

**AN ENHANCED MITIGATION TECHNIQUE FOR
ECONOMIC DENIAL OF SUSTAINABILITY (EDOS)
ATTACK**

BY

Mohammed Yahya Alkaff

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

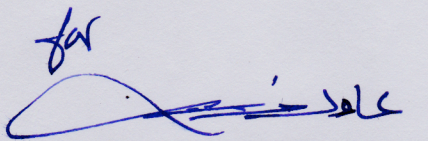
DECEMBER, 2013

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

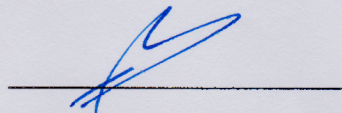
DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammed Yahya Alkaff** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.



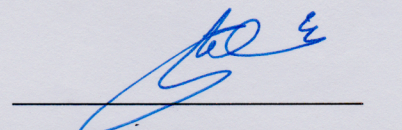
Dr. Basem Al Madani
Department Chairman

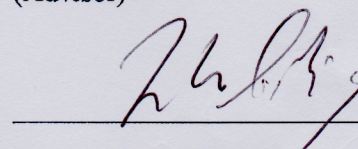


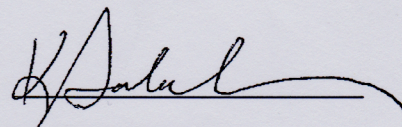
Dr. Salam A. Zummo
Dean of Graduate Studies

13/5/14
Date




Dr. Mohammed Houssaini Sqalli
(Advisor)


Dr. Zubair Ahmed Baig
(Member)


Dr. Khaled Salah
(Member)

© Mohammed Yahya Alkaff

2013

Dedication

I dedicate this thesis to my parents, my wife, my brothers, and sister.

ACKNOWLEDGMENTS

Always, I thank Allah for helping and guiding me to achieving all my success. Second, I would like to thank my advisor Dr. Mohammed Sqalli for his support and help, as well as all my thesis committee members for their guidance and time to improve this work.

I would also like to thank the King Fahd University of Petroleum and Minerals (KFUPM) for creating the research environment for the students and for supporting them until they accomplish their objectives.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	V
TABLE OF CONTENTS.....	VI
LIST OF TABLES.....	IX
LIST OF FIGURES.....	X
LIST OF ABBREVIATIONS.....	XI
ABSTRACT.....	XII
ملخص الرسالة.....	XIV
1 CHAPTER 1 INTRODUCTION.....	1
1.1 Background and Terminology	4
1.2 Economic Denial of Sustainability (EDoS)	5
2 CHAPTER 2 LITERATURE REVIEW.....	9
3 CHAPTER 3 EDOS ATTACK DEFENDER MITIGATION TECHNIQUE	22
3.1 Proposed Architecture and Algorithms	30
4 CHAPTER 4 SIMULATOR.....	38
4.1 Simulator's Design	38
4.1.1 Simulator's Assumptions	41
4.2 Simulation Measures	41
4.2.1 Response Time	41
4.2.2 Computing Resources Utilization.....	44

4.2.3	Throughput	45
4.2.4	Cost	45
4.3	Simulator's Validation	46
4.3.1	Response Time	47
4.3.2	Computing Resources Utilization	48
4.3.3	Cost	48
5	CHAPTER 5 SIMULATION RESULTS AND DISCUSSION	50
5.1	Normal Mode Results	50
5.1.1	Response Time Evaluation	51
5.1.2	Resources Utilization Evaluation	51
5.1.3	Cost Evaluation.....	52
5.1.4	Throughput Evaluation	53
5.2	Flash Crowd Mode Results	54
5.2.1	Response Time Evaluation	55
5.2.2	Resources Utilization Evaluation	56
5.2.3	Cost Evaluation.....	57
5.2.4	Throughput Evaluation	58
5.3	Attack Mode Results and EDoS Shield Comparison	59
5.3.1	Response Time Evaluation	61
5.3.2	Resources Utilization Evaluation	62
5.3.3	Cost Evaluation.....	63
5.3.4	Throughput Evaluation	64
5.4	Detailed Results	65
5.4.1	Resources Utilization Evaluation	66
5.4.2	Response Time Evaluation Including Load Balancer Delay	68

