ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3862

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche: *Laboratoire Mathématiques, Informatique et Applications – Sécurité de l'Information*
Discipline: *Informatique*
Spécialité: *Informatique, Sécurité Informatique.*

Présentée et soutenue le : 09/11/2023 par :

Nouhad Sanoussi

Enhancement of SDN Security Based on Intrusion Tolerance and Security Games

JURY

Faissal OUARDI	*PES*	*Université Mohammed V, Faculté des Sciences, Rabat*	*Président*
Lahoucine BALLIHI	*PH*	*Université Mohammed V, Faculté des Sciences, Rabat*	*Rapporteur/ Examinateur*
Jalal LAASSIRI	*PES*	*Université Ibn Tofail, Faculté des Sciences, Kénitra*	*Rapporteur/ Examinateur*
Hicham BENSAID	*PH*	*Institut National des Postes et Télécommunications, Rabat*	*Rapporteur/ Examinateur*
Said EL HAJJI	*PES*	*Université Mohammed V, Faculté des Sciences, Rabat*	*Invité*
Ghizlane ORHANOU	*PH*	*Université Mohammed V, Faculté des Sciences, Rabat*	*Directrice de thèse*

Année Universitaire : 2023-2024

Dedication

To My Beloved Parents!!

Acknowledgments

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have made it through my master's degree!

I am grateful to the Laboratory of Mathematics, Computing and Applications - Information Security (LabMIA-SI), Faculty of Sciences, Mohammed V University in Rabat, under the supervision of **Mrs. Ghizlane ORHANOU** and **Mr. Said EL HAJJI**, for providing me with the opportunity to conduct my research and for all of the resources and support they provided.

First of all, I would like to express my sincere gratitude to my patient and supportive supervisor **Mrs. Ghizlane ORHANOU** Habilitated Professor from the Faculty of Sciences of Rabat who has supported me throughout this research project, and generously provided knowledge and expertise. I am extremely grateful for our friendly chats at the end of our meetings that helped me in my personal life, and your personal support in my academic endeavors.

From the bottom of my heart I would like to say big thank you to my co-supervisor **Mr. Said EL HAJJI** Professor of Higher Education from the Faculty of Sciences in Rabat for all his precious efforts, professional and pedagogical orientations. I have been extremely fortunate to be his student in the Master's journey and his Ph.D. student afterward that I could not have undertaken this journey without his support. I wish him a better and more relaxing retirement.

I would particularly like to express my deepest appreciation to **Mr. Faissal OUARDI**, Professor of Higher Education from the Faculty of Sciences in Rabat for agreeing to chair the jury for my thesis defense and for examining and judging this work. And for allowing me into his Lab's class, it truly has been a very, very good time in this experience.

Besides, I would like to acknowledge and give my warmest thanks to **Mr. Lahoucine BALLIHI**,

Habilitated Professor from the Faculty of Sciences of Rabat for having accepted to participate in this jury, and for agreeing to judge this work as a reporter. And for letting me supervise labs in "OS and Applications Security" where I could apply my Knowledge.

Also, I would like to express my sincere gratitude to **Mr. Jalal LAASSIRI**, Professor of Higher Education from the Faculty of Sciences, Ibn Tofail University, Kenitra for accepting to evaluate and examine this work as a reporter. And for giving me the opportunity to be a part of the professors' team in IICIA Master with the aim to strengthen my competencies.

Furthermore, I am deeply indebted to **Mr. Hicham BENSAID**, Habilitated Professor from the Institut National des Postes et Télécommunications for agreeing to be a jury member of this defense and for examining and judging this work as a reporter. I would like to express my gratitude to you and assure you of my deep respect.

Of course, I would like to thank all the professors at the Laboratory of Mathematics, Computing and Applications - Information Security. And I would like to express my thanks to all the members and research colleagues of the laboratory for their coffee discussions and for the creation of a friendly working environment that brings out my best performance.

I would also like to give special thanks to my parents, my family as a whole, and my friends for their continuous support and understanding when undertaking my research and writing my project. Your prayer for me was what sustained me this far.

And for my husband, thanks for all your support, without which I would have stopped these studies a long time ago. You have been amazing!!

Nouhad.

Abstract

The wide spread of the Internet leads to the expansion of networks and the increase of data. However, traditional network systems miss global visibility of the network state and have trouble deploying and maintaining coherent network-wide policies. It gets difficult to Maintain stable and robust network security with this complexity and weaknesses in integration. In addition, traditional security mechanisms of defense necessitate an enormous investment and an accurate study of the network to secure it effectively. The weak point of traditional network security solutions is their lack of a quantitative decision framework. To solve these issues, firstly SDN simplifies network management, offers flexibility, and makes communication networks easier. However, SDN faces several security challenges, since the controller is considered as a single point of failure that may return the whole network down in case of a security compromise. Indeed, the single point of failure is partially tackled by the use of multi-controller mechanisms. However, simply using these mechanisms cannot avoid compromising vulnerable controllers due to their visible nature. In fact, the attacker needs just much more time to compromise the whole system. Thus, the use of intrusion tolerance is necessary. Secondly, security games proved their efficiency in the second issue. This thesis focuses on modeling a game theoretic approach to formalize the attack-defense interaction by taking into account both internal and external attacks and analyzing the effect of Intrusion Tolerant Systems (ITS) on the payoff of both internal and external attackers and the defender, then helping in decision making. Besides that, we propose to approach the issue of intrusion tolerance in the SDN control plane by applying a Recovery Based model which assumes that as soon as a system comes online it is compromised; therefore, periodic restoration to a good state is necessary. Then, we aim to establish Moving Target Defense (MTD) that provides a proactive defense against adaptive adversaries. The goal of the MTD in the Dispatcher is to constantly shift between multiple controllers with diverse configurations in order to increase the uncertainty for the attacker, in effect, diminishing the information gathered from the control plan during the reconnaissance phase of a potential attack. Finally, we put in place probabilistic models that can contribute to the perception of the performance of self-cleansing intrusion tolerance in the SDN control plane.

Keywords: Software Defined Network, SCIT, Moving Target Defense, Multicontroller, Urn model, Network Security, Intrusion Tolerant System, Game theory, Attacker, Bayesian game.

Résumé

La large diffusion de l'internet conduit à l'expansion des réseaux et à l'augmentation des données. Cependant, les architectures de réseau traditionnelles manquent de visibilité globale sur l'état du réseau et ont des difficultés à déployer et à maintenir des politiques cohérentes à l'échelle du réseau. Cette complexité et ces faiblesses en matière d'intégration nuisent au maintien d'une sécurité de réseau stable et robuste. En outre, les mécanismes de défense traditionnels exigent un investissement considérable et une étude précise du réseau pour sécuriser efficacement l'infrastructure. La faiblesse des solutions traditionnelles est qu'elles ne disposent pas de méthodes de décision quantitative. Pour résoudre ces problèmes, le SDN simplifie tout d'abord la gestion du réseau, offre de la flexibilité et facilite les réseaux de communication. Cependant, le SDN confronte à plusieurs défis en matière de sécurité, car le contrôleur se considére comme un point de défaillance unique qui peut entraîner l'arrêt de l'ensemble du réseau en cas de compromission de la sécurité. En effet, le point de défaillance unique se résout partiellement par l'utilisation de mécanismes multi-contrôleurs. Cependant, la simple utilisation de ces mécanismes ne peut pas éviter de compromettre les contrôleurs vulnérables en raison de leur nature visible. En fait, l'attaquant a besoin de beaucoup plus de temps pour compromettre l'ensemble du système. L'utilisation de la tolérance aux intrusions est donc nécessaire. Deuxièmement, les approches fondées sur la théorie des jeux prouvent leur efficacité dans le deuxième problème. Cette thèse se concentre sur la modélisation d'une approche théorique des jeux pour formaliser l'interaction attaque-défense en prenant en compte les attaques internes et externes et en analysant l'effet des systèmes tolérants aux intrusions sur le gain de l'attaquant interne et externe et du défenseur, aidant ainsi à la prise de décision. En outre, nous proposons d'aborder le problème de la tolérance aux intrusions dans la couche de contrôle SDN en appliquant tout d'abord un modèle basé sur la récupération qui suppose que dès qu'un système est mis en ligne, il est compromis; par conséquent, une restauration périodique à un bon état est nécessaire. Ainsi, nous visons à établir une défense des cibles mobiles (Moving Target Defense - MTD) qui fournit une défense proactive contre les adversaires adaptatifs. L'objectif de MTD dans le répartiteur est de passer constamment d'un contrôleur à l'autre avec diverses configurations afin d'augmenter l'incertitude pour l'attaquant, ce qui a pour effet de diminuer les informations recueillies à partir de la couche de contrôle pendant la phase de reconnaissance d'une attaque potentielle. Enfin, nous mettons en place des modèles probabilistes qui peuvent contribuer à la perception de la performance de la tolérance aux intrusions auto-nettoyante dans la couche de contrôle SDN.

Keywords: SDN, SCIT, MTD, Multicontroller, Modèle de l'urne, Sécurité réseau, Système de tolèrance aux intrusions, Théorie des jeux, Attaquant, Les jeux Bayesiansh.

Contents

List of Figures

List of Tables

List of Acronyms

A

ACT	Adaptive Cluster Transformation
AI	Artificial Intelligence
APIs	Application Programming Interfaces
ARM	Adaptive Reconfiguration Module

B

BR	Best Response

C

CC	Command and Control
CN	Conventional Network
COTS	Commercial Off-The-Shelf
C-SCIT	Cloud-based Self Cleansing Intrusion Tolerance
CV	Cluster Vector

D

DDoS	Distributed Denial of Service
DISC	Dynamic Intrusion Signature Configuration
DNS	Domain Name System
DODAG	Destination Oriented Directed Acyclic Graph
DoS	Denial of Service

F

FW	Firewall

G

GTM-CSec Game Theoretic Model for Cloud Security

I

IDM Intrusion Detection Module

IDPS Intrusion Detection and Prevention System

IDS Intrusion Detection System

INSENS INtrusion-tolerant routing protocol for wireless SEnsor NetworkS

IoT Internet of Things

IP Internet Protocol

IPS Intrusion Prevention System

IT Information Technology

ITS Intrusion Tolerant System

ITSRP Intrusion Tolerant Secure Routing Protocol

J

JVM Java Virtual Machine

M

MAFTIA Malicious and Accidental Fault Tolerance for Internet Applications

MEC Mobile Edge Computing

ML Machine Learning

MTD Moving Target Defense

MVMP Multi-Version Multi-Path

N

NBI Northbound Interface

NE Nash Equilibrium

NIB Network Information Base

NOS Network Operating System

O

ONOS Open Network Operating System

OS Operating System

Q

QoS Quality of Service

R

RHSM Random Host and Service Multiplexing

RPL Routing Protocol

S

SBI Southbound Interface

SCIT Self Cleansing Intrusion Tolerance

SDN Software Defined Network

SITAR Scalable Intrusion Tolerant ARchitecture

T

TTCB Trusted Timely Computing Base

V

VM Virtual Machine

VRRP Virtual Router Redundancy Protocol

W

WAN Wide Area Network

WSN Wireless Sensor Network

1

Introduction

Network systems are of vital importance and become an indispensable part of our lives with the development of Machine Learning (ML), Artificial Inteligent (AI), and cryptocurrencies ...etc. Concurrently with Corona pandemic spread, the world suddenly found itself in front of machines; communications, work, commerce, and entertainment... briefly, people obliged themselves to change their habits with the aim to adapt to the pandemic and the quarantine.

Network security is an important concern for researchers and specialists in the domain. These last years registered a significant increase in the variety and frequency of cyber threats, that cost great damage to governments, society, organizations...etc [1]. The third quarter of 2023 saw a 6.5% increase in cyber attacks with 1,108 events [2]. Security issues get worse once such systems become open to general public life, to be used in communications, business, and even entertainment. To complicate matters further, the majority of users are such blind customers, not trained enough to cognize the nature, limitations, and breaches of these systems. As an example, Internet of Things (IoT) emerges as the newest technology to render people's life easier, IoT devices improve network connectivity, health...etc. Unfortunately, it becomes the best area of interest when it comes to attacks.

Furthermore, the complexity and the interconnected nature of information systems and networks make security problems profoundly complex [3]. Firstly, the complexity increases with the enormous revolution of both hardware and software. In fact, computers today become more and more powerful. As a consequence, the software running on these computers become more layered and complex.

Secondly, the systems' interconnected nature makes the administrator's task to control the network very difficult if not impossible, especially with distributed networks. The issue is summarized in the phrase "The truly secure computer is the one unplugged from the network" [3].

Security threats and defenses mechanisms

Attackers and Defenders and their motives

The term **attacker** is used in this dissertation to personalize a person or a group with malicious intent to compromise information systems. It could be a hacker, a cracker, a whacker...etc .
Primarily, the notion of the hacker was not malignant. **A hacker** [4] was a curious person about computer systems, he tried to understand and learn as much as possible by developing and improving software.
A cracker [4] is a person using his skills to attack information systems. Over time, the hacker took the same connotation as the cracker.
A phreaker [4] is a hacker focusing his skills on communication systems to access communication devices, to steal calling card numbers.
A whacker [4] is a novice hacker who attacks Wide Area Networks (WANs) and wireless networks.
A script/kiddie [4] is usually a young individual without programming skills using attack software that is freely available on the Internet and from other sources.
A cyber-terrorist [4] is an individual who works for a government or terrorist group that is engaged in sabotage, espionage, financial theft, and attacks on a nation's critical infrastructure.
Malicious users inside an organization. These types of users predominantly try to access unauthorized networks and systems. They can potentially cause great damage, by stealing secrets and customers' data or making modifications.
Hactivist [4] is a person or a group of persons that have motivations and justifications behind their activities. They target any organization that practices social injustice from their point of view.

The term **defender** is used to the person trying to find weaknesses in the information systems and defend them from attacks by employing different defense mechanisms and configuring constantly the systems to ensure confidentiality, integrity, and availability.
An ethical hacker [4] uses the skills of a hacker but without causing harm with the aim to find

weaknesses in the system and give recommendations to the owners in an attempt to defend their information system.

Security threats

Security threats and attacks can be defined as attempts to compromise information systems by targeting the basic principles of information systems security known as **CIA triad** which are: Confidentiality, Integrity, and Availability.

Compromising **confidentiality** means getting unauthorized access to the confidential system of information.

Compromising **integrity** is achieved by any modification of information by unauthorized users or unauthorized or unintentional modification of information by authorized users.

Compromising **availability** by launching Distributed Denial-of-Service attacks (DDoS) that impact the availability of online services.

Attackers use some steps described in frameworks like the Cyber Kill Chain framework, developed by Lockheed Martin (2022) [5], to launch attacks against their targets. The steps of this framework are :

Reconnaissance is a preliminary activity in which an attacker attempts to gather information about a target preparatory to launching an attack. It includes scanning the network from the inside or outside, finding vulnerabilities...etc.

Weaponization is the creation of malware to be used against an identified target.

Delivery can take the form of hacking into an organization's network and exploiting a hardware or software vulnerability to infiltrate it.

Exploitation, in this stage, attackers take advantage of the vulnerabilities they have discovered in the reconnaissance stage to further infiltrate a target's network and achieve their objectives.

Installation, attempting to install malware and other cyberweapons onto the target network to take control of its systems and exfiltrate valuable data.

Command and Control (CC), in this stage, the attackers communicate with the malware they have installed onto a target's network to instruct cyberweapons or tools to carry out their objectives.

Actions on objectives, by compromising confidentiality, integrity, and availability.

To properly defend an information system, it is necessary to understand the steps and techniques used by the attackers.

Defense mechanisms Vs ITS (briefly)

Several defense mechanisms have been developed for network security. The most popular and significant tools used to secure networks are Firewalls, Antivirus Software, Intrusion Detection and Prevention systems (IDPS), and Intrusion Tolerant Systems (ITS).

Firewalls [6] are devices or programs that control the flow of network traffic between networks or hosts that employ differing security policies. While firewalls are often discussed in the context of Internet connectivity, they may also have applicability in other network environments. For example, many enterprise networks employ firewalls to restrict connectivity to and from the internal networks used to service more sensitive functions, such as accounting or personnel. By employing firewalls to control connectivity to these areas, an organization can prevent unauthorized access to its systems and resources.

IDPS [7]: Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can in addition attempt to stop possible incidents.

Antivirus [8]: This tool is used to detect, identify, or remove malwares performing proactive detection, active detection, or reactive detection. That is, they detect a virus before it executes, during execution, or after execution. Identification and removal modules are more straightforward in their application; neither are of use until a malware has been detected.

The mechanisms above are considered traditional security mechanisms used to defend the systems reactively to known and detected attacks, however, these technologies are still ineffective against unknown and undetected attacks, then can not guarantee the total security of the system.

ITS: Intrusion tolerance is the ability of a system to continue operating (possibly degraded) after a successful attack. This means, designing a system in such a way that the harm caused by the intruder is restrained, further, automatically repaired, rather than shutting the system down and completely losing the service it provides.

Software Defined Network

What is it?

Since the emergence of new technologies such as cloud computing, virtualization, and the Internet of Things, the use of networks imposes several criteria such as high accessibility, less complexity, and dynamic management. Unfortunately, the complexity in traditional networks leads to a lack of global visibility of the network state, thus, suffers from reliability, scalability, flexibility, and manageability.

To solve the limitations and issues resulting from existing networks, the SDN architecture has emerged as a logically centralized control software [9], by decoupling the network logic from underlying forwarding devices. This architectural recomposition places the 'brain' of the network on a specialized central controller, enabling centralized management and a global view of the network. The data plane is composed of 'dummy' devices, forwarding packets based on rules specified remotely. These rules may be specified by the application running atop the controller and triggered according to packet-level extracted information [10].

SDN Planes

As depicted in Fig. 1 [11], SDN architecture consists of three main layers. The Data plane in the lowest level, the application plane in the highest level, and between them there is the control plane. The communication between these layers is maintained via Southbound Interface (SBI) and

Northbound Interface (NBI). The first one is located in data plane and ensures the communication between data and control plane, and the second, located in the control plane, ensures communication between control and application plane. In contrast, conventional networks (CNs) have the control plane, i.e., network control functions such as routing protocol implementations, running inside each network device to learn forwarding tables in a distributed fashion[12].

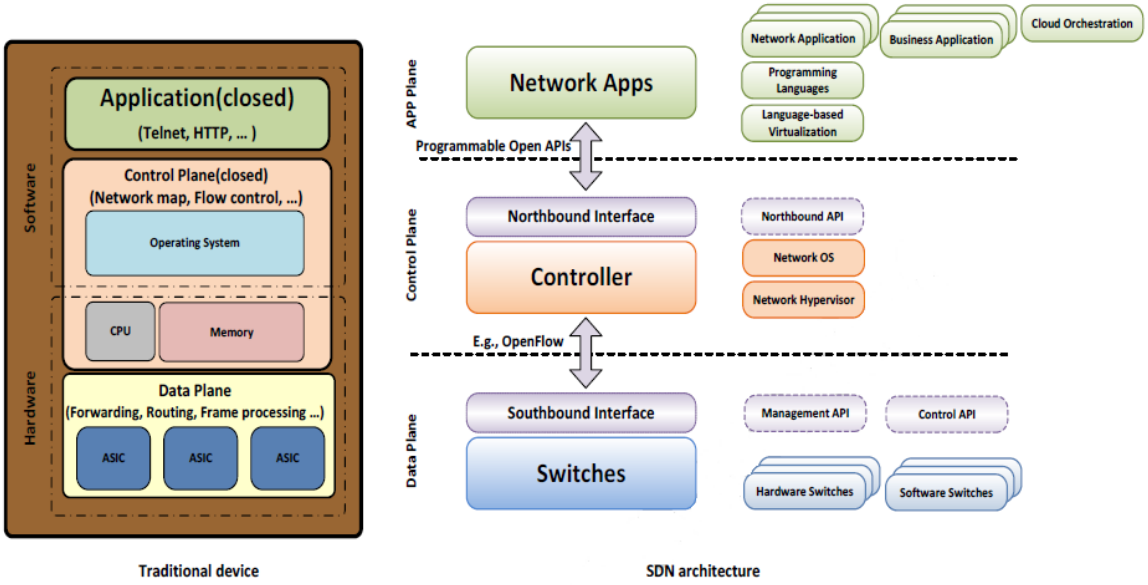

Figure 1: Component of traditional device and SDN structure.

- **Data plane:** [13] The data forwarding layer consists of numerous SDN switches, which are physically connected by wired or wireless media. Each switch is a simple device in charge of forwarding network packets and has a forwarding table, named the Flow Table, which contains thousands of rules that are used to formulate forwarding decisions.
- **Control plane:** In this plane, we find the controller which is the brain of SDN. It manages and controls the entire network. The control plane is composed of two components; network operating system (NOS) that acts as the SDN controller and applications which represent the software programs installed in the controller. Network could have more than one controller, so each controller is responsible to control a group of network switches, but may interfere with each other.
- **Application plane:** [11] is at the top of the SDN architecture, which includes all the applications that exploit the services provided by the controller in order to perform network-related tasks in various areas.

Security issues

Compared to traditional network architectures, SDN approach seems more robust to malicious attacks. Due to its nature design, the characteristics of SDN architecture impact on its security [13]:

- Effective monitoring of abnormal traffic. The SDN architecture can be exploited to enhance network security with the provision of a highly reactive security monitoring, analysis and response system. The central controller can perceive the entire network traffic simultaneously. Traffic analysis or anomaly-detection methods deployed in the network generate security-related data, which can be regularly transferred to the central controller. Applications can be run at the controller to analyze and correlate this feedback from the complete network. Based on the analysis, new or updated security policy can be propagated across the network in the form of flow rules [14].

- Timely dealing with vulnerabilities. This consolidated approach can efficiently speed up the control and containment of network security threats. Once a new threat has been detected, operators can program new software to analyze and deal with the vulnerability immediately, without spending time to wait for an update of the operating system or application software integrated in the manufacturer-proprietary devices.

However, the same attributes of centralized control and programmability associated with the SDN platform introduce network security challenges [13]:

- Vulnerable controller. Increased potential for Denial-of-Service (DoS) attacks due to the use of a centralized controller. Most functions, such as network information collection, network configuration, and routing calculation, are concentrated in the SDN controller. The SDN architecture provides a more concentrated target and greatly reduces the difficulty of such attacks.

- Risks caused by open programmable interfaces. SDN is more susceptible to security threats. First, it makes the software vulnerabilities of the SDN controller fully exposed to attackers, as the latter will have enough information to formulate an attack strategy. Second, the SDN controller provides a large number of programmable interfaces for the application layer and this level of openness may lead to an abuse of the interface, such as embedding malicious code, such as a virus.

- More attack points. As the SDN is divided into three layers, the entities of each layer may be spread across different locations of the network; communication between these entities will be necessary and frequent. Hence, compared to traditional networks, SDN provides more possible attack points for attackers, such as the SDN switch, the links between SDN switches, the SDN Controller, the links between the controller and the switches, the links between controllers, and the application software.

Security and decision making

Decision-making has a considerable function in network security. It provides several advantages and simply corresponds to following the scientific method, numerous books and articles that have an interest in network security approach it to theoretical models to study the problems analytically.

Feedback Loop

Generally, real-life problems need observation and collection of information from practitioners on a daily basis. Such as planning and implementing network security need to ask some questions like, What are you trying to protect or maintain? Are your objectives compatible with your security infrastructure, operations, and tools? What risks are associated with inadequate security? What are the implications of not implementing security? What is your tolerance for risk?...etc, with the aim to understand the real problems and provide novel algorithms and solutions by performing a feedback loop [3] between high-quality theoretical research and real-life problems as depicted in Fig. 2.

Quantitative approach

The quantitative approach based on mathematical models expresses the decision-maker in a transparent and durable manner and decision-making can be made on a large scale, in addition to checking the model (decision-making process) experimentally by multiple experts.

Game theory provides a rich set of mathematical tools and models for investigating multi-person strategic decision-making where the players (decision makers) compete for limited and shared resources.

As a special case, security games study the interaction between malicious attackers and defenders. Security games and their solutions are used as a basis for formal decision-making and algorithm

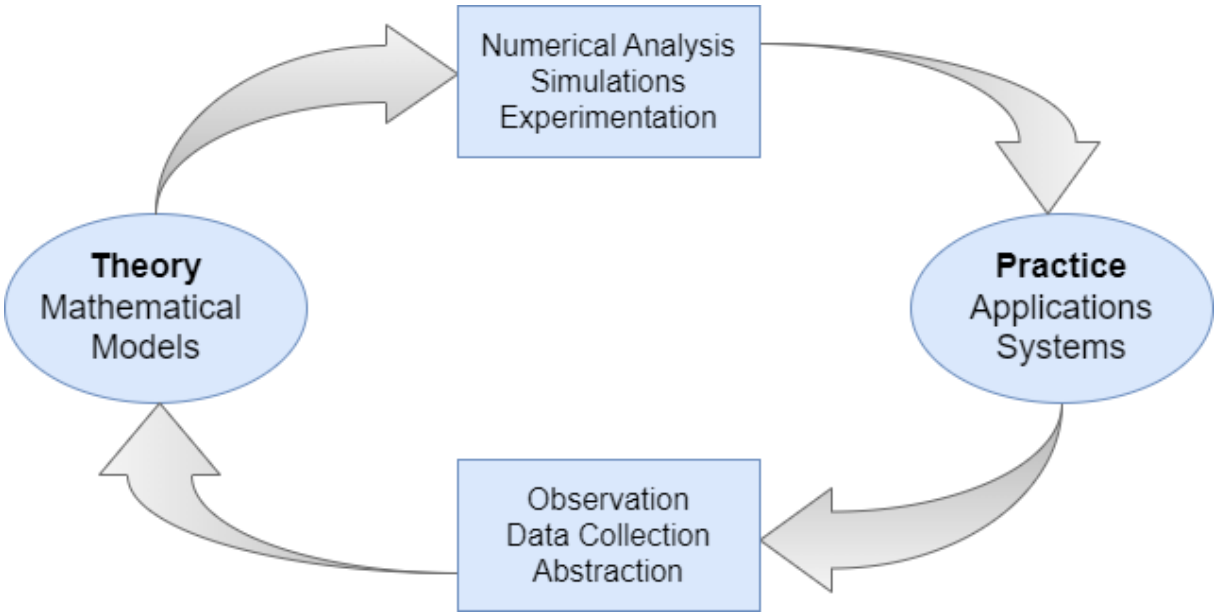

Figure 2: Feedback loop between theory and practice.

development as well as to predict attacker behavior. Security games vary from simple deterministic ones to more complex stochastic and limited information formulations and are applicable to security problems in a variety of areas.

