, $\Theta_{trans\_rate}$ and $\Theta_{load}$ are variable threshold values, provided by the user, for desired maximum number of hops, desired transfer rate (which is directly related to the quality of transmission), and nodal processing time/delay respectively. These threshold values are service specific, and depends on the quality demanded by the service running on a specific host.

***Formalization of Intermediate Hops Constraint:*** The formal definition of QoS constraint, for maximum number of intermediate hops to be used, is as follows:

$$QoS_{inter\_hops} : \forall_{x,y} \left( \exists_{i,j,k} \left( \#_{hops} \leq \Theta_{hops} \right) \right)$$

where $\#_{hops}$ can be defined as follows:

$$\begin{aligned} \#_{hops} : \Bigg( & \sum \Big( H(x,i), H(i,j), H(j,k), H(k,y) \Big) \vee \\ & \sum \Big( H(x,i), H(i,j), H(j,y) \Big) \vee \\ & \sum \Big( H(x,i), H(i,y) \Big) \Bigg) \end{aligned}$$

The uninterpreted function $H : \mathbb{N}^2 \mapsto \mathbb{N}$ represents the number of hops between a pair of source and destination, therefore, it accepts two arguments. $\#_{hops}$ represents the maximum number of hops, regardless of how many intermediate peers have been chosen to reach a certain destination. The above mention constraint is a generalized version, as the proposed approach can decide to choose one, two, or three intermediate peers, depending on the situation and availability, which results in the form of a disjunction of three different cases. The proposed formalization limits that the number of hops on chosen route, in any case, should be at most $\Theta_{hops}$ for any pair of source and destination.

***Formalization of Link band-width Constraint:*** Following is the formal rep-

resentation of link band-width constraint:

$$l^a(x,y) : \mathbb{N}^2 \mapsto \mathbb{R}$$

$$QoS_{band\_width} : \forall_{x,y}\left(\bigvee_{i:1}^{3}\left(\Phi_i\right)\right)$$

$$\Phi_1 : \exists_{i,j,k}\Big(l^a(x,i) > \Theta_{trans\_rate} \wedge l^a(i,j) > \Theta_{trans\_rate}$$

$$\wedge\, l^a(j,k) > \Theta_{trans\_rate} \wedge l^a(k,y) > \Theta_{trans\_rate}\Big)$$

$$\Phi_2 : \exists_{i,j}\Big(l^a(x,i) > \Theta_{trans\_rate} \wedge l^a(i,j) > \Theta_{trans\_rate}$$

$$\wedge\, l^a(j,y) > \Theta_{trans\_rate}\Big)$$

$$\Phi_3 : \exists_{i}\Big(l^a(x,i) > \Theta_{trans\_rate} \wedge l^a(i,y) > \Theta_{trans\_rate}\Big)$$

To avoid a overloaded links, which may be under attack, link available bandwidth constraint is very effective. This constraint must be checked on run time (online) before selecting a sequence of peers. The main reason for checking this constraint online/on run-time is that the peers may be off-line at any instance of time and thus making them unable to process a request, therefore, this is one of few main steps in protocol which must be checked online. Formal description of online-phase of the protocol is explained in the next section. The above mentioned constraint is a generalized version, which covers a path of length three. For better understanding of the readers it can easily be broken into three parts by removing the disjunction symbol. $l^a(x,y)$ refers to the cumulative link available bandwidth between peer $x$ and $y$ (including all hops) and $\Theta_{trans\_rate}$ refers to the desired transfer rate of the source. Therefore, at any given time the algorithm must find a satisfiable path/route for which the available bandwidth is more than the desired transfer rate of the source.

***Formalization of Load Constraint:*** Formal definition of load constraint on

intermediate peers is as follows:

$$l(u) : \mathbb{N} \mapsto \mathbb{R}$$
$$Total_{load} : \sum \Big( l(i), l(j), l(k) \Big)$$
$$QoS_{load} : \forall_{x,y} \Big( \exists_{i,j,k} \Big( Total_{load} \leq \Theta_{load} \Big) \Big)$$

Where *l(u)* is a function which evaluates load on an intermediate peer. This *load* constraint implies that only those intermediate peers should be selected which have collective load less then certain threshold $\Theta_{load}$ to maintain quality of transmission and control delays.

### 3.5.3 Formal Description of EPRM Protocol

In this section we formally describe the structure of the protocol along with the algorithm of proposed approach and some basic definitions which are fundamental for better understanding of the readers.

**Definition 1** ***Potential Peers Set (PPS):*** *A PPS is a set of possible logical paths for a pair of source and destination satisfying reachability criteria. PPS for a network in Fig.3.3 contains:* $\{(P_1, P_2, P_3), (P_4, P_5, P_6), (P_7, P_8, P_9), ...\}$

**Definition 2** ***Mutation Set (MS):*** *A MS is a set of logical paths (constituting peers) which meet the aforementioned constraints in order to connect a pair of source and destination, such that MS*$\subseteq$ *PPS. An instance of MS is a logical path* $L_i \in MS$. *MS from an example network given in Fig. 3.3 is,* $\{L_1 : (P_1, P_2, P_3)$, $L_2 : (P_4, P_5, P_6)$, $L_3 = (P_7, P_8, P_9)\}$

**Definition 3** ***Mutation Sequence (MSeq):*** *A MSeq is a set with an ordered sequence of mutations (logical paths) for a pair of source and destination, such that:* $MSeq \subseteq MS$, *which is calculated at run-time based on the real-time available bandwidth updates from the peers.*

**Definition 4** ***Mutation Cycle:*** *A time duration after which a logical path repeats, once every possible member of MSeq is used, is referred as Mutation Cycle duration.*

#### 3.5.3.1 Protocol Description

*Off-line Phase: Formation of Mutation Set*:

The first and foremost step of EPRM protocol is to efficiently compute the mutation set given a set of source and destination according QoS requirements. For our implementation and evaluation we consider E2E reachability using at most three intermediate peers (length=3) and present the corresponding Algo. 1 to compute mutation set[1], considering three intermediate peers (length=3). Nevertheless, the algorithm can easily be extended and generalized up to any desired length.

*Online Phase: Calculation of Mutation Sequence*:

- *Broadcast Association Information:* Once a source computes MS, it broadcasts to all of potential peers in MS, an activation order/sequence.
- *Liveness Response:* Immediately after receiving activation order/sequence each selected peer must send available path bandwidth and information/packet drop rate, from itself to its successor and next peer to the source. This information is used to finalize the mutation sequence (MSeq).
- *Calculating Mutation Sequence:* Based on the information received in liveness response phase, the mutation sequence of active intermediate peers is efficiently calculated for every flow (or a set of flows) such that: $L_i \in$ MSeq. Moreover, every logical path in MSeq exhibit minimum overlapping with its predecessor and successor logical paths. Furthermore, if there exist a set of some critical physical links ($\beta$) that must be avoided, our approach avoids all such logical paths that uses these critical links. This phase can be repeated iteratively, until the mutation

[1]We use iterative algorithm in actual and this recursion based algorithm (Algo. 2) is presented here for simplicity and limitation of space.

sequences start repeating or according to user requirement for a certain number of flows. Algorithm 2 also guarantees that the flow distribution of each peer and intermediate link will be same if there are only disjoint routes in the network.

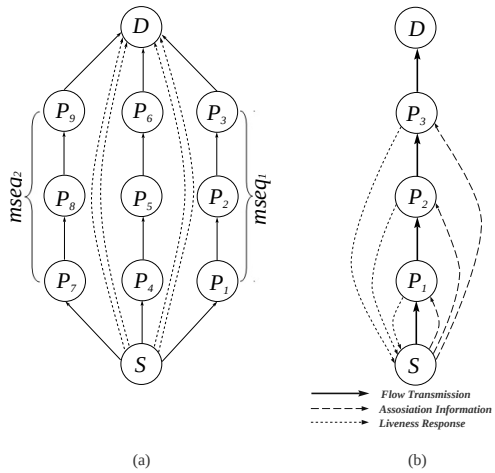



Figure 3.3: (a) EPRM flow transmission using multiple MSeq of intermediate peers. (b) Schematic diagram for interaction between source and elements of MS.

***Discussion*:** Our approach involves two phase mutations which makes it difficult and chaotic for an attacker to learn about insights of the network. The first phase mutation is the selection of best suited candidates for a MS, which can be shuffled and re-selected after a certain time in an iterative way. The second phase mutation becomes active when different mutation sequences ($mseq_i$) are selected from MS for different set of flows in an iterative manner. Fig. 3.3 presents a schematic diagram of interaction between source and other selected peers in order to obtain information from neighboring peers.

One of the main goals in MTD research, is to minimize the overlap of physical paths between consecutive flows. Reducing the overlap will increase the agility in the system, as the traffic will be evenly distributed in the network, making it harder for an attacker to identify critical links. We also present an algorithm in Algo. 2 to minimizing path overlapping between multiple flows. Unlike previous methods [39, 52], we do not completely negate the previously used logical paths. Instead,

**Algorithm 1** Computing Mutation Set

```
1: P = {(P_i), (P_i, P_j), (P_i, P_j, P_k)}
2: C = {  Set of constraints  }
3: procedure LAB(P,C)
4:     (result, model) = check_satisfiability(P, C)
5:     if result==SAT then
6:         MS.insert(model)
7:         P' = model.P
8:         for each peer p ∈ P' do
9:             assertion_stack.push()
10:            C.insert(¬p)
11:            Lab(P, C)
12:        end for
13:    else
14:        assertion_stack.pop()
15:        return
16:    end if
17: end procedure
```

we randomize the mutation cycle of a logical path to uniformly distribute its reuse probability. Every time a logical path is inserted into MSeq (Line-13, Algo. 2), it is assigned a random counter $t_i$. For each insertion in MSeq, this counter decrements for all logical paths already inserted (Line-16). When the counter $t_i$ expires (becomes zero) for a logical path $L_i$ then this logical path becomes a contender to be inserted back in MSeq provided all other conditions are satisfied. To the best of our knowledge this is the first run time optimization algorithm for minimizing the overlap between multiple flows in a network using SMT in the domain of MTD research.

In Algo. 2, we use *Route* to store list of physical links along a logical path and *findLinks()* function uses `traceroute` to find these links. The *physical_path* variable represents a set of physical links along the physical path between source and destination. The *If* condition (Line 20) ensures that if there exists any logical path that was skipped in the last iteration due to the overlapping condition, it can be checked again to see if it becomes valid again in the new sequence.

**Algorithm 2** Computing Mutation Sequence

**procedure** FINDNEXT
  Route = $\emptyset$
  **for** each logical_path $L_i \in MS$ **do**
    Route.insert(findLinks($L_i$)) ▷ Find links along the logical path.
  **end for**
  prev = physical_path
  MSeq = $\emptyset$
  iteration = 0
  **while** True **do**
    **for** each route $r_i \in Route$ **do**
      ratio = $\frac{|r_i \cap prev|}{|r_i|}$ ▷ $|r_i|$ counts set elements.
      **if** ($r_i \notin$ MSeq or $t_i = 0$) and (ratio $< \Theta_{ratio}$) and ($\beta \cap r_i = \emptyset$) and ($Total^i_{load} \leq \Theta^i_{load}$) **then**
        MSeq.insert($L_i$)
        prev = $r_i$
        $t_i$ = Rand($\Theta_{length}$)
        Decrement all other $t$
      **end if**
    **end for**
    iteration = iteration + 1
    **if** count(Selected) $\geq \Theta_{length}$ **then**
      break While Loop
    **end if**
    **if** iteration $\geq \Theta_{reduce\_ratio}$ **then**
      Reduce $\Theta_{ratio}$ ▷ Allowing more overlapping.
    **end if**
  **end while**
**end procedure**

### 3.5.4 Complexity Analysis of EPRM

For a fixed source and destination if we only want to find one satisfiable path, then the complexity (either the worst case or the average case) of Dijkstra algorithm is $O(n^2)$, and the average complexity algorithm in the thesis is $O(\Delta^\eta)$, where $\Delta$ is the maximum node degree of the graph (physical topology), and $\eta$ is the upper bound of the length (in terms of hops) between two peers in the graph. For most practical topologies, $\Delta$ and $\eta$ are small numbers. If one wants to enumerate all satisfiable paths, then the possible number of paths is $O(n!)$ but our approach has complexity $O(n^3)$, since one needs to test all possible combinations of containing only three intermediate peers, and reduces the overall complexity.