5.4.3	Response Time Evaluation in Flash Crowd	69
6	CHAPTER 6 CONCLUSION AND FUTURE WORK.....	71
6.1	Conclusion	71
6.2	Future Work.....	72
	REFERENCES.....	74
	VITAE.....	79

LIST OF TABLES

Table 1 Response time Equations for different queuing model [55]	42
--	----

LIST OF FIGURES

Figure 1 Auto Scaling on the cloud Services.....	6
Figure 2 EDoS Attack on the cloud Services.....	7
Figure 3 EDoS Attack Defender Components.....	24
Figure 4 EDoS Attack Defender Modes.....	25
Figure 5 EDoS Attack Defender, Suspicion Mode Thresholds.....	26
Figure 6 Normal Mode.....	27
Figure 7 Suspicious and Attack Mode, Attacker Case.....	29
Figure 8 Suspicious and Attack Mode, Legitimate User Case.....	29
Figure 9 Cloud Service Component Flowchart.....	31
Figure 10 EDoS Defender Component Flowchart.....	32
Figure 11 the EDoS-Shield Architecture [34].....	36
Figure 12 Simulation Model Flowchart.....	39
Figure 13 Queuing model for EDoS attack against a cloud service [34].....	41
Figure 14 Traffic flow paths from sources to a destination.....	43
Figure 15 simulation results for EDoS Shield and our simulations, Response Time.....	47
Figure 16 simulation results for EDoS Shield and our simulations, Utilization.....	48
Figure 17 simulation results for EDoS Shield and our simulations, Cost.....	49
Figure 18 Response Time evaluation Normal Mode.....	51
Figure 19 Resources Utilization evaluation Normal Mode.....	52
Figure 20 Cost evaluation Normal Mode.....	53
Figure 21 Throughput evaluation Normal Mode.....	54
Figure 22 Response Time evaluation Flash Crowd Mode.....	56
Figure 23 Resources Utilization evaluation Flash Crowd Mode.....	57
Figure 24 Cost evaluation Flash Crowd Mode.....	58
Figure 25 Throughput evaluation Flash Crowd Mode.....	59
Figure 26 Response Time evaluation Attack Mode.....	62
Figure 27 Resources Utilization evaluation Attack Mode.....	63
Figure 28 Cost evaluation Attack Mode.....	64
Figure 29 Throughput evaluation Attack Mode.....	65
Figure 30 Resources Utilization evaluation Attack Mode.....	67
Figure 31 Response Time with Load Balancer Delay evaluation Attack Mode.....	69
Figure 32 Response Time using five instances for scaling up Flash Crowd Mode.....	70

LIST OF ABBREVIATIONS

EDoS	:	Economic Denial of Sustainability
DDoS	:	Distributed Denial of Service
LRC	:	Legitimate request counter
TRC	:	Total Request Counter
BYOD	:	Bring Your Own Device
GTT	:	Graphic Turning Test
CAPTCHA	:	Completely Automated Public Turing test to tell Computers and Humans Apart

ABSTRACT

Full Name : Mohammed Yahya Alkaff
Thesis Title : AN ENHANCED MITIGATION TECHNIQUE FOR ECONOMIC DENIAL OF SUSTAINABILITY (EDOS) ATTACK
Major Field : COMPUTER NETWORKS
Date of Degree : December 2013

Cloud computing is a promising technology for the future of IT industry. Many organizations and companies are moving towards this technology. Cloud computing is a suitable solution for organizations and companies looking for saving money in IT and improving performance and availability. Cloud computing security is a big challenge for the provider of the cloud services and it is a big concern for the customers of these cloud services. Cloud computing has attractive features such as elasticity, auto scaling, and utility computing. These features could help the adopters maximize resource utilization and minimize their operating costs. However, if the attacker takes advantage of these features and launch a Distribute Denial of Service (DDoS) attack on the cloud computing environment, this attack could change to a new attack, namely Economic Denial of Sustainability (EDoS) attack. The DDoS attack will trigger the elasticity and auto-scaling features on the cloud so the resources will grow according to the demand of the attack and due to the “pay as you go” model of the cloud, the adopters will be charged for the scaling of the resource until it reaches a point that it cannot sustain economically. The aim of this work is to study several existing mitigation techniques that prevent or mitigate the EDoS attack and state major drawbacks of the existing mitigation techniques. Then, a

new approach is proposed to mitigate the EDoS. This new mitigation technique takes into account most of the drawbacks for the existing mitigation techniques. The effectiveness of the proposed mitigation technique is evaluated using simulation. In addition, we conduct a comparison between our new approach and the EDoS-Shield technique.