Security risk-management

Nowadays, most organizations have an awareness of security risks, because of the nightmares that many of them endure, represented in big loss of money due to network downtime, data breaches...etc. For this reason, they are learning how to manage these risks using security risk management, which is best exercised in some fields, such as finance, and is relevant in the network security context. Security risk management promotes the way of looking at network security as a problem to be managed rather than solved once and for all through pure engineering and empirically. The situation has changed as the field is enriched by quantitative approaches and models. Risk management processes are modeled by analytical and mathematical frameworks that provide a foundation for computer-assisted decision-making capabilities. In consequence, besides that just improving scalability and efficiency of solutions, these frameworks also increase transparency and manageability. Such a mathematical abstraction is useful to combine seemingly different problems, facilitate future research, and develop computer-based scalable solutions that rely on rational principles and transparency. On the other hand, risk management is not purely technical problem and cannot be addressed by purely technical solutions. Organizational and human aspects play at least as important a part as the technical ones.

The risk management process can be presented as follow [3]:

- **Risk assessment:** A process for observing the systems closely to understand what happens and identifying the weaknesses and vulnerabilities that could affect the system and could be exploited by any threat.
- **Risk analysis and decision making:** Provided by mathematical frameworks such as game theory that have multiple benefits ranging from efficiency, prioritization, and clear expression of tradeoffs in a quantitative manner to scalability and transparency.
- **Execution of measures:** Involves dynamic allocation of existing resources, organizational changes, and future investments.

Research motivation and goals

Motivation

P1: Which security mechanisms are used in critical networks to keep functioning properly even if the intrusions have occurred and are successful?

The conventional security mechanisms such as Firewalls, Intrusion Detection, or Prevention Systems seem to be insufficient against the new types of attacks and intrusions. In addition, the availability of emergent environments like SDN, Cloud Computing, or Wireless Sensor Networks (WSN) is very crucial and an interruption during a few minutes or a few seconds could have a big technical and financial impact. Thus, the use of new mechanisms of security is a must, especially the Intrusion Tolerant System ITS, which ensures the continuity of services even in the presence of intrusions.

P2: How could the defender survey cyber-attack behavior in network security and take the best actions?

Despite the enormous efforts made to guarantee a great level of security in the network, this issue is still far from being completely solved. Hence, to continue providing proper services in threatening environments there is a need for intrusion tolerance. The purpose of an intrusion tolerant system (ITS)

is to survive against every intrusion, rather than to prevent them. Unfortunately, these mechanisms of defense require a huge investment and an accurate study of the network to effectively secure the infrastructure. The weakness of traditional network security solutions is that they lack a quantitative decision framework. Security game approaches proved their efficiency in this issue. Security game is a remarkable concept in various security situations and has found important application in cyber security. It provides a quantitative measure of the quality of defense using Nash equilibrium concept, where both the defender and the attacker try to get optimal strategies, and their conflict for security objectives is their stimulant to swerve unilaterally from their equilibrium strategies. The concept of equilibrium helps to offer a quantitative prediction of the payoffs of the scenario the security game model captures. With the quantitative measures of security, game theory makes security manageable beyond the strong qualitative assurances of cryptographic protections.

P3: How to avoid the SDN single point of failure?

The use of SDN leads to new security challenges in the control plane and in the two other planes, such as man-in-the-middle attacks, denial-of-service attacks (DoS), and saturation attacks, etc. As a centralized controller responsible for managing the whole network is considered as a single point of failure, that can render the network compromised.
Unfortunately, the existing multi-controller architectures lack security and safety mechanisms. Using a multi-controller mechanism alone cannot avoid the SDN at a single point of failure. Besides, the visible nature of the control plane makes it more vulnerable to attacks, particularly to DoS and DDoS (Distributed DoS) attacks

Goals

The main goal of this Thesis is to **study methods, tools, and techniques to improve SDN control plane intrusion tolerance and provide a quantitative approach to perform security networks at low cost.**

To achieve this goal, we will focus on the following three general objectives:

- Study the state of the art of ITSs and security games, the approaches existing and their techniques as well as a review of technologies using ITSs and game theory to improve their

security,

- Develop a game theoretic approach to model the attack-defense interaction in taking into account both internal and external attacks and analyze the effect of intrusion tolerant system on the payoffs of both the internal and external attacker as well as the defender.
- Enhance intrusion tolerance in SDN control plane and increase the uncertainty for the attacker, in effect, diminishing the information gathered from the control plan during the reconnaissance phase of a potential attack.

Research Contribution and Organization

This Thesis will provide several contributions, either published in international journals, [15] (Sanoussi et al., 2020), [16] (Sanoussi et al., 2023), and [17] (Sanoussi et al., 2024), or through the chapters of this thesis, to the enhancement of SDN control plane intrusion tolerance aligned with the development of a game-theoretic approach to model the attack-defense interaction. These contributions are presented in the 4 chapters of this document, as follows:

- Chapter 2 presents an overview of ITSs techniques using three separate lines of approaches: Detection-Triggered, Algorithm-Driven, and Recovery Based, as well as analyzes and contrasts several works in three different environments: Cloud computing, Software Defined Networks (SDNs), and Wireless Sensor Networks (WSNs) to proceed toward Intrusion Tolerant Systems (ITSs) (Sanoussi et al., 2024) [17].

- Chapter 3 introduces the foundations of game theory in general and security games in particular, its classifications, its forms, and some basic definitions. Through this chapter, we provide a review of the applications of security games in several emerging technologies such as Cloud Computing, Software Defined Networks (SDN), and the Internet of Things (IoT), in addition to its use in improving the mechanisms of defense as Intrusion Detection System (IDS) and Moving Target Defense (MTD).

- Chapter 4 presents our contribution to security games involving intrusion tolerance. The chapter develops a game theoretic approach to model the attack-defense interaction taking into

account both internal and external attacks and analyzes the effect of intrusion tolerant system on the payoffs of both the internal and external attacker as well as the defender (Sanoussi et al., 2020) [15].

- Chapter 5 focuses on our second and main contribution. We propose to approach the issue of intrusion tolerance in the SDN control plane by first applying a Recovery Based model which assumes that as soon as a system comes online it is compromised; therefore, periodic restoration to a good state is necessary. Secondly, the proposed approach aims to establish Moving Target Defense (MTD) that provides a proactive defense against adaptive adversaries. The goal of the MTD in the Dispatcher is to constantly shift between multiple controllers with diverse configurations in order to increase the uncertainty for the attacker, in effect, diminishing the information gathered from the control plan during the reconnaissance phase of a potential attack. Then, we put in place probabilistic models that can contribute to the perception of the performance of self-cleansing intrusion tolerance in the SDN control plane (Sanoussi et al., 2023) [16]. Finally, the conclusion and future research directions are described in Chapter 6.

2

Survey on Intrusion Tolerant Systems

This chapter presents a general overview of Intrusion Tolerant Systems, including the principle behind it, its techniques, and the different approaches used in ITS. In addition, it surveys the employment of ITSs in several environments such as cloud computing, SDNs, and WSNs. This chapter is the subject of Our paper (Sanoussi et al., 2024) [17].

2.1 Introduction

Currently, so many technologies are very quickly becoming significant topics in both the scientific community and the general public. Especially with the growth of network applications, even the most isolated people have developed strong bonds with one another, which leads to the expansion of networks and the increase of data. Consequently, the development of new environments is a must such as **Cloud Computing** which ensures that different applications have access to computing power, storage space, and various software services [18] as required, in order to increase the efficiency of using computing resources, minimize power consumption per service. Furthermore, **Software Defined Network (SDN)** has emerged as one of the most significant network architectures for streamlining network management and facilitating communication networks, due to its primary characteristic of separating the network control from the data forwarding layer. As well, **Wireless Sensor Networks**

(WSNs) arise due to their distinctive characteristics: limited energy lifetime, slow embedded processors, severely constrained memory, and low bandwidth radios [19].

Nevertheless, the use of these new environments leads to new security challenges such as man-in-the-middle attacks, denial-of-service attacks (DoS), saturation attacks, etc. The traditional security mechanisms such as firewalls, intrusion detection or prevention systems seem to be insufficient against the new types of attacks and intrusions. In addition, the unavailability of these emergent environments is very crucial and an interruption during few minutes or few seconds could have a big technical and financial impact.

Thus, the use of new mechanisms of security is a must, especially the Intrusion Tolerant System ITS, which ensures the continuity of services even in the presence of intrusions.
Whereas existing surveys already address ITSs in general technologies. This survey focuses on ITSs in cloud computing, SDN, and WSNs, and their classification compared with the approaches proposed in [24].

The remainder of this chapter is structured as follows: Section 2 presents the aim of intrusion tolerance systems, techniques, and architectures. Section 3 overviews ITS in Cloud Computing. Section 4 gives a survey of ITS in SDN. Section 5 reviews ITS in WSNs. Finally, a comparison and a conclusion are given in Section 6.

2.2 Intrusion Tolerance Systems

2.2.1 Intrusion Tolerance Need

Vulnerability and attack are the substantial causes of an intrusion. Vulnerability represents every anomaly in a system components that can be exploited to hack that system and an attack is a malicious attempt to hack the system. In a first stage, an attacker tries to find any vulnerability of the target system which seldom ends in failure. In a second stage, as shown in Fig. 3, he inserts an attack which leads the system components to behave abnormally. Finally, this abnormal way of behaving causes security failure. In order to prevent a security failure, the use of intrusion tolerance is necessary to block the effectiveness of malicious attacks, which means that intrusion tolerance aid

the system to recover to a proper state before it falls into a security failure as shown in Fig. 4

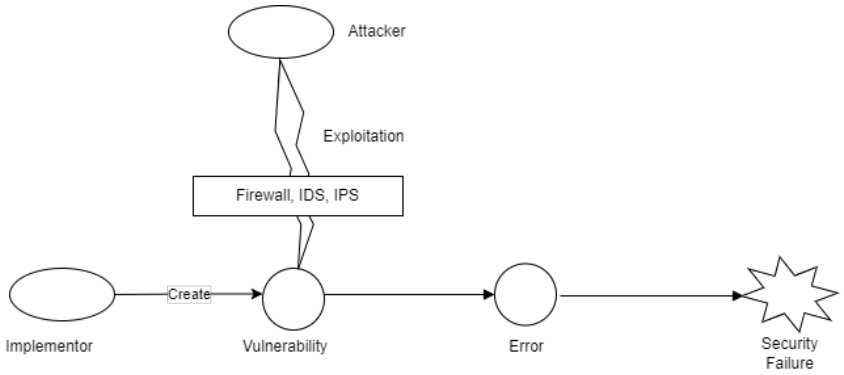

Figure 3: Attack leading to security failure

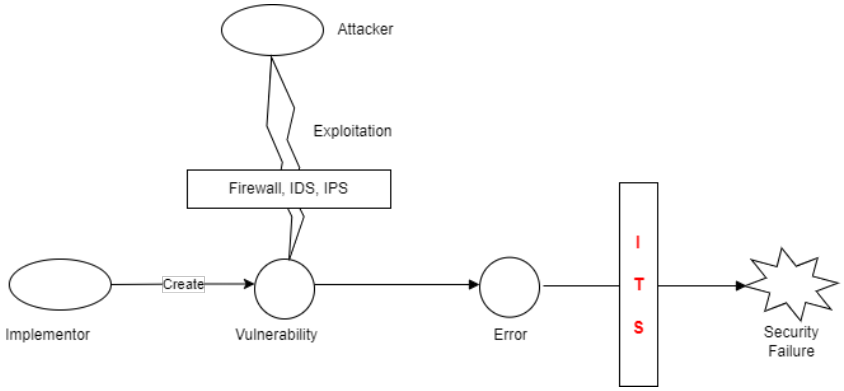

Figure 4: Security failure blocked by ITS

2.2.2 Fundamental Techniques

The goal of this section is to present the key techniques used in most researches working on Intrusion Tolerant Systems. The major part of these techniques are extracted from Fault Tolerance techniques:

- **Redundancy**: As the authors of [20] said, no redundancy, no tolerance. i.e, redundancy is the main technique used in tolerance, and the other techniques are complementary. But the use of redundancy alone is scanty. On the whole, redundancy is laying additional resources to a system to use them in need. It can be used in both hardware (replication) and software.

- **Diversity per se needs redundancy** [20]: to achieve diversity, redundant components should be significantly distinct, such as hardware diversity. The used hardware in the redundant components must be different. The same issue to software diversity by using distinct software and

in operating system diversity by using different OS that present, in case they exist, different types of vulnerabilities. The higher the diversity, the lower the risk of system failure, but the greater the complexity of the system.

- **Voting** [20]: Diverse redundant responses are the result of redundant components in the presence of intrusion in the system. Voting is the solution for differences in output data, it's based on comparing the redundant responses and come to consensus to choose the proper response.

- **Acceptance Test**: To tolerate faults generally or intrusion specifically, we need to detect them. So the use of acceptance test. The concept involves a series of statements that will manifest an exception if the system fails [21].

- **Threshold Scheme and Distributed Trust** [20]: is also called secret sharing. The concept principally focuses on splitting data D into p chunks that could be dispersed in many emplacements. So it requires k or more chunks to rebuild the data D. Otherwise, it reveals nothing.

- **Dynamic Reconfiguration** [20]: The principle is to dynamically reconfigure the system without any downtime for the system, so the service can be uninterrupted.

- **Indirection** [20]: It's generally designed as layers, in which designers insert barriers between clients and servers. The most used indirection systems by ITSs are: virtualizations, proxies, wrappers.

The three following techniques could be used in several environments, especially in WSN and IoT.

- **Intrusion-Tolerant Routing** [22]: It relies on multi-path routing in which multiple paths are used to send data, or various paths are considered as backups to be utilized in case of the primary path drops out. Then, they are combined with cryptographic mechanisms to ensure

Intrusion-Tolerance.

- **Key Management** [22]: As sensor nodes suffer from resources derisory and are susceptible to failures, key distribution and management in WSN faces high difficulties and challenges. Symmetric key based mechanism is the most used for WSN. And the most used techniques to pre-distribute the keys are:
  - Pairwise Key Sharing between the base station and each node.
  - Pairwise key sharing between nodes.
  - A global group keying.
  - Hybrid.

- **Cryptographic algorithms suitability** [22]: The choice of using Symmetric-Key encryption is returned basically to the resource constrained of WSN. The block ciphers are the fundamental cryptographic primitives for WSN, like RC5, RC6, SKipjack and AES (Rijndael), etc. As evaluated in [23], Rijndael is the best Symmetric-Key encryption for its high security and energy efficiency needs.

2.2.3 Intrusion Tolerance Architectures

In this section, we study three different approaches to Intrusion Tolerant architectures (SITAR, MAFTIA, SCIT), focusing on three different approaches [24]: Detection-triggered, Algorithm driven and recovery-based system.

- **Detection-Triggered** based on multiple levels of defense to increase system survivability. One of them is Intrusion Detection that call up recovery mechanisms to work. One of these architectures is Scalable Intrusion Tolerant ARchitecture (SITAR) [25] that relies principally on redundancy, diversity, voting and automatic reconfiguration.

  As shown in Fig. 5, SITAR architecture contains the following components:

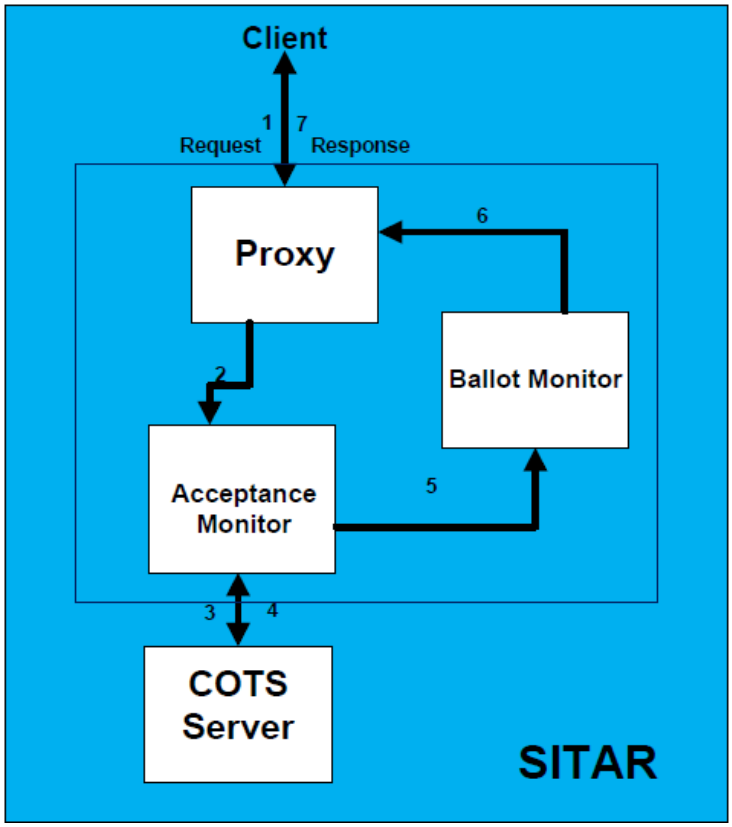

Figure 5: SITAR Architecture

1. Proxy servers, the first point of connection between the clients and the (Commercial Off-The-Shelf) COTS servers, are responsible for accepting or rejecting requests based on security policies.

2. Acceptance monitors, put acceptance test on the requests previously accepted by the proxy servers. If accepted, the requests would be forwarded to the specified Commercial Off-The-Shelf (COTS) server.

3. COTS servers, receive requests, treat them and generate adequate responses, then transmitted them again to the acceptance monitors to validate them. Validated responses are delivered to the ballot monitors.

4. Ballot monitors Applies a voting and agreement process to the forwarded responses, the result is the final response that will be sent to the proxy then to the client.

5. Adaptive reconfiguration module (ARM), collects intrusion signs from other components, evaluates the threats, then applies necessary reconfiguration to ensure system security and performance.

6. Audit control module, controls resources use by the components to prevent intrusions.

- **Algorithm-Driven** based on the principle of augmenting the level of trustworthiness by constructing layers of more trusted components in a progressive way. It uses intrusion tolerant techniques such as voting algorithm, threshold cryptography, etc.

There are several architectures that fit the algorithm driven approach, but Malicious and Accidental Fault Tolerance for Internet Applications (MAFTIA) [26] is more appropriate to study in that case.

MAFTIA architecture employs intrusion tolerant techniques to construct layers that contain more trusted components and middleware founded on untrusted hosts and networks.

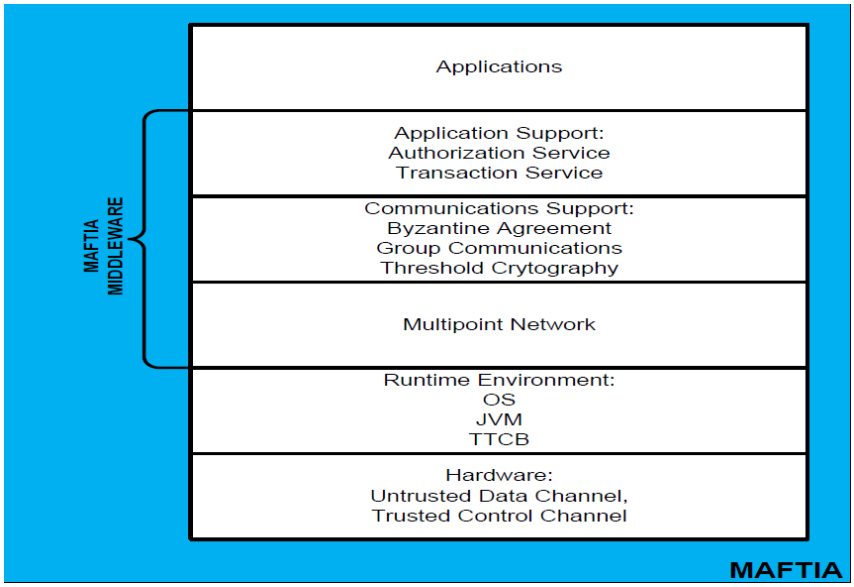

Figure 6: MAFTIA Architecture

As shown in Fig. 6, MAFTIA architecture is divided in at least three layers.

1. The hardware layer contains hosts and networking devices that are generally untrusted such as usual personel computers for example, however, they might include pieces of trusted hardware.

2. The runtime support comprises the Operating System (OS), Java Virtual Machine (JVM), and Trusted Timely Computing Base (TTCB) which is a distributed trusted component that give a set of trusted time to middleware protocols for communication.

3. The middleware layer consists firstly of multipoint network based on the physical infrastructure that provides basic secure channels, multipoint addressing, etc. Secondly, the communications support module that ensures principaly byzantine agreement (voting), group communications, and threshold cryptography. Finally, the application support module ensures transactions and key management and authorization services.

MAFTIA's intrusion tolerance approach is based on the notion of trust and trustworthiness relationship, by constructing layers of security service to secure applications, he intrusion tolerant strategy that include voting and secret threshold cryptography, covers host machines, networking devices, protocols and OS extensions.

- **Recovery Based** system founded on the principle that as soon as a system comes online, it is compromised, therefore, periodic restoration to a good state is necessary [24].

Self Cleansing Intrusion Tolerance (SCIT) [27] is a recovery based model workable to servers that are open to the internet such as web and DNS servers. The main advantages of the SCIT architecture are:

  - It ensures diversity of the server, so the attacker should make more efforts to compromise a system.
  - It shortens the exposure time of the server, therefore, rendering the information gathered from the reconnaissance phase inaccurate for the attacker.
  - It reduces losses by controlling the time that a server is exposed to the network.

The SCIT architecture is founded on two fundamental components as shown in Fig. 7.

The server cluster is a group of nodes doing the same tasks and affording the same services but running in different operating system platforms to ensure diversity, thus making malicious exploitation more difficult.

Each node is continuously routed through the following life-cycle states as:

  - **Active:** Node is online and accepts/processes any incoming requests.

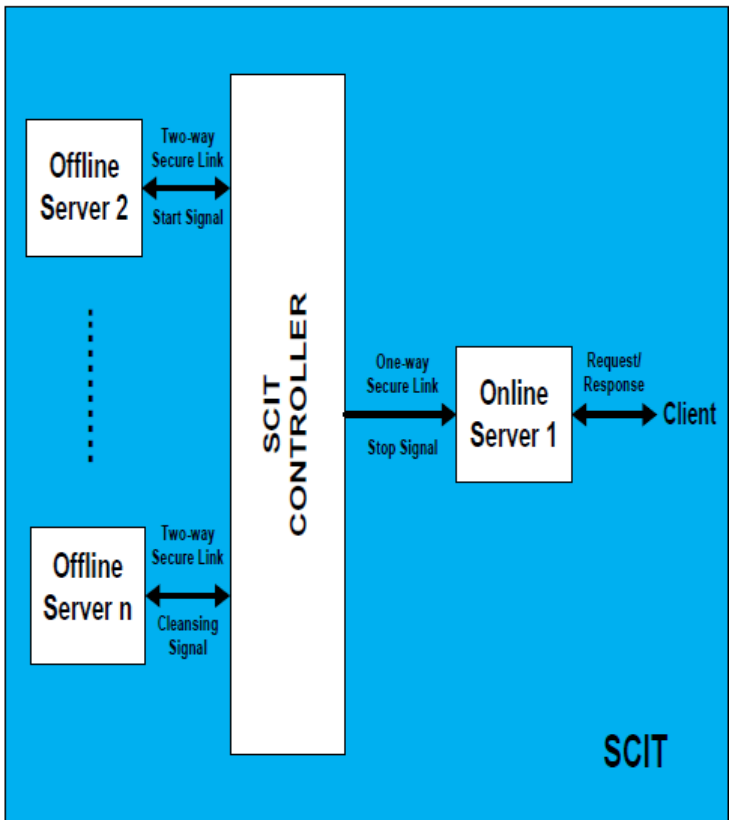

Figure 7: SCIT Architecture

- **Grace Period:** Node processes any existing requests, but doesn't accept any new ones.
- **Cleansing:** Node is offline and undergoes the cleansing to get to a known good state.
- **Live Spare:** Node has been restored and is ready to come online.

The SCIT controller is the center part that manages server rotation in and out of the cleansing mode.

- **Hybrid** is the last possible approach that could combine two or more of the approaches discussed before.

2.2.4 Analysis

On the whole, SITAR is effective in tolerating intrusions. In fact, the architecture depends especially on acceptance testing that looks for known intrusion types. While balloting helps the system be resilient against unidentified attackers. Whereas validation of incoming requests has the advantage of allowing the system to continue operating even after a hacked COTS server. Additionally, this

design is effective against attacks like cross-scripting, SQL injection, access, violation, DoS attack, and data ex-filtration. Unfortunately, this process will endure additional latency on service time.

Generally, MAFTIA provides a full intrusion-tolerant design based on the concept of trust and trustworthiness from bottom components and other trusted security services. However, the use of Byzantine agreement protocol and secret-sharing cryptosystems impacts the response delay. Additionally, the integration of the system into other architectures can be costly and complex in terms of using specified APIs and protocols. MAFTIA design offers tolerance against attacks like access violation, impersonation, web defacing, DoS, and data ex-filtration.

Unlike the two other approaches, SCIT differs significantly because it is basically a recovery-based strategy and doesn't include an intrusion detection element. The plan is predicated on the notion that the system would continue to have vulnerabilities, making successful attacks inevitable. SCIT does not require extra latency time. In addition, the intrusion tolerance is determined by the situation of how much the design may restrict the attacks' impact and keeps the services offered to users available, which depends on the exposure time.