## 3.6 Evaluation and Effectiveness of EPRM

### 3.6.1 Experimental Setup

In our design and evaluations of EPRM, we use AWS and R2Lab architectures for testing the feasibility of EPRM. As mentioned earlier, each user can reserve a slice in AWS and R2Lab and user can add nodes in this slice from all over the world. We use this node as a peer that hosts a virtual router. We create a virtual network that connects a source and its destinations by creating tunnels from source to its peers and between peers (Fig. 3.2).

There are two common methods to create and manage a VN within a slice. The first implements VNs in user space. This requires the creation of a virtual router in each peer and connecting the virtual routers together with UDP tunnels. Second possible way is to implement the VNs in kernel space by setting up packet forwarding in kernel space. Although, implementing/installing virtual router in kernel space is more efficient approach, but specifying privileges using Vsys, is unfortunately tedious and error-prone procedure. Therefore, We use former method to install virtual routers scripts in each peer within user space.

To setup VNs, we use a Python Vsys API package provided by NEPI [3]. According to the example topology, given in Fig. 3.2, we connect the physical nodes with point-to-point tunnels through the Vsys API. We assign private IP addresses within the assigned subnet of our slice to the virtual interfaces for those tunnels. Then we install the pre-computed routing table entries (from Algo. 1) to the forwarding tables of the physical nodes through the Vsys API and use Algo. 2 to change their priorities to achieve desired sequence.

### 3.6.2 Delay Overhead Feasibility of EPRM

As EPRM may result in a virtual path with increased hops as compared to the actual physical path. Therefore, we initially conducted experiments to test the delay introduced by EPRM, when it uses 2 peers to increase the defense against the DDoS attack and also compare its results to existing peer to peer (P2P) routing mechanism ToR. We performed three experiments using the topology from Fig. 3.2 where source sends `ping` commands to the destination. In our first experiment, we configure the network such that the `ping` command follows the static physical path. In second experiment, we configure the system to follow the path $\langle \text{source}, p_1, p_3, \text{destination} \rangle$, and in third experiment we install ToR implementation on source node and ping the end-host/destination in 3.2, via anonymous ToR routers. The results of the $2,000$ measured delay samples and their Cumulative Distributed Function (CDF) are illustrated in Fig. 3.4(a), and 3.4(b). These results show that by adding even two peers in the path, the increase in delay for EPRM is minimal, i.e., it increases from average 70 milliseconds to only 85 milliseconds and we do not experience any significant variations. We observed minimum to no difference when we increased the peers from 2 to 3. Whereas the delay overhead for ToR implementation is significantly larger than the overhead of static path (2-3 fold), while it only preserves anonymity. Whereas EPRM defends against infrastructure level attacks while adding minimal overhead. These results clearly demonstrate the suitability of EPRM for critical

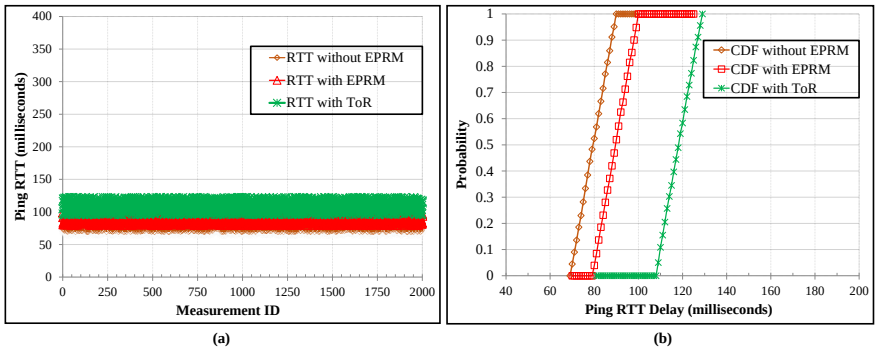



Figure 3.4: (a) Impact of EPRM on the bounded delay and its CDF analysis in (b) communications.

### 3.6.3 Simulation Setup

For SMT formalization, we use Z3 solver [34], and for evaluation we use real life peer to peer data sets available at SNAP [57]. As the real life data sets are not sufficient in numbers, we also use BRITE [40] as network generator to generate additional topologies closer to real life characteristics. In BRITE we use the Waxman model to generate random and power law based preferential attachments. The two parameters used for Waxman model are $\alpha = 0.2$ and $\beta = 0.15$ and the network growth type is set to be incremental. All the experiments are conducted on Core i7 machines with 3.4GHz processor, and 16GB memory.

### 3.6.4 Effectiveness and Resilience of EPRM

The ideal metric available in literature to analyze the effectiveness of the MTD or route mutation approaches is (*Mutation Protection Effectiveness (MPE)*)[39] (which assumes that defender is following Shamir's criteria [59]). *MPE* metric is represented as follows:

$$MPE = 1 - \sum_{1=1}^{m} \frac{R}{N} \frac{1}{M} d_i$$

MPE can be defined to be the percentage of packets in a transmission which do not pass through any intermediate nodes (links) that are being compromised or eavesdropped. Note that this metric is application specific and minimum requirement of

MPE can vary from application to application. MPE calculates the average case effectiveness of a deployed mechanism, which is why it is appropriate to compare between different route mutation techniques. This metric assumes that defender and attacker are randomly choosing routes regardless of the actions taken by each other.

In our R2Lab based experiment, we calculate the physical topology using `traceroute` probes and used this in our experiments. Attacker can target $R$ routes out of $N$ possible disjoint routes, in $M$ mutation intervals; where $d_i$ is the damage caused by the degradation of the flows within the same interval. Suppose if attacker is targeting same routes which are being used by the defender in a certain interval, and this causes a degradation of *40%* traffic/flows, then $d_i$ can be 0.4.

Our approach is rather mixture of proactive and reactive mechanisms as we mutate peers in a proactive fashion, but after collecting information from our potential peers; and based on that we take decision about which links to avoid (if under attack) to maintain QoS, this highlights the resilient features of the proposed strategy. We setup our experiment such that, an adversary has limited knowledge about critical links and limited resources (an adversary has capacity to cause maximum *80%* degradation in traffic/flows). We cannot assume an adversary model with infinite resources as this is not a practical assumption. We target/attack different percentages of routes/intermidiate links (R) in *Mutation Set* (as adversary has knowledge about critical links) to observe the effectiveness of EPRM. We also assume that the adversary's behavior is dynamic, but in a certain interval he/she is only targeting certain percentage of links. We compare the effectiveness of our proposed approach against RRM and non-RRM techniques. We choose network with 100 nodes and 256 edges (with length L=3), and run experiments for single flow and multiple flows. Fig. 3.5 (a) shows the MPE effectiveness of EPRM, in comparison with RRM and non-RRM approaches (for the network of same characteristics). For small number of flows EPRM seems to have significant benefit over RRM, and non-RRM approaches. It is

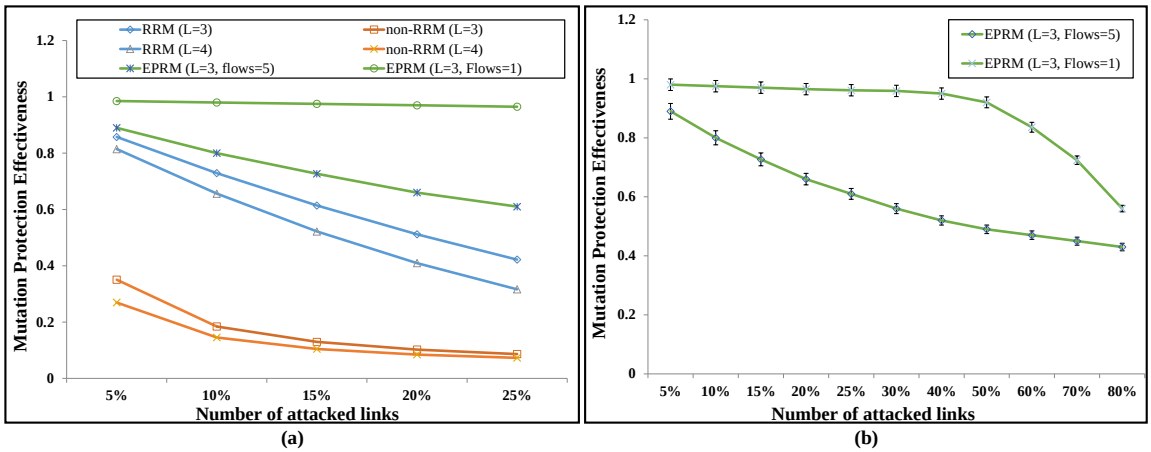



Figure 3.5: (a) MPE comparison of RRM, non-RRM, and EPRM; (b) Extended MPE of EPRM for Single and multiple-flows

20% more efficient than RRM (when L=3 and flows=5) and 51% more efficient than tradition non-RRM approaches (with L=3 and flows=5, when 25% of the network is under attack). The effectiveness of EPRM seems to decrease as number of flows with-in the network increase. This justifies the fact that potentially EPRM can be considered as an ideal candidate for mission critical services. Although it can operate in proactive mode, but as the network size becomes larger (which is the case in real life), its overhead can increase. Therefore, the ideal operational choice for EPRM is its utilization for mission critical services in defensive/reactive mode.

Figure. 3.5 (b) shows the extended evaluation of EPRM, where even if *80%* links are under attack, more than *45%* flows are safely transmitted for multiple flow mode as far as network is not fragmented, which potentially means unless adversary attacks the whole infrastructure. The proposed approach reactively changes a path/route if there is a blockage in any intermediate link. The *5%* degradation of flows in single-mode experiment is due to the reaction time of the approach, which is solely dependent on how frequently the members of *Selected Mutation Set* send update to the source (about the drop rate from them to their successors). Assuming that adversary is also mutating attacked links ($attack\_set \subset MS$ at any interval) will not reduce the effectiveness, as online phase of the proposed approach have tendency to rapidly

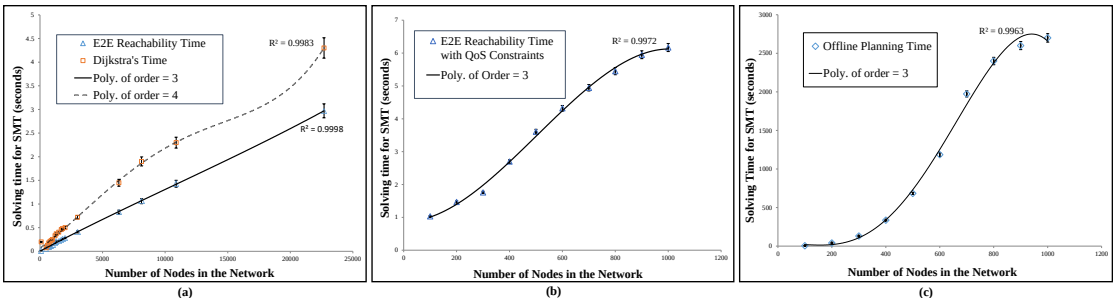



Figure 3.6: (a) Evaluation of resilient E2E reachability approach, (b) Time Required by SMT for E2E reachability with QoS constraints,(c) Time required by SMT for off-line computation of Candidate Mutation Set

mutate between multiple choices, unless $attack\_set \supseteq MS$, which means the whole infrastructure is under attack.

In figure. 3.5 (b), the MPE graph for multiple flows (flows=5) remains intact whereas for the same number of attacked links, the graph for single flow declines rapidly. Multiple flow graph includes cumulative MPE for all of the flows, even if one degrades rapidly, if other flows have less degradation the overall MPE will decrease less rapidly.

### 3.6.5 Overhead of SMT formalization for E2E Reachability

We have used simulation based experiments for the rest of the evaluation using large network topologies. We claim that the proposed formalism for reachability is efficient then previously proposed approaches [39, 52]. We also compare our approach with Dijkstra's algorithm [36], the most efficient algorithm available for shortest path calculation. Fig. 3.6 (a) shows the time required by SMT for finding set of potential neighbors/peers (a single logical path with $l$=3), which a source can contact for help if under attack. We can see that the time required by SMT increases when the network size increases, but E2E reachability formalization is still efficient than Dijkstra's algorithm. Note that E2E reachability formalization doesn't not incorporate the QoS or security constraints, and only finds a satisfiable solution for reachability, a path/route through which a source can reach destination under divers situation.