The proposed technique is based on reactive mitigation schemes and it has three phases. In the first phase, we are monitoring the auto-scaling feature and suspicion mode thresholds to detect if there is an EDoS attack. In the second phase, once an attack is detected, the cloud service will trigger the checking component by forwarding all requests to this component. This component is responsible for differentiating between legitimate users and automated attackers (Zombies). This component will differentiate the traffic by sending Graphic Turing Tests such as CAPTCHA to the request generator. In the third phase, the checking component will drop all traffic that cannot respond to the CAPTCHA. However, the checking component will forward all requests that pass the CAPTCHA validation to the cloud service.

ملخص الرسالة

الاسم الكامل: محمد يحيى عبد الله عبد الرحمن الكاف

عنوان الرسالة: تقنية محسنة لتخفيف هجمات الحرمان الاقتصادي للاستدامة في الحوسبة السحابية

التخصص: هندسة شبكات الكمبيوتر

تاريخ الدرجة العلمية: 1 ديسمبر 2013

الحوسبة السحابية هي تكنولوجيا واحدة لمستقبل صناعة تكنولوجيا المعلومات. العديد من المنظمات والشركات تتجه نحو هذه التكنولوجيا. الحوسبة السحابية هي الحل المناسب للمؤسسات والجامعات الأكاديمية التي تبحث لتوفير المال في مجال تكنولوجيا المعلومات وتحسين الأداء والتوافر والخدمات. أمن الحوسبة السحابية يشكل تحديا كبيرا لمقدم الخدمات السحابية فذلك هو مصدر قلق كبير للعملاء هذه الخدمات السحابية. الحوسبة السحابية لديها ملامح جذابة مثل المرونة، التوسع التلقائي، والحوسبة الخدمية. هذه الميزات يمكن أن تساعد المتبنين تعظيم استخدام الموارد وتقليل تكاليف التشغيل الخاصة بهم. ومع ذلك، إذا كان المهاجم يستفيد من هذه الميزة وإطلاق هجوم الحرمان من الخدمة الموزعة على بيئة الحوسبة السحابية، يمكن تغيير هذا الهجوم إلى هجوم جديد، وهو هجوم الحرمان الاقتصادي للاستدامة. هجوم الحرمان من الخدمة الموزعة يستهدف مرونة والتوسع التلقائي على السحابة وبالتالي فإن الموارد سوف تنمو وفقا لطلب من المهاجم وبوجود الية الدفع للاستخدام في نموذج السحابة فان المشغل سيضطر الى دفع كل التكاليف الناتجة عن الهجوم الى ان يصل الى درجة لا يقدر بعدها الاستمرار اقتصاديا بسبب الخسارة الكبيرة. الهدف من هذا العمل هو دراسة عدة تقنيات التخفيف القائمة التي تمنع أو تخفف من هجوم الحرمان الاقتصادي للاستدامة والاستفادة من كل السلبيات الموجودة في تقنيات التخفيف الموجودة حاليا. ثم، سوف نقدم تقنية جديدة لتخفيف هجوم الحرمان الاقتصادي للاستدامة. هذه التقنية الجديدة التخفيف ستأخذ في عين الاعتبار معظم عيوب تقنيات التخفيف القائمة. يتم تقييم فعالية تقنيات التخفيف المقترحة باستخدام المحاكاة. بالإضافة إلى ذلك، نحن نخطط ل إجراء مقارنة بين نهجنا الجديد وتقنية درع هجوم الحرمان الاقتصادي للاستدامة لتقييم أسلوبنا بشأن التقنيات الموجودة.