The classification of the next ITS architectures will depend on the approaches discussed before.

2.3 ITS in Cloud Computing

In this section, we present and analyze three Intrusion Tolerant Systems in Cloud computing.

2.3.1 Adaptive Cluster Transformation ACT with proactive recovery based ITSs

ACT, presented in [28], firstly aims to maintain a good quality of service (QoS) by using a variable cluster size, and secondly defends the system against DoS attacks by early predicting the incoming massive packets, then replacing the compromised clusters with new ones.

ACT architecture is presented in Fig. 8 as follows:

- **Virtual Machines (VMs):** have the same functionality such as providing services, connecting to a secure database, then delivering the same responses to the same requests in the case they are not compromised.
- **HW servers:** are the point of connection between the VMs and the Internet.

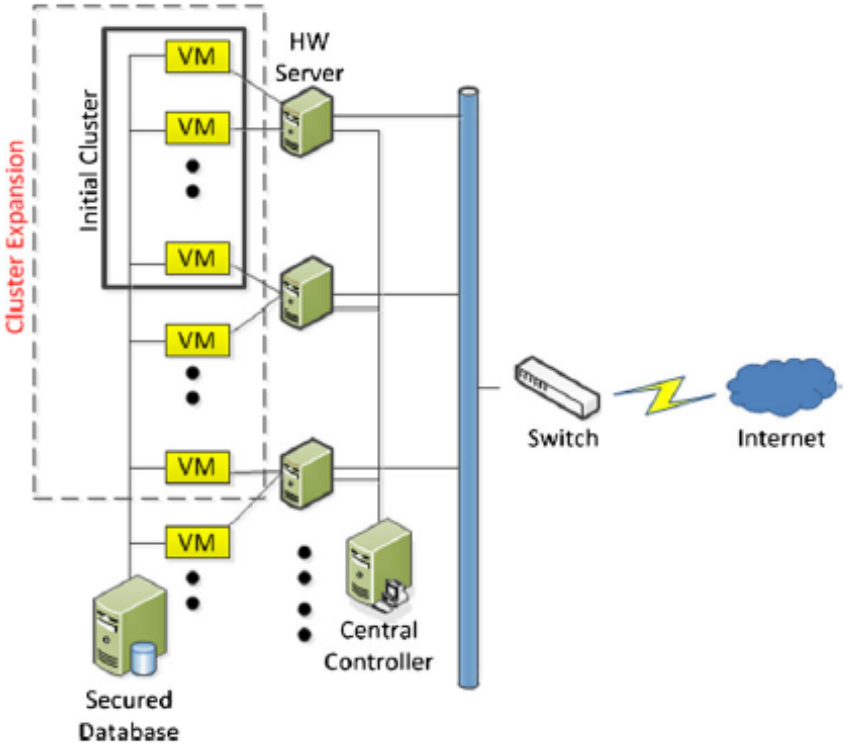

Figure 8: ACT Architecture

- **The central controller:** checks the system performance and the external threat, then decides the cluster's size (expansion or reduction) and the rotation state of each VM.

The authors consider that the attacker never modifies any system configuration and all attacks except the DoS attack are stopped by security mechanisms (Firewall, IDS, IPS).

Two main schemes in the ACT architecture are considered:

- **Adaptive Cluster Expansion and Reduction:** To maintain the Quality of Service (QoS), the central controller checks continuously the request-response delay. Then, it decides to increase the cluster size if the delay is longer, otherwise, decreases the cluster size.
- **Adaptive Cluster Substitution:** To avoid system failure due to the sudden increase of incoming packets such as DoS attack. The central controller checks continuously the current response delay of each VM and the number of unprocessed requests.

2.3.2 Hybrid Recovery-Based Intrusion Tolerant System

Proactive recovery-based ITS introduced in [29] depends on the periodic recovery of virtual machines to mitigate intentional intrusions or potential errors. Unfortunately, this solution suffers from managing the exposure time, and the VMs could be attacked easily within their working period. To tackle this problem, the authors in [29] propose a novel hybrid recovery-based ITS that utilizes a scheduling mechanism for availability-driven recovery to decrease the exposure time that leads to higher availability and to prevent performance degradation caused by DDoS attacks.

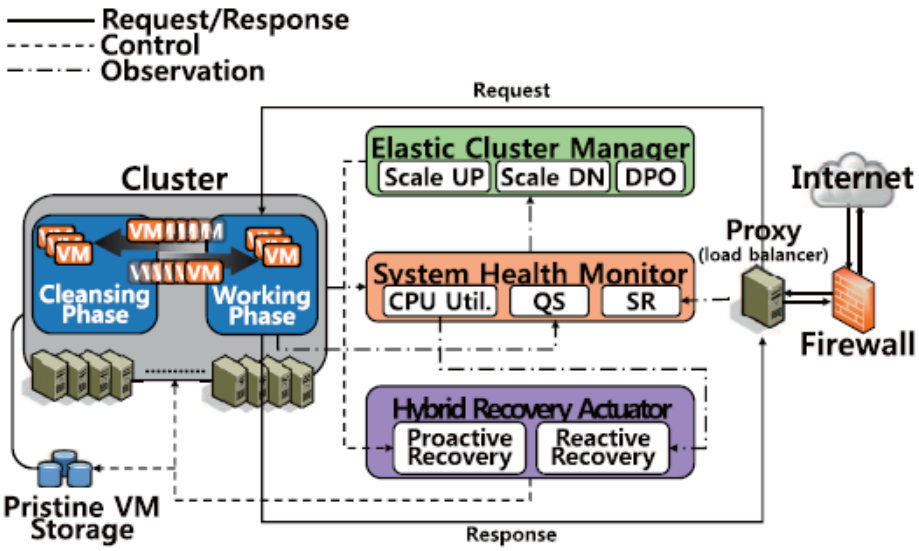

Figure 9: Hybrid Recovery Based ITS Architecture

The architecture of hybrid recovery-based ITS in Fig. 9 is separated into three primary modules:

- **Hybrid recovery actuator:** uses proactive and reactive recovery on the availability-driven recovery scheduling to ensure the lowest level of exposure time besides preventing the damage occurred by DDoS attacks.
- **Elastic cluster manager:** ensures providing services even when there is a volumetric DDoS. It contains three processes: Cluster scale-up, Cluster scale-down, and double plus one expansion.
- **The system health monitor:** monitors information concerning the states of VMs.

2.3.3 A Framework for Intrusion Tolerance in Cloud Computing

The Framework for intrusion tolerance in Cloud Computing in [30] is obtained by utilizing and adapting the existing MAFTIA intrusion tolerance framework in the Cloud Computing environment to ensure availability, integrity, and confidentiality.

In [30], the authors propose intrusion tolerance framework illustrated in Fig. 10, which is based on the layered design of Cloud Computing architecture: User Level, Middleware, and System Level.

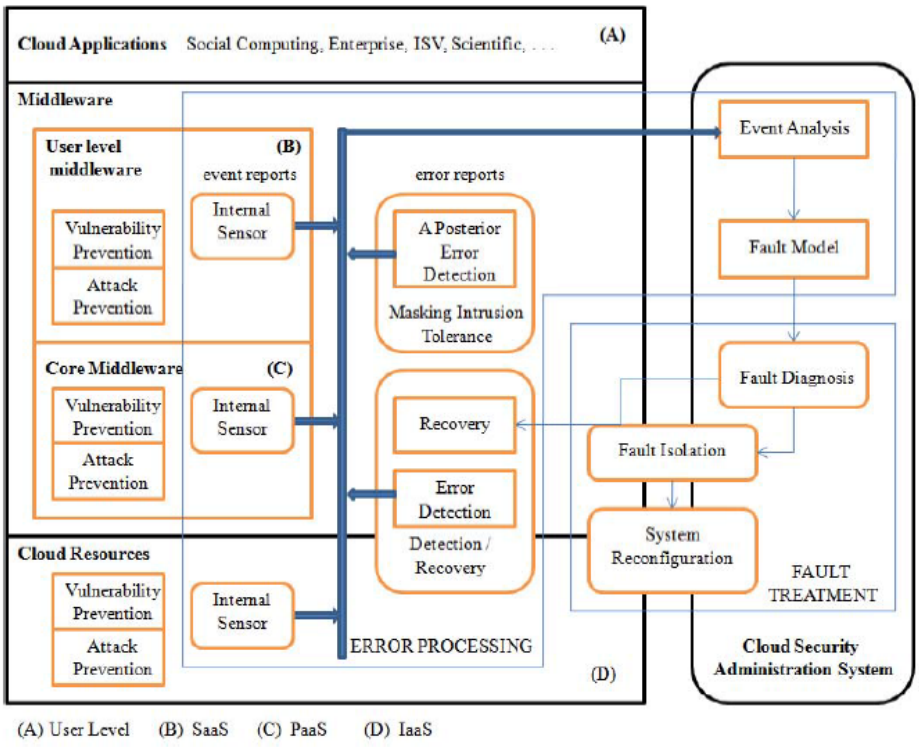

Figure 10: The framework for Intrusion Tolerance Architecture for Cloud Computing

The framework is composed of components such as:

- **Vulnerability and attack prevention:** This component is present in all the layers of Cloud Computing. It introduces mechanisms that ensures authentication, authorization, in addition to firewalls, which prevent attacks.
- **Error Processing:** consists principally of three components: Event Analysis, Error Detection, and Fault Model.
- **Fault Treatment:** is in charge of fault treatment at the middleware and system level of cloud

computing.

- **Cloud Security Administration System:** for managing and treating security vulnerabilities in the cloud environment.

2.3.4 Analysis

The Quality of Service (QoS) is improved by ACT in conjunction with proactive recovery-based ITSs when there is a flash crowd problem or a volumetric DDoS. However, this method did not address how to regulate exposure time when the cluster size was changed. Additionally, it lacked any defenses against stealthy attacks like application-layer DDoS attacks.

On the other hand, Hybrid Recovery-Based Intrusion Tolerant System employs dynamic cluster resizing and availability-driven recovery, efficiently managed the exposure time to maximize service availability, and maintained a particular level of appropriate reaction time despite workload fluctuations. Additionally, this architecture reduced the impact of volumetric and application-layer DDoS attacks.

Finally, the framework for Intrusion Tolerance in Cloud Computing is based on MAFTIA architecture adapted to Cloud Computing built on the notion of Trust and Trustworthiness connection, which enables to create layers of security services that operate as a top-to-bottom fortress to safeguard applications. The performance rises with the boost in the number of hosts in a datacenter. However, the increase in failed hosts leads to an increase in the total execution time.

After classifying and discussing the efficiency of these intrusion tolerant designs in cloud computing, we move to present other ITSs in Software-Defined Networks (SDN).

2.4 ITS in Software Defined-Networking

Software-Defined Networks (SDN) is an emerging technology that separates the control plane from the data plane. The controller in the SDN control plane has network-wide control capabilities. There are several efforts to secure and defend SDN. However, there is lack of works on intrusion tolerance

in SDN especially the control plane. For this reason we decided to enhance ITS in SDN control plane as detailed in Chapter 5.

Two ITS designs elaborated in this section concern intrusion tolerance at the level of the control plane.

2.4.1 Intrusion Tolerant Architecture for SDN Networks Through Flow Monitoring

The benefits of SDN and intrusion tolerance are both utilized in the proposed system [31]. Indeed, in the Opendaylight SDN controller, an Intrusion Detection Module (IDM) has been developed. IDM's major goal is to change the flows and tolerate intrusion after it has been discovered. This is accomplished by keeping track of the switch's packet count for flows.

In fact, after collecting flow information from the switch and tracking packet counts at various intervals, the controller determines the packet rate. A higher priority flow with the same match criteria is introduced with the action of punting to the controller if the rate exceeds the threshold value. To avoid overburdening the controller with packets, this flow will have a 5-second idle timeout. Thus, the controller receives the subsequent few packets.

If a packet is from a known sender, IDM will do deep packet inspection on the punted packets and add a new flow with match criteria of 5-tuple information and action of dropping.

Otherwise, a new flow is introduced with the source IP, and an action of the drop is performed if the packet comes from an unidentified source. This will prevent any packets from being delivered by an unknown sender, safeguarding the system.

2.4.2 A Fault-Tolerant and Consistent SDN Controller

Performance and consistency can be balanced through the design of a reliable and fault-tolerant Master-Slave SDN controller. The major goal of this work in [32] is to get the best feasible performance between an SDN Master-Slave controller and a single controller.

The proposed architecture of the Master-Slave SDN controller consists of a master controller and a distributed datastore used to store the Network Information Base (NIB) data by backup SDN controllers as shown in Fig. 11:

The master attends messages similarly to how a single controller does so, but with the following

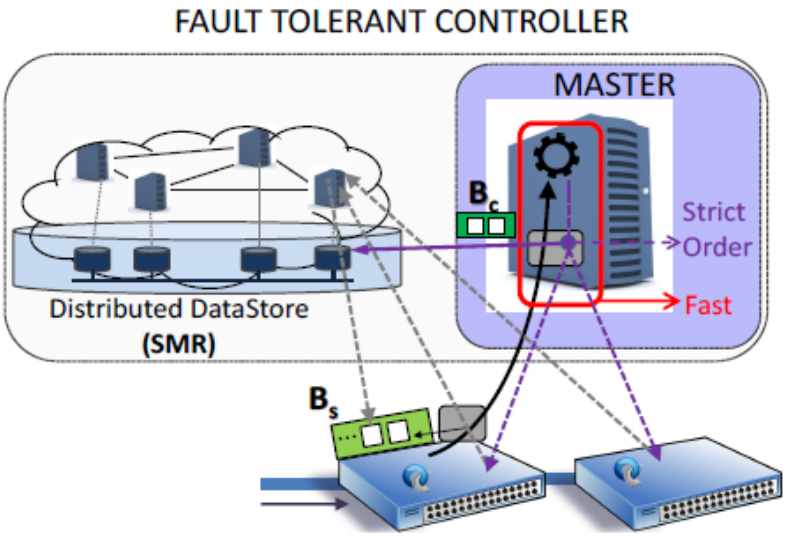

Figure 11: Master-Slave SDN Controller architecture

two exceptions:

- Each NIB update is divided, stored, and transmitted to the common datastore right away once each incoming message has been processed. A further, successively-increasing ID is utilized for consistency checks.
- Consistency assurance processes rely on NIB-updates divided and buffered from a single reference point, enabling a certain amount of Master independence, as well as communication between switch buffers and the slave platform. The goal of the suggested adjustments is to minimize the overhead on the master operation.

2.4.3 Analysis

The concept of intrusion-tolerant architecture for SDNs is involved in monitoring the packet count for flows by using IDM. If the packet rate (flow statistics from the switch) exceeds the threshold value, an idle timeout of 5 secs is set to prevent overload of the controller. Additionally, the process of deep packet inspection will endure additional latency on service time. This design is especially effective against DoS attacks, but it can lead also to availability problems because of dropping all the packets sent from unknown senders.

SDN Master-Slave controller uses a replication scheme joined with a consistency check and a correction mechanism to avoid the single point of failure and to increase the performance when needed.

2.5 ITS in Wireless Sensor Networks (WSN)

WSNs are formed of a considerable number of small sensor devices that are exploited in hard surroundings. These latter are characterized by their limited power, limited computation, and wireless communication capabilities. In general, these sensor nodes are susceptible to failure, in addition to being vulnerable to malicious attacks. As a consequence, using fault tolerance and intrusion tolerance are two fundamental mechanisms to prevent the WSNs from failing.

As mentioned already in the fundamental techniques subsection, the techniques used to tolerate intrusions in WSNs are:

- Intrusion Tolerant Routing,
- Key Management,
- Cryptographic algorithms suitability.

2.5.1 Multi-Version Multi-Path (MVMP)

MVMP mechanism, proposed in [22], is a fault-intrusion tolerant routing mechanism for WSN. Its main goal is the security of data transmission. In fact, it guarantees data authenticity, integrity, and confidentiality.
MVMP mechanism integrates data fragmentation, multiple paths, and multiple versions of cryptographic algorithms to achieve secure and reliable data transmission in WSN.

The process used in this mechanism is shown in Fig. 12

Sending sensor node: Data packets are divided into pieces, before being encrypted using different cryptographic algorithms supported by the sending node, then, transmitted to the destination via multiple disjoint paths as seen in Fig. 13.

Receiving sensor node:

The encrypted fragments pass the authentication and integrity check, before the reconstruction of the encrypted data packet. As a result, the original data packet can be obtained after decryption.

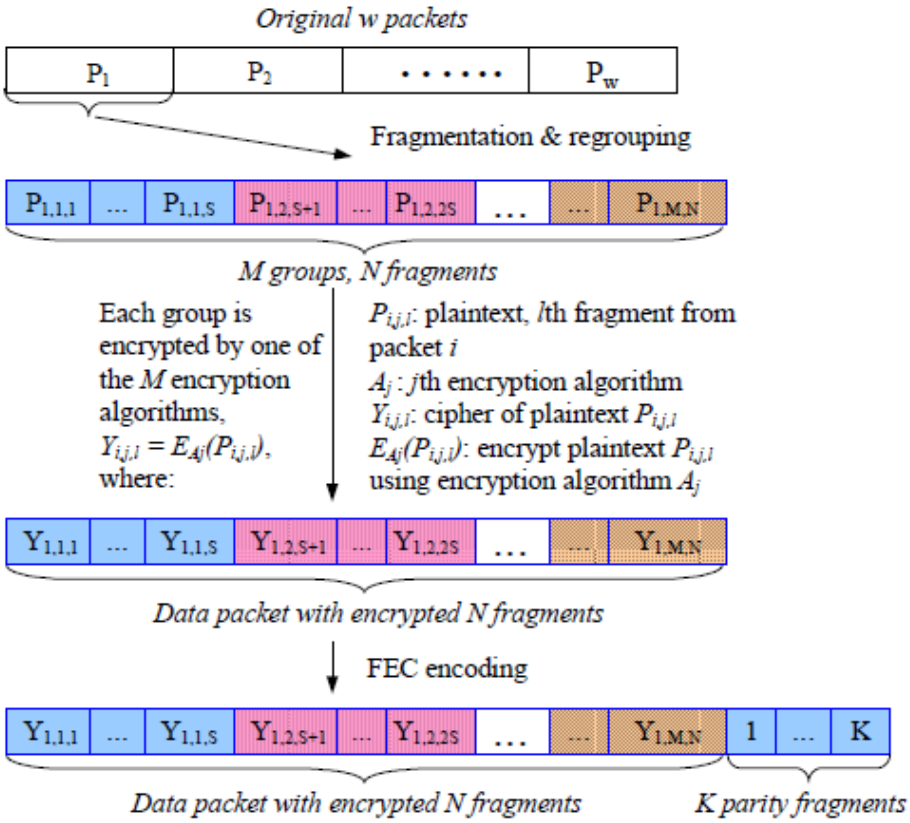

Figure 12: MVMP Mechanism

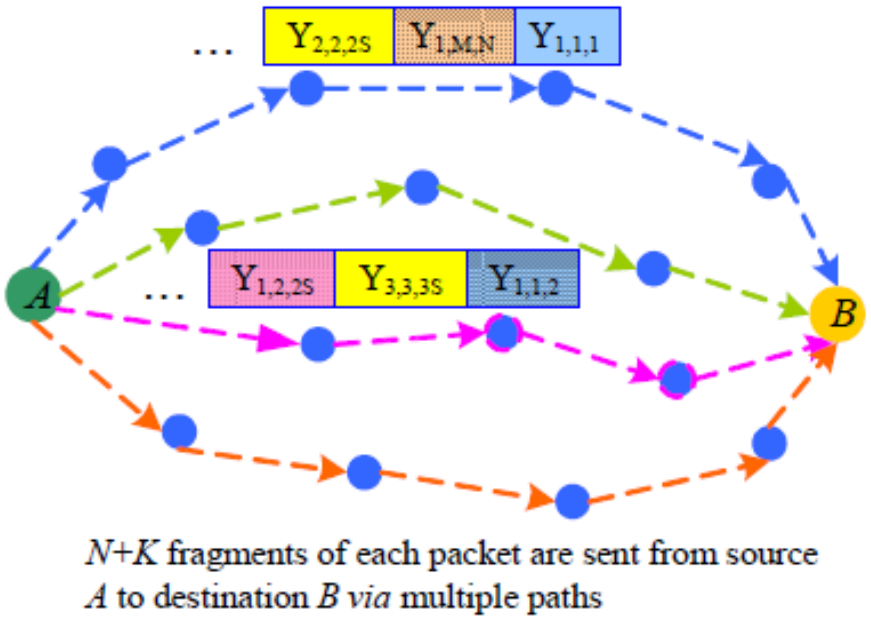

Figure 13: MVMP Data Packets Sending Process

2.5.2 A Secure Fault Tolerant Routing Protocol for IoT

The Secure Fault Tolerant Routing Protocol, proposed in [33], enhances performance and decreases energy consumption.

The proposed mechanism uses RPL routing protocol to transmit data. In addition, it employs Dijkstra's algorithm to determine the minimal path from Base Station to its destination and uses the Blowfish algorithm to transmit data securely, as illustrated in Fig. 14 and Fig. 15.

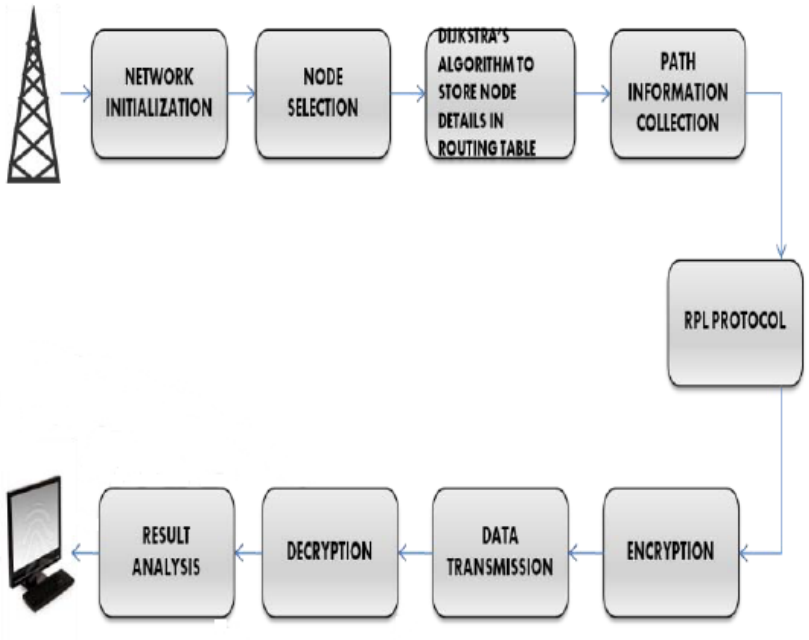

Figure 14: A Secure Fault Tolerant Routing Protocol Architecture

2.5.3 INSENS- INtrusion-tolerant routing protocol for wireless SEnsor NetworkS

INSENS, proposed in [34], is based on three principles:

- Firstly, using redundancy in routing to avoid intruders while transmitting data. In fact, the difficulty of detecting intrusions in a suitable time and way pushes the authors to propose a strategy of routing mechanism that tolerates intrusions rather than detects intrusions.
- Secondly, reducing the functions of sensor nodes that suffer from resource constraints by executing the big computations at the base stations that are resource-rich.

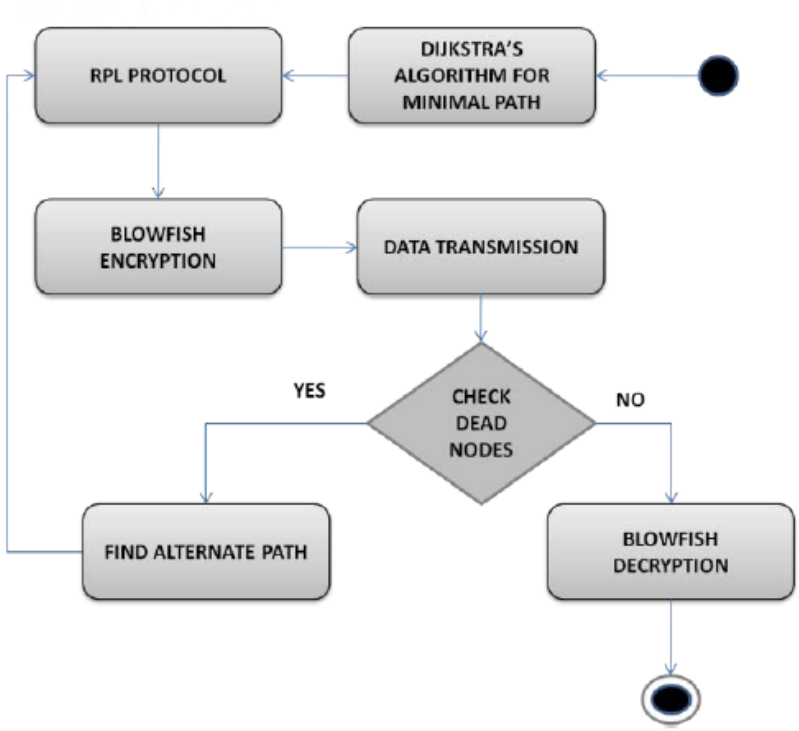

Figure 15: A Secure Fault Tolerant Routing Protocol Data Transmition

- Finally, using suitable authentication and limiting flooding by implementing symmetric-key cryptography to restrain damage occurring by (undetected) intruders.

INSENS mechanism is separated into 2 phases:

- Route Discovery phase includes gathering topological knowledge and building suitable forwarding tables at each node. Route Request, Route Feedback, and Computing and Propagating Multi-path Routing Tables are the three rounds that make up this process.

- Data Forwarding phase allows each sensor node to send data to the base station and vice versa. It is important to note that all node-to-node communication is one-way forwarded (unicast) through the base station.