### 3.6.6 Evaluation of Off-line Phase

Fig. 3.6 (b) shows the time required by SMT for E2E reachability with QoS constrains. The time overhead in this case increases many fold, as compared to the E2E reachability without *QoS Constraints*, due to the large number of constraints (as SMT has to explore more choices to find a satisfiable solutions). Although the overhead increases with the networks size, but it is still feasible for practical life, as this computation is done as a part of off-line phase in EPRM protocol. Fig. 3.6 (c) shows the time required to compute complete *Mutation Set*.

### 3.6.7 Evaluation of Online Phase

Maintaining the quality of service and defending an infrastructure efficiently depends on the deployed run-time defense mechanisms. All of the above steps of the protocol are suppose to run off-line repetitively, after certain period of time. Therefore, little lag or overhead is acceptable, but the run-time defense mechanisms should be much more efficient. We compare the efficiency of online phase of our approach with recent approach in MTD domain [39]. Due to the huge scale differences we separately present RRM graphs.

Figure 3.7 (a) shows the SMT solving time for route selection using RRM in a network that have 5 consecutive flows, whereas, w is the number of intervals that the new route should not repeat [39]. We can see that the required time increases when the network size increases, especially when the number of routers in the network reaches 300.

We use same statistics to evaluate our approach. We choose 5 nodes as source at the same time to transmit their flows, and use intermediate routers and peers to forward their traffic to the desired unique destinations. We repeat experiment 10 times and calculate the average time taken by a source for complete and distinct route selection. Note that we do not rely on human selection to configure manually

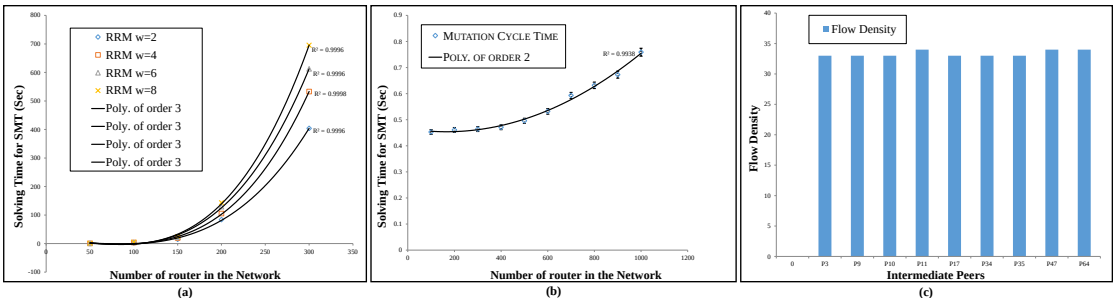



Figure 3.7: (a) Time required by RRM for Route selection, (b) Time required by SMT for on-line computation of Mutation Cycle, (c) Flow density of utilized intermediate peers

and select for what period of time a route should not repeat, rather we rely on our algorithm to minimize the route intersection, and evenly distribute traffic on all links. As algorithm can efficiently find the mutation cycle, therefore we do not rely on manual selection. Figure 3.7 (b) shows the efficiency and overhead of online phase for the proposed approach. We calculate the duration for *Mutation Cycle* computation for different size of networks and in each network the cardinality of MS is five at most, which means MS contains at most five virtual paths. Note that the aim of online-phase of the protocol is not to just select a virtual path (set of intermediate peers), rather to select and mutate the possible sequences of peer in non-orderly fashion for different set of flows. Experimentation shows that the overhead of the proposed approach is significantly less than RRM, which makes it suitable for practical usage, as mutation is performed in milliseconds as a result of the liveness response. The main reason for less overhead of our approach is that it is distributed in nature and all the computations are neither performed online nor by a single centralized controller.

We also perform experiments to analyze whether the flow density is evenly distributed over the links, or not. The proposed approach aims to maximize the even distribution of flows, in order to defend the maximum percentage of flows and to deceive reconnaissance. Fig. 3.7 (c) shows the distribution of flows over the peers and intermediate links. The network (with nodes=100 and edges=256) used for the

investigation, only contains disjoint routes/paths for reaching destination.

## 3.7 Conclusion

In this chapter, we addressed the limitations of existing route mutation approaches and proposed to extend route mutation architecture by considering end-hosts as routing elements. The resultant architecture exhibited significant advantages over existing approaches, in terms of run time efficiency, and resiliency against persistent attacks. The evaluation have showed that our algorithm is 40% more efficient than the Dijkstra's algorithm, 20% more efficient than RRM (in worst-case scenario) and more than 45% flows can be protected in multi-flow mode, even when the adversary is attacking 80% of the links. This makes EPRM more suitable for mission critical applications, when service availability and routing resilience for fraction of flows (such as emergence responding, communication in critical infrastructures, etc.) is the foremost objective. Ideally, our model if integrated with ToR infrastructure can address the issues of security/defense against DDoS attacks (by integrating adaptiveness) and can enhance the efficiency of ToR infrastructure.

In future we aim to: (1) incorporate additional operational constraints, (2) further analyze and evaluate the effectiveness of the end-point routing agility under adaptive attack models based on game theory, (3) deploy the proposed approach to other related architectures e.g. wireless ad-hoc networks, and (4) extend the existing approach to defend against reconnaissance attacks. We also notice absence of fairness for utilization of the resources in multi-flow cases which perhaps is the reason for rapid drop of MPE when more then 50% links are attacked, therefore, in future (5) we also aim to exploit the potential of centralized SDN controller for fair distribution of resources (e.g. bandwidth) among peers. Finally, (6) we aim to develop and integrate quantitative metrics in our model to measure how exposed/vulnerable an organization (or end-host) is against modern DDoS attacks, and quantify that how much resilience our approach can offer to a certain end-host or an organization.

# CHAPTER 4: Nature-Inspired Defense Approach Against Insider Threats

## 4.1 Backgroun & Motivation

According to the recently published studies on insider threat assesment [17][18], the biggest challenges faced by cyber-security community today, comes from insider threats. During 2017, the reported number of incidents in 159 organizations were 3,269, with insider attacks accounting for 76% of total cyber attacks. With ever increasing incident numbers and high affiliated cost (avg. $500K/incident), there is an imminent need to address the challenges faced by insider threat detection and response mechanisms [49].

The ultimate objective of forensic and cyber security community, is to effectively deter large range of cyber threats in real-time and autonomous manner. While in reality, the existing state-of-the-art solutions, only address the facet of plethora of these issues. The focus of research community, in the last decade, has been towards developing Security Information and Event Management (SIEM), or Security Event Management (SEM) systems. These systems have tendency to efficiently collect event related information (in the form of logs) from network devices (e.g., firewalls) or operating systems. This event based information is then fed into analytical tools, for detecting malicious activities via analyzing correlations between event logs, or signature-based testing. Finally, these systems generate alarm to update the security analyst, who is solely responsible for checking the legitimacy of an on existing threat (going alarm). The next step for a security analyst is to either report a detected threat (in case it is legitimate) to the authorities who have permission to change access authorization, or implement security policies in semi-automatic/manual ways, to cope with the ongoing threat. In most of the organizations, security analysts only

report IT facilities, and do not have control over tuning access control policies, adding another barrier and time lag, which could be beneficial for adversaries.

For detailed understanding we construct time-line comparison, between an insider and the operational capability of existing deterrence mechanisms (c.f. figure 4.1). For an insider, the attack can be divided into four phases (red-line in figure 4.1). In the first phase, the aim of the attacker is to gain access to the information/asset. In the second phase, the attacker performs initial steps for stealing or transferring data. It could involve using an external drive, a secure SSH connection establishment to a remote server, or connection to a local machine. This is the stage which can raise flags or alarms which must be timely detected by deployed defense mechanisms. Third stage is when the attack is launched and is in process. This stage could take from several seconds to minutes depending on the type of attack. Final, and fourth stage is when an attack is completed, which could last or overlap with any of the deterrence stages. The defending mechanism may only be in alert generation phase or an analyst might be investigating the detected anomaly. The figure also shows the operational capability of existing state-of-the art technologies, which eventually needs to rely on one or several human analysts to understand the root cause on an incident which has taken place already. The time-line shows what advantages an adversary has over existing mechanisms in place. The attackers mostly achieve their goals, while defenders are only in the detection phase. This technological gap provides an insider with asymmetric advantages, as only 70% of the reported organizations are using SIEM systems to deal with insiders, and among them only 25% are technologies to observe user behavior. Remain 30% organizations do not even have any alert generation or anomaly detection mechanisms in place to cope with insiders [17][18].

As for SIEM mechanism in place, these technologies (SIEM/ SEM) perform well for log and information aggregation purposes, but due to inherent limitations they lack capability of providing any assistance against an ongoing attack, in policy syn-

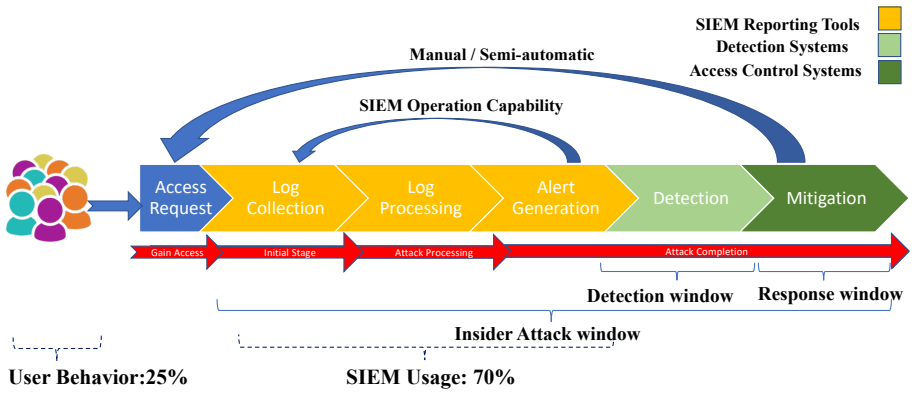



Figure 4.1: Threat Neutralization: Illustration of Insider Attack Time-line Vs. Insider Threat Deterrence Capability

thesis, or a legitimate user turned into an insider. The inability to integrate detection mechanism with policy synthesis procedure, and hence utilize *Threat Analytics* in a timely manner, are the main causes which do not allow state-of-the-art methods to deter against an ongoing malicious activity in real-time. These inherent limitations are expected be eradicated in the next generation architectures, by efficiently integrating *Machine Learning* based *Threat Analytics* with standalone monitoring technologies. These integrated units, can then be deployed to various nodes in the network/infrastructure focusing on different tasks. Whereas, little to none efforts are being spent in designing such autonomous integrated technologies, which can not only detect anomalous behavior of an insider, but can also trigger necessary actions against a threat in near real-time, via incorporation of actionable *Threat Intelligence* with policy enforcement procedure.

On the other hand, as a result of evolution, biological systems depict promising features and intrinsic appealing characteristics. These characteristics mainly include, inherent resiliency to failures and perturbations, adaptability to varying environmental conditions, and collaborative behavior based on a limited set of rules. Cellular regulation mechanism, in an example of previously mentioned attributes, which mitigates the perturbations (unusual protein concentration rate) at cellular level via signal transduction mechanism, by maintaining the optimal amount of protein concentra-

tion, due to abnormal behavior of certain proteins. On the basis this biological inspiration, we 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. In the scope of this article, the threat refers to a behavioral anomaly, specially in case when legitimate employee turns into a malicious insider, a scenario which makes it difficult for the existing tools to differential between normal and abnormal employee.

### 4.1.1 Problem Statement

As discussed earlier, SEM and SIEM are the state of the art solutions to deal with the threats that insider pose to an organization. This state of the art solutions to deal with insider threats involve either Security Event Management (SEM) or Security Information and Event Management (SIEM) systems. This technology works efficiently, when it comes to the collection of event related information, but due to the unavailability of any internal support for *logic of causation, reputation/risk metrics, and learning*, it does not provide any hint about ongoing insider attacks, based on user behavior anomalies [76]. Second main limitation is the involvement of human decision maker in the phase of policy implementation [76]. Despite the presence of state of the art monitoring and analysis tools, by the time human administrators take actions the damage is already done. The average response time of state of the art tools is between 13-15 minutes to raise an alarm, which later (response time) is extended by the human response time to take a reactive measure (by implementing reactive policy) [77][60].

Hence the focus of this thesis is to eliminate the inherent limitations and present a cellular regulation inspired policy regulation framework, by learning from the principles of nature, in which monitoring, analytic and policy regulation mechanisms are integrated to thwart against an insider.