وتستند هذه التقنية المقترحة على وضع خطط لتخفيف رد الفعل ولها ثلاث مراحل. في المرحلة الأولى، ونحن نراقب ميزة التحجيم التلقائي للكشف عما إذا كان هناك هجوم الحرمان الاقتصادي للاستدامة. في المرحلة الثانية، بمجرد اكتشاف هجوم، فإن خدمة سحابة تحريك عنصر فحص من قبل إحالة جميع الطلبات إلى هذا العنصر. هذا العنصر هو المسؤول عن التفريق بين المستخدمين الشرعيين والمهاجمين الآلي (Zombies)، وهذا عنصر التفريق بين حركة المرور عن طريق إرسال الاختبارات الجرافيك اختبارات رسم تورينج مثل ارسال كابيتشا (CAPTCHA) لمرسل الطلب. في المرحلة الثالثة، سيقوم عنصر التحقق بإسقاط جميع حركة او طلبات المرور التي لم تستجب للكابيتشا (CAPTCHA). عنصر التحقق سيقوم بتوجيه كافة الطلبات التي استجابت بنجاح للكابيتشا إلى الخدمة السحابية.

CHAPTER 1

INTRODUCTION

Cloud computing technology is a result of urgent needs for low cost, high utilization, and efficient management of the available resources in the information technology industry. Cloud computing is a utility that provides services on demand. All services provided by the cloud are elastic and could be rented or released by the subscribers of these services via a web-based tool accessed via the Internet. These services are based on a model called “pay per use” model, which allows the clients or subscribers of the services to request resources on demand and pay only for their usage. The cloud computing services can be categorized based on the services delivered to the end users. These categories include the Infrastructure offered as a service (IaaS), Platform offered as a service (PaaS), and Software offered as a service (SaaS). There is another classification of cloud computing, which depends on the location and the services offered by the cloud. This classification is divided into public cloud, private cloud, hybrid cloud, and community cloud [1]. In addition, a private cloud consists of two types based on the location of the cloud and these are on premise private clouds and externally hosted private cloud (virtual private cloud) [1].

According to a recent survey conducted by the International Data Corporation (IDC) [2], security is ranked first as the greatest challenge of cloud computing, as about 87% of IT executives cited security as the top challenge preventing their adoption of the cloud services model. Security concerns have led organizations to hesitate to move critical resources to the cloud. Corporations and individuals are often concerned about how security and compliance integrity can be maintained in this new environment. Also, moving critical applications and sensitive data to public and shared cloud environments is of great concern for corporations since their data center's network boundary defense is not on hand. With security being one of the top concerns that hinders cloud computing [3]–[8] it has become a major field of study.

The aspect of cloud computing security is wide and general. Therefore, it is imperative to introduce two types of network security threats, Denial of Service (DoS) and Distributed Denial of Service (DDoS). The DoS and the DDoS attacks overwhelm a network infrastructure or service by employing a distributed number of malicious or infected machines to perform unwanted operations intended to cause damage to the IT infrastructure of an organization. For example, a botnet (defined as a collection of malicious machines participating in an attack) is activated to overwhelm a web server, using an asynchronous attack that makes the site unavailable to end users, due to an exhaustion of its computing or network resources.

Cloud computing allows us to scale our servers in magnitude and availability in order to provide service to a greater number of requests from end users. Moreover, adopters of the cloud service model are charged based on a pay-per-use basis of the cloud's server and network resources, aka utility computing. Such a service model may appear to overcome

the effects of a DDoS attack, i.e., resource bottlenecks are eliminated. However, these clouds merely transform a conventional DDoS attack on server and network resources to a new breed of attacks that target the cloud adopter's economic resource, originally labeled as Economic Denial of Sustainability attack (EDoS) [9], [10]. Therefore, unlike conventional DDoS attacks, EDoS targets the financial constraint of an organization, but not its physical network or server constraints. EDoS occurs when zombie machines (part of a botnet) send a large amount of undesired traffic towards the cloud, exploiting the cloud's elasticity, to chalk up an exorbitant amount of cost on a cloud adopter's bill, leading to large-scale service withdrawal or bankruptcy. In our work, we study several existing mitigation techniques that prevent or mitigate the EDoS attack and state all drawbacks of the existing mitigation techniques. Then, a new approach is proposed to mitigate the EDoS. This new mitigation technique takes into account most of the drawbacks of the existing mitigation techniques.