2.5.4 ITSRP- Intrusion Tolerant Secure Routing Protocol

ITSRP, proposed in [35], is a secure routing protocol that prioritizes the design of a few fields to highlight the security accounting of the key exchange without increasing the protocol's complexity. The primary goal of the ITSRP is to tolerate damage from an intrusion that has compromised deployed

sensor nodes and is motivated to inject, modify, or block packets while maintaining an acceptable cost in terms of energy factor.
ITSRP mechanism consists of three steps:

- Path discovery: the sink node notifies the other nodes that it requires network topology.
- Path reverse: nodes provide topology data to sink nodes so they can create routing tables.
- Data transfer: based on routing tables.

The process of establishing a secret session key whenever a source node wishes to transmit a private message M to the sink node but does not have a shared session key with the sink node is where the overall idea to accomplish intrusion tolerance in ITSRP originates. It is important to note that each node stores a local route table and that it is first assigned a Distributed Key that is only shared between itself and the sink node (such as the base station).

2.5.5 Analysis

MVMP mechanism uses data fragmentation over many paths and multiple versions of encryption to ensure that an attacker cannot obtain all of the data at once. based on the supposition that it is extremely unlikely that attackers are aware of all encryption techniques.

In the secure Fault Tolerant Routing Protocol for IoT, the use of RPL routing protocol to find the path for data transmission, the Dijkstra's algorithm to detect the shortest minimal path, and the blowfish algorithm technique have the potential to improve system security and communication in wireless networks by reconciling cryptography. These techniques acting as the algorithm-driven approach works to grant a mechanism that could guarantee trust.

Battery drain attacks, memory fatigue attacks, DOS/DDOS attacks, and other serious intrusions

can all be handled with INSENS. Rushing attacks are still a problem, though.

The ITSRP can withstand certain severe intrusions and attacks, Sink Hole, Sybil, HELLO Flood, and Rushing attacks. Wormhole attacks can nevertheless still happen despite the low chance.

2.6 Comparison Summary and Conclusion

In this chapter, we have presented an analysis of several Intrusion Tolerant Systems in three emergent environments (cloud computing, SDN, WSNs). In sum, we have described each architecture, its characteristics, and its techniques used to tolerate intrusions, and have made a brief analysis compared to the different approaches proposed in [24]: Detection-Triggered, Algorithm-Driven, and Recovery Based, in a try to classify each approach.

Table 1 recapitulates the functional distinctions, techniques, and performance between all the ITS designs elaborated above, including cloud computing, SDN, and WSN. In addition to submitting their intrusion tolerance efficiency and complexity.

Table 1: Functional Distinctions and Efficiency of ITS designs in several emerging technologies.

Ref N	*Approach*	*Techniques*	*Performance Impact*	*Attacks Handled*	*Complexity*	*In Which Technology Used*
SITAR [25]	Detection-Triggered	Redundancy, Diversity Voting, Acceptance Test Dynamic Reconfiguration Indirection	Impact on Response time	XSS, DoS SQL Injection Data ex-filtration	High	–
MAFTIA [26]	Algorithm-Driven	Redundancy, Diversity Voting, Threshold Scheme	Impact on response time	Access violation Impersonation web defacing Data ex-filtration	High	–
SCIT [27]	Recovery Based	Redundancy Diversity (Optional)	Computing cycles for starting a new server instance	limits DoS, web defacing, and Data ex-filtration (Exposure short = Attack limited)	Low	–
ACT [28]	Recovery Based	Redundancy Diversity (Optional)	Computing cycles for cluster expansion and reduction	Volumetric DDoS	Medium	Cloud Computing
Hybrid Recovery-Based ITS [29]	Hybrid	Redundancy, Diversity Indirection Dynamic reconfiguration	Impact on Response time	Stealthy resource exhaustion attacks DDoS attacks	High	Cloud Computing
Framework for IT [30]	Algorithm-Driven	Redundancy, Diversity Reconfiguration, Voting Threshold Cryptography	Impact on Response time	Authentication and authorization attacks	High	Cloud Computing
IT Through Flow Monitoring [31]	Detection-Triggered	Acceptance test Indirection	Response Time	DoS attacks	High	SDN
Fault-Tolerant and Consistent SDN Controller [32]	Detection-Triggered	Redundancy Threshold scheme	Response time	DoS attack	High	SDN
MVMP [22]	Algorithm-Driven	Intrusion Tolerant Routing Cryptographic Algorithms	Response time	Access Violation	High	WSN
Secure Fault Tolerant Routing Protocol for IoT [33]	Algorithm-Driven	Cryptographic Algorithms	Response time	Access Violation	Medium	WSN
INSENS [34]	Algorithm-Driven	Intrusion Tolerant Routing Key Management Cryptographic Algorithms	Response time	Battery drain attack Memory exhaustion DoS/ DDoS attack	Medium	WSN
ITSRP [35]	Algorithm-Driven	Key Management Cryptographic Algorithms	Response time	Sink hole, Sybil attacks HELLO flood attack	Medium	WSN

3

Security Games Overview

This chapter introduces the foundations of game theory in general and security games in particular, its classifications, its forms, and some basic definitions, and also provides a review of the applications of security games in several emerging technologies.

3.1 Introduction

The security of crucial infrastructures in particular is becoming a critical concern and the preoccupation of worldwide researchers in the field. To complicate matters further, the increasing spread of public connectivity of today's information systems that emerge with new security challenges. Traditional security has accomplished a long way toward ensuring confidentiality, integrity, and availability. Unfortunately, with the rise of strategic attacks and the complexity of the systems, the defense using traditional methods could be cost-prohibitive [36]. Thus, how the defender could ensure the most effective protection against sophisticated attacks, especially with limited security resources?

The past decade has seen an explosion of research in an attempt to address this fundamental question which has led to the development of the well-known model of **Security Game** [37].

Security game is a remarkable concept in various security situations and has found important application in cyber security. It provides a quantitative measure of the quality of defense using Nash

equilibrium concept, where both the defender and the attacker try to get optimal strategies, and their conflict for security objectives is their stimulant to swerve unilaterally from their equilibrium strategies. The concept of equilibrium helps to offer a quantitative prediction of the payoffs of the scenario the security game model captures. With the quantitative measures of security, game theory makes security manageable beyond the strong qualitative assurances of cryptographic protections.
Security game adopts a different and more economic viewpoint: security is not the absence of threats, but the point where attacking a system costs more than not attacking [36]. Indeed, exactly this difference is what brings an interesting viewpoint on security, if we no longer consider security as the absence of threats -a state that would not be reachable anyway-, but rather as a state in which the expenses for an attack outweigh the gain from it. This economic view on security is not new, but somewhat surprisingly, much research on security is still focused on preventing all known attacks at any cost, rather than optimizing the defender's efforts and limited resources to gain the maximum security achievable. In fact, cybercrime has grown into a full-featured economy, maintaining black markets, supply chains, and widely resembling an illegal counterpart of the official software market. Traditional security remains an important foundation to tackle the issue from below, but the security game offers a top-down view by adopting the economic and strategic view of the attackers too, and as such complements purely technological security means.
The objective of this chapter is to introduce the foundations of game theory in general and security games in particular, its classifications, its forms, and some basic definitions, also provide a review of the applications of security games in several emerging technologies such as Cloud computing, Software Defined Network (SDN), and Internet of Things (IoT), in addition to its use in improving the mechanisms of defense as Intrusion Detection System (IDS) and Moving target Defense (MTD).

3.2 Security Game Model and Definitions

Game theory is the elaboration of multi-person decision situations as a game, each player of the game picks out his actions in order to ensure his best possible payoffs, obviously according to the rational actions picked out by the other players. Game theoretic models represent a quantitative assessment of the quality of payoffs with the concept of Nash equilibrium where the players seek optimal strategies.
Game theory used in cyber security called security game is a two-player game established between a

defender and an attacker. The attacker seeks to exploit the defender's weaknesses or vulnerabilities with the aim to launch attacks and the defender wrestles to assign his limited security resources to protect the critical locations of his system - why not the whole system if the security resources allow to- from the attacker's attack. Two broad categories of application of game theory in cyber security are [38]:

- The Cyber-Attack-Defense Analysis.
- The Cyber Security Assessment.

By modeling the defense behaviors as games the actions of cyber attacker can be predicted in Cyber-Attack-Defense analysis. It also analyses the possible states of attack-defense equilibrium. The counter defense strategies can be determined ideally based on the state of equilibrium.
The equilibrium state of cyber-attack-defense can be scrutinized and the prognosis of the attack and defense strategies can be used as the rationalization of cyber security and assessment. Owing to the quantitative facets of game analysis security and reliability is viewed as a quantitative assessment which gives a computation of cyber security and reliability.

3.2.1 Security Game Model

A security game's components are defined by responding to the following questions:

- Who are the decision makers (players)?

  This work centers fundamentally on two players; one of them is the attacker that represents one or multiple persons planning to compromise the system. The other is the defender that tries to protect the system or the network as much as possible.

  Players: N $=\{1, ..., n\}$ in a finite set of n, indexed by i.

- What can the players do?

  Represent the actions that are available to them. The collection of possible attacks for the attacker. And the protective or defensive actions of the defender to prevent the attacker from compromising the system.

  Action set for player i $A_i : a = (a_1, ..., a_n) \in A = A_1 \times ... \times A_n$ is an action profile.

- What motivates players?

  Estimating payoffs to the players, the costs and benefits values for each possible interaction

of the players.

Utility function or Payoff function for player i: $u_i : A \rightarrow \mathbf{R}$

$u = (u_1, ..., u_n)$ is a profile of utility functions.

- What information do different players know when they take action?

3.2.2 Games Classifications

Games can be classified into different categories based on perspectives:

- **Cooperative Vs Noncooperative:** [3] The two types differ in the agreement between players, to take actions or decisions collaboratively and collectively and consider all the players trustworthy (Cooperative game). Otherwise, disagreement and trustlessness are the bases of the situation (Noncooperative game).
- **Zero-sum Vs Nonzero-sum:** [3] Take into consideration the sum of the objective function (payoffs) of the two players, if this sum is equal to zero then the game is called a zero-sum, if not, the game is a nonzero-sum.
- **Finite Vs Infinite:** [3] Focuses on the action sets; If the sets are finite then the game is finite, also known as a matrix game. Otherwise, the game is infinite.
- **Deterministic Vs Stochastic:** [3] The two games differ in the determination of the outcome; if this latter is settled uniquely by the player's actions, then, it's called a deterministic game. If the outcome needs additional parameters such as state of nature to be established, then, the game is called stochastic.
- **Complete Information Vs Incomplete Information:** [3] Differ in the description of the game, if it's mutual information to all players, is called a complete game. Otherwise, it's incomplete.
- **Static Vs Dynamic:** [3] Determined by the number of times the game is played; Static game is played once for each player without knowing the actions played by any other player. Otherwise, the game is dynamic.

3.2.3 Basic Definitions

- **Dominant Strategies [39]**

  Is the strategy that generates the highest payoff of any strategy available for every possible action by the other players.

  That is, a strategy $a_i \in A_i$ is a dominant (or weakly dominant) strategy for player i if $u_i(a_i, a_{-i}) \geq u_i\left(a_i^{'}, a_{-i}\right)$ for all $a_i^{'}$ and all $a_{-i} \in a_{-i}$.

  A strategy is a strictly dominant strategy if $u_i(a_i, a_{-i}) \succ u_i\left(a_i^{'}, a_{-i}\right)$ for all $a_i^{'} \neq a_i$ and all $a_{-i} \in a_{-i}$.

  The existence of 'Dominant strategies' facilitates the predictions.

- **Best Response [39]**

  Let $a_{-i} = \langle a_1, ..., a_{i-1}, a_{i+1}, ..., a_n \rangle$. And $a = (a_{-i}, a_i)$

  A strategy a_i^* is a best response of player i to a profile of strategies a_i $a_i^* \in BR(a_{-i})$ iff $\forall a_i \in A_i, u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i})$.

  A best Response of player i to a profile of strategies of the other players is said to be a strict best response if it is the unique best response.

- **Nash Equilibrium [39]**

  A profile of strategies $a \in A$ is a pure strategy Nash Equilibrium if a_i is a Best Response to a_{-i} for each i.

  $a = \langle a_1, ..., a_n \rangle$ is a ('pure strategy') Nash Equilibrium iff $\forall i, a_i \in BR(a_{-i})$.

  Theorem 3.2.1 (Nash, 1950) *Every finite game has a Nash Equilibrium.*

- **Mixed Strategies [39]**

  A strategy s_i is defined as any probability distribution over the actions A_i for agent i. Playing only one action with positive probability called pure strategy. On the other hand, mixed strategy is playing more than one action with positive probability in the aim to confuse the opponent by playing randomly.

  Following mixed strategy for all the players, the use of payoffs would not be allowed, instead, the idea of expected utility is imposed:

$$u_i(s) = \sum_{a \in A} u_i(a) P(a|s) \tag{3.1}$$

Utility $u_i(s)$ under mixed strategy profile s $\in$ S, where S is the set of all possible mix strategy profiles. $u_i(a)$: all action profiles in the game (all of the cells in the normal form of the game.). Multiplied by the probability of getting to strategy profile a given strategy profile s.

$$P(a|s) = \prod_{j \in N} s_j(a_i) \tag{3.2}$$

$P(a|s)$: The probability we well get to a given action profile given a strategy profile s, is just the product of the probability of each player playing his part of an action profile. So, for example, if a player was playing with probability 0.5 on each action and the other player was playing with probability 0.5. Then the probability that we get to this action is 0.25.

3.3 Game Forms

3.3.1 Normal Game

The standard representation of a game is the Normal form game, Known also as a Strategic form or Matrix form, which is often represented by a table (Matrix).
Finite, n-person normal form game: $\langle N, A, u \rangle$:

- Players: N $=\{1, ..., n\}$ in a finite set of n, indexed by i.
- Action set for player i A_i : $a = (a_1, ..., a_n) \in A = A_1 \times ... \times A_n$ is an action profile.
- Utility function or Payoff function for player i: $u_i : A \to \mathbf{R}$
  $u = (u_1, ..., u_n)$ is a profile of utility functions.

3.3.2 Bayesian Game

Bayesian games model lack of information about the properties of players in a noncooperative game using a probabilistic approach. In a Bayesian game, the players are usually assumed to be one of many specific types. A special nature player is introduced to the game which assigns a predetermined probability distribution to each player and type combination, which constitutes its fixed strategy. Subsequently, the original players compute their own strategies by taking into account each

possible player-type combination weighted by the predetermined probability distribution [3].

Bayesian game: a set of games that differ only in their payoffs, a common prior defined over them, and a partition structure over the games for each player. Is defined by the tuple $\langle N, A, \theta, p, u \rangle$ where:

- N is a set of players.
- $A = (A_1, ..., A_n)$, where A_i is a set of actions available to player i.
- $\theta = (\theta_1, ..., \theta_n)$, where θ_i is the type space of player i.
- $p : \theta \to [0, 1]$ is the common prior over types.
- $u = (u_1, ..., u_n)$, where $u_i : A \times \theta \to \mathbb{R}$ is the utility function for player i.

3.3.3 Extensive Game

Players in these games move sequentially in some pre-specified order, at most a finite number of times. And each player is completely aware of all moves that have been made previously. These games can be represented by simple trees as in Fig. 16.

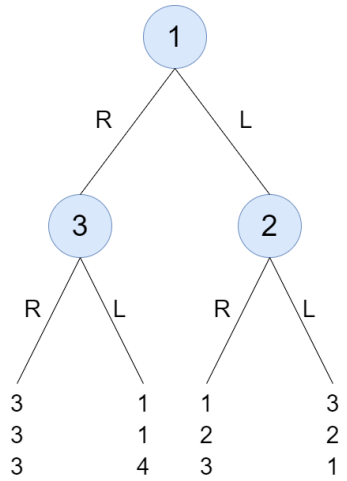

Figure 16: An Extensive Game with 3 Players and Two Actions Each.

A finite perfect information game in extensive form is defined by the tuple $\langle N, A, H, Z, \chi, \rho, \sigma, u \rangle$ where:

- Players: N $=\{1, ..., n\}$ in a finite set of n players.

- Actions: A is a single set of actions.
- Choice nodes and labels for these nodes:

  - Choice nodes: H is a set of non-terminal choice nodes.

  - Action function: $\chi : H \to 2^A$ assigns to each choice node a set of possible actions.

  - Player function: $\rho : H \to N$ assigns to each non-terminal node h a player $i \in N$ who chooses an action at h.
- Terminal nodes: Z is a set of terminal nodes, disjoint from H.
- Successor function: $\sigma : H \times A \to H \cup Z$ maps a choice node and an action to a new choice node or terminal node such that for all $h_1, h_2 \in H$ and $a_1, a_2 \in A$ if $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ then $h_1 = h_2$ and $a_1 = a_2$.

  - Choice nodes form a tree: nodes encode history.

3.3.4 Repeated Game

A repeated game is a game based on a normal form game that is played over and over again, either finitely or infinitely times. Three variants of this model are presented:
The finitely repeated game, in which each player attempts to maximize his average payoff.
The infinitely repeated game, in which each player attempts to maximize his long-run average payoff.
The infinitely repeated game, in which each player attempts to maximize his discounted payoff.
For each of these models a Folk Theorem is proven, which states that under some technical conditions the set of equilibrium payoffs is (or approximates) the set of feasible and individually rational payoffs of the base game.

3.4 Security Games Modeling Review Applications

In the last few years, the world exploit the applications of network technologies extremely. It facilitates data storage and access to information and provides a sufficient way to communicate between users. However, network security challenges have increased including cyber crimes, data stealth, illegal data, access, DDoS attacks...etc, causing serious threats in so many areas such as national security, energy and economic industries... etc.
As attacks become more and more sophisticated, so understanding the cyber security issues is a big

challenge in the process of avoiding or solving any system compromise. Cyber security techniques, such as Firewalls, and IDS/IPS... are unilaterally passive and static security defenses devoted to a specified attack scenario or method [40], and lack quantitative analysis and decision framework [41]. Game theory is considered as a framework that affords to model interactions between greedy/selfish, competitive players, malicious attackers, and system administrators. It is used by researchers in the last few years to address network security problems by providing a mathematical frame for modeling problems for so many adversaries, and analyzing many possible scenarios before determining the appropriate strategy to take. This way could increase greatly the system administrator's decision-making [42].

Game theory is applied largely to intrusion detection research areas due to its attack-defense interactive nature. The authors in [43] propose a non-cooperative game between the cooperative intruders and the Intrusion Detection System (IDS) to reach an optimal solution to the IDS for detecting intrusions, consequently reducing the attack chances from the colluding adversaries. While in [44], a non-cooperative zero-sum game with incomplete information is proposed for Dynamic Intrusion Signature Configuration (DISC) to solve the problems occurring during activating all possible intrusion signatures by finding the optimal strategy for DISC in the Signature-Based IDS. Whereas in [45], the authors formulate a robust game theoretic approach and demonstrate the existence and uniqueness of Nash Equilibrium (NE) in the system based on Dirichlet density. Also using the Markov Decision Process to reinforce learning approaches for multiple simultaneous attacks.

Cloud computing is known as the most recent technology adopted for storage, sharing, processing, and providing other services widely used by companies, organizations, and the public. Due to its complicated and different infrastructure elements, it faces several security issues, especially sophisticated attacks. Unfortunately, traditional security solutions are not appropriate for these kinds of challenges. Therefore, the use of game theoretic methods can be effectively used for analyzing the security of cloud computing. As proposed in [46], a game theory-based model is provided, called Game Theory Cloud Security Deep Neural Network. The model utilizes Deep Neural Network for the classification of attacks and normal data for security in the cloud. The authors in [47] propose a security-aware virtual machine (VM) allocation approach in the public cloud using game theory. They show that there are multiple Nash Equilibria for the public cloud security game and we can allow the players' Nash equilibrium profile to not be dependent on the probability that the hypervisor

is compromised, reducing the factor externality plays in calculating the equilibrium. It is proved that using this allocation method, the negative externality imposed onto other players can be brought to a minimum compared to other common VM allocation methods. The work in [48] a scalable security risk assessment model is proposed for cloud computing using game theory, so it can be evaluated whether the risk in the system should be fixed by the cloud provider or tenant of the system. In [49] a game-theoretic model GTM-CSec has been proposed. The proposed model intelligently selects the most suitable module out of the signature, anomaly, and honeypot based detection to detect the attack. The selection of a particular detection module instead of using all in parallel not only leads to the reduction of energy consumption but also increases the overall efficiency of the defender system. The strategies for both the defender and attacker have been evaluated and the best one has been delineated with Nash Equilibrium (NE).

Game theory have been used to tackle the network security issues, particularly handling the dynamic defense mechanism. Regrettably, the most efforts concentrate on traditional network, while few researches touch on the Software Defined Network (SDN). Among the few works, [50] design a dynamic defense mechanism for SDN based on game theory to solve the problem that static defense mechanisms in the existing methods cannot describe the uncertain and continuous change in the process of the network attack and defense. The framework for SDN adopts multistage dynamic defense strategies with the help of a quantization method of attack. The defender may find the behaviors of a particular attacker and make an adaptive response. This framework seeks to support the defender to interact with an attacker following the initial deployment of cyber defenses. Additionally, the authors in [51] took advantage of many modern technologies such as SDN, Mobile Edge Computing (MEC) and tools like Machine Learning (ML) to propose a new architecture to rectify the limitations of existing architectures. The added value of this architecture presented by the addition of Smart Node, it is an intelligent entity allows the collection of radio information, the prediction of necessary needs and the efficient allocation of network resources. A new game theory mechanism is also proposed in this work for the study of the optimal placement of Smart Node in a large network. Using game theory to guide Moving Target Defense (MTD) decision-making can maximize the effect of MTD and minimize the cost [52]. In fact, game theory is a theory that studies how to make decisions when the behaviors of decision-making subject interact directly with each other.The authors in [53] explained an important branch of MTD mechanisms named IP address hopping mechanism

using detection systems for attack mitigation in the cloud environment and proposed a game theory approach to model the attack-defense interaction and analyze the effect of MTD on the payoff of both the attacker and the defender. [54] proposes a method to generate a switching strategy for real-world web applications based on the Moving Target Defense (MTD) architecture. To find an effective switching strategy, The authors model the system as a repeated Bayesian game. They develop methods to assign attack actions to attacker types and generate realistic utilities based on the expertise of security professionals. The work in [55] suggest a concept of deception attack surface to illustrate deception-based moving target defense. Moreover, the authors propose a quantitative method to measure deception, which includes two core concepts: exposed falseness degree and hidden truth degree. they further formulate a deception game model between an attacker and a defender, in which the defender attempts to protect the entry points on the attack surface by creating or changing a deception attack surface. Furthermore, they provide a detailed example scenario and analyze the deception game's equilibrium.

Internet of Things (IoT) is an emerging technology due to the rapidly increasing number of devices and their connectivity to the internet. Indeed, IoT faces several security challenges because of its complication and different infrastructure elements. Therefore, the use of game theoretic methods can be effectively used for analyzing the security of IoT. In this study [56], to compensate for this limitation, the authors propose a game-theory-based vulnerability quantification method using attack tree, which consists of three steps: game strategy modeling, cost-impact analyzation, and payoff calculation. they present a case study for a social-IoT-based network environment. Using the proposed method, they believe social IoT network system security experts will be able to cope with security incidents more effectively. [57] this work attempts to integrate the DODAG-specific contextual trust model, and RPL-specific rank variance factor, named as Secure RPL using non-cooperative game MOdels and DEmpster Shaffer Theory in IoT (S-MODEST), such that it can detect the attackers accurately and significantly reduces the resource consumption. The non-cooperative game model consists of two interrelated formulations in the proposed solution. Firstly, non-zero sum game constructs the trust model and extracts the contextual information from DODAG structure to strengthen the trustworthiness. Secondly, the evolutionary game is utilized for trusted router selection. The non-zero sum game theory formulates the interaction of nodes and assists the trust measurement by applying direct and restricted Dempster–Shaffer theory.

3.5 Conclusion

Game theory has been an important concept in various security situations and has found great application in cyber security. Security games provide a quantitative framework for modeling the interaction between attackers and defenders. Security games and their equilibrium solutions are used as a basis for formal decision-making and algorithm development as well as to predict attacker behavior.

However, even if the game structure is realistic, the output of the security game crucially depends on the input values, defined for example in the game matrix. This input–output relationship, or how a variation (uncertainty) in the input of a security game affects its outcome. Thus practical implementation of game theory concepts is still an open research area. In this chapter, many aspects, and applications of game theory are discussed especially in the areas of Cloud Computing, SDN, IoT, IDS, and MTD.

4

Game theoretic approach based on Intrusion Tolerant System

In this Chapter, we propose a game theoretic approach to model the attack-defence interaction in taking into account both internal and external attacks and analyse the effect of intrusion tolerant system on the payoff of both the internal and external attacker and the defender. A MATLAB simulation is used to illustrate the game model and calculate the frequency of attack strategy and invest in tolerance strategy.

This chapter is the subject of Our paper (Sanoussi et al., 2020) [15].

4.1 Introduction

Due to the wide spread of Internet, a large number of applications in a variety areas, e.g, government, health, eCommerce, etc., are deployed, as well, it lead to the growth of cyber attacks against organizations and companies, which obligate them to invest in protecting networks and platforms containing critical and important information.