### 4.1.2 Contributions

Our first main contribution is defining a temporal criteria to detect behavior based insider threats. Second major contribution of this article is the integration of threat analytics in policy regulation mechanism, which results in our proposed bio-inspired framework. Our third main contribution is to formalize the problem of access control as *state transition problem* such that the formal tools can be leveraged for access control synthesis. Finally, our fourth main contribution is to leverage formal tools (i.e., theorem provers) to implement policy regulation problem, not only to verify the correctness of access control policy, but also to thwart against an insider attack in timely manner.

## 4.2 Related Work

The following table summarizes our findings in the literature. In our evaluation we conduct a four-property check against each contribution. First, we check if a contribution belongs to anomaly detection domain or signature based detection domain. Second, we evaluate if a proposed method/contribution considers behavioral attributes in detection mechanism or not. The third most important thing we consider in our literature review is whether a proposed method incorporates, risk, trust, or reputation based metrics in access control problem. Finally we evaluate a contribution against whether the proposed method, links risk based analysis to policy regulation process or not. As our aim is to figure out if a proposed method can be effectively used to deter against known/unknown insider attacks. Hence, proposed evaluation criteria, specially, consideration of behavioral attributes, risk attributes and automated regulation mechanism forms the basis of our evaluation criteria. Due to the nature of contributions in 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 [45]. Due to the involvement of non-

Table 4.1: Existing techniques and limitations

| Approach | Signature | Anomaly | Behavioral Attributes | Risk/Reputation Indicator | Policy Regulation |
|---|---|---|---|---|---|
| Agraotis et. al. (2016) | ✓ | ✗ | ✗ | ✗ | ✗ |
| Bishop et. al. (2016) | ✓ | ✗ | ✗ | ✓ | ✗ |
| Oliver et. al. (2012) | ✗ | ✓ | ✗ | ✗ | ✗ |
| Chen et. al. (2015) | ✗ | ✓ | ✗ | ✓ | ✗ |
| Ted et. al. (2013) | ✗ | ✓ | ✓ (single indicator) | ✗ | ✗ |
| Zhang et. al. (2014) | ✗ | ✓ | ✓ (single indicator) | ✗ | ✗ |
| Rashid et. al. (2016) | ✗ | ✓ | ✓ | ✗ | ✗ |
| Legg et. al. (2017) | ✗ | ✓ | ✓ | ✗ | ✗ |
| Nissanke et. al. (2004) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Aziz et. al. (2006) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Sudip et al. (2006) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Ma et. al. (2010) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Liang et. al. (2012) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Feng et. al. (2017) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Le et. al. (2020) | ✗ | ✓ | ✓ | ✗ | ✗ |

technical factors for understanding behavior of a legitimate individual/employee, this domain faces rare and unique challenges, as compared to the challenges faced by the other detection domains [70]. 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 complex, making an *insider threat* one of the biggest existing challenge in cyber-security domain [28, 18].

### 4.2.1 Signature-based Insider 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 [21]., proposes a tripwire

solution based on the policies defined over alarming behaviors, and attack-patterns, to predict/detect actions that are indicators of insider threat [70]. The proposed approach neither provides any understanding about how the result of detection will reinforce the policy regulation, nor any real life threat test dataset based experimental evaluation. IBM SIEM solution, QRadar, works on the same principles, via incorporation of offenses (signatures) implementation. These signatures based offense repository is then used to detect threats in general [8].

Attack trees based approaches, to detect insider threats, have also been proposed in literature [24]. The authors leverage 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 to compute minimum cut set. Minimum cut set, once computed against each scenario, can then be used to help designing countermeasures. The approach has significantly high dependency on accuracy of modeled process (attack graph), whereas such models can only incorporate know vulnerabilities, and are not useful when it comes to unknown threats. The proposed approach also do not provide any basis to account for behavioral anomalies, and access regulation.

### 4.2.2 Anomaly-based Insider Threat Detection Systems

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

The focus of these systems is towards 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 propose, as using such indicators can enhance the chances of understanding psycological state of an insider for detection purposes [20, 65].

One such approach which utilizes non-technical indicators, to predict individuals behavior is proposed by Brdiczka et al. [26]. The authors use a game dataset (World of Warcrafts), along with activity related information from social network to evaluate their approach. Although it they provide grounds for developing a detection mechanism to incorporate non-technical indicators, but in reality the relevance between a game player and an insider is closer to none due to the differences in nature of attributes, motivation and setting in which an individual behave [70]. The proposed research also do 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 anomaly. Chen et al. [30], are the first to propose a formal framework based on probabilistic theory to deal with insider threats. The authors use belief based *Bayesian* modeling approach to first construct belief about the intention of an insider to attack, and then use probabilistic model checking to calculate the probability of an attack by a potential insider. The proposed approach suffers with fundamental limitations when it comes to calculating the probability values and Markove Desision 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 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 dealing 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 MDP model. Where in reality the actions of an insider do not follow Bernoulli's distribution, as this only allows outcome to have two possibilities (yes/no), hence cannot predict complex unknown behavior (as insider attacks are series of unknown events). The approach also do not use technical attributes,which

could enhance the chances of detection [70].

Another such approach proposed by Brdiczka et al. [26], also leverages automated technique, and incorporate non technical attributes (by collecting sensitive data, i.e., social network surfing) for the analysis. The aim of the authors was to increase detection accuracy by incorporating the non-technical indicators, whereas, the evaluation could only deliver 82% of accuracy. Although the approach performs reasonably batter than existing approaches, but unavailability of mapping mechanism from detection to policy regulation, technical challenges, and low accuracy makes it highly impractical for real life usage [70].

In literature, some methods mainly focus on specific threat detection rather than a generic detection method. For instance, Zhang et al. [66], analyze document access pattern to understand users intention, based on the contents of the document. The authors propose to construct profiles of all users and define anomaly as a deviation of current access pattern from the history profile. The proposed method is focused towards monitoring only a single indicator (access file type), resulting in capability to only analyze specific type of insider threat, i.e., information leakage. Whereas according to recently publised research and technical reports, combining multiple attributes for detection purposes can enhance the detection accuracy [28, 17, 70].

Another similar approach which works on the same concept has been proposed by Senator et al., which observes an insider based on his/her database access behavior [78]. Although the authors incorporate multiple attributes to deal with the low signal-to-noise ratio challenge, they do not provide any information about how their detection results can be leveraged to tune policy against an insider, and their approach finally rely on security manager/analyst for necessary (access control reporting/tuning) actions.

Legg et al. [56], proposes Principle Component Analysis (PCA) based solution to analyze behavioral anomalies by observing online activities of the employees. The

authors propose to construct user activity vectors on hourly basis and build a 24-hour activity matrix for employees. This activity matrix is passed onto PCA module to calculate distance between activity vectors to classify users in to different groups identifying the insiders with the high variance. These methods require a user-defined threshold/criteria to classify the data into certain number of groups. Hence, limiting the ability of a security manager/analyst to interpret information and trigger necessary actions in a timely manner to deal with an ongoing threat. Another limitation of the approach is that it clusters the identical users in the same group, which makes it difficult to integrate it with policy regulation mechanism, as traditional policies are defined in hierarchical manner and all identified malicious individuals may not belong to one hierarchy, hence, highlighting the need for automated policy optimization/synthesis process.

Finally, Rashid et al. [67], propose a detection method based on Hidden Markov Models (HMM). The proposed method learns the normal (behavioral) profiles of employees and analyze deviations from the normal profiles, to detect insider threat. The normality is considers as sequential events. in the context of the paper. Although, the proposed approach performs well while learning from data that is sequential in nature, but the complexity of the problem and computational cost for training the models increases as the number of states increase, hence, effecting the efficiency of the proposed approach in real life scenarios.

### 4.2.3 Policy Regulation in RBAC

RBAC is the most intensively used architecture in access control domain. Though it has certain benefits, but this control architecture does not allow automated synthesis of security policies at run time. Due to this inherent limitations, several extensions to this architecture have been proposed over the recent years [29, 41, 37] to eliminate this problem. These extension in RBAC propose to integrate risk/trust notion in access control mechanism. Although, these approaches propose to tune access policy against

the changes in risk or trust levels, they neither provide a comprehensive details about how the risk or trust values are calculated, nor how the access control policy can be synthesized satisfying all constraints in a fully automated manner.

Sudip et. al [29], propose to associate a trust interval to each roles, 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 very fundamental property of RBAC, according to which the user is assigned a role according to his/her function in the hierarchy of the organization, as compared to trust levels.

Feng et al. [41], proposes 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. [37], in which permissions are labeled with risk and trust thresholds. If the trust of the user requesting permission balances the risk of the action, the permission is granted. Although these approaches incorporate notion of trust and risk in access control mechanism, but they do not provide any clear answers to the most fundamental questions in insider threat deterrence domain, such as: *how the trust and risks are is computed*, and *how the risk can be minimized on run-time.*

Ma et. al. [61], proposes to assign roles with confidence level, 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 certain role) trying to access an assets is calculated. However, the the proposed approach do not provide any way to mitigate insider threats, as the parameters representing trust and risk are not dynamic and are not linked with user's behavior. In addition, the authors also do not provide any experimental evaluation of the proposed mechanism.

In [72], the authors propose to assign users with a certain budget depending upon their roles in the organization. They also assign cost factor to each access permission. Through an authorized role, when a user makes a request to access an asset, the

value of the budget is consumed (as there is a cost affiliated with permission) and access is granted only if cost is under the available budget of the user. The authors claim that, this approaches can force individuals to spend their budget in a more cautious manner. However, the proposed approach may provide great advantages to disgruntled employee. For instance, a legitimate employee turned into a malicious insider, may not care about spending his all budget to steal critical file/assets, hence, leaving the organizational assets vulnerable [70].

Similar approaches which aim at minimizing risk exposure have also been proposed in the literature [64, 23]. For instance, Nissanke et. al. [64] propose an approach for risk analysis in which permission set is labeled with the risk values, and role hierarchy is organized based on the risk. The author aim at assisting a security analyst such that he/she may only assign permissions to the roles after considering the risk affiliated with the permission. The proposed approach do not allow to maintain a role hierarchy, which is somewhat unrealistic, as roles in an organization are defined and required to maintain in an hierarchy and RBAC allows inheritance in a hierarchical manner.

Aziz et. al. [23], propose a method to optimize risk exposure against the evolution of the system. 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. At any time number of obligations can increase or decrease against a user. Although the authors incorporate notion of risk in the system, but do not provide information about calculation of risk, behavioral analysis, and fulfillment check against imposed obligations.

### 4.2.4 Commercial Tools

On the other hand, commercially available products (by Beta Systems, Oracle, IBM and SAP) in the market for insider threat analytics and management, do incorporate notion of risk [11, 12, 14, 13]. The focus of these products is towards analysing the access usage of permission with high risk by closely monitoring, auditing, and generating alerts for assisting security analysts. Although the notion of risk, one way

or another, exists in these products, however, these risk based analytic is not used in policy regulation or implementation process. This limits the capability of the defenders and shifts the scale in the favour of adversaries. With this motivation in mind, our aim to propose an integrated system, which is not only capable of performing threat analytic by incorporating behavioral indicators, but can also efficiently optimize the access control policy. In the forthcoming section, we discuss our inspiration which allows us to develop the design of an integrated deterrence system.

## 4.3 Cellular Regulation via Signal Transduction

The evolution of morphological (structural) features and all the operational capabilities of an organism is highly dependent on genes and are controlled at interacellular scale [32, 79, 68]. Genes and their by products (proteins) are the active players in this controlled mechanism, which play in a programmed way to perform various tasks within an organism. Genes are the informative regions of Deoxyribonucleic acid (DNA), and can be classified as per their involvement in different organismic activities. When a certain region of DNA (gene) is active, information flows from genetic to proteomic level as complex processes of transcription and translation. These regions (genes) on DNA influence each-other and the nature of influence can be categorized as activation, or inhibition. This interaction among genes and their by product forms a complex dynamic network which is referred as Gene/Biological Regulatory Network (GRN/BRN). Hence, these networks of complex interaction are responsible for maintaining normal functionalities within a cell against any external or internal perturbation after receiving signals from cellular receptors [68]. Figure 4.2 provides an overview of traditional representation of DNA and its coiled and unfolded format. As a first step towards understanding about the nature of interactions in this complex network we briefly discuss the biological phenomenon of blood pressure regulation.