The proposed technique is based on reactive mitigation schemes. In the first phase, we are monitoring the auto-scaling feature to detect if there is an EDoS attack. The monitoring parameters are based on the auto scaling parameters because EDoS tackles the auto-scaling feature. In this phase, we monitor the traffic bandwidth and the average CPU utilization using an upper level threshold and a lower level threshold in order to avoid the fluctuation around one value.

In the second phase, once an attack is detected, the cloud service will trigger the checking component by forwarding all requests to this component. This component is responsible for differentiating between legitimate users and automated attackers (Zombies). This

component will differentiate the traffic by sending Graphic Turing Tests [11], such as CAPTCHA [12]–[15], to the request generator.

In the third phase, the checking component will drop all traffic that cannot respond to the CAPTCHA. However, the checking component will forward all requests that pass the CAPTCHA validation to the cloud service.

1.1 Background and Terminology

Cloud computing is a technology model which is making a revolution in the computing environment. According to the official NIST definition [16], "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Cloud computing is based on some existing technologies such as virtualization, utility computing, grid computing, and automatic computing. However, the virtualization technology is mainly form the foundation of cloud computing technology [17]. Virtualization is the responsible layer to pool the computing resources from the available hardware. It allocates and reallocates virtual resources based on demand [17].

There are four types of could computing based on the location and the services offered by the cloud; these types are private, public, hybrid, and community clouds. A public cloud is a cloud where the infrastructure layer is offered by a third party and shared between customers. In addition, the client has no control on the data, network, and security settings. A private cloud can be of two types, the first one is on premise private cloud, in

which the infrastructure layer is dedicated to a specific organization and not shared or common with other organizations. In addition, the organization has full control on data, network, and security settings. The second type of private cloud is an externally hosted private cloud (virtual private cloud), in which the infrastructure layer is used by one organization but offered by a third party. This totally hosted private cloud outside the organization is less costly than the on premise private cloud.

A hybrid cloud is defined as a mixture of private and public clouds. The customer could host his critical applications on his private cloud and host non-critical applications on the public cloud. Also, this type of cloud computing is used for the infrastructure layer. For example, private computing infrastructure could be used for daily and normal activity. However, if there is a high activity in the network such as flash-crowd effect, then the computing infrastructure could be expanded by renting more resources from the public cloud infrastructure. Then, these resources may be released when not needed or the activity returns to normal. A community cloud is used to share the infrastructure layer between organizations or companies of the same community. For example, Ministries in Saudi Arabia could share the infrastructure layer on the cloud to get access to data related to citizens of Saudi Arabia.

1.2 Economic Denial of Sustainability (EDoS)

The Economic Denial of Sustainability (EDoS) attack is a major threat in the cloud-computing environment. This attack is not only causing the service to be unavailable or down like a Denial of service (DoS) attack or Distributed Denial of service (DDoS) attack but also it is causing a tremendous economic loss. The EDoS takes advantage of

some attractive features in the cloud-computing environment such as elasticity, auto scaling, and pay as you go model. The main source of the EDoS attack is the DDoS attack targeting the cloud resources. Then, because of the elasticity and auto scaling features, the resources will grow according to the demand of the attack and due to the pay as you go model of the cloud, adopters will be charged for the scaling of the resource until it reaches a point that it cannot sustain economically [9], [10]. DDoS could prevent the legitimate users from accessing the service for a certain amount of time but EDoS could prevent the service provider from delivering the service forever if the attack leads to bankruptcy.

The following scenarios illustrate the idea of the EDoS attack and the normal auto-scaling feature. Scenario 1 illustrates the Auto Scaling concept on the cloud services and it is shown in Figure 1.