Large companies use protective mechanisms such as firewalls and/or reactive mechanisms such as Intrusion Detection Systems (IDSs) that monitor the events occurring in the network and analyze them to detect the anomalous behavior of the system. But these mechanisms are still ineffective

against unknown attacks and internal threats which represent about 58 % coming from employees, ex-employees, and third parties [58]. The research focus of IDS is therefore on how to detect as many attacks as possible, as soon as possible. Nevertheless, there is a great amount of crucial applications that require providing (lower) services continuously even when they are partially compromised, that's why intrusion tolerance [59], is motivated by the recognition of the fact that intrusion will occur and some will be successful. The concern of intrusion tolerance is not how to defend or detect the intrusion, but how to mask or restrain the intrusion when the network has been intruded.
Intrusion tolerance is the ability of a system to continue operating (possibly degraded) after a successful attack. This means, designing a system in such a way that the harm caused by the intruder is restrained, further, automatically repaired, rather than shutting the system down and completely losing the service it provides [60]. Common techniques are used by Intrusion Tolerant systems [59] [61], are presented as follow:

- **Redundancy and Diversity:** Redundancy mainly alludes to the additional resources assigned to a system to be used in need [62]. Whilst, diversity means that the redundant resources should be implemented differently from each other.
- **Voting:** Is a process that requires comparing the redundant responses and reaching agreement on the results to find the correct response, thus providing integrity of the data.
- **Acceptance test:** Is generally defined as a series of declarations that will evoke an exception if a module have failed or been compromised, thus the state of the system is not acceptable.
- **Threshold scheme and distributed trust:** It usually consists of storing data shares in physically distributed locations (Redundant Components) such that even if n-k-1 shares were attacked and compromised, the confidentiality is still kept and the original data can be reconstructed, therefore, tolerance can be achieved.
- **Dynamic reconfiguration:** It general purpose after the detection of an intrusion is helping in prevention, elimination as well as tolerance. There is reactive reconfiguration generally performed manually by the administrator, thus, involves some downtime, while the proactive one reconfigures the system dynamically and adaptively so that the service can be uninterrupted.
- **Indirection:** Is using additional layers as protection barriers and fault logic to separate clients and servers.

On the other hand, in order to control future threats in security systems, game theory is useful in suggesting various probable actions and in predicting their related outcomes [63].

This chapter attempts to use game theory to model the cyber security environment based on Intrusion Tolerant Systems focusing on both internal and external attacks, where security game allows a quantitative framework to model the interaction between attackers and defenders .
Our objective is to tolerate attacks using Intrusion tolerant Systems and to survey cyber attack behavior in network security by formulating an original Bayesian game to model conflicting and rational confrontation between the defender and External and Internal attackers.

The remainder of this paper is organized as follows. The related work from the state of the art on the game theoretic approaches based security is covered in the next section. Section 3 discusses the system and the threat models. In Section 4, we propose our game theoretic model and its analysis, while Section 5 presents the numerical results. Finally, the conclusion is drawn in Section 6.

4.2 Literature Review

Security under a game theoretic framework have attracted huge research attention. Several probable actions along with the predicted outcome can be suggested through game theoretic methods in order to control future threats [63].

Wang et al. in [64] presented an overview about Scalable Intrusion Tolerant Architecture (SITAR), and introduced a state transition model specific to the SITAR that describes the dynamic behavior of multiple intrusion tolerance strategies existing in the SITAR system as well transforming this model into continuous time Markov chain.

Guo et al. in [65] exploited the design diversity available from off-the-shelf operating system products and proposed a game theoretic approach for studying the optimal diversification strategy of configurations of virtual replicas for improving the resilience of the service in the presence of attacks.

El mir et al. in [66] proposed a novel approach that performs proactive and reactive measures to ensure a high availability and to minimize the attack surface using VM migration and formulated a game theoretic approach to present the interaction between attack and defense systems.

Njilla et al. in [67] introduced a game theoretic framework for security and trust relationship in cyberspace for users, service providers, and attackers.

Kamouha et al. in [68] discussed a game theoretic model to help the online social network users determine the optimal policy in terms of data sharing using a zero-sum game model.

El mir et al. in [53] explained an important branch of MTD mechanisms named IP address hopping mechanism using detection systems for attack mitigation in the cloud environment and proposed a game theory approach to model the attack-defense interaction and analyze the effect of MTD on the payoff of both the attacker and the defender.

In short, the previous researches are mostly focused on defense against external attacks, obviously, the intrusion can occur in network system not only from the external attackers but also from internals. Moreover, internal attackers have a greater level of access contrary to the external attackers who have to disable the external defenses before they can log into a company's network. In this paper, we are interested in modeling a game theoretic approach of the defender and both types of attackers (Outsider and Insider) and analyzing the effect of ITS on the payoffs of the players.

4.3 System and Game Theoretic Models

4.3.1 Studied Architecture

Figure 1 illustrates our studied system: A network security of a company, with the use of mechanisms of defense such as firewalls and IDSs.

After detecting an intrusion, the employed IDS alert the network administrator to take an action to stop or mitigate the attack [69]. However, current IDSs mechanisms may prove effective for defending against casual attackers using well known techniques, they are still ineffective against unknown and

undetected attacks.

Furthermore, the intrusion can occur in network system not only from the external attackers but also from the legitimate users as internal attackers[70]. The intruders are defined externals in the sense that they are not registered as users of network system. Their worries are how to detect and bypass the mechanisms of authentication and authorization. The internal attackers such as employees in the company which are already registered and they are looking for violating the system integrity for instance can modify or destroy the sensitive information without any authorization or through malicious actions [71].

For this purpose, Intrusion Tolerant Systems have been added to insure a continuation of services even when the system is under active attack or partially compromised.

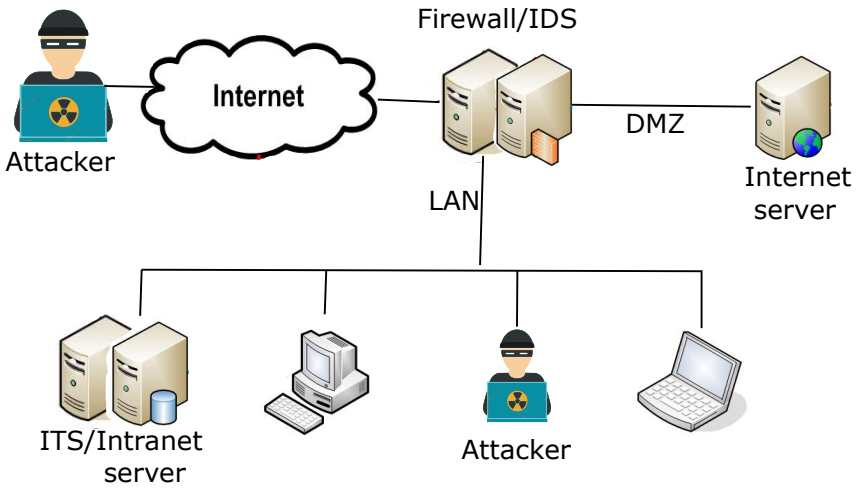

Figure 17: Architecture system

4.3.2 The Bayesian Game Model

This section considers a Bayesian game with two players: An attacker and a defender, where the attacker's type can be an outsider or an insider. Our assumption is that the two players are rational. Therefore, they understand the system in place and can perform the necessary calculation to only take the actions that improve their expected payoff.

The attacker has two actions: launch an Attack (Attack) on the defender's infrastructure or Not to launch an Attack (Not Attack). Only one of the two actions can be used at a time by an attacker.

The defender's actions are defined as follow: to either Invest in Tolerance (Invest Tolerance) or Not to Invest in Tolerance (Not Invest Tolerance).

A strategy profile for this game is represented as a 2-tuple indicating the action taken by each player. For example, the strategy profile (Attack, Invest Tolerance) means that the attacker launches an attack to breach the defender's system, and the defender invest in tolerance to mitigate attacks in case of a successful one.
Let us examine the payoff/utility structure of the game, which is the player satisfaction. We normalize the payoffs to the players following the strategy profile. the parameters used are:

- The parameter β represents the probability that critical data in the defender's system are compromised given a successful attack.
- The parameter L represents the estimated loss to the defender.
- The parameter C_T represents the cost of tolerance.
- The parameter C_D represents the cost to defend the system from an outside attack.
- The parameter $cost_O$ represents the cost of attacker of attacking from outside.
- The parameter $cost_I$ represents the cost of attacker of attacking from inside. with $cost_O \succ cost_I$
- The parameter λ represents the probability of an attacker getting detected or caught on the defender's system.
- The parameter P represents the loss to the attacker from getting detected while launching an attack on the defender's system.

We suppose that the defender invests in the defense system against the attacks from the outside.

Table I and Table II show the game model in a normal form, where the table I illustrates the game model when the attacker's type is Outsider, and table II shows the game model when the attacker's type is Insider. The payoffs of the two players are represented in each block in two lines. The first line in the block is the defender's payoff and the second line is the attacker's payoff.

The payoffs are calculated as follows: If the player choose the strategy profile (Attack, Invest Tolerance), where the attacker choose to launch an Attack and the defender choose to Invest in

Table 2: Table game in case of outsider attacker

	Attach	Not Attack
Invest Tolerance	$-\beta L - C_T - (1-\lambda)C_D$ $\beta L - cost_O - \lambda P$	$-C_T - C_D$ 0
Not Invest Tolerance	$-L - (1-\lambda)C_D$ $L - cost_O - \lambda P$	$-C_D$ 0

Table 3: Table game in case of insider attacker

	Attack	Not Attack
Invest Tolerance	$-\beta L - C_T$ $\beta L - cost_I$	$-C_T$ 0
Not Invest Tolerance	$-L$ $L - cost_I$	0 0

Tolerance, the payoffs if an attacker is outsider differ from the insider attacker.

The defender will mitigate his loss β L for investing in tolerance which cost a cost C_T and a cost in investing in defense systems against the outsiders $(1-\lambda)C_D$,which is not the case against the insider attackers. The attacker payoff is the benefit βL of launching an attack without getting caught or detected less the cost of investing in attack ($cost_O$ for an outsider attacker and $cost_I$ for an insider attacker) less the probability λ of getting detected:

$$U_{def}(A, IT) = -\beta L - C_T - (1-\lambda)C_D \tag{4.1}$$

$$U_{att}(A, IT) = \beta L - cost_O - \lambda P \tag{4.2}$$

4.4 Game and Equilibrium Analysis

The principal aim of this analysis is to extract the different Nash Equilibrium of the game in both tables I and II and understand their impact on both players. Per definition of Nash Equilibrium (NE) profile: no player can increase his payoff by a unilateral deviation. Beside this, players are rational, necessarily each of them is playing his Best Response (BR) to the other BR. Solving the NE is leading to predict the behavior of rational players in the game.

Case 1: if $\beta L - cost_O - \lambda P \succ 0$

Then the strategy (Not Attack) of the Outsider/Insider attacker is strictly dominated by the strategy (Attack). Thus the Best Response for Outsider/Insider attacker is to launch an Attack (Attack).

- If $-L \succ -\beta L - C_T$:

  Then $U_{def}(Attack, NotInvestTolerance) \succ U_{def}(Attack, InvestTolerance)$, the defender prefers not to invest than invest. Thus, the strategy profile (Attack, Not Invest T)olerance is a pure NE because neither the defender nor the attacker can increase their payoff by unilateral deviation.

- $-L \prec -\beta L - C_T$:

  Then $U_{def}(Attack, InvestTolerance) \succ U_{def}(Attack, NotInvestTolerance)$, the defender prefers to invest than not to invest. Thus, the strategy profile (Attack, Invest Tolerance) is a pure NE because neither the defender nor the attacker can increase their payoff by unilateral deviation.

Case 2: if $L - cost_I \prec 0$

Then the strategy (Attack) of the Outsider/Insider attacker is strictly dominated by the strategy (Not Attack). Thus the Best Response for Outsider/Insider attacker is not to launch an Attack (Not Attack).

$U_{def}(NotAttack, NotInvestTolerance) \succ U_{def}(NotAttack, InvestTolerance)$, the defender prefers not to invest than invest.

Thus The strategy profile (Not Attack, Not Invest Tolerance) is the pure NE.

Case 3: if $L - cost_O - \lambda P \prec 0$ **and** $\beta L - cost_I \succ 0$

Then, the strategy (Attack) of the Outsider attacker is strictly dominated by the strategy (Not Attack). Thus the Best Response for Outsider attacker is not to launch an Attack (Not Attack).And

the strategy (Not Attack) of the insider attacker is strictly dominated by the strategy (Attack). Thus the Best Response for insider attacker is to launch an Attack (Attack).
We suppose that the type of the game is α, i.e, the probability of getting an Outsider attacker is α. Otherwise, the probability is equal to $(1-\alpha)$.

$$\alpha(-C_T - C_D) + (1-\alpha)(-\beta L - C_T) = -\alpha C_D - (1-\alpha)L \tag{4.3}$$

$$\alpha = \frac{L - \beta L - C_T}{L - \beta L} \tag{4.4}$$

- If $\alpha \succ \frac{L-\beta L-C_T}{L-\beta L}$:

  Then the attacker's type is Outsider, his Best Response is (Not Attack).
  $U_{def}(NotAttack, NotInvestTolerance) \succ U_{def}(NotAttack, InvestTolerance)$ Thus, the strategy profile (Not Attack, Not Invest Tolerance) is a pure NE.

- If $\alpha \prec \frac{L-\beta L-C_T}{L-\beta L}$:

  Then the attacker's type is Insider, his Best Response is (Attack). We have $\alpha \succ 0$, i.e, $L-\beta L-C_T \succ 0$ which means, $U_{def}(Attack, InvestTolerance) \succ U_{def}(Attack, NotInvestTolerance)$
  Thus, the strategy profile (Attack, Invest Tolerance) is a pure NE.

- If $\alpha = \frac{L-\beta L-C_T}{L-\beta L}$:

  There are no pure strategy profiles for a NE in the game. However, a mixed strategy NE is highly plausible, and it's defined as follow:

  $$NE_{mixed} = (x.Attack + (1-x).NotAttack, y.InvestTolerance + (1-y).NotInvestTolerance) \tag{4.5}$$

  Where x is the probability that the defender plays Invest Tolerance strategy and y the probability that the attacker plays Attack strategy. We formulated the expected payoff of the defender when he plays Invest Tolerance action as follow:

  $$\begin{aligned} EP_{def}(InvestTolerance) = \alpha(y(-\beta L - C_T - (1-\lambda)C_D) + (1-y)(-C_T - C_D)) \\ +(1-\alpha)(y(-\beta L - C_T) - (1-y)C_T) \end{aligned} \tag{4.6}$$

  And his expected payoff when he plays Not Invest Tolerance strategy:

$$EP_{def}(NotInvestTolerance) = \alpha(y(-L-(1-\lambda)C_D)-(1-y)C_D)-(1-\alpha)yL \quad (4.7)$$

The expected payoff of the attacker when he plays Attack strategy is:

$$\begin{aligned} EP_{att}(Attack) = \alpha(x(\beta L - cost_O - \lambda P) + (1-x)(L - cost_O - \lambda P)) \\ +(1-\alpha)(x(\beta L - cost_I) + (1-x)(L - cost_I)) \end{aligned} \quad (4.8)$$

And his expected payoff playing Not Attack strategy is:

$$EP_{att}(NotAttack) = 0 \quad (4.9)$$

From basic principles of game theory, the defender optimal strategy is to choose x in such a way the attacker is indifferent when deciding between Attack and Not Attack. Similarly, the attacker choose y such that the defender is indifferent in choosing Invest Tolerance and Not Invest Tolerance. This is translated by:

$$U_{def}(InvestTolerance) = U_{def}(NotInvestTolerance) \Leftrightarrow y = \frac{C_T}{L - \beta L} \quad (4.10)$$

$$U_{att}(Attack) = U_{att}(NotAttack) \Leftrightarrow x = \frac{\alpha(cost_O + \lambda P - cost_I) - L}{\beta L - L} \quad (4.11)$$

4.5 Numerical Results

In this section, we derive the numerical results from the game analysis equilibrium that provided an explanation of the game model. Indeed, the variables used thereafter are: L, $C_T, C_D, cost_O, cost_I, P, \lambda$. The values allocated will remain fixed during the MATLAB simulation to illustrate concrete examples, while the value of β is variable.

$L = 2000, C_T = 200, C_D = 150, cost_O = 2000, cost_I = 300, P = 600, \lambda = 0.5$

For the scenario, when $\alpha = 0.8$, we conclude that the attacker's type is Insider when $0.15 \leq \beta \prec 0.2$ and Outsider when $0.5 \leq \beta$.

According to Figure 24, the defender's best response is Invest Tolerance when $0.15 \leq \beta \prec 0.2$

while his best response when $0.5 \leq \beta$ is Not Invest Tolerance. As illustrated in the same figure, we have two mixed Nash equilibrium when $\beta \prec 0.15$and $0.2 \prec \beta \prec 0.5$.

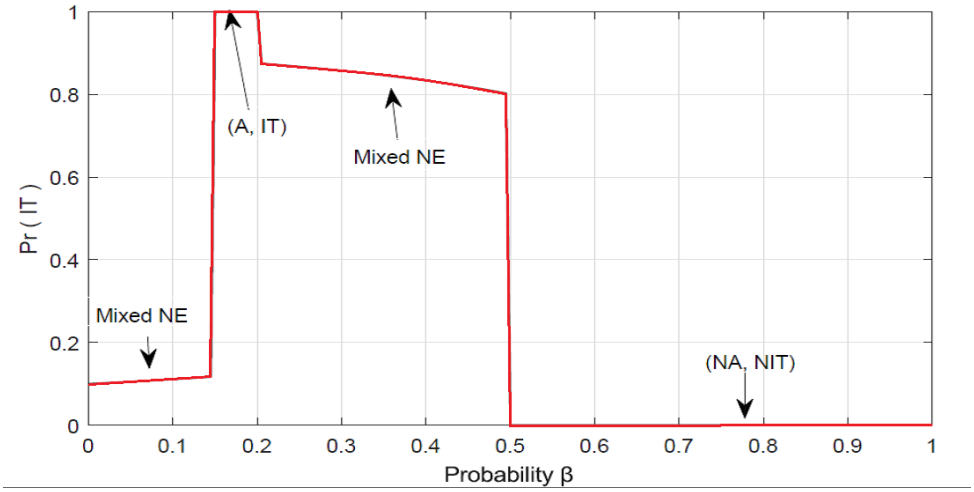

Figure 18: The probability of Investing in Tolerance

Figure 26, shows that the best response of the attacker is Attack when $0.15 \leq \beta \prec 0.2$ while his best response when $0.5 \leq \beta$ is Not Attack. In the same case, we have two mixed Nash equilibrium when $\beta \prec 0.15$and $0.2 \prec \beta \prec 0.5$.

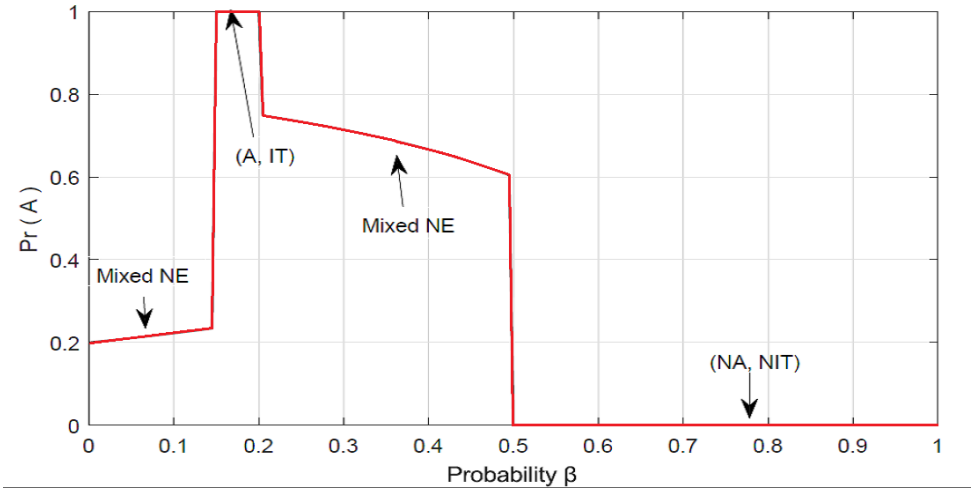

Figure 19: The probability of Attacking

Figures 20, 33 and **??** show that the utility of the defender respectively when $\beta \prec 0.15$, $0.15 \leq \beta \prec 0.2$ and $0.2 \prec \beta \prec 0.5$ decreases considerably by the increase of β.

On the level of the variation of the defender's utility represented in figure 21, it is clear that the latter decreases by the increase of β. In addition for values of $0.15 \leq \beta \prec 0.2$ the defender's utility

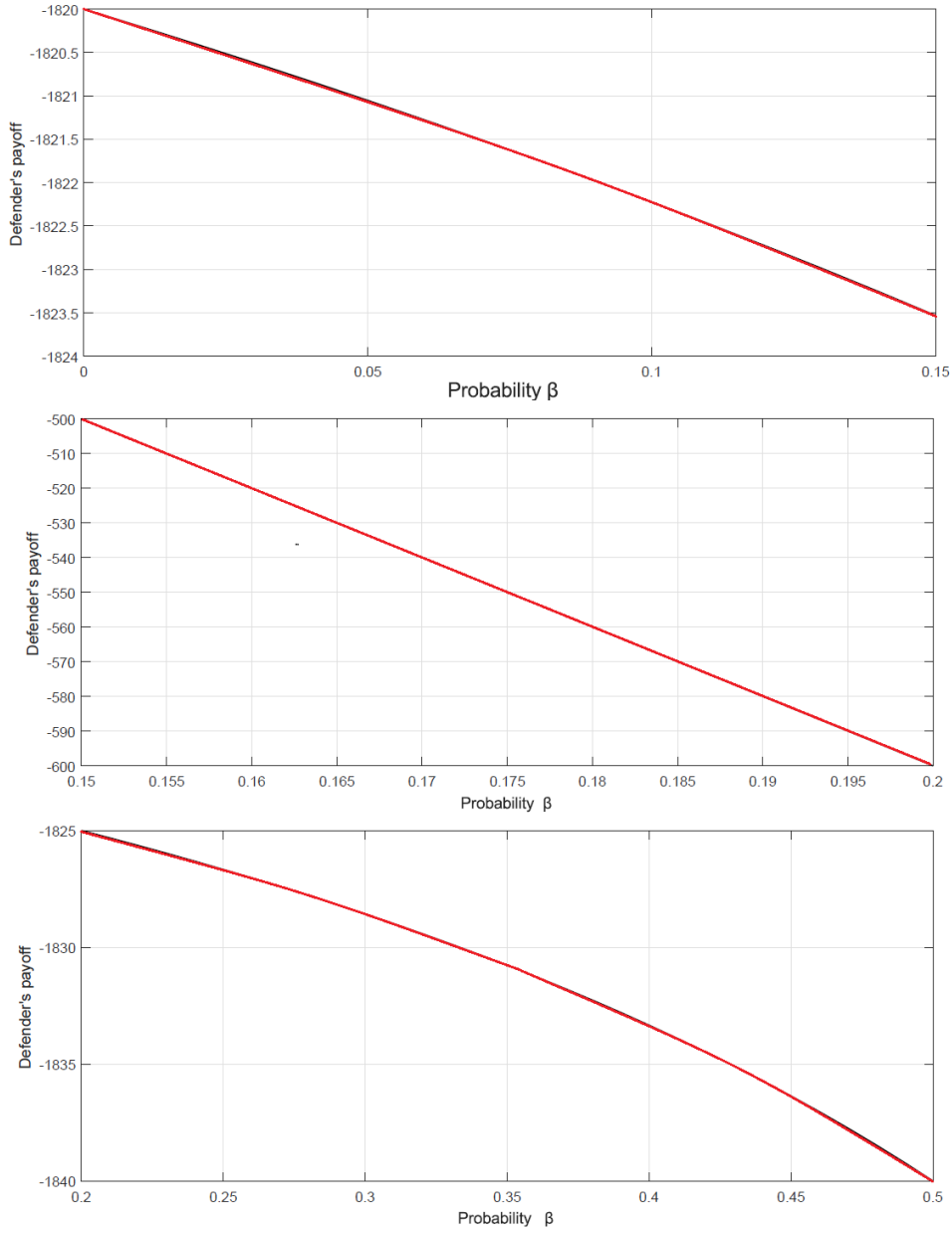

Figure 20: The probability of Investing in Tolerance

is considerably higher because the defender is investing in tolerance.

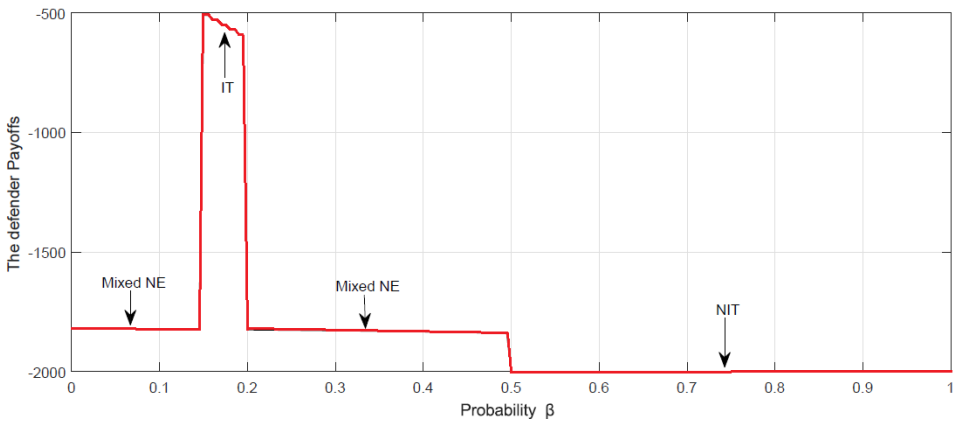

Figure 21: The probability of Investing in Tolerance

4.6 Conclusion and Future Work

In this chapter, we are interested in introducing ITSs in the network security and using game theoretic approach to model the attack-defense interaction, then analyze the effect of intrusion tolerant system on the payoffs of both players.

The defender and the attacker provide a quantitative approach to perform a cost analysis of the security investment. This research take into account the actions of the two players. The game has multiple Nash equilibrium that can be converted into pure strategy or mixed strategy under specific conditions. We have calculated the probability of attacking strategy and investing in tolerance strategy that both increase with the increase in the probability of compromising the system, which means that the defender should invest in tolerance with the increase of attacks.

For future work, we plan to study the dynamic Bayesian version of the extended proposed scenario that includes different attacker profiles, in order to derive a more realistic decision.