### 4.3.1 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 futher 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 Falko Dressler [38]. 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“ [70].

### 4.3.2 Mapping: Biological DNA Vs. Access DNA

To proceed further with the idea of integrating auto-resiliency characteristics in access control architecture, there must exist some analogy between biological and cyber elements/parameters. The following figure 4.2 provides an illustrative representation of our mapping concept. We propose the concept of *Access DNA* as compared to biological DNA, which contains information about cyber parameters/elements. In *Access DNA* each region corresponds to specific functionality. For instance, sub-regions (genes) in control related category over *Access DNA* are attributed to different policy configurations. Whereas sub-regions in *Threat Analytical* category may correspond to threat related information (threat vectors), against a user/machine. Finally, the regions which receive activity related information, in the form of activity vectors (via log collection) can be refereed as receptor regions (genes).

These regions over *Access DNA* representing cyber parameters, influence each-other in the same way they influence in a real DNA, forming a complex regulatory network. For instance, receptor regions play a crucial role in increasing/decreasing (activating/inhibiting) the the values/levels of the regions which are related to threat analytics. By interpreting information in *Threat Analytical* regions, it can be determined if a system is perturbed or not. As a result if the system is perturbed (means if a user is behaving maliciously and threat has crossed corresponding threshold), then there is an imminent need to activate a certain gene (policy configuration), to deal with an ongoing threat. Selecting a specific policy configuration impacts the access/actions available to users/machines, hence, influencing directly or indirectly *Threat Analytical*

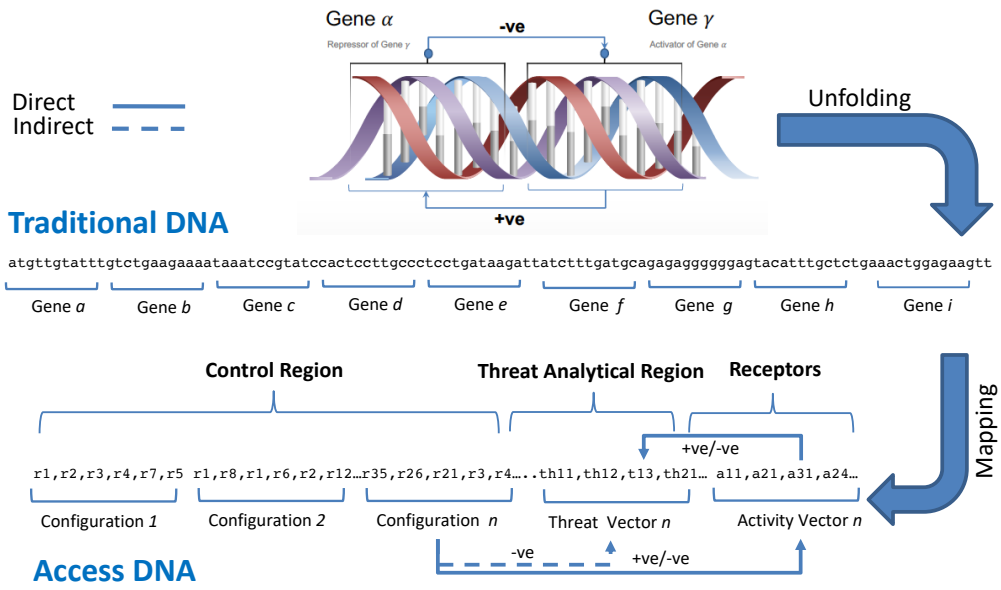



Figure 4.2: Our Proposed Concept of Access DNA & Analogy with Gene Regulation

region, to eliminate the perturbation. This mapping concept of representing cyber parameters, as *Access DNA* forms the basis of our proposed framework and is very crucial for understanding of cyber regulation.

In the next section, we discuss the functionality of each region in the context of our proposed framework, and provide formalization of their functionality as well.

## 4.4 Cellular Regulation Inspired Mapping & Proposed Framework

In this section, we propose cellular regulation inspired framework to deal with insider threats. 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. Though, with minor modification the proposed framework can also be extended to deal with external threats, but we limit our focus to the internal/insider threats only, as we can observe the variations in insider's behavior via activity log collection. Figure 4.3 presents a detailed description of our proposed framework along with side by side comparison

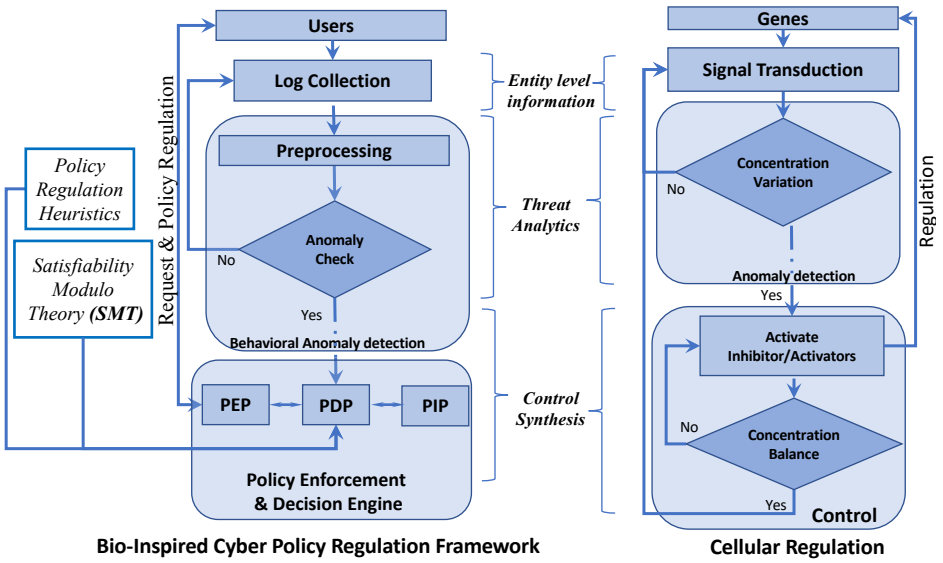



Figure 4.3: Mapping of Cellular Regulation Mechanism to the Proposed Cyber Policy Regulation Framework

to the working principles of cellular regulation process. We discuss each component in details in the forthcoming discussion.

### 4.4.1 Sensing Module

As we established earlier that in cellular regulation, sensing is performed by receptor genes, which monitor continuously the concentration levels of proteins, and disperse that information internally (within a cell), so that other task specific genes can be made aware of the perturbations in their environment. In our case this functionality is replicated by a sensing mechanism, which collects event logs of user's activities and convert them in to activity vectors, $av_{ij} : av_{11}, av_{12}, av_{13}.....av_{nm}$. Whereas there can be $\boldsymbol{m}$ users, having $\boldsymbol{n}$ activity vectors $\boldsymbol{(av)}$.Rather than dispersing information to every entity in the system (as in case of cellular regulation), we propose to collect

logs and activity vectors 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 do not consider behavioral attributes, hence, in this research we propose to integrate behavior based threat analytics with access control mechanism by following the principles of cellular regulation. Most of the recent recommendations [20, 65, 26]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 [70].

### 4.4.2 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 activities 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 main issue we address during feature engineering phase is the time interleaving based aggregation of logs against different activities. This aggregation allows us to understand an event as a sequence of activities over time, as an attack is not just an individual activity, rather it is a sequence of malicious activities over time. The following snapshot 4.4 shows the resultant dataset. The second most important challenge is to map each log entry from merged dataset of all users to a specific string of bits, such that only one row encompassing all the activity attributes against a user is active at a time. Such mapping is necessary to understand an individual's activity behavior at a certain time. We use OneHotEncoding method for this purpose. As OneHotEncoding method allows us to distinctly recognize each activity

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| | date | user_x | activity | web | Device |
| 0 | 1/4/10 7:35 | DTAA/JSH05 | Logon | http://force. | Connect |
| 1 | 1/4/10 7:35 | DTAA/LBD09 | Logon | http://force. | Connect |
| 2 | 1/4/10 7:35 | DTAA/CGT01 | Logon | http://force. | Connect |
| 3 | 1/4/10 7:35 | DTAA/DAW0 | Logon | http://force. | Connect |

Figure 4.4: Log Aggregation

| A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| | date | user_x | activity | user_y | web | DTAA/AAA0371 | DTAA/AAC0344 | DTAA/AAC0599 |
| 0 | 1/4/10 7:35 | DTAA/JSH05 | Logon | DTAA/DBM0 | http://force. | 0 | 0 | 0 |
| 1 | 1/4/10 7:35 | DTAA/LBD09 | Logon | DTAA/DBM0 | http://force. | 0 | 0 | 0 |
| 2 | 1/4/10 7:35 | DTAA/CGT01 | Logon | DTAA/DBM0 | http://force. | 0 | 0 | 0 |

Figure 4.5: Construction of Activity Vectors via OneHot-Encoding Method

vector as a unique event, differentiating it from the other ensemble. This results in the construction of unique activity vectors ($av_{ij}$). The following snapshot 4.5 provides the overview of the resultant dataset containing activity vectors, where each user is assigned a column. The final challenge is to map these event based activity vectors to a temporal variable, such that this information can be utilized by machine learning methods. Therefore, we convert time-stamps into a cyclic temporal variable which varies between 0-24Hrs. Such temporal representation of activities 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). We also provide the resultant dataset for the convenience of readers and further analysis [9].

***Anomaly Check:*** 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.

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 (activity vectors) 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. Hence, our main contribution is to define behavioral anomoly as significant deviation in a user's activity vectors, from its previous profile. To measure deviation our proposed metric rely on computing euclidean (or any other) distance between activity vectors. Following formalization shows our mathematical representation of behavioral deviation.

$$\boxed{\delta^{b}_{Avg} = ||\gamma\left(\frac{\sum_{j}^{n}(D^{j}_{Avg})}{n}\right) - X^{i}_{Avg}||} \tag{4.1}$$

$\delta^{b}_{Avg}$ represents average behavioral deviation of an activity vector ($av_i$), from normal profile in terms of Euclidean distance. Whereas, $X^{i}_{Avg}$, represents average distance of $av_i$ for which anomoly detection is under consideration, from other activity vectors. The fraction ($\frac{\sum_{j}^{n}(D^{j}_{Avg})}{n}$) represents the weighted mean of the average distances ($D^{j}_{Avg}$) of each activity vector ($av_j$) from other activity vectors (where i$\neq$ j). This allows us to calculate average deviation of an activity vector from average normal/previous behavior, rather than relying on a distance or similarity metric, resulting in high number of false positives. Another benefit of this weighted average deviation is the incorporation of $\gamma$ factor, which allows user to define a strict/relaxed criteria against

an anomaly. As for anomoly, we define anomaly ($\mathcal{A}$) as significant deviation from average weighted mean as follows:

$$\boxed{\mathcal{A} : X^i_{Avg} \gg \gamma\left(\frac{\sum_j^n (D^j_{Avg})}{n}\right)} \tag{4.2}$$

Once average weighted deviation of a behavioral vector is calculated, as a next step towards finding anomalies, we propose to use unsupervised clustering methods to predict labels (as the data is not labeled) and classify the behavioral vectors on the basis of aforementioned metric. Now as the data grows, the complexity of clustering algorithms grows alongside. Hence to avoid delays is decision making procedure, once the classification is performed and cross validated, we use supervised machine learning algorithms, to train models against a user's profile (long term or short term behavior), and then predict their activities every time they try to access resources.

Such learning profiles (containing trained classifiers) can be stored in pickle formats (.pkl), and can be called efficiently to compute run-time threat analytics for knows scenarios, e.g., if user is significantly deviating from normal behavior, threat analytic module considers it as user behavioral 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 section 4.5.

### 4.4.3 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 [33]. 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 prospective threat level has altered. 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, using the notion of a *8-tuple*: $\mathcal{M} = (S, s_0, r_{ij}, dec_k^{ij}, \mathcal{T}_{ij}^k, C(\mathcal{T}_{ij}), \hookrightarrow, \delta)$:

- 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;

- $r_{ij}$ is the rule defining access of an asset $j$ against a user $i$
- $\mathcal{T}_{ij}^{k}$ is the threat level/impact of a given request by user $i$ to access asset $j$, which can be calculated as: $\mathcal{T}_{ij} = L_i \times Imp_j$. 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 $\mathbb{B}$, such that it represents the following mapping $r_{ij} \mapsto \mathbb{B}$;
- $\mathrm{C}(\mathcal{T}_{ij})$ is a set of constraints over the threat vector/values;
- $\hookrightarrow$ is a finite set of transitions such that: $\hookrightarrow \subseteq (S \times \mathbb{N} \times \mathbb{N} \times \mathbb{B})^2 \times \mathrm{C}(\mathcal{T}_{ij})$;
- $\delta$ is a finite set of transition rules which maps $\mathrm{C}(\mathcal{T}_{ij})$ 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 ($A_j$).