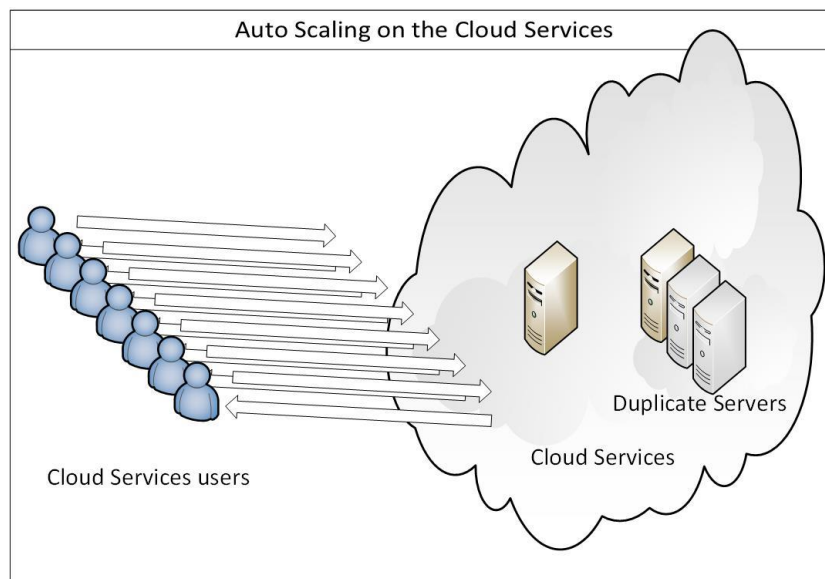


Figure 1 Auto Scaling on the cloud Services

Auto scaling is one of the attractive features of a cloud-computing environment. This feature allocates automatically more instances or resources to handle the high load (scale up) and release automatically these resources when the load or traffic returns back to normal (scale down). The auto scaling could be activated by monitoring some parameters such as CPU utilization, Memory usage, response time, and network bandwidth. In scenario one, we illustrate the auto scaling feature activation, when more requests are coming to the cloud computing environment. In this case, the resources scale up and duplicate the servers to handle the high traffic.

Scenario 2 illustrates the EDoS Attack targeting the cloud services and it is shown in Figure 2.

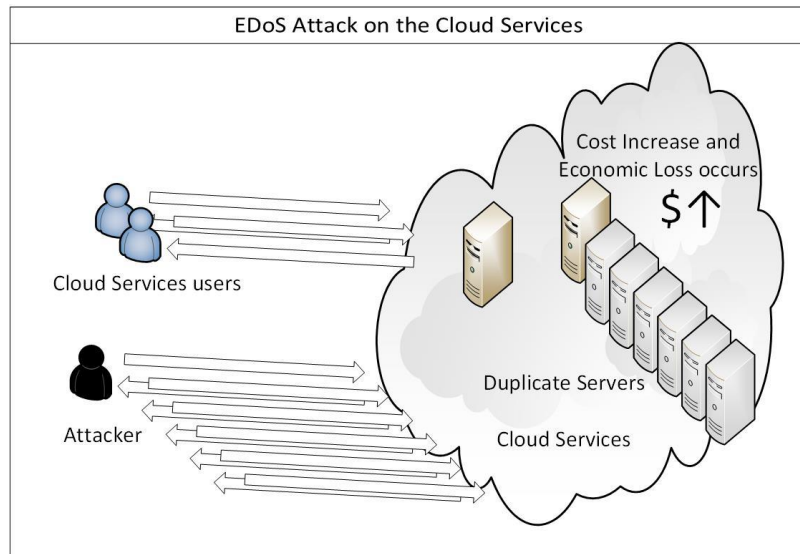


Figure 2 EDoS Attack on the cloud Services

In scenario two, the attacker launches a high number of requests to the cloud computing services, and then the servers scale up by the auto-scaling feature to handle this high traffic. Then, the adopter has to pay for all duplicated servers due to the pay-as-you-go

model. The cost for the adopter increases and economic loss occurs that it cannot be sustained economically.

Our study in this thesis focuses on the evaluation of the EDoS attack on the SaaS cloud service such as a web application service, which could be considered as a single-class service. For most of the web applications, the number of client requests and the service rate are considered to be random variables having Poisson distribution [18], [19]. We are focusing on the EDoS attack targeting a single-class service where all cloud customers' requests have the same processing procedure as it is in the web service that delivers content, such as web pages, using the Hypertext Transfer Protocol (HTTP) over the Internet. Thus, considering a Poisson distribution for the service rate in our case is a valid assumption that also helps in simplifying our performance model.