5

Intrusion Tolerant Controller

In this Chapter, we propose to approach the issue of intrusion tolerance in the SDN control plane by first applying a Recovery Based model which assumes that as soon as a system comes online it is compromised; therefore, periodic restoration to a good state is necessary. Secondly, we aim to establish Moving Target Defense (MTD) that provides a proactive defense against adaptive adversaries. The goal of the MTD in the Dispatcher is to constantly shift between multiple controllers with diverse configurations in order to increase the uncertainty for the attacker, in effect, diminishing the information gathered from the control plan during the reconnaissance phase of a potential attack. Finally, We put in place probabilistic models that can contribute to the perception of the performance of self-cleansing intrusion tolerance in the SDN control plane.

This chapter is the subject of our paper (Sanoussi et al., 2023) [16].

5.1 Introduction

With the continuous development of network applications, the most disconnected individuals have become actually highly connected with each other [72]. However, the traditional network has revealed a deficiency of their inflexible structure and complex configuration and cannot keep up the network innovation [73]. Furthermore, conventional network architectures lack global visibility of

the network state and have difficulties in deploying and maintaining coherent network-wide policies. This complexity and weaknesses in integration make it worse for maintaining stable and robust network security. For example, changing or updating security policies in these systems in the case of abnormal behavior or intrusions is practically unmanageable and priced [74]. Thus, Software-Defined Networking (SDN) is suggested to outdo these weaknesses.

SDN become one of the most significant network architectures that simplify network management and make communication networks easier and its main feature is separating the network control from the data forwarding plane. To elaborate on the employment of the SDN architecture, three principal SDN layers are presented as follows:

- **The application plane:** which contains applications such as policy implementation, network management, and security services.
- **The control plane:** It is a logically centralized control framework that runs the Network Operating System, maintains a global view of the network, and provides hardware abstractions to SDN applications.
- **The data plane:** that forwards traffic flows based on instructions from the control plane.

Nevertheless, the use of SDN leads to new security challenges in the control plane and in the two other planes, such as man-in-the-middle attacks, denial-of-service attacks (DoS), and saturation attacks, etc. As a centralized controller responsible for managing the whole network is considered as a single point of failure can render the network compromised.

Unfortunately, the existing multi-controller architectures lack security and safety mechanisms. Using a multi-controller mechanism alone cannot avoid the SDN at the single point of failure: If a controller fails, its load should be carried by other controllers, which leads to exceeding their capacity, and then cascading failure of controllers will occur [75].

Motivation. SDN security tackles the problem of bottleneck control plane by suggesting multi-controllers architectures to achieve higher scalability and availability. Unfortunately, two main problems remain:

- Simply using multi-controllers in SDN cannot avoid the threat of the single point of failure

because, when a controller fails, his load will turn to other working controllers, which may transcend their capacity. So resulting in a global failure.

- Besides, the visible nature of the control plane makes it more vulnerable to attacks, particularly to DoS and DDoS (Distributed DoS) attacks that are the most threatening security challenges for the SDN control plane [74]. As demonstrated in [76] the authors use a primary and a secondary controller, and the attacker bombs IP packets with random headers to render the primary controller in a non-responsive state. However, the secondary controller is also susceptible to DoS and DDoS attacks since a detection mechanism for these attacks is not employed.

Contributions. This chapter suggests a new architecture based on intrusion tolerance in SDN-enabled networks. The design of this solution is based on using two proposed mechanisms, one used at the level of the control plane, the other underneath.

- The first mechanism based on a hierarchical multi-controller uses the Self Cleansing Intrusion tolerance approach. This consists of a root controller managing other regular controllers with diverse operating systems which creates a diversity protection issue. In addition, it maintains its recovery circle as described in the Self Cleansing Intrusion Tolerance (SCIT) architecture subsection.

- The second mechanism is to move the regular controllers in a continuous way by the employment of the Moving Target Defense technique, to render their visible nature obscure for attackers to exploit.

- Finally, we introduce a set of probabilistic models to investigate the performance of Self Cleansing Intrusion Tolerance in the SDN Control plane.

The remainder of this chapter is structured as follows: Section 5.2 presents the pertinent literature review. Section 5.3 overviews the multi-controller designs, the SCIT architecture, and the MTD technique. Section 5.4 describes in detail the architecture and the mechanisms that we propose to ensure intrusion tolerance in SDN. In Section 5.5, we show the analysis of probabilistic models in SDN. Finally, the conclusion and future work are summarized in Section 5.6.

5.2 Literature Review

The adoption of SDN technology has increased in the latest years, due to its benefits that attract the attentiveness of the industry and research community [9].

In this section, we present a literature review in relation to three concepts on which our proposal is based.

Multi-controller. Redundancy is one of the most significant principles of tolerance in any system. A single controller is a single point of failure that could fail anytime leaving the network without its control plane. Consequently, multi-controller architectures are crucial for SDN, as mentioned in [77]. This paper proposes fault tolerance mechanisms for basic SDN by using a slave controller architecture. The first mechanism uses a virtual controller redundancy based on the existing Virtual Router Redundancy Protocol (VRRP) used on routers in traditional networks. The second mechanism is a light synchronization of information about the network view and flow decisions between the controllers.

In [78] the authors describe existing fault-tolerant SDN controller solutions and put forward a mechanism to design a consistent and fault-tolerant Master-Slave SDN controller by a replication scheme combined with a consistency check and a correction mechanism.

The following works refer to controller clustering architectures to achieve load balancing. These strategies are generally built of one super controller and multiple regular controllers. The super controller is used in managing all regular controllers loads. BalanceFlow [79] is a controller clustering design based on a hierarchical model. All controllers in BalanceFlow preserve their own load information and share it periodically with each other through a cross-controller communication system. [80] and [81] are similar to BalanceFlow by defining a super controller to manage controllers' load. However, [80] proposes a load-balancing scheme for hierarchical controller configurations. The authors in [81] break the usual dependency between super controller and regular controller, where this latter has its own Cluster Vector (CV), which helps the regular controllers to find each other and query about loads of each other.

On the other hand, in [82] the authors propose a dormant mechanism model based on the flat design for multi-controller to save network resources, reduce energy by letting some controllers enter the dormant state to be inactive or power off when the network's load is light.

Some works in multi-controllers are concentrated on identifying the optimal location of the controllers. In [83] the authors propose a hybrid metaheuristic algorithm to deploy multiple controllers effectively. The concept is based on selecting the optimal controller based on controller features using an optimization algorithm, then placing multiple controllers based on the selected controller using a hybrid metaheuristic algorithm. The authors in [84] work on detecting first the number of controllers needed, then discover the optimal location of controllers in the network, and after that implement controller migration.

These architectures provide only redundancy (multi-controller) through different designs to achieve performance, scalability, and load balancing but not intrusion tolerance. Such as an attacker could compromise the SDN control plane since he could easily succeed in the reconnaissance phase. Our approach aims to let this step more difficult for the attacker in different manners.

Intrusion Tolerance. There are various solutions to secure and defend SDN, however, the researchers working on intrusion tolerance in SDN are bounded. Some of the architectures such as those described in [31] are treating the issue of intrusion tolerance by integrating an intrusion detection module in the OpenDayLight SDN, that monitors the flow as detailed in Chapter 2.In addition, the authors in [85] propose an SDN fault-tolerant and a resilient SDN controller design approach achieved by a three-tier approach: Risk assessment, Intrusion detection, and self-healing.

Undetected intrusions are unavoidable and ought to be handled as an inherent problem in any system. Consequently, using a Recovery Based technique is a must. [86] describes the Self Cleansing Intrusion Tolerance (SCIT) and develops an algorithm that manages intrusion tolerance through server rotation and cleansing. [87] is one of the papers working on Recovery Based techniques by designing a Cloud-based Self Cleansing Intrusion Tolerance (C-SCIT) architecture that ensures intrusion tolerance for services and applications utilized in the Cloud. They analyze the challenges in

their approach compared to traditional implementations. Besides that, the authors in [88] propose Duo, an ITS integrated in SDN in order to reduce the exposure time without consuming computer resources by assorting network traffic into two groups with the help of SDN: Suspicious traffic is transmitted to Grey servers and benign traffic forwarded to White servers.

In our proposed approach, since the redundancy alone is insufficient to ensure intrusion tolerance, we suggest adding to the SDN multi-controller architecture a Recovery Based Model (SCIT) to minimize the probability to be compromised. In addition, the controllers involved in the SCIT cycle host different Operating Systems (OS), so that the attacks that are supposed to succeed for one OS (Windows, macOS, Linux, etc) will not necessarily threaten another different type of OS, which will ensure the diversity issue in protecting the SDN controllers.

Moving Target Defense (MTD). MTD is used to acquire dynamic and proactive network defense. It aims to render the information collected by the attacker invalid by changing the attack surface, thus, frustrating the attack operation. In [89], the authors propose an SDN-based MTD mechanism employing the shuffling of IP (Internet Protocol) addresses and port numbers to obscure both the network and the transport layers in pursuit of defending the system against network reconnaissance and scanning attacks. This mechanism named Random Host and Service Multiplexing (RHSM) allows each host to use random, multiple virtual IPs to be dynamically and periodically shuffled. The work in [90] similar to RHSM, aims at designing an SDN-based MTD architecture using multi-controllers in SDN and strengthens their security by implementing an IP shuffling technique utilizing ONOS SDN controllers.
To get acquainted with the techniques used in our work, the next section provides an overview of them.

5.3 Technical Background

In this section, we present in the first subsection the two multi-controller designs. In the second subsection, we introduce the SCIT architecture, its advantages, and its life cycle. After that, we describe the MTD mechanism.

5.3.1 Multi-Controller Overview

In the original SDN design, the whole network is handled by a single controller. This latter is responsible for forwarding all sorts of rules and policies to the data plane that contains the forwarding devices such as switches, load balancers, etc. As a result, a single controller in SDN is defined as a single point of failure or called sometimes a bottleneck. On the other hand, with the expansion of the network, a single controller will fail in managing the huge amount of the OpenFlow processing requests and all other responsibilities. Thus, the deployment of the multi-controller design is necessary.

A multi-controller architecture is composed of a collection of controllers in the control plane to manage all together the Software Defined Network to obtain a certain performance, scalability, and tolerance. Some surveys such as [91] categorized the multi-controller architecture into flat and hierarchical designs. In both of them, the network topology is structured into a group of domains, each domain is locally managed by a controller. However, in the flat model, as mentioned in Figure 1, the control plane includes just one layer, where the controllers are on the same level and communicate with each other to get a thorough vision of the SDN topology.

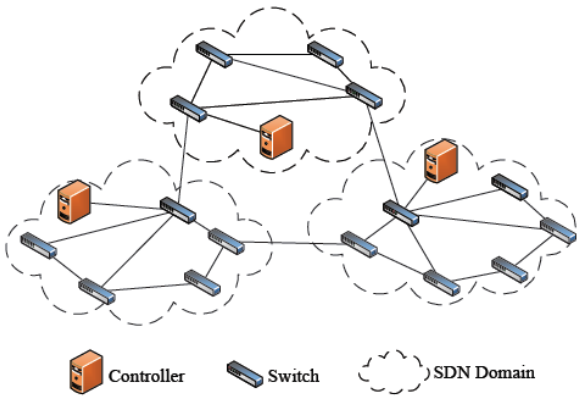

Figure 22: Flat Design

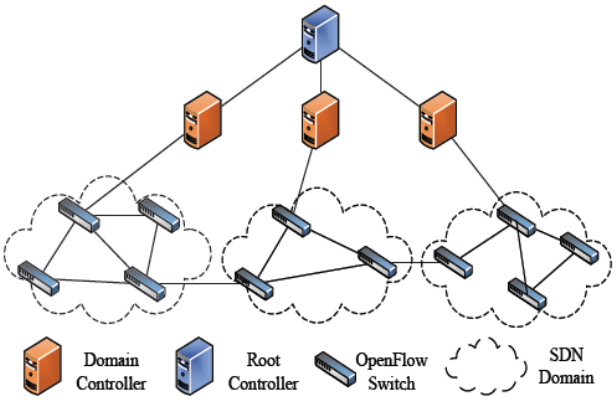

Figure 23: Hierarchical Design

While in the hierarchical model, the control plane contains various layers, every layer includes at least one controller. As an example in Figure 2, the control plane is composed of two layers, the controller on the top layer (Root controller) manages the controllers in the layer beneath (domain controllers) and maintains the global vision of the network topology [92]. In its turn, each domain controller takes care of its local domain.

5.3.2 SCIT Architecture Description

Self Cleansing Intrusion Tolerance (SCIT) is a recovery-based system founded on the principle that as soon as a system comes online, it is compromised, therefore, periodic restoration to a good state is necessary [87].

The main advantages of the SCIT architecture are:

- It ensures the diversity of the server, so the attacker should make more efforts to compromise a system.
- It shortens the exposure time of the server, therefore, rendering the information gathered from the reconnaissance phase inaccurate for the attacker.
- It reduces losses by controlling the time that a server is exposed to the network.

The SCIT architecture is founded on two fundamental components as shown in Figure 3.

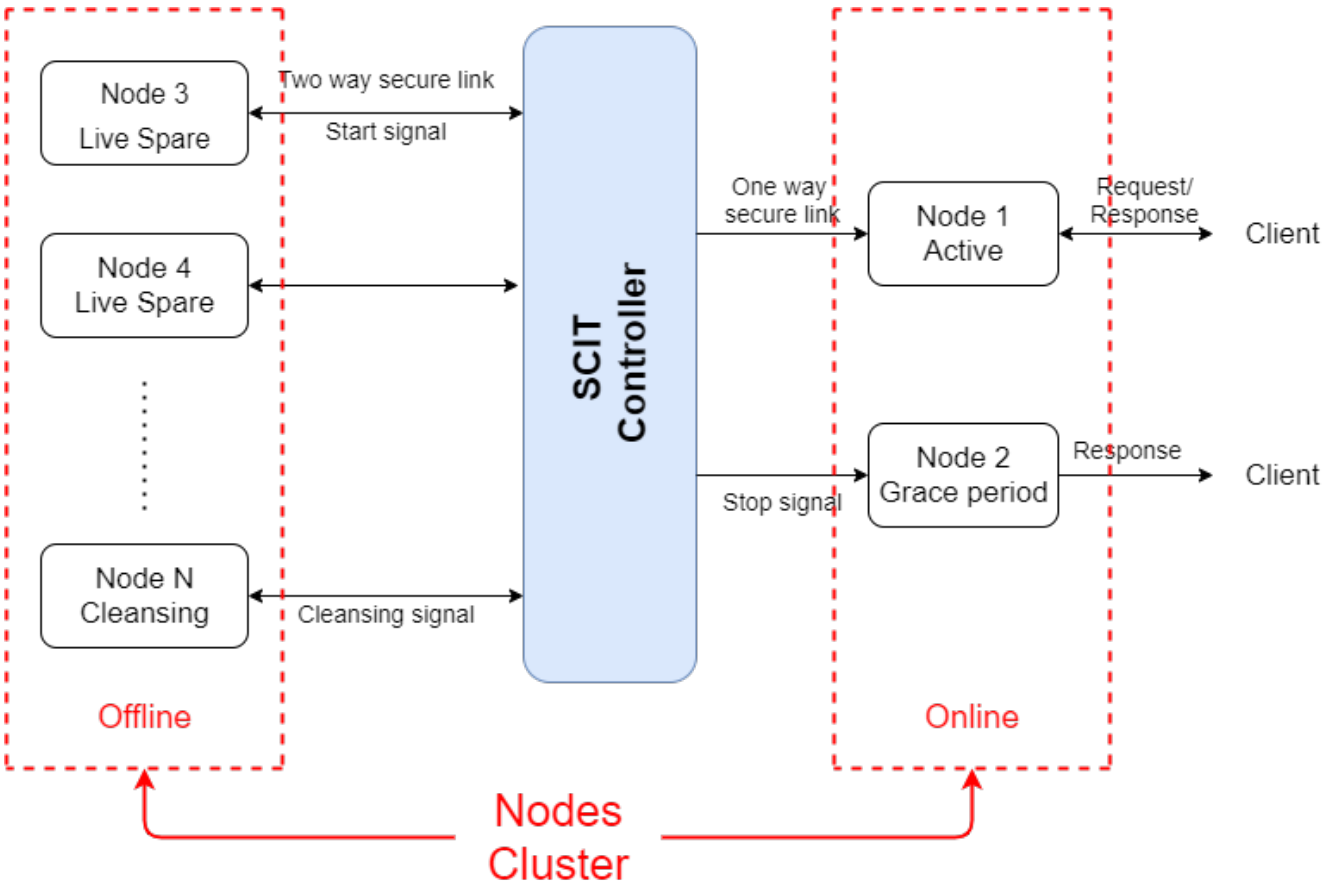

Figure 24: SCIT Architecture

A server cluster is a group of nodes doing the same tasks and affording the same services but running in different operating system platforms to ensure diversity, thus making malicious exploitation more difficult.

Each node is continuously routed through the following life-cycle states [93] as:

- **Active:** Node is online and accepts/processes any incoming requests.
- **Grace Period:** Node processes any existing requests, but doesn't accept any new ones.
- **Cleansing:** Node is offline and undergoes the cleansing to get to a known good state.
- **Live Spare:** Node has been restored and is ready to come online.

The SCIT controller is the center part that manages server rotation in and out of the cleansing mode.

5.3.3 Moving Target Defense

Network defense techniques based on static defense fail, because of the limitations of their nature. The attackers typically have asymmetric advantages and the defenders are always disadvantaged by being passive. To change the asymmetric situation between attacks and defenses, Moving Target Defense (MTD) has emerged as a technique that supplies enhanced security based on the dynamically re-configuring of the underlying systems. Its goal is to move the components of a system in a continuous way. This makes the exploitation phase harder and/or costlier and increases the uncertainty in the attacker strategy. More specifically, the information gathered by the attacker in the reconnaissance phase might become useless during the attack phase if the defender has moved to a new configuration in that time [94].

The substantial concept for evolving MTD techniques resides in the conclusion of these three questions: What to move, How to move, and When to move. [95].

"What to move": Deals with system surfaces that can be dynamically altered to perplex attackers, which are defined as follows [94]:

- **Exploration Surface Shifting:** Ensures the inexactness and the faultiness of the information gathered by an attacker, like open ports, system topology, vulnerabilities, etc. Therefore, the attacker, with this inaccurate information from the reconnaissance phase, will have to execute attacks randomly to exploit the system.

- **Attack Surface Shifting :** Guarantees the invalidity of an attack action chosen after some exportation, by switching between attack surfaces. For example, An attack to exploit a Linux Based OS will be useless if it is launched against a machine running a Windows OS.

- **Detection Surface Shifting and Prevention Surface Shifting:** Confuse the attacker's mind about the defense mechanisms used. So, that renders the attack process expensive.

- **Multi-Surface Shifting:** Simply created by combining two or more of the described surfaces shifting above.

In this paper, we will use the Exploration Surface Shifting and the Attack Surface Shifting.

"How to move": Describes how to switch the attributes to increase the uncertainty and confusion to the attacker. This is related to the MTD techniques classified into [96]:

- **Network Address shuffling:** This Is a technique that aims to change the IP address (and port number) of the target periodically or randomly.

- **Diversity:** This Is the ability to generate the same services with different implementations such as operating systems.

- **Redundancy:** Creating multiple replicas of a node, a service, or data.

"When to move": Defines the optimal time to change from the current state of an MTD system to a new state to ruin the progress gained by an attacker during the current state via rendering the

information gathered invalid [95]. The fundamental policy to reach that is by applying one of the three following approaches:

- **Time-Based:** Presented as a proactive adaptation, where the MTD operation shift the system surfaces (exploration surface, attack surface...) periodically on a Constant Period Switching which is based on a fixed time interval, or on a Variable Period switching based on a variable time interval.

- **Event-Based:** Defined as a reactive adaptation of the MTD mechanism executed when a particular event happens such as an indication of accessing a system or attempting to launch a certain attack by an intruder.

- **Hybrid:** Performing the MTD operation by combining the Time-Based and Event-Based in an adjustable way.

In the following section, we will present our proposition of an Intrusion Tolerant Controller (ITC) for a Multi-Controller SDN architecture, based on the SCIT architecture combined with the MTD mechanism.

5.4 Intrusion Tolerant Controller (ITC) Architecture

Three mechanisms are combined to ensure intrusion tolerance in an SDN control plane architecture. The first subsection presents the global architecture of our proposal while the second one goes into detail about the technical aspects.

5.4.1 Global Architecture Description

In this section, we present our proposed Intrusion tolerant Controller architecture. As shown in Figure 4, a hierarchical design of a multi-controller scheme is settled down. The control plane contains three layers:

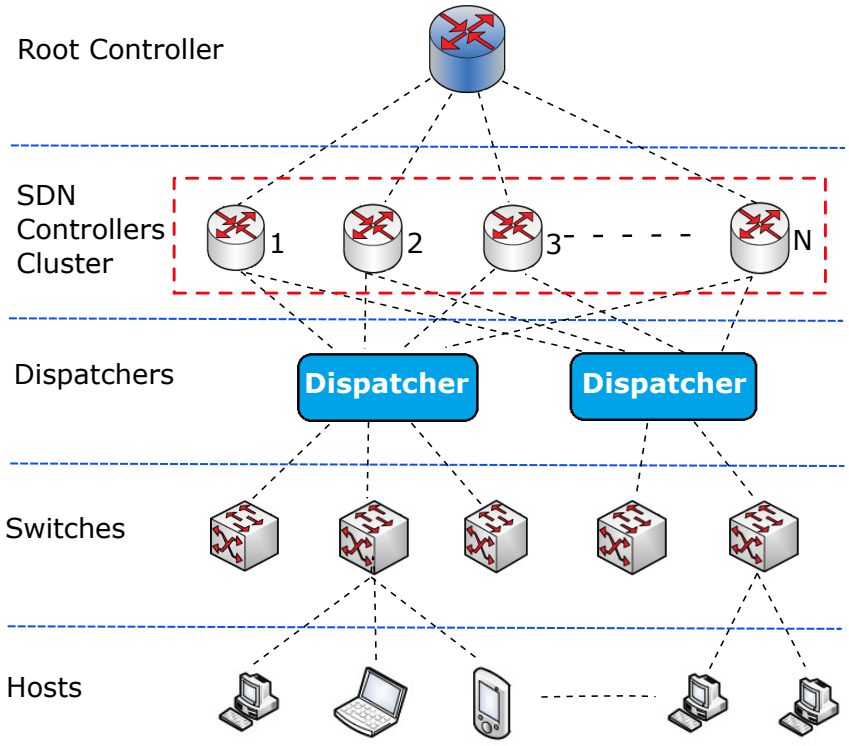

Figure 25: The system architecture

- The Root controller on the top layer manages the flow request handled by each domain controller and maintains a global vision of the network topology. Besides, it performs the role of the SCIT controller by managing the controllers' rotation in and out of the cleansing mode. Note that the Root controller is replicated, and the internal controller network is completely separated from the external network.

- The domain controller manages its temporary local domain, which changes periodically by using the SCIT rotation (detailed in the paragraph below).

- The dispatcher laid in the layer beneath the controllers' cluster, is in charge of forwarding the flow to the active controllers applying a load balancing mechanism to avoid the overload of the active controllers, in addition to representing an interface between the control plane and the data plane. In other words, the switches are unaware of the architecture and the layers laying above the dispatcher.

This architecture ensures redundancy by using this collection of controllers based on self-cleansing intrusion tolerance in the control plane to manage all together the Software Defined Networking

by changing the states of every controller and ensuring reconfiguration to obtain more performance, scalability, and availability. Besides, using diversity in controllers' Operating Systems and applications renders it more difficult for the attacker to deal with all these different OSs. In addition, MTD is used to ensure dynamic protection against probing and scanning attacks.

5.4.2 SCIT and MTD Architecture Description

We apply a Recovery Based Model [24] which assumes that as soon as a system comes online it is compromised to minimize the controllers' failure. Figure 5 shows the diagram of the SCIT architecture used in our proposed control plane.

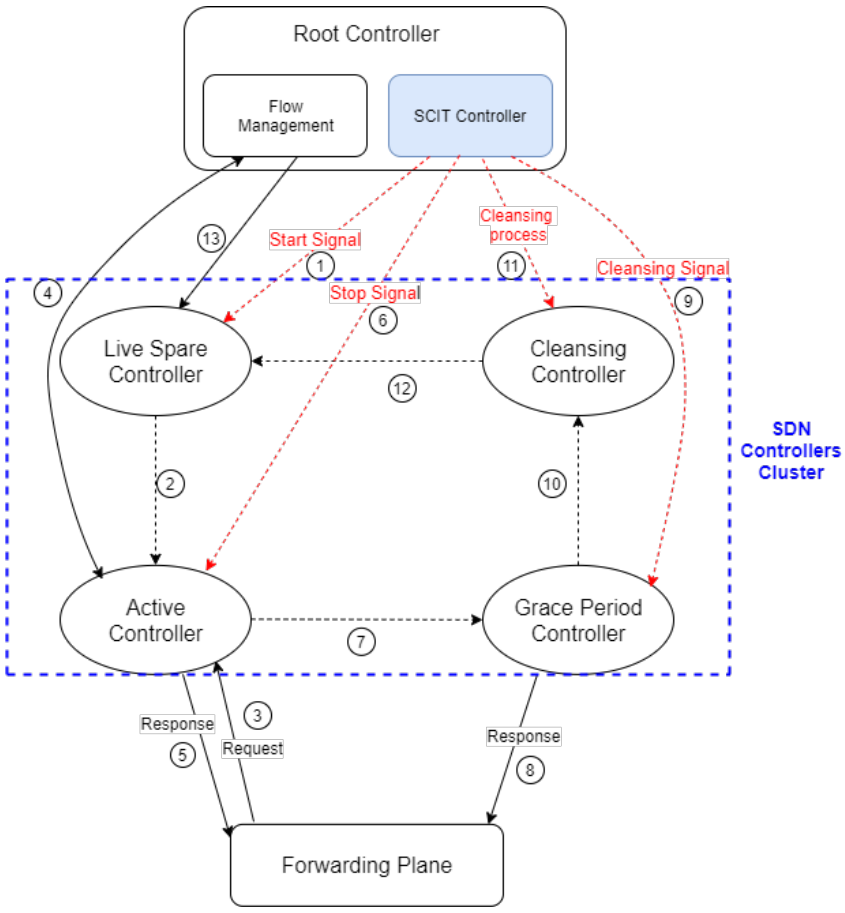

Figure 26: SCIT cycle in the SDN control plane

Each domain controller receives signals from the root controller periodically to rotate the states:

a The SCIT controller installed in the root controller sends a start signal to a controller in the live spare (1). This latter changes the state to active (2). So it starts accepting the incoming packets from the data plane (3), and treating them (5) by exchanging with the flow manage-

ment in the root controller (4).

b After a while, which is a constant period switching except when a particular event happens, the SCIT controller sends a stop signal to a controller in the active state (6), and switches the state to grace (7) by treating only the existing packets in his line (8) and refusing any other coming packets.

c Subsequent to treating the existing packets, the controller switches to the cleansing state after receiving the cleansing signal (10), where it becomes offline and not visible in the control plane. In this state, it will be cleansed and configured if there are any configurations needed to minimize vulnerabilities.

d After the cleansing period, the controller is restored, it exchanges with the flow management (11) in the root controller to get the minimum knowledge and waits to get the start signal from the SCIT controller to change again to the Live spare controller (12).