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

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

$$\boxed{(s, r_{ij}, \mathcal{T}_{ij}, dec_k^{ij}) \xrightarrow{\mathcal{T}_{ij} >= \theta;\ dec_k^{ij} = 0} (s', r_{ij}, \mathcal{T}_{ij}, dec_k^{ij'})}$$

$$\boxed{(s, r_{ij}, \mathcal{T}_{ij}, dec_k^{ij}) \xrightarrow{\mathcal{T}_{ij} < \theta;\ dec_k^{ij} = 1} (s'', r_{ij}, \mathcal{T}_{ij}, dec_k^{ij''})}$$

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:

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

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 $\mathcal{T}_{ij}$ becomes higher than the acceptable threshold ($\theta$), 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. In the following section we discuss the evaluation and effectiveness of our approach.

### 4.4.4 Implementation of PRM as PRTS

In this section we briefly discuss the implementation related details of PRTS. The following algorithm 3, shows, how PRTS switches states once it finds a change in users threat related attribute. The algorithm starts with accepting inputs from three different files, containing information about access permissions (***constraints.txt***), policy configurations (**config.txt**), and threat related information (***threat.log***), which is computed and updated by *Threat Analytic Module*. Since we are using Python API for implementation, these files should be in a format which is interpret-able by Python

Z3 API. In case if an SMT solver does not accept regular expression (which Z3 does), the input files should be in SMT-LIB format. Along with input files, a set of constraints is also required to implement user specific criteria for synthesizing policy. In our case the set $\psi$ contains $\eta$, and bounded constraints over higher order unbounded functions $(dec_{ij}, r_{ij}, \mathcal{T}_{ij})$. Function $SYN$ then computes a satisfiable configuration as per the criteria mentioned in 4.3 or 4.4, if it exists, otherwise returns to hazard mode (configuration of the policy), where by default the access to any request is denied.

The algorithm runs until it finds a satisfiable configuration which neutralizes all the threat vectors $(\Delta(\mathcal{T}_{ij} = 0))$ against all users. To better understand the nature of parameters and schematics of the algorithm, we incorporate the following figure 4.6 for the readers. The figure shows the nature of activity vectors based on various behavioral attributes which construct the receptor part of CyberDNA. This information, along with our previously proposed behavioral anomaly metric $(\mathcal{A})$ helps modifying part of CyberDNA which contains the information about threat levels against different access permissions. Threat related information is then fed into the above presented algorithm along with the satisfiability constraint $\eta$, and policy configurations. Eventually PRM uses SMT solver to synthesize the satisfiable version of the policy.

In the forthcoming section, we also formalize a variation of $\eta$, which can be fed into the same algorithm, for the purposes of safety verification of an access control system. In the next section we test our proposed approach in rigorous manner to understand its strength an weaknesses.

## 4.5 Evaluation of Bio Inspired Policy Regulation Framework

### 4.5.1 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 [47, 16]. The dataset contains the log activities for one thousand employees and contains five different types of insider threat scenarios,

**Algorithm 3** Computing Policy Configuration

**while** $\Delta(\mathcal{T}_{ij}) \neq 0$ **do**
  $P = \bigvee_k(p_k) \bigcup \mathcal{H} \longleftarrow$ {set of policy configurations}
  $Cp_k \in P \longleftarrow$ current configuration
  $\mathcal{T}_{ij} \mapsto \mathbb{R} \longleftarrow$ {set of threat values}
  $dec_{ij} \mapsto \mathbb{B} \longleftarrow$ {set of decision vectors against $r_ij$}
  $\psi \longleftarrow$ {set of constraints}
  **procedure** SYN(P, $Cp_i$, $\mathcal{T}_{ij}$, $r_{ij}$, $dec_{ij}$, $\psi$ )$\longleftarrow$ {Evaluation of property 4.3/4.4}
    (Result, Model) = check_satisfiability(SYN())
    **if** Result==SAT **then**
      $Cp_i$=Model[$p_i$]
      Select($Cp_i$)
    **else**
      Select($\mathcal{H}$)
    **end if**
  **end procedure**
**end while**

for detailed description of the scenarios we refer our reader to the dataset details (*file: scenario.txt*) in [16] . In the context of this thesis, 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 [16], we construct our own threat test dataset which can be accessed and used for machine learning purposes [9]. 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

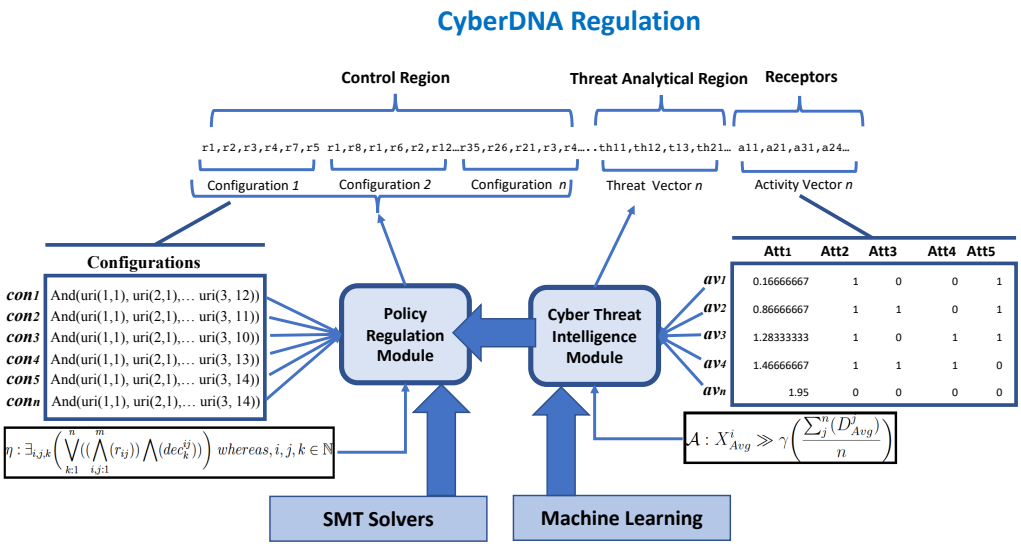


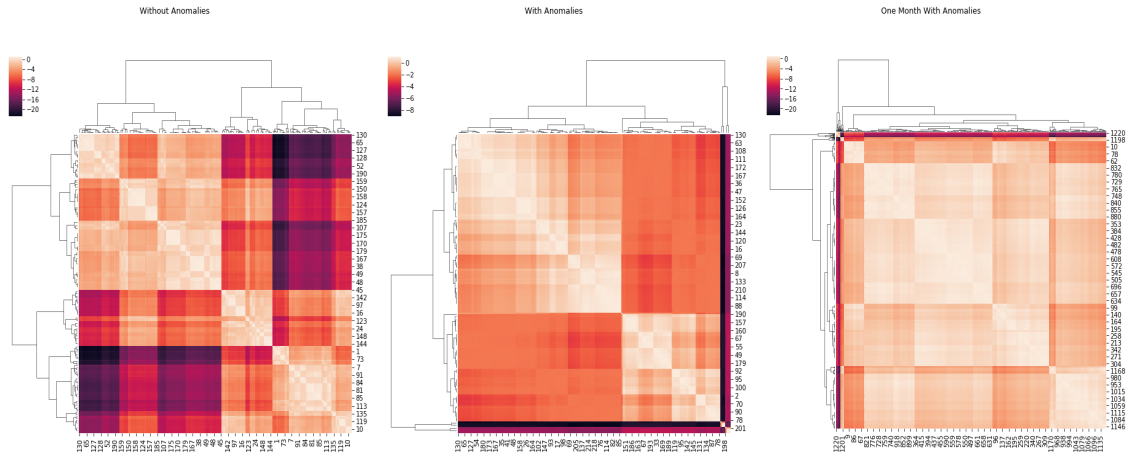

Figure 4.6: Schematics of Cyber-Regulation

hours 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.

As our first goal is to establish a criteria and efficiently classify a behavioral anomaly from the normal activity vector against a user, towards this objective, we extract the profile of the user (**DTAA/KEE0997**), from our refined dataset, which has highest frequency in *scenario 1* threat test dataset [16].

#### 4.5.1.1 Unsupervised Classification based Analysis: Measuring Accuracy of Labeling

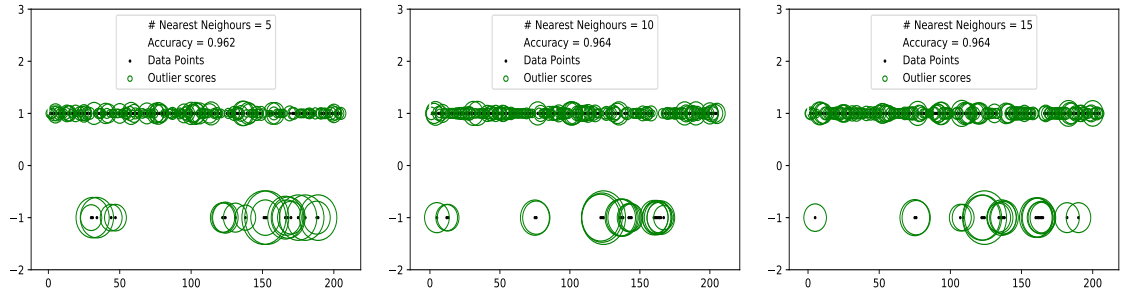
In the first phase of evaluation, as mentioned earlier, based on our proposed average deviation metrics 4.1, we calculate pairwise distances of behavioral/activity vectors of the user under consideration (**DTAA/KEE0997**) . Once we have pairwise distances for given activity data set containing activity vectors, we use unsupervised machine learning methods, for classification of activity vectors into normal and abnormal categories. We use DBSCAN, and linkage based hierarchical clustering methods for this purpose. Figures 4.10 show the clusters which we were able to obtain by applying un-




(a) (b) (c)

Figure 4.7: Clustering of behavioral threat vectors: (a) weekly analysis without anomalies, (b) weekly analysis with anomalies, and (c) monthly analysis with anomalies

supervised classification methods on our refined dataset. We perform three different experiments, in the first experiment, we aim to understand if the anomalous samples ranked as per our proposed anomaly metrics will be clustered together or not. For this purpose we deliberately remove anomalous samples, and perform clustering. Figure 4.10 (a) shows the clustering of weekly data samples in which we deliberately remove anomalous samples (activity vectors). The figure shows two major clusters which are in accordance to the activities performed by the user during normal office hours and during late hours. Now to test if the anomalous samples will be clustered separately or not, we add anomalous samples back to the weekly data and perform clustering again. Figure 4.10 (b) shows that after adding anomalous samples, the cluster distribution changes, and anomalous samples are cluster separately (tightly together), which can be termed as an anomalous class. In the third experiment we extend data set to consider monthly samples along with 15% noise, and anomalous samples. Figure 4.10 (c) shows that our proposed metric is accurate enough to separate the anomalous samples, from the normal ones, even in the monthly data containing noise.

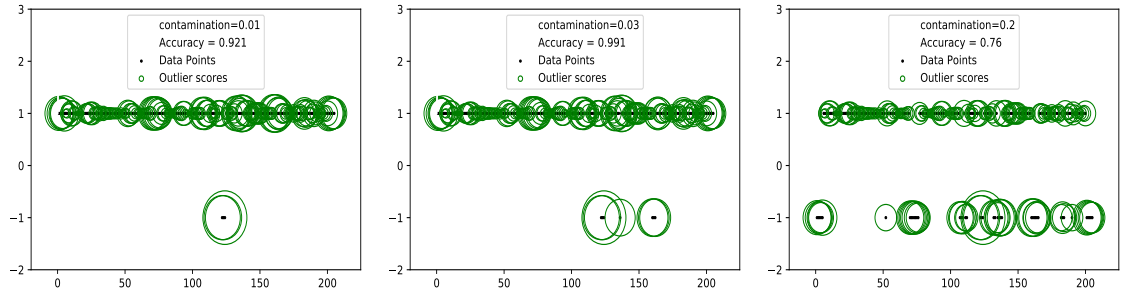



(a) (b) (c)

Figure 4.8: Effect of neighboring element on the accuracy of classification.