CHAPTER 2

LITERATURE REVIEW

Cloud computing security is one of the very important issues and some researchers have attempted to address it. They consider the DDoS and EDoS attacks as flooding attacks but due to the architectures of the cloud computing these attacks could increase the risk and the damage caused to the cloud computing systems. In this section, we discuss the related work including the security concern related to EDoS and the mitigation techniques that tackle the EDoS. In addition, we state the advantage and disadvantage of the existing mitigation techniques.

Zhang et al. [17] provide a brief summary about some research challenges in cloud environments such as virtual machine migration, energy management, traffic management and analysis, automated service provisioning, data security, server consolidation, software frameworks, storage technology and data management, and novel cloud architectures. In this work, we tackle an important research challenge that is the security of the cloud.

Cloud computing security is one of the very important issues and some researchers have attempted to address it. Hoff discussed the concept of transforming DDoS attack to EDoS in his blog in 2008 [9], [10]. Then, many researchers started investigating the EDoS attack and included this attack as a major threat and risk for the cloud-computing environment. The following works consider the EDoS attack as a security issue in cloud computing systems.

Khosravani et al. [20] considered the EDoS attack as one of the technical risk that target the adoption of the cloud. Shah et al. [21] proposed a detailed study of SaaS, PaaS, and IaaS in terms of security, privacy, and trust; and determines risk and their solution directives. They consider EDoS and DoS as IaaS medium risk. The European Network and Information Security Agency (ENISA) considers the EDoS attack as a high impact attack in their report “Cloud Computing: Benefits, Risks and Recommendations for Information Security” [22]. The report stated the vulnerabilities that could cause the EDoS attack which are AAA vulnerabilities (authentication, authorization and accounting vulnerabilities), User provisioning vulnerabilities, User de-provisioning vulnerabilities, Remote access to management interface, and No policies for resource capping. In addition, the report stated that Company reputation, Customer trust, real time services, and Service delivery are affected assets by the EDoS attack [22]. Yu et al. [23] considered the EDoS attack as a new attack that targets the cloud platforms. They defined the EDoS attack as, “where a large number of bots act as benign clients to enjoy the service of the victim to financially bankrupt a cloud customer using the “pay-as-you-use” billing mechanism”. Lemoudden et al. [24] described how a traditional DoS attack could affect the cloud services and reach a complicated, resourcefully demanding premeditated goal. Sudha et al. [5] listed the EDoS attack as an application level attack that could overcome the cloud service budget and increase the cloud utility bill. Therefore, they considered this attack to create huge impact on the companies’ economic resources. Buyya et al. [25] proposed innovative management techniques to support SaaS layer applications and conceptual architecture of autonomic management of Clouds. This conceptual architecture has security and attack detection component, which is responsible

for differentiating between legitimate requests and DDoS attack requests. The main objective of this component is avoiding unnecessary waste of energy and budget due to dynamic provision of resources for attack requests. The authors [25] considered the DoS attack as a critical security threat because of dynamic provision resources that allocate more resources for the incoming requests whether it is legitimate request or an attack request. This allocating of unnecessary resources could increase the cost for the providers and increase the energy waste. Yu et al. [26] divided the DDoS targeting the cloud environment into two attacks based on the resource provisioning plans that are provided by the Cloud Service Provider. The first attack is the EDoS attack targeting the cloud when the resource-provisioning plan is a short-term on-demand one. The second attack is the traditional DDoS attack when the resource-provisioning plan is a long-term reservation. Xiao et al. [27] presented the potential vulnerabilities in the cloud-computing environment that may be exploited by the attackers. They consider the EDoS attack as a vulnerability of the cloud-pricing model. They stated that the EDoS attack needs more investigation and study [27].

In addition, there are many other works that consider and count the EDoS attack as a real threat for the cloud computing environment [28]–[30].