This rotation of states is necessary for a periodic restoration to a good state, and it is important in changing the domain controller that manages each local domain periodically. This measure makes the exploitation phase for an attacker that seeks to compromise the domain controller, in a certain local domain, harder if it is not impossible especially if the exposure time during which every domain controller will be online is managed effectively. In our proposal, we opt for a hybrid rotation by applying a proactive adaptation. The root controller rotates the states of the domain controllers periodically on a constant period switching and performs a reactive adaptation when a particular event happens. Besides, every controller of the cluster uses a different operating system based on several criteria and studies like [97] with the aim to ensure diversity, so compromising the whole system would be difficult for attackers.

In the level of the dispatcher, the use of SCIT architecture in the layer above provides a Moving Target Defense mechanism. The objective is to make the information gathered by the attacker in

the control plane reconnaissance phase useless.

The MTD techniques used in this architecture are:

- What to move? Is a combination of Exploration surface shifting and attack surface shifting, the rotation state of the domain controllers generates a dynamic alteration of the system surfaces of the active controllers. This means that at every alteration, the open ports, the Network Operating System (NOS), the vulnerabilities, etc. will change with the change of the active domain controller at every local domain.

- How to move? The Network Address shuffling is gained with the complete alteration of the domain controller at every local domain. The redundancy is reached by the use of a multi-controller design. Using different NOS in the controllers such as ONOS, NOX, POX, FloodLight, RYU, etc. could guarantee diversity at every alteration.

- Finally, when to move? The optimal time to change from the current state of an MTD system to a new state depends on the time defined by the Root controller for every controller to be online. As explained above, we opt for a hybrid alteration, that combines the time based and the event-based approach. The Root controller uses a constant period switching for the rotation unless a particular event happens such as an indication of accessing a domain controller or attempting to launch a certain attack by an intruder.

In the following section, we aim to analyze the performances of SCIT applied to the SDN control plane depending on different criteria.

5.5 Analysis of SCIT in SDN Control plane

In this section, we discuss the scenario in which the attacker seeks to fulfill reconnaissance on the control plane of the SDN, while we try to keep safe our control plane by applying Self Cleansing Intrusion Tolerance. Modeling the system will help to determine if and when SCIT is advantageous.

To do this, we are inspired by the modeling approach used in [98] for analyzing the performance and the efficiency of using Network address shuffling as an MTD defense to protect a network from probing attacks.

Assume the following is true:

- There are m total domain controllers available in the SDN Control plane, o offline controllers, and n=m-o online controllers.
- $v \leq m$ vulnerable controllers in the general case and $v' \leq n$ are vulnerable controllers in SCIT case (vulnerable controllers among those in the online state).
- A SCIT event pseudo-randomly and uniformly remaps and changes the states of all m domain controllers in the SDN.
- The attacker will serially attempt k probes.
- The goal of the attacker is to compromise at least one of the vulnerable controllers in k attempts.

To model the system, we opt to use a **Urn Model** consisting generally of a bowl or urn holding a set of colored balls. The player randomly withdraws a ball from the bowl and notates its color.
In our case, deem a bowl containing n balls, v red balls, and $n - v$ green balls. The bowl is our control plane, red balls represent vulnerable domain controllers, while the green ones represent non-vulnerable domain controllers.
The player (attacker) withdraws one at a time, k balls from the bowl. If the player withdraws at least one red ball (vulnerable controller) then he succeeds. Using our intrusion tolerant mechanism discussed in the section 5.4 especially the SCIT mechanism between every withdrawal is considered a defense strategy. To demonstrate that, we evaluate the extremes in our case: Static multi-controller SDN and SCIT in multi-controller SDN.

5.5.1 Modeling Static Multi-Controller SDN Case

As previously mentioned, the hierarchical design of a multi-controller SDN consists of a Root controller that manages the domain controllers, in their turn manage local domains. Presume the domain controllers assigned to every local domain are fixed. The attacker's strategy to detect a vulnerable

controller is to sequentially shift through them.

The urn model used in this case is called "Sampling without replacement" [99].

- If $k \geq m$, the attacker is certain to discover all υ vulnerable domain controllers.
- If $k < m$, the probability for the attacker to find one vulnerable controller in k attempts is:

$$Pr(\upsilon) = \frac{\binom{\upsilon}{1}\binom{m-\upsilon}{k-1}}{\binom{m}{k}} \tag{5.1}$$

In the general case, let's take X_υ the random variable representing the number of vulnerable domain controllers that could be found by the attacker, so the probability is given by:

$$Pr(X_\upsilon = x) = \frac{\binom{\upsilon}{x}\binom{m-\upsilon}{k-x}}{\binom{m}{k}} \tag{5.2}$$

$$\sum Pr(X_\upsilon) = 1 \qquad where \quad X_\upsilon \in \{0, 1, 2, ...\} \tag{5.3}$$

Then, the probability for the attacker to find at least one vulnerable domain controller is:

$$Pr(X_\upsilon > 0) = 1 - Pr(X_\upsilon = 0) = 1 - \frac{\binom{m-\upsilon}{k}}{\binom{m}{k}} \tag{5.4}$$

5.5.2 Modeling SCIT in Multi-Controller SDN Case

SCIT is the approach proposed in this paper to apply a Recovery Based Model in the control plane, which guarantees that at every state's rotation the attacker loses any information gathered about the domain controllers.

As discussed in the section 5.4, the exposure time of every active controller impacts the attacker's success probability.

The attacker withdraws, one at a time, k balls from the bowl, to get at least one red ball (vulnerable online controller). But with the use of SCIT, he will repeat this move T times, which represents the T rotation state cycle. Because at every rotation state cycle, the attacker will attempt k probes to find the online vulnerable domain controller before changing the states of the domain

controllers by SCIT controller (the end of the exposure time). If the states changed before finding the online vulnerable controller, the attacker will repeat the test T times until getting at least one online vulnerable controller.

The model used in this case is a repeated experiment, which is composed of T tests (one test represents one rotation states cycle) independent from each other. The result of every test is event A: Finding or not finding at least one vulnerable online domain controller.
$X_{v'}$ the random variable equals the times in which the attacker finds at least one online vulnerable domain controller (Success of the event A). The attacker success probability is determined through a binomial distribution given by:

$$Pr(X_{v'} = s) = \binom{T}{s} p^s (1-p)^{T-s} \qquad where \quad s \in \{0, 1, 2, ...T\} \tag{5.5}$$

p is the probability for the attacker to find at least one online vulnerable domain controller in one test. So, as discussed in the subsection 5.4:

$$p = 1 - \frac{\binom{n-v'}{k}}{\binom{n}{k}} \tag{5.6}$$

Then, the probability for the attacker to succeed at least one time at every rotation states cycle is:

$$Pr(X_{v'} > 0) = 1 - Pr(X_{v'} = 0) = 1 - (1-p)^T = 1 - (\frac{\binom{n-v'}{k}}{\binom{n}{k}})^T \tag{5.7}$$

In the next section, we discuss the efficacy of SCIT in the multi-controller SDN using these urn models.

5.6 Results and Discussion

In the section 5.5, we have presented the analysis of static multi-controller SDN and SCIT in multi-controller SDN. The results of this analysis depend on various parameters as follows:

- The percentage of vulnerable domain controllers in controllers cluster.
- The number of probes to find a vulnerable online controller.

- The exposure time to replace an online controller.

Then, we discuss the performance of SCIT in multi-controller SDN and its impact.

5.6.1 Static Multi-Controller SDN

To show the impact of vulnerable controllers' number in a controllers' cluster, and the number of probes allowed for the attacker, assume the use of a static multi-controller with the percentage of vulnerable controllers varying between 0 percent and 100 percent.

Figure 27 shows the attacker success probability as the percentage of vulnerable controllers and the number of domain controllers probed increase.

As seen in the graphs, in the case the attacker makes only 10 probes, he could detect a little less than 50 percent of the vulnerable controllers. Furthermore, the attacker's best chance to find a vulnerable domain controller is when the number of vulnerable domain controllers increases and the allowed probes in the cluster are augmented.

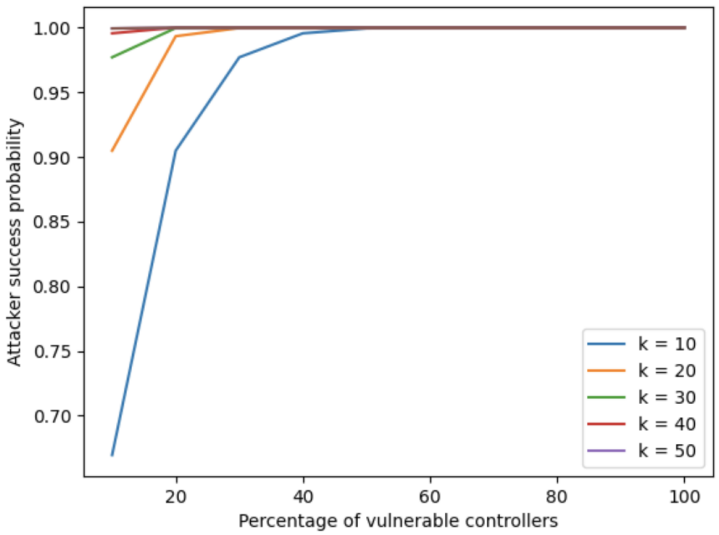

Figure 27: The attacker success probability for finding the vulnerable domain controller in static multi-controller SDN

5.6.2 Static Vs SCIT

In this part, we analyze the efficiency of SCIT in multi-controller SDN compared with the static multi-controller SDN.

As discussed in subsection 5.6.1, the success of the attack depends on the augmentation of vulnerable controllers and the number of probes. Assume there is 25 percent of the online domain controllers in the SCIT multi-controller SDN, due to SCIT nature, back to figure 26, the least number of controllers to use is 4 which should be in different states, so the SCIT architecture could work normally. Thus, consider the least case in which online domain controllers represent 25 percent of the controllers' cluster.

The vulnerable online domain controllers vary from 0 percent to 100 percent. Assume also T takes values: 1, 2, 5, 10.

Figure 28 illustrates the graphs of a static multi-controller and SCIT multi-controller SDN with different values of T, where the attacker has k=10 which represents the number of allowed probes.

As noticed, the attacker success probabilities in the graphs of the SCIT multi-controller with T=1 and T=2 are lesser than the graph of the static multi-controller. For example, when there is 20 percent of vulnerable online controllers and the attacker has k=10 probes allowed, the attacker success probability in a static multi-controller is equal to 0.9 while in SCIT multi-controller it's equal to 0.4 when T=1 and 0.65 when T=2. And it's closer to 1 when the percentage of vulnerable controllers gets bigger.

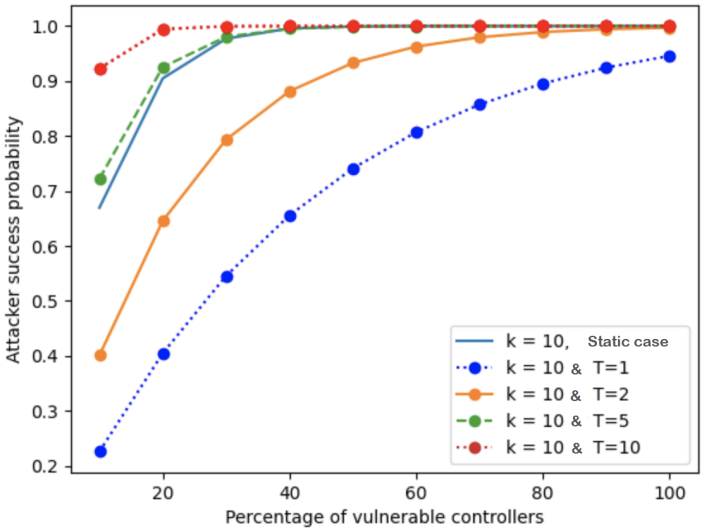

Figure 28: The attacker success probability for finding the vulnerable domain controller with k=10

Figure 29 illustrates the graphs of a static multi-controller and SCIT multi-controller SDN with different values of T, where the attacker has k=20 which represents the number of allowed probes.

As noticed, the attacker success probabilities in the graphs of the SCIT multi-controller with T=1 and T=2 are lesser than the graph of the static multi-controller. For example, when there is 20 percent of vulnerable online controllers and the attacker has k=20 probes allowed, the attacker success probability in a static multi-controller is much bigger than when T=1 and T=2. And it's closer to 1 when the percentage of vulnerable controllers gets bigger.

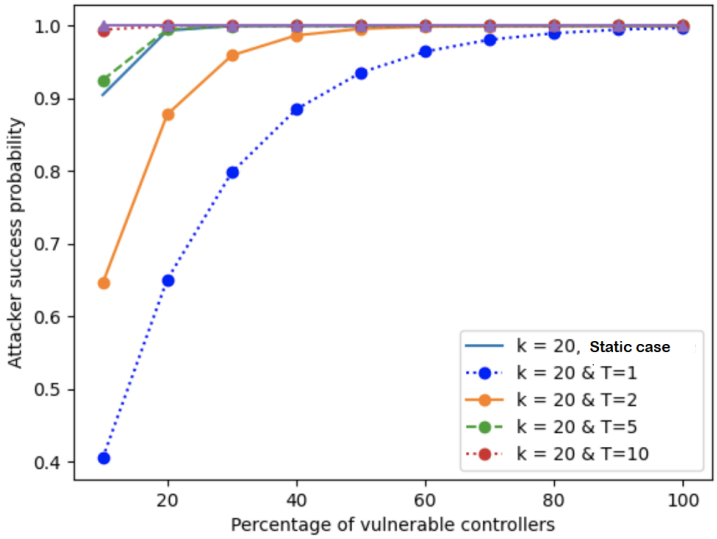

Figure 29: The attacker success probability for finding the vulnerable domain controller with k=10

Figure 30 illustrates the graphs of a static multi-controller and SCIT multi-controller SDN with different values of T, where the attacker has k=30 which represents the number of allowed probes.

As noticed, the attacker success probabilities in the graphs of the SCIT multi-controller with T=1 and T=2 are lesser than the graph of the static multi-controller. For example, when there is 20 percent of vulnerable online controllers and the attacker has k=30 probes allowed, the attacker success probability in a static multi-controller is much bigger than when T=1 and T=2. And it's closer to 1 when the percentage of vulnerable controllers gets bigger.

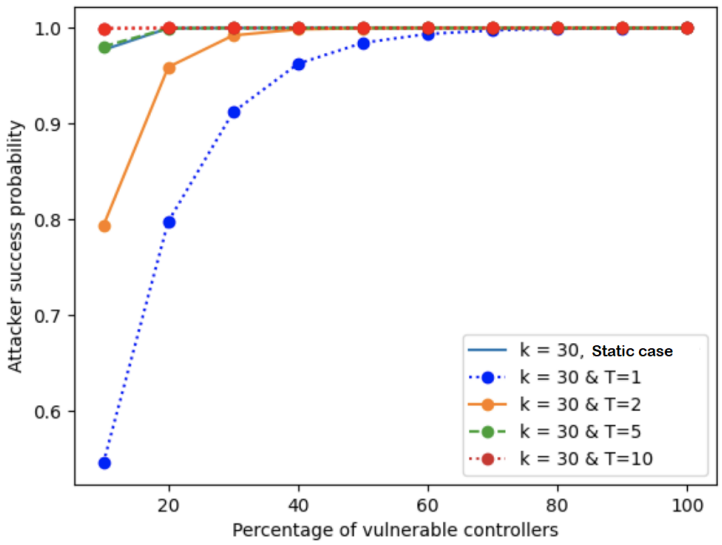

Figure 30: The attacker success probability for finding the vulnerable domain controller with k=10

Figure 31 illustrates the graphs of a static multi-controller and SCIT multi-controller SDN with different values of T, where the attacker has k=40 which represents the number of allowed probes.

As noticed, the attacker success probabilities in the graphs of the SCIT multi-controller with T=1 and T=2 are lesser than the graph of the static multi-controller. For example, when there is 20 percent of vulnerable online controllers and the attacker has k=40 probes allowed, the attacker success probability in a static multi-controller is much bigger than when T=1 and T=2. And it's closer to 1 when the percentage of vulnerable controllers gets bigger.

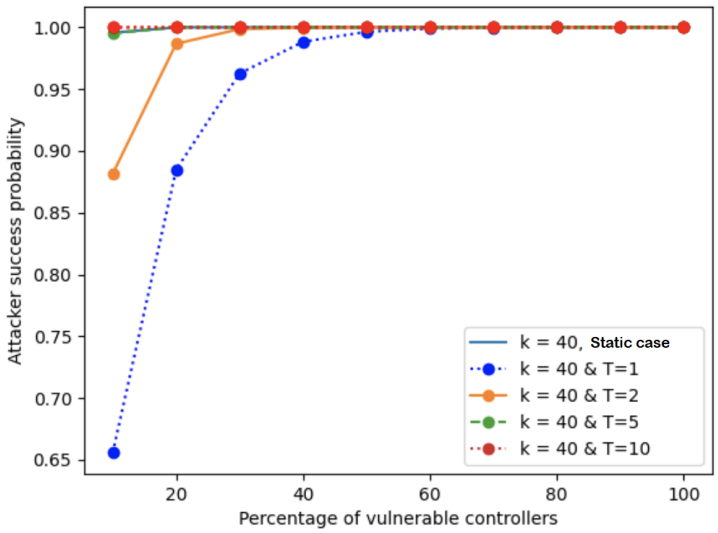

Figure 31: The attacker success probability for finding the vulnerable domain controller with k=10

Figure 32 illustrates the graphs of a static multi-controller and SCIT multi-controller SDN with different values of T, where the attacker has k=50 which represents the number of allowed probes.

As noticed, the attacker success probabilities in the graphs of the SCIT multi-controller with T=1 and T=2 are lesser than the graph of the static multi-controller. For example, when there is 20 percent of vulnerable online controllers and the attacker has k=50 probes allowed, the attacker success probability in a static multi-controller is much bigger than when T=1 and T=2. And it's closer to 1 when the percentage of vulnerable controllers gets bigger.

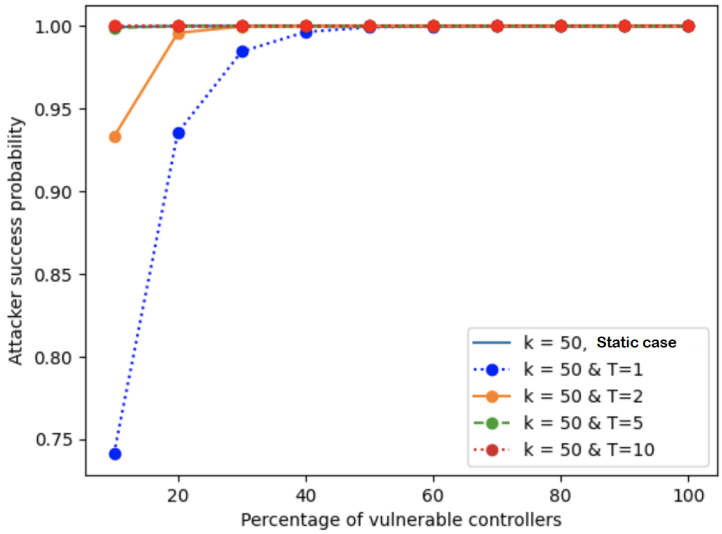

Figure 32: The attacker success probability for finding the vulnerable domain controller with k=10

However, we observe that when T increases, the attacker's success probability also increases. In fact, in a constant time t, if the number of rotations of SCIT increases, the exposure time diminishes. And this situation leads to decreasing the number of allowed probes by the attacker and diminishing the number of vulnerable online domain controllers. This issue was not taken into consideration in the above graphs and will be considered and analyzed in the following subsection.

5.6.3 Exposure Time

As previously discussed, the graphs in subsection 5.6.2 do not consider the advantage of reducing the exposure time when the number of SCIT rotations increases.

Exposure Time is the time in which a domain controller stays online. The longer the domain controller stays online, the more exposed to attacks it will be. In fact, the vulnerable and /or compromised controllers are contained if the exposure time is shorter.

In Figure 33, we consider the cases of ITC (SCIT SDN multi-controllers) operation when the T that represents the rotation number of states in the SCIT mechanism increases, then the exposure time decreases, consequently the number of probes allowed to the attacker k decreases. So we were interested in the following chosen cases as examples: k=50 and T=1, k=25 and T=2, k=10 and T=5, k=3 and T=10, in addition to the static case.

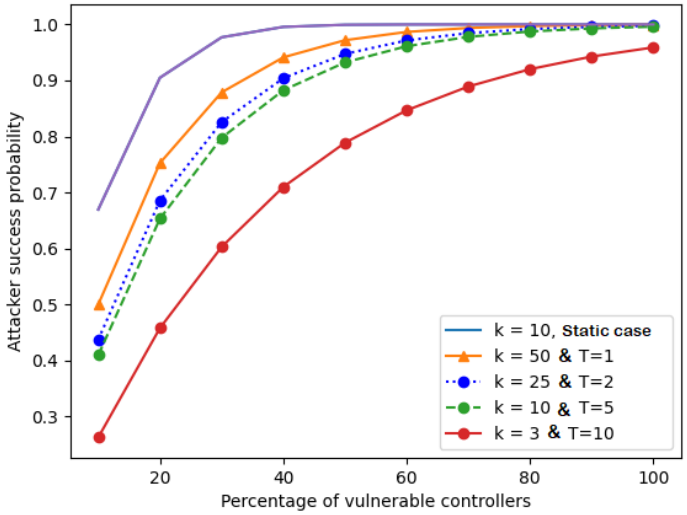

Figure 33: Impact of exposure time decrease on the attacker success probability for finding the vulnerable domain controller in SCIT multi-controller SDN

We can see clearly that the attacker's success probability in SCIT decreases considerably compared with the static case without the intrusion tolerance approach. Furthermore, the attacker's success probability decreases with the increase of the number of rotations in SCIT, i.e shortening of exposure time.

5.6.4 Comparison and limitations

Works in [31] and [85] already discussed in the related work section aim to place various levels of defense to increase SDN survivability, including intrusion detection that triggers recovery mechanisms. And this leads to additional latency in the service time. Furthermore, the authors in [31] apply an idle timeout of 5 seconds to the flows exceeding the threshold value to prevent overburdening of sending a great number of packets to the controller or causing unintentional unavailability due to dropping packets from an unknown source. Our proposed architecture based on SCIT is extremely different from the two works. In fact, it is a recovery-based approach that relies on the concept of accepting the existence of vulnerabilities in the control plane, and the fact that successful attacks are unavoidable. However, the process of changing a controller from state to state, and self-cleansing could mitigate attacks. Furthermore, using diversity in controllers' OS and applications renders it more difficult for the attacker to tackle with. In addition, MTD is used to ensure dynamic protection

against probing and scanning attacks.

The proposed solution could face a security problem with a very long exposure time and a performance problem with a very short exposure time, which needs more attention in future work: the loss of connections coming from the data plane would be more frequent when the frequency of the rotation states increases. Indeed, when an active controller changes his state to the grace period state, it becomes unable to receive new requests and then it only treats the existing packets in his line and refuses any other coming packets. These latter will be redirected to the new active controller. When changing from state to state becomes more frequent, many requests could be lost. Consequently, this situation could cause a high bandwidth load because of the generation of error messages or the resending of new requests by the Clients. As well, this requires more controllers to run which leads to consuming more computing resources. As there is a conflicting interest in choosing the exposure time, short for security and long for performance, the ITC parameters should be tuned correctly to avoid these problems.

5.7 Conclusion

The control plane is considered the most critical zone of SDN. In order to resolve this dilemma, so many researchers propose multi-controllers architectures rather than a single centralized controller. Unfortunately, simply using multi-controllers cannot avoid the threat of a single point of failure. In addition to the visible nature of the control plane makes it more vulnerable to attacks. In this chapter, we first use the SCIT approach based on a hierarchical multi-controller architecture to ensure intrusion tolerance. It is composed of a root controller managing other regular controllers with diverse operating systems and maintaining their recovery circle. The second mechanism is to move the regular controllers in a continuous way by the use of MTD techniques, to confuse the attacker. To do this, we adopt a hybrid alteration by combining time-based and event-based approaches.

After that, we utilize an Urn model to model our system that illustrates the theoretical measurement of the performance of our proposed architecture by taking into consideration several parameters including the percentage of vulnerable domain controllers, the number of probes allowed to the at-

tacker, and the frequency of our SCIT rotation that affects on the exposure time and the previous parameters.

In conclusion, the Intrusion Tolerant Controller would be a good solution to enhance security, especially the availability of the SDN controllers due to the integration of the intrusion tolerance approach.