Parametric Analysis of Unsupervised Classification

Our next objective is to find out the optimal values for parameters which could impact the accuracy of of classification. We use *Local Outlier Fitting* algorithm to understand how the classification could be impacted. Using LoF, two key parameters can be tuned to understand if our proposed metric is resilient enough or not, these parameters are: number of nearest neighbors, and contamination value. No. of nearest is a parameter which defines minimum number of samples in the smallest cluster, whereas contamination represents percentage of outliers. Following figures show the impact of both parameters and the change in accuracy accordingly. Figure 4.8 shows that the accuracy remains same even if we change number of neighbors in a cluster. In the second experiment we change the contamination value, and resultant figures 4.9 shows that the clustering/classification is indeed sensitive to the contamination value. For instance, figure 4.9 (b) shows, if we set its value at 0.02 which means the number of anomalous samples in the dataset are 2%, we acheive 99% accuracy, as all the anomalous samples are correctly identified. But as we increase its value to 20%, the LoF algorithm punishes the distances less intensively and hence allowing the increase in the number of false positives resulting in decrease of accuracy (c.f. figure 4.9 (c)). Therefore, we recommend using a lower value for contamination parameter.




(a) (b) (c)

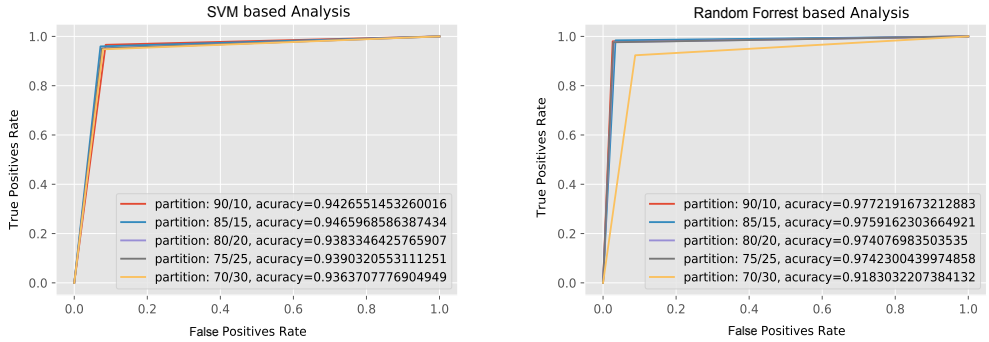
Figure 4.9: Effect of contamination on the accuracy of Classification

#### 4.5.1.2 Supervised Learning based Analysis: Measuring Accuracy of Behavioral Predictions

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 figure. 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 $\approx 98\%$ 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.

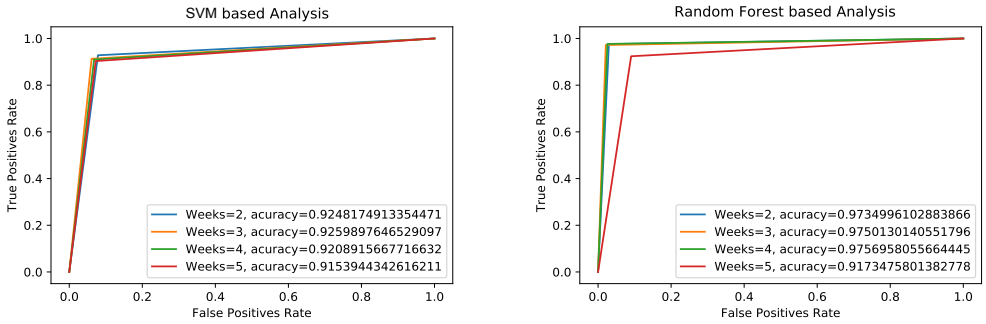



(a) (b)

Figure 4.10: Evaluation of train-test sample size partitioning to find optimized partitioning 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)

Figure 4.11: Temporal evaluation to find optimal number of weeks for prediction analysis

### 4.5.2 Evaluation of Policy Regulation Module

#### 4.5.2.1 Experimental Setup

In this section, we conduct a detailed evaluation of our proposed *Policy Regulation Module* to assess the efficiency and real life viability of our proposed approach. This evaluation helps us to understand and answer the fundamental questions, which we set forth earlier (i.e., can automation of access control synthesis mechanism be rendered in real time, with reasonable time overhead). We conduct three different types of experiments. First we analyze the time complexity of PRM to synthesize a correct configuration given a set of rules in a policy and threat related information from *Threat Analytic Module*. Second we analyze the time required for PRM to verify safety property. Finally we demonstrate how, PRM can effectively select transit from one state to another, by accurately selecting the correct policy configuration.

To compute satisfiable configuration of access control policy, and to verify safety/hazard property, we use Z3 as an underlying SMT solver [34]. For implementation of PRM we use Python API, integrated with z3 solver. All the experiments are conducted on Core i5 machine with 2.4GHz processor and 16Gb of memory.

#### 4.5.2.2 Synthesis based Evaluation of PRM

The result presented in the figure 4.12 (a) shows how the time require to compute a satisfiable configuration by PRM varies with the increase in the number of rules (in a policy). The performance analysis of PRM shows that the time complexity trajectory remains stable/flat as number of access rules increase, having a spike as the number of access rules approach 350, overall averaging around $\approx 0.15 Secs$. We run experiments for different number of time and average our results over 100, 250, 500, 1000 iterations, to make sure that our result have more accuracy.

Figure 4.12 (c) depicts the idea of state transition within the system. The underlying example contains 62 configurations of a policy. The figure shows, at t=0, the active configuration is *C57*, which remains active up untill t=2, where we intentionally raise the threat value associated to a certain access, and PRM deactivates that configuration after reading the updated values from TAM, and a new configuration *C58* is activated, which remains active till t=6. Our experiment results show that PRM can effectively mitigate any insider threat by synthesizing access control policy in real-time with in the fraction of seconds.

#### 4.5.2.3 Safety Verification of PRM

Subsequently, another benefit of using formal methods is the ability to verify the correctness of the proposed system over entirety of its state space. We also formalize two different versions of safety property, and verify it using Z3 theorem prover. We use existential and universal quantifiers to explore all states of the system and verify the possibility, that no user should have access to a certain asset, while its risk of accessing the corresponding asset, increases beyond bearable threshold ($\theta$).

$$\forall_k \bigvee_{k:1}^{n} \left( \exists_{(i,j)} \left( \left(r_{ij}\right) \bigwedge \left(dec^{ij}\right) \bigwedge \left(\mathcal{T}_{ij} < \theta_{ij}\right) \right) \right) \tag{4.3}$$

$$\boxed{\forall(i,j,k)\bigvee_{k:1}^{n}\Big(\big(r_{ij}\big)\bigwedge\neg\big(dec^{ij}\big)\bigwedge\big(\mathcal{T}_{ij}>\theta_{ij}\big)\Big)} \tag{4.4}$$

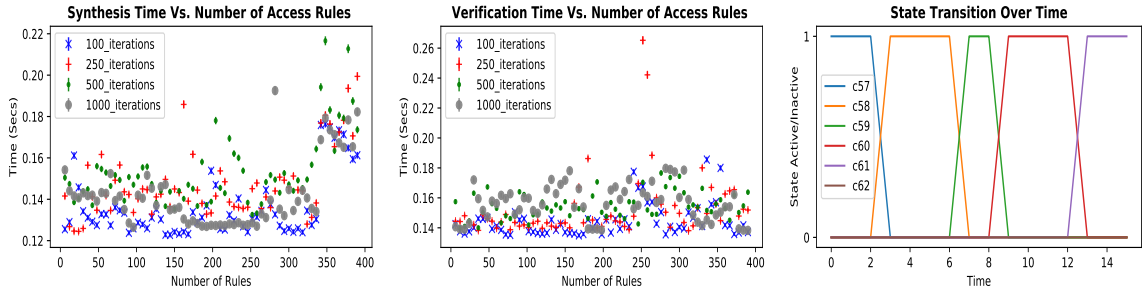
The first property (4.3), using existential quantifier ($\exists$) over bounded variables to express a diagnose-able system state, checks whether for all $k$ states (configurations of given policy), there exists a rule against each $k_{th}$ configuration, which can be selected by *Policy Regulation Module* where the threat ($\mathcal{T}_{ij}$) has crossed bearable threshold ($\theta_{ij}$) and user (i), still has access to critical asset (j) ($\big(dec^{ij}\big)$. The result of this property is the configuration itself, if any such configuration exists. If the system is well established then the result of this property would be UNSAT, indicating that there is no such scenario where high risk access is granted.

The second property uses a different universal approach to conduct a safety check: where for all configurations (k) which PRM can select at a given time, if the threat value goes above the threshold ($\mathcal{T}_{ij} > \theta_{ij}$), then allowable configurations should contain a rule in which asset (j) can be accessed by user (i). Whereas the result of second property is only true and false, as it conducts a safety check. Our approach provides a flexible way, where not only safety of the system can be verified with click of a button, by using property 4.4, but a error prone configuration (which could result due to human error), can also be diagnosed using property 4.3.

We verify both properties over increasing sample space of rules, and calculate average time complexity of PRM for verifying the correctness of the system. We run experiments for 100, 250, 500, and 1000 iteration and then average the results. Figure 4.12 (b) shows that the trajectory of time complexity to verify safety properties, remain stable with the average value around $\approx 0.155 Secs$.

## 4.6 Conclusion

In this chapter, we present a novel *Cellular Regulation inspired Access Control Policy Regulation Framework*, which not only observes the variation in the environment




(a) (b) (c)

Figure 4.12: (a) SMT time to calculate Satisfiable Configuration, (b)SMT time to verify Safety/Hazard Property, (c) State Transition Example in Policy Regulation Module

(behavioral anomalies of insiders), but also triggers necessary responses (policy regulation) to defend against an insider threat. Our first main contribution is to present formalization of a feature (activity) based temporal criteria for understanding behavioral anomalies. Our second main contribution is integration of threat analytics with policy regulation module (resulting in bio-inspired policy regulation framework). Our third main contribution is to model policy regulation problem as state transition system and formally present its semantics to incorporate threat analytics against an insider. Our fourth major contribution is to leverage SMT based theorem prover (Z3) to implement *Policy Regulation Module* module, which not only provides us a way of synthesizing correct configuration (given a set of constraints), but also provides an analysts a way to verify the correctness of access control system. We also present formalization of safety/hazard properties of such systems in section 4.5.2.3. Finally we rigorously test our proposed system to evaluate its efficiency and effectiveness.

For evaluating the efficiency and effectiveness, we use real-life threat dataset provided by CERT. Our evaluation illustrates that we were able to achieve an accuracy of 99% while correctly labeling the dataset using our proposed behavioral anomaly metric. As for prediction based analysis, we were able to achieve 98% (92% in worst case scenario) in case of behavioral anomaly detection. The evaluation related to PRM shows that proposed system was able to synthesize the regulated policy in fraction of

a second ($\approx 0.15 Secs$). Whereas the reported time for a well trained security analyst is $\approx 15 minutes$ to investigate and report an insider threat using state of the art SIEMs technologies (i.e., Splunk), let alone the time required by IT staff to revoke or limit the access of a certain user. 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] https://cran.r-project.org/web/packages/rismed/index.html.

[2] https://github.com/akshaynagpal/rgscholar.

[3] https://r2lab.inria.fr/.

[4] https://ropensci.org/packages/.

[5] https://www.elsevier.com/solutions/scopus/features/api.

[6] https://www.planet-lab.org/.

[7] https://www.singerinstruments.com/resource/what-are-genetic-mutation/.

[8] IBM QRadar, SIEM.

[9] www.dropbox.com/s/rerwekvuji12icm/logon_hotencoded_cleaned_data.csv?dl=0.

[10] www.incapsula.com.

[11] Access Risk Management. Technical report, 2012.

[12] Application Access Controls Governor. Technical report, 2012.

[13] Identity and Access Governance. Technical report, 2012.

[14] Resource Access Control Facility (RACF). Technical report, 2012.

[15] *Framework for Improving Critical Infrastructure Cybersecurity*. National Institute of Standards and Technology, USA, 2014.

[16] CERT threat test dataset. CERT, 2016.

[17] Defending Against the Wrong Enemy. Technical report, SANS Insider Threat Survey, 2017.

[18] Insider Threat Report. Technical report, CA Technologies, 2018.