6

Conclusions & Perspectives

The network security in general and the security of crucial infrastructures in particular is becoming a critical concern and the preoccupation of worldwide researchers in the field. To complicate matters further, the increasing spread of public connectivity of today's information systems emerges with new security challenges. Traditional security has accomplished a long way toward ensuring confidentiality, integrity, and availability. But these mechanisms are still ineffective against sophisticated attacks and internal threats. As mentioned earlier, ITSs are motivated by the recognition of the fact that intrusion will occur and some will be successful. The concern of intrusion tolerance is not how to defend or detect the intrusion, but how to mask or restrain the intrusion when the network has been intruded. This thesis uses the intrusion tolerance mechanisms to tackle the security issues in SDN control plane and employ security games to improve the decisions making in network security.

This Chapter provides the conclusions of this dissertation. We first summarize the main contributions and discuss how they meet the objectives established. Next, we identify and discuss a number of challenging open issues that should be tackled in future work. Finally, we list various scientific publications that have resulted from this Thesis.

Contributions

We summarize the contributions made in this work and discuss the main conclusions that arise from them.

First, a comprehensive study was conducted in Chapter 2 to understand the different approaches around ITSs and the mechanisms involved in their functionality. We first touched on the aim of

intrusion tolerant systems, their techniques, and the different approaches used to tackle the issues of successful attacks in different situations. We next overviewed ITSs in Cloud Computing, SDN, and WSNs. This helped us come up with an analysis and a comparison of the mechanisms and taxonomies given in several environments.
Second, a comprehensive study was detailed in Chapter 3 to have an idea about security games. We introduce the security games model, its basic definitions, and its forms. We extended the study by making a review of security games providing a quantitative framework for modeling the interaction between attackers and defenders in several situations and environments.

Third, a game theoretic model based on intrusion tolerant system is detailed in Chapter 4. This was done by implementing a game theoretic approach to model the attack-defense interaction, then analyzing the effect of an intrusion tolerant system on the payoffs of both players. The defender and the attacker provide a quantitative approach to perform a cost analysis of the security investment. This research takes into account the actions of the two players. The game has multiple Nash equilibrium that can be converted into pure strategy or mixed strategy under specific conditions. We have calculated the probability of attacking strategy and investing in tolerance strategy that both increase with the increase in the probability of compromising the system, which means that the defender should invest in tolerance with the increase of attacks.

Finally, Chapter 5 focused on improving the intrusion tolerance in SDN control plane. we first use the SCIT approach based on a hierarchical multi-controller architecture to ensure intrusion tolerance. It is composed of a root controller managing other regular controllers with diverse operating systems and maintaining their recovery circle. The second mechanism is to move the regular controllers in a continuous way by the use of MTD techniques, to confuse the attacker. To do this, we adopt a hybrid alteration by combining time-based and event-based approaches. After that, we utilize an Urn model to model our system that illustrates the theoretical measurement of the performance of our proposed architecture by taking into consideration several parameters including the percentage of vulnerable domain controllers, the number of probes allowed to the attacker, and the frequency of our SCIT rotation that affects on the exposure time and the previous parameters.

Open Issues and Future Work

The works in this thesis have taken a step towards modeling and making decisions to secure networks in general, besides enhancing and evaluating intrusion tolerance in SDN. However, during the preparation and writing of this thesis, we realized that many of the presented works open up new research questions and perspectives that require more attention. Some of these research tracks are:

- Enhancement of intrusion tolerance in other environments like IoT, Fog Computing... and other SDN planes.
- Using Dynamic security games in modeling intrusion tolerance in SDN to give more realistic scenarios.
- Using machine learning and game theory to find the best exposure time that could ensure intrusion tolerance, either the best performance of the system.
- Developing other approaches to intrusion tolerant systems that show more efficiency and less complexity in implementing.

Publications

We list all publications that arise from this Thesis organized by chronological order:

- Nouhad Sanoussi, Ghizlane Orhanou, and Said El Hajji. 2020. A game theoretic approach based on intrusion tolerant systems. Int. J. Secur. Netw. 15, 3 (2020), 175–181. https://doi.org/10.1504/ijsn.2020.109698
- Nouhad Sanoussi, Kaouthar Chetioui, Ghizlane Orhanou, Said El Hajji, ITC: Intrusion tolerant controller for multicontroller SDN architecture, Computers and Security, Volume 132, 2023, https://doi.org/10.1016/j.cose.2023.103351.
- Nouhad Sanoussi, Ghizlane Orhanou, Survey on Intrusion Tolerant Systems in Emerging Technologies, ITM Web Conf., 60 (2024) 00011. DOI: https://doi.org/10.1051/itmconf/20246000011

Bibliography

[1] Yuchong Li, Qinghui Liu, A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments, Energy Reports, Volume 7, 2021, Pages 8176-8186.

[2] hackmageddon web site, https://www.hackmageddon.com/category/security/cyber-attacks-statistics/ [accessed on 24th january 2024]

[3] Tansu Alpcan and Tamer Baar. 2010. Network Security: A Decision and Game-Theoretic Approach (1st. ed.). Cambridge University Press, USA.

[4] Ronald L. Krutz, Russell Dean Vines. THE CEH PREP GUIDE, THE COMPREHENSIVE GUIDE TO CERTIFIED ETHICAL HACKING. Wiley India Pvt. Limited, 2007.

[5] https://www.eccouncil.org/cybersecurity-exchange/threat-intelligence/cyber-kill-chain-seven-steps-cyberattack/

[6] Scarfone, K. and Hoffman, P. (2009), Guidelines on Firewalls and Firewall Policy, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD,

[7] Scarfone, K. and Mell, P. (2007), Guide to Intrusion Detection and Prevention Systems (IDPS), Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD,

[8] Polk, W. and Bassham, L. (1992), A Guide to the Selection of Anti-Virus Tools and Techniques, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/nist.sp.800-5

[9] Gong, Yili and Huang, Wei and Wang, Wenjie and Lei, Yingchun: A survey on software defined networking and its applications. Frontiers of Computer Science, N. 6, Vol. 9, pp. 827–845. (2015).

[10] Shaghaghi, A., Kaafar, M.A., Buyya, R., Jha, S. (2020). Software-Defined Network (SDN) Data Plane Security: Issues, Solutions, and Future Directions. In: Gupta, B., Perez, G., Agrawal, D., Gupta, D. (eds) Handbook of Computer Networks and Cyber Security. Springer, Cham.

[11] Rahim Masoudi, Ali Ghaffari, Software defined networks: A survey, Journal of Network and Computer Applications, Volume 67, 2016, Pages 1-25,

[12] A. Abdou, P. C. van Oorschot and T. Wan, "Comparative Analysis of Control Plane Security of SDN and Conventional Networks," in IEEE Communications Surveys and Tutorials, vol. 20, no. 4, pp. 3542-3559, Fourthquarter 2018.

[13] Shu, Z., Wan, J., Li, D. et al. Security in Software-Defined Networking: Threats and Countermeasures. Mobile Netw Appl 21, 764–776 (2016).

[14] S. Scott-Hayward, G. O'Callaghan and S. Sezer, "Sdn Security: A Survey," 2013 IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, 2013, pp. 1-7, doi: 10.1109/SDN4FNS.2013.6702553.

[15] Nouhad Sanoussi, Ghizlane Orhanou, and Said El Hajji. 2020. A game theoretic approach based on intrusion tolerant systems. Int. J. Secur. Netw. 15, 3 (2020), 175–181. https://doi.org/10.1504/ijsn.2020.109698

[16] Nouhad Sanoussi, Kaouthar Chetioui, Ghizlane Orhanou, Said El Hajji, ITC: Intrusion tolerant controller for multicontroller SDN architecture, Computers and Security, Volume 132, 2023.

[17] Nouhad Sanoussi, Ghizlane Orhanou, Survey on Intrusion Tolerant Systems in Emerging Technologies, ITM Web Conf., 60 (2024) 00011. DOI: https://doi.org/10.1051/itmconf/20246000011

[18] Nagarajan, Ajay. "Realizing Cyber Resilience with Hybrid Intrusion Tolerance Architectures.",George Mason University, 2017

[19] La V.H., Cavalli A. A study of Intrusion-tolerant routing in Wireless Sensor Networks. Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS). 2014;26(6):99-110. (In Russ.)

[20] Wang, F. and Raghavendra Uppalli and Killian, C. "Analysis of techniques for building intrusion tolerant server systems" IEEE Military Communications Conference, 2003. MILCOM 2003, vol. 2, pp. 729–734, 2003.

[21] Koren, I., Krishna, C.M." Fault-tolerant systems." Elsevier/Morgan Kaufmann (2007).

[22] R. Ma, L. Xing and H. E. Michel, "Fault-Intrusion Tolerant Techniques in Wireless Sensor Networks," 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, 2006, pp. 85-94.

[23] Y. W. Law, J. Doumen and P. Hartel, "Survey and Benchmark of Block Ciphers for Wireless Sensor Networks", TR-CTIT-04-07, Jan. 2004.

[24] Q. Nguyen and A. Sood, "A Comparison of Intrusion-Tolerant System Architectures," in IEEE Security and Privacy, vol. 9, no. 4, pp. 24-31, July-Aug. 2011.

[25] Dazhi Wang, Bharat B. Madan, and Kishor S. Trivedi. 2003. Security analysis of SITAR intrusion tolerance system. In Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security (SSRS '03). Association for Computing Machinery, New York, NY, USA, 23–32.

[26] Paulo E. Veríssimo, Nuno F. Neves, Christian Cachin, Jonathan Poritz, David Powell and Yves Deswarte, Robert Stroud, and Ian Welch. "Intrusion-Tolerant Middleware: The Road to Automatic Security", IEEE Security and Privacy, 2006.

[27] Yih Huang, David Arsenault, and Arun Sood. "Secure, Resilient Computing Clusters: Self-Cleansing Intrusion Tolerance with Hardware Enforced Security (SCIT/HES)", The Second International Conference on Availability, Reliability, and Security, ARES 2007.

[28] J. Lim, Y. Kim, D. Koo, S. Lee, S. Doo, and H. Yoon, "A novel Adaptive Cluster Transformation (ACT)-based intrusion tolerant architecture for hybrid information technology," J. Supercomput., vol. 66, no. 2, pp. 918–935, Nov. 2013.

[29] B. Jang, S. Doo, S. Lee, and H. Yoon, "Hybrid Recovery-Based Intrusion Tolerant System for Practical Cyber-Defense," IEICE Trans. Inf. Syst., vol. E99.D, no. 4, pp. 1081–1091, 2016.

[30] V. M. Karande and A. R. Pais, "A Framework for Intrusion Tolerance in Cloud Computing," in Advances in Computing and Communications, vol. 193, A. Abraham, J. L. Mauri, J. F. Buford,

J. Suzuki, and S. M. Thampi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 386–395.

[31] Manu, B. and Koundinya, Anjan K.: Intrusion Tolerant Architecture for SDN Networks Through Flow Monitoring. 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), pp. 1–5. (2017).

[32] Gonzalez, Andres J. and Nencioni, Gianfranco. and Helvik, Bjarne E. and Kamisinski, Andrzej: A Fault-Tolerant and Consistent SDN Controller. 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. (2016).

[33] Chaithra.S, Gowrishankar S, "Study of Secure Fault Tolerant Routing Protocol for IoT", International Journal of Science and Research (IJSR), Volume 5 Issue 7, July 2016, pp. 1833-1838.

[34] Jing Deng, Richard Han, and Shivakant Mishra. 2006. INSENS: Intrusion-tolerant routing for wireless sensor networks. Comput. Commun. 29, 2 (January, 2006), 216–230.

[35] Jiliang Zhou, Caixia Li, Qiying Cao and Yu Shen, "An intrusion-tolerant secure routing protocol with key exchange for wireless sensor network," 2008 International Conference on Information and Automation, Changsha, 2008, pp. 1547-1552.

[36] Quanyan Zhu and Stefan Rass. 2018. Game Theory Meets Network Security: A Tutorial. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). Association for Computing Machinery, New York, NY, USA, 2163–2165. https://doi.org/10.1145/3243734.3264421

[37] Xu, H. (2016). The Mysteries of Security Games: Equilibrium Computation Becomes Combinatorial Algorithm Design. Proceedings of the 2016 ACM Conference on Economics and Computation.

[38] Patil, A. P., Bharath, S., Annigeri, N. M. (2018). Applications of game theory for cyber security system: A survey. International Journal of Applied Engineering Research, 13(17), 12987-12990.

[39] Jackson, Matthew O., A Brief Introduction to the Basics of Game Theory, SSRN Electronic Journal, (December 5, 2011), https://dx.doi.org/10.2139/ssrn.1968579.

[40] Y. Wang, Y. Wang, J. Liu, Z. Huang and P. Xie, "A Survey of Game Theoretic Methods for Cyber Security," 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), Changsha, China, 2016, pp. 631-636, doi: 10.1109/DSC.2016.90.

[41] T. Alpcan and T. Baser, "An intrusion detection game with limited observations," Proc. 12th Int. Symp. on Dynamic Games and Applications, 2006.

[42] X. Liang and Y. Xiao, "Game Theory for Network Security," in IEEE Communications Surveys and Tutorials, vol. 15, no. 1, pp. 472-486, First Quarter 2013, doi: 10.1109/SURV.2012.062612.00056.

[43] P. Chakraborty, K. Majumder and A. Dasgupta, "A game theoretic model to detect cooperative intrusion over multiple packets," Proc. ICAIECES, Springer India: 895-907, 2016.

[44] X. Punithan, J. Kim, D. Kim, and Y. Choi, "A game theoretic model for dynamic configuration of large-scale intrusion detection signatures," Multimedia Tools and Applications: 1-17, 2015.

[45] S. Paul et al., "Extended game theoretic Dirichlet based collaborative intrusion detection systems," Proc. Computational Intelligence, Cyber Security and Computational Models, Springer, 2016.

[46] A. Jain, K. Tripathi, A. Jatain and M. Chaudhary, "A Game Theory based Attacker Defender Model for IDS in Cloud Security," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 190-194.

[47] L. Kwiat, C. A. Kamhoua, K. A. Kwiat, J. Tang and A. Martin, "Security-Aware Virtual Machine Allocation in the Cloud: A Game Theoretic Approach," 2015 IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 2015, pp. 556-563, doi: 10.1109/CLOUD.2015.80.

[48] E. Furuncu and I. Sogukpinar (2015). "Scalable risk assessment method for cloud computing using game theory," Computer Standards and Interfaces, 38: 44-50, 2015.

[49] Komal Singh Gill, Sharad Saxena, Anju Sharma, GTM-CSec: Game theoretic model for cloud security based on IDS and honeypot, Computers and Security, Volume 92, 2020,

[50] Mao, D., Zhang, S., Zhang, L., Feng, Y. (2019). Game Theory Based Dynamic Defense Mechanism for SDN. In: Chen, X., Huang, X., Zhang, J. (eds) Machine Learning for Cyber Security. ML4CS 2019. Lecture Notes in Computer Science, vol 11806. Springer, Cham.

[51] M. Abderrahim, A. Ben Letaifa, A. Haji and S. Tabbane, "A Game Theory-Based Effective Network Management in SDN Networks," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, Poland, 2018, pp. 390-395, doi: 10.1109/WAINA.2018.00117.

[52] Sun Yan, Ji Weifeng, Weng Jiang, Zhao Beiying. (2020). Overview on MTD technology based on game theory. MATEC Web of Conferences. 309. 02012. 10.1051/matecconf/202030902012.

[53] El Mir, I., Haqiq, A., Kim, D.S. (2018). A Game Theoretic Approach for Cloud Computing Security Assessment Using Moving Target Defense Mechanisms. In: Ben Ahmed, M., Boudhir, A. (eds) Innovations in Smart Cities and Applications. SCAMS 2017. Lecture Notes in Networks and Systems, vol 37. Springer, Cham.

[54] Sailik Sengupta, Satya Gautam Vadlamudi, Subbarao Kambhampati, Adam Doupé, Ziming Zhao, Marthony Taguinod, and Gail-Joon Ahn. 2017. A Game Theoretic Approach to Strategy Generation for Moving Target Defense in Web Applications. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 178–186.

[55] Duohe Ma, Zhimin Tang, Xiaoyan Sun, Lu Guo, Liming Wang, and Kai Chen. 2022. Game Theory Approaches for Evaluating the Deception-based Moving Target Defense. In Proceedings of the 9th ACM Workshop on Moving Target Defense (MTD'22). Association for Computing Machinery, New York, NY, USA, 67–77.

[56] Seokcheol Lee, Sungjin Kim, Ken Choi, Taeshik Shon, Game theory-based Security Vulnerability Quantification for Social Internet of Things, Future Generation Computer Systems, Volume 82, 2018, Pages 752-760,

[57] Kiran, V., Rani, S. and Singh, P. Towards a Light Weight Routing Security in IoT Using Non-cooperative Game Models and Dempster–Shaffer Theory. Wireless Pers Commun 110, 1729–1749 (2020). https://doi.org/10.1007/s11277-019-06809-w

[58] Newgenapps: https://www.newgenapps.com/blog/internal-and-external-security-threats

[59] Wang, F., Uppalli, R., Killian, C.: Analysis of techniques for building intrusion tol- erant server systems. In: IEEE Military Communications Conference, 2003. MILCOM 2003. vol. 2, pp. 729–734 Vol.2 (2003)

[60] Stavridou, V., Dutertre, B., Riemenschneider, R.A., Saidi, H.: Intrusion tolerant software architectures. In: DARPA Information Survivability Conference and Exposition II, 2001. DISCEX'01. Proceedings. vol. 2, pp. 230–241. IEEE (2001)

[61] Ouffoue, G., Ortiz, A.M., Cavalli, A.R., Mallouli, W., Domingo-Ferrer, J., Sanchez, D., Zaidi, F.: Intrusion Detection and Attack Tolerance for Cloud Environments: The CLARUS Approach. pp. 61–66. IEEE (2016)

[62] Koren, I., Krishna, C.M.: Fault tolerant systems. Elsevier/Morgan Kaufmann (2007)

[63] A. Attiah, M. Chatterjee and C. C. Zou, "A Game Theoretic Approach to Model Cyber Attack and Defense Strategies," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1-7.

[64] Wang, D., Madan, B.B., Trivedi, K.S.: Security analysis of SITAR intrusion tolerance system. In: Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security. pp. 23–32. ACM (2003)

[65] Guo, M., Bhattacharya, P.: Diverse virtual replicas for improving intrusion tolerance in cloud. pp. 41–44 (2014)

[66] Elmir, I., Mehdi, K.E., Mohamed, H., Abdelkrim, H., Kim, D.S.: A Game Theoretic approach based virtual machine migration for cloud environment security. International Journal of Communication Networks and Information Security (IJCNIS) 9(3) (2017)

[67] Njilla, L.Y., Pissinou, N., Makki, K.: Game theoretic modeling of security and trust relationship in cyberspace: Game Theoretic Modeling Security Trust Cyberspace. International Journal of Communication Systems 29(9), 1500–1512 (2016)

[68] Kamhoua, C.A., Kwiat, L., Kwiat, K.A., Park, J.S., Zhao, M., Rodriguez, M.: Game Theoretic Modeling of Security and Interdependency in a Public Cloud. 2014 IEEE 7th International Conference on Cloud Computing pp. 514–521 (2014)

[69] Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q.: A Survey of Game Theory as Applied to Network Security. In: 2010 43rd Hawaii International Conference on System Sciences. pp. 1–10 (2010)

[70] Mir, I.E., Kim, D.S., Haqiq, A.: Security modeling and analysis of a self-cleansing intrusion tolerance technique. In: 11th International Conference on Information Assurance and Security (IAS). pp. 111–117 (2015)

[71] Huang, Y., Ghosh, A.K.: Introducing Diversity and Uncertainty to Create Moving Attack Surfaces for Web Services. In: Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats, pp. 131–151. Springer New York (2011)

[72] Lei, Cheng and Zhang, Hong-Qi and Tan, Jing-Lei and Zhang, Yu-Chen and Liu, Xiao-Hu: Moving Target Defense Techniques: A Survey. Security and Communication Networks. Hindawi. (2018)

[73] Benson, Theophilus and Akella, Aditya and Maltz, David: Unraveling the Complexity of Network Management. Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, ser. NSDI'09, Berkeley, CA, USA, pp. 335–348. (2009).

[74] Ahmad, Ijaz an Namal, Suneth and Ylianttila, Mika and Gurtov, Andrei: Security in Software Defined Networks: A Survey. IEEE Communications Surveys & Tutorials, N. 4, Vol. 17, pp. 2317–2346. (2015).

[75] Guang Yao, Jun Bi and Luyi Guo, "On the cascading failures of multi-controllers in Software Defined Networks," 2013 21st IEEE International Conference on Network Protocols (ICNP), 2013, pp. 1-2.

[76] Fonseca, P. and Bennesby, R. and Mota, E. and Passito, A.: A replication component for resilient OpenFlow-based networking. 2012 IEEE Network Operations and Management Symposium, pp. 933–939. (2012).

[77] Sidki, Liran and Ben-Shimol, Yehuda and Sadovski, Akiva: Fault tolerant mechanisms for SDN controllers. 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 173–178. (2016).

[78] Gonzalez, Andres J. and Nencioni, Gianfranco. and Helvik, Bjarne E. and Kamisinski, Andrzej: A Fault-Tolerant and Consistent SDN Controller. 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. (2016).

[79] Hu, Y. and Wang, W. and Gong, X. and Que, X. and Cheng, S.: BalanceFlow: Controller load balancing for OpenFlow networks. 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Vol. 02, pp. 780–785. (2012).

[80] Selvi, Hakan and Gur, Gurkan and Alagoz, Fatih: Cooperative load balancing for hierarchical SDN controllers. 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), pp. 100–105. (2016).

[81] Sufiev, Hadar and Haddad, Yoram: A dynamic load balancing architecture for SDN. 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE), pp. 1–3. (2016).

[82] Yonghong, Fu and Jun, Bi and Jianping, Wu and Ze, Chen and Ke, Wang and Min, Luo: A dormant multi-controller model for software defined networking. China Communications, N. 3, Vol. 11, pp. 45–55. (2014).

[83] Radam, Neamah S. and Al-Janabi, Sufyan T. Faraj and Jasim, Khalid Sh.: Multi-Controllers Placement Optimization in SDN by the Hybrid HSA-PSO Algorithm. Computers, N. 7, Vol. 11. (2022).

[84] Ramya, G., Manoharan, R. Enhanced optimal placements of multi-controllers in SDN. J Ambient Intell Human Comput, Vol. 12, pp. 8187–8204. (2021).

[85] Athanasios Liatifis, Christos Dalamagkas, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Evangelos Markakis, Valeri Mladenov, and Panagiotis Sarigiannidis: Fault-Tolerant SDN Solution for Cybersecurity Applications. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22) (2022).

[86] Yih Huang and Arsenault, D. and Sood, A.: Closing cluster attack windows through server redundancy and rotations. Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), Vol. 2, pp. 12–21. (2006).

[87] Nguyen, Quyen L. and Sood, Arun: Designing SCIT architecture pattern in a Cloud-based environment. 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 123–128. (2011).

[88] Y. Lee, S. Lee, H. Seo, C. Yoon, S. Shin, and H. Yoon, "Duo: Software Defined Intrusion Tolerant System Using Dual Cluster," Secur. Commun. Netw., vol. 2018, pp. 1–13, 2018.

[89] Sharma, Dilli P. and Cho, Jin-Hee and Moore, Terrence J. and Nelson, Frederica F. and Lim, Hyuk and Kim, Dong Seong: Random Host and Service Multiplexing for Moving Target Defense in Software-Defined Networks. ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6. (2019).

[90] Narantuya, Jargalsaikhan and Yoon, Seunghyun and Lim, Hyuk and Cho, Jin-Hee and Kim, Dong Seong and Moore, Terrence and Nelson, Frederica: SDN-Based IP Shuffling Moving Target Defense with Multiple SDN Controllers. 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S), pp. 15–16. (2019).

[91] Gong, Yili and Huang, Wei and Wang, Wenjie and Lei, Yingchun: Multi-controller Based Software-Defined Networking: A Survey. IEEE Access, Vol. 6, pp. 15980–15996. (2018).

[92] Hossein, Amir and Watts, Michael J and Ahmadi, Kourosh: An Overview of Multi-Controller Architecture in Software-Defined Networking. Conference: CITRENZ 2019. (2019).

[93] Mir, Iman El and Kim, Dong Seong and Haqiq, Abdelkrim: Security modeling and analysis of a self-cleansing intrusion tolerance technique. 2015 11th International Conference on Information Assurance and Security (IAS), pp. 111–117. (2015).

[94] Sengupta, Sailik and Chowdhary, Ankur and Sabur, Abdulhakim and Alshamrani, Adel and Huang, Dijiang and Kambhampati, Subbarao: A Survey of Moving Target Defenses for Network Security. IEEE Communications Surveys and Tutorials, N. 3, Vol. 22, pp. 1909-1941. (2020).

[95] Cho, Jin-Hee and Sharma, Dilli P. and Alavizadeh, Hooman and Yoon, Seunghyun and Ben-Asher, Noam and Moore, Terrence J. and Kim, Dong Seong and Lim, Hyuk and Nelson, Frederica F.: Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. arXiv. (2019).

[96] Hong, Jin B. and Kim, Dong Seong: Assessing the Effectiveness of Moving Target Defenses Using Security Models. IEEE Transactions on Dependable and Secure Computing, N. 2, Vol. 13, pp. 163–177. (2016).

[97] Salman, Ola and Elhajj, Imad and Kayssi, Ayman and Chehab, Ali: SDN controllers: A comparative study. 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1-6. (2016).

[98] Carroll, Thomas E. and Crouse, Michael and Fulp, Errin W. and Berenhaut, Kenneth S.: Analysis of network address shuffling as a moving target defense. 2014 IEEE International Conference on Communications (ICC), pp. 701-706. (2014).

[99] Mahmoud, Hosam M: Pólya urn models and connections to random trees: a review. Journal of the Iranian Statistical Society (JIRSS). (2003).