[19] International Standards Organization ISO/IEC 27005: 2008. *Information Technology-Security Techniques-Information Security Risk Management*. International Standards Organization, Geneva, Switzerland, 2008.

[20] McCormac. A, Parsons. K, and Butavicius. M. Preventing and Profiling Malicious Insider Attacks. Technical report, Defense Science and Technology Organization, 04 2012.

[21] Ioannis Agrafiotis, Arnau Erola, Michael Goldsmith, and Sadie Creese. A tripwire grammar for insider threat detection. In *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, MIST '16, pages 105–108. ACM, 2016.

[22] Ian F. Akyildiz, Özgür B. Akan, Chao Chen, Jian Fang, and Weilian Su. Interplanetary internet: State-of-the-art and research challenges. *Comput. Netw.*, 43(2):75–112, October 2003.

[23] Benjamin Aziz, Simon N. Foley, John Herbert, and Garret Swart. Reconfiguring role based access control policies using risk semantics. *J. High Speed Netw.*, 15(3):261–273, July 2006.

[24] M. Bishop, H. M. Conboy, H. Phan, B. I. Simidchieva, G. S. Avrunin, L. A. Clarke, L. J. Osterweil, and S. Peisert. Insider threat identification by process analysis. In *2014 IEEE Security and Privacy Workshops*, pages 251–264, May 2014.

[25] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA, 1999.

[26] Oliver Brdiczka, Juan Liu, Bob Price, Jianqiang Shen, Akshay Patil, Richard Chow, Eugene Bart, and Nicolas Ducheneaut. Proactive insider threat detection through graph learning and psychological context. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pages 142–149, 2012.

[27] Scott Camazine, Nigel R. Franks, James Sneyd, Eric Bonabeau, Jean-Louis Deneubourg, and Guy Theraula. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA, 2001.

[28] Dawn M Cappelli, Andrew P Moore, and Randall F Trzeciak. *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley, 2012.

[29] Sudip Chakraborty and Indrajit Ray. 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 '06, pages 49–58, New York, NY, USA, 2006. ACM.

[30] Taolue Chen, Florian Kammüller, Ibrahim Nemli, and Christian W. Probst. A probabilistic analysis framework for malicious insider threats. In *Human Aspects of Information Security, Privacy, and Trust*, pages 178–189. Springer International Publishing, 2015.

[31] A. Clauset., C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703, November 2009.

[32] Eric H. Davidson and Douglas H. Erwin. Gene Regulatory Networks and the Evolution of Animal Body Plans. *Science*, 311(5762):796–800, February 2006.

[33] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960.

[34] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.

[35] Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[36] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[37] Nathan Dimmock, András Belokosztolszki, David Eyers, Jean Bacon, and Ken Moody. Using trust and risk in role-based access control policies. In *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies*, SACMAT '04, pages 156–162, New York, NY, USA, 2004. ACM.

[38] Falko Dressler. Self-organized network security facilities based on bio-inspired promoters and inhibitors. In *Advances in Biologically Inspired Information Systems*, pages 81–98. Springer, 2007.

[39] Qi Duan, E. Al-Shaer, and H. Jafarian. Efficient random route mutation considering flow and network constraints. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 260–268, Oct 2013.

[40] Muddassar Farooq and MyiLibrary. *Bee-inspired protocol engineering : from nature to networks*. Natural computing series. Springer, Berlin, DE, 2009.

[41] F. Feng, C. Lin, D. Peng, and J. Li. A trust and context based access control model for distributed systems. In *2008 10th IEEE International Conference on High Performance Computing and Communications*, pages 629–634, Sept 2008.

[42] L. Garber. Denial-of-service attacks rip the internet. *Computer*, 33(4):12–17, Apr 2000.

[43] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[44] Moti Geva, Amir Herzberg, and Yehoshua Gev. Bandwidth distributed denial of service: Attacks and defenses. *IEEE Security & Privacy*, (1):54–61, 2014.

[45] Iffat A. Gheyas and Ali E. Abdallah. Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. *Big Data Analytics*, 1(1):6, Aug 2016.

[46] Fida Gillani, Ehab Al-shaer, Samantha Lo, Qi Duan, Mostafa Ammar, and Ellen Zegura. Agile virtualized infrastructure to proactively defend against cyber attacks. In *INFOCOM 2015*, volume 1, pages 270–280 vol.1, April 2015.

[47] J. Glasser and B. Lindauer. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*, pages 98–104, May 2013.

[48] J. N. Haack, G. A. Fink, W. M. Maiden, A. D. McKinnon, S. J. Templeton, and E. W. Fulp. Ant-based cyber security. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*, pages 918–926, April 2011.

[49] Ponemon Institute. Cost of insider threats: Global. 2018.

[50] J. H. Jafarian, E. Al-Shaer, and Q. Duan. Adversary-aware ip address randomization for proactive agility against sophisticated attackers. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 738–746, April 2015.

[51] Jafar Haadi Jafarian, Ehab Al-Shaer, and Qi Duan. Openflow random host mutation: Transparent moving target defense using software defined networking. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN, pages 127–132, New York, NY, USA, 2012. ACM.

[52] JafarHaadi Jafarian, Ehab Al-Shaer, and Qi Duan. Formal approach for route agility against persistent attackers. In Jason Crampton, Sushil Jajodia, and Keith Mayes, editors, *Computer Security, ESORICS 2013*, volume 8134 of *Lecture Notes in Computer Science*, pages 237–254. Springer Berlin Heidelberg, 2013.

[53] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell. Catch me if you can: A cloud-enabled ddos defense. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 264–275, June 2014.

[54] M. S. Kang, S. B. Lee, and V. D. Gligor. The crossfire attack. In Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13. , DC, USA: IEEE Computer Society*, pages 127–141, 2013.

[55] Min Suk Kang, Soo Bum Lee, and Virgil D. Gligor. The crossfire attack. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 127–141, Washington, DC, USA, 2013. IEEE Computer Society.

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

[57] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[58] Xiaoyun Liao, Paul Lochhead, Reiko Nishihara, Teppei Morikawa, Aya Kuchiba, Mai Yamauchi, Yu Imamura, Zhi R. Qian, Yoshifumi Baba, Kaori Shima,

Ruifang Sun, Katsuhiko Nosho, Jeffrey A. Meyerhardt, Edward Giovannucci, Charles S. Fuchs, Andrew T. Chan, and Shuji Ogino. Aspirin Use, Tumor PIK3CA Mutation, and Colorectal-Cancer Survival. *N Engl J Med*, 367(17):1596–1606, October 2012.

[59] Zhenyu Liu, Marta Z Kwiatkowska, and Costas C Constantinou. A swarm intelligence routing algorithm for manets. In *Communications, Internet, and Information Technology*, pages 484–489, 2004.

[60] LogRhythm. Logrhythm ueba. 2017.

[61] J. Ma, K. Adi, M. Mejri, and L. Logrippo. Risk analysis in access control systems. In *2010 Eighth International Conference on Privacy, Security and Trust*, pages 160–166, Aug 2010.

[62] Douglas C. MacFarland and Craig A. Shue. The sdn shuffle: Creating a moving-target defense using host-based software-defined networking. In *Proceedings of the Second ACM Workshop on Moving Target Defense*, MTD '15, pages 37–41, New York, NY, USA, 2015. ACM.

[63] Neustar. Worldwide ddos attacks & cyber insights. May, 2017.

[64] Nimal Nissanke and Etienne J. Khayat. Risk based security analysis of permissions in rbac. In *WOSIS*, 2004.

[65] J. R. C. Nurse, O. Buckley, P. A. Legg, M. Goldsmith, S. Creese, G. R. T. Wright, and M. Whitty. Understanding insider threat: A framework for characterising attacks. In *2014 IEEE Security and Privacy Workshops*, pages 214–228, May 2014.

[66] Zhang R., Chen X., Shi J., Xu F., and Pu Y. Detecting insider threat based on document access behavior analysis. In *Web Technologies and Applications*, volume 8710, pages 98–104. Lecture Notes in Computer Science, Springer, 2014.

[67] Tabish Rashid, Ioannis Agrafiotis, and Jason R.C. Nurse. 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 '16, pages 47–56, New York, NY, USA, 2016. ACM.

[68] Usman Rauf. A taxonomy of bio-inspired cyber security approaches: Existing techniques and future directions. *Arabian Journal for Science and Engineering*, Feb 2018.

[69] Usman Rauf, Fida Gillani, Ehab Al-Shaer, Mahantesh Halappanavar, Samrat Chatterjee, and Christopher Oehmen. Formal approach for resilient reachability based on end-system route agility. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, MTD '16, pages 117–127, New York, NY, USA, 2016. ACM.

[70] Usman Rauf, Mohamed Shehab, Nafees Qamar, and Sheema Sameen. Bio-inspired approach to thwart against insider threats: An access control policy regulation framework. In *Bio-inspired Information and Communication Technologies*, pages 39–57. Springer International Publishing, 2019.

[71] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.

[72] F. Salim, J. Reid, E. Dawson, and U. Dulleck. An approach to access control under uncertainty. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 1–8, Aug 2011.

[73] Verizon Enterprize Solutions. Data breach investigations report. 2015.

[74] Angelos Stavrou, Angelos D. Keromytis, Jason Nieh, Vishal Misra, and Dan Rubenstein. Move: An end-to-end solution to network denial of service. In *Proceedings of the Internet Society (ISOC) Symposium on Network and Distributed Systems Security (SNDSS)*, San Diego, CA, February 2005.

[75] Ahren Studer and Adrian Perrig. The coremelt attack. In *ESORICS, 2009: 14th European Symposium on Research in Computer Security, Saint-Malo, France, September 21-23, 2009. Proceedings*, pages 37–52. Springer Berlin Heidelberg, 2009.

[76] TechTarget. Siem technology primer: Siem platforms have improved significantly. 2012.

[77] TechTarget. User behavioral analytics tools can thwart security attacks. 2015.

[78] E Ted, Henry G Goldberg, Alex Memory, William T Young, Brad Rees, Robert Pierce, Daniel Huang, Matthew Reardon, David A Bader, Edmond Chow, 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*, pages 1393–1401, 2013.

[79] Louis C. Thomas and Richard d' Ari. *Biological feedback*. CRC Press, 1990.

[80] Huangxin Wang, Quan Jia, Dan Fleck, Walter Powell, Fei Li, and Angelos Stavrou. A moving target {DDoS} defense mechanism. *Computer Communications*, 46:10 – 21, 2014.

[81] Thomas N. Williams, Tabitha W. Mwangi, David J. Roberts, Neal D. Alexander, David J. Weatherall, Sammy Wambua, Moses Kortok, Robert W. Snow, and Kevin Marsh. An Immune Basis for Malaria Protection by the Sickle Cell Trait. *PLoS Med*, 2(5), May 2005.

[82] Paul Wood, Christopher N. Gutierrez, and Saurabh Bagchi. Denial of service elusion (dose): Keeping clients connected for less. In *34th IEEE Symposium on Reliable Distributed Systems, SRDS 2015, Montreal, QC, Canada, September 28 - October 1, 2015*, pages 94–103, 2015.

## 4.1 Appendix

### 4.1.1 Search Attributes for Metadata Analysis

| Survey | Swarm Intelligence | Immunology |
|---|---|---|
| Bio-inspired cyber security | Swarm intelligence cyber security | Immune inspired cyber security |
| Bio inspired cyber Security | Swarm intelligence network security | Immune inspired network security |
| bio inspired cyber defense | Ant colony optimization network security | immunology inspired cyber security |
| Bio-inspired network security | Ant colony optimization cyber security | immunology inspired network security |
| Bio inspired network security | ACO inspired network security | immune system inspired cyber security |
| Bio inspired security survey | ACO inspired cyber security | immune system inspired network security |
| Bio inspired security taxonomy | particle swarm optimization cyber security | HIS inspired cyber security |
| - | particle swarm optimization network security | - |
| - | PSO inspired network security | - |
| - | PSO inspired cyber security | - |

| Genetic Mutation | Biological Regulation |
|---|---|
| gene mutation inspired network security | gene regulation inspired network security |
| gene mutation inspired cyber security | gene regulation inspired cyber security |
| genetic mutation inspired cyber security | gene regulatory networks inspired network security |
| genetic mutation inspired network security | gene regulatory network inspired cyber security |
| - | cell regulation inspired network security |
| - | cell regulation inspired cyber security |