Reddy et al. [31] connect the security concerns and issues to the several service architecture levels of the cloud computing. They considered security of client data and information as the most important requirement for any services provided by any cloud-computing environment. They classify the security concerns and issues based on Software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS). They describe the flooding attacks and their effect on the cloud-computing

platform. They consider a DoS attack as one of the flooding attacks that target the infrastructure of the cloud-computing platform. They also state that the risk of a DoS attack increases when it targets a cloud computing platform, due to the shared resources as a result of the virtualization feature of the cloud computing. This type of flooding attacks could cause an Economic Denial of Sustainability (EDoS) attack. In our work, we concentrate on the application level security and the EDoS attack is a major concern. Therefore, we propose a new mitigation technique, for the EDoS, namely the EDoS Attack Defender.

Modi et al. [32] surveyed the intrusion detection techniques that can be used in a cloud computing environment. They stated many types of attacks that target the cloud-computing environment. They describe the flooding attack and how this attack could raise the usage bills drastically as the cloud would not be able to distinguish between the normal usage and fake usage. They described the EDoS attacks and they claimed that some of the proposed mitigation techniques are not efficient since they only use traditional firewalls because firewalls cannot distinguish good traffic from DoS attack traffic. In addition, they state that research is still needed to detect an EDoS attack in the cloud. In this work, we focus on the EDoS attack due to its high impact on the cloud-computing environment. We propose a new mitigation technique for the EDoS attacks, namely the EDoS Attack Defender.

Khor et al. [33] proposed an On-Demand Cloud-based EDoS Mitigation Mechanism named Self-verifying Proof of Work (sPoW). This mitigation technique is supposed to handle the EDoS attack before it triggers the billing mechanism. They introduced an asymmetric step before committing the server's resources. The server requires a proof of

work from the client, before committing its resources to the client. Clients expend their resources to solve a “crypto-puzzle” and submit a proof of the solution as an embedded signal (capability) within the packets. A server has to generate a “crypto-puzzle” to protect the connection server channel. A crypto-puzzle consists of both the encryption of channel information (such as IP address and port number) and the concealed encryption key with k bits representing the puzzle difficulty. A puzzle requester running on the client-side expands the client resources by brute forcing these k bits to discover the server channel information where it can submit an initial connection request. It prioritizes the legitimate traffic based on the difficulty of the puzzle. This mitigation technique has some weaknesses regarding the puzzle, requiring a computation power from the clients to solve the puzzles and from the server to generate the puzzles especially in case of difficult puzzles. In addition, it could lead to a puzzle attack if the attacker responds with high difficult unsolved puzzles to the server. In our work, we avoid requiring a proof of work from the clients when they ask for cloud resources, and only check whether the clients are humans by sending CAPTCHA if the demand of resources exceeds a pre-defined threshold.

Sqalli et al. [34] proposed a new mitigation technique against EDoS attacks in cloud computing, namely EDoS-Shield. The EDoS-Shield architecture consists of two main components. The first component is a virtual firewall (VF) and it works as a filter mechanism based on white and black lists that hold IP addresses of the originating clients. The second component is a verifier cloud node (V-Node) and it uses the graphic Turing tests such as CAPTCHA to verify legitimate requests. Then, it updates the whitelist and blacklist based on the results of the verification process. This technique is

based on the IP address and it does not consider IP spoofing. The authors of this paper improved the EDoS-Shield and proposed an Enhanced EDoS-Shield that takes into account the Spoofed IP Addresses [35]. This technique adds a Time to Live (TTL) value of the packet's IP header and a counter of unmatched TTL values to the white and black lists to distinguish whether the packet is having a spoofed IP address. In addition, it adds the attack start time in the blacklist. This work includes results obtained from a discrete event simulation model. These results show that this technique is an effective approach to mitigate EDoS attacks originating from spoofed IP addresses and outperforms the original EDoS-Shield. These results include performance and cost metrics.

These two mitigation techniques, EDoS-Shield and Enhanced EDoS-Shield, are based on the IP addresses lists approach that has many drawbacks such as blocking an entire NAT network if one of the public IP address is caught as an attacker and added to the black list. In addition, cloud services are accessible from everywhere, so it is difficult to recognize clients fingerprint and their TTL values; and some attack tools that could change the value of TTL so this value is not always correct. Finally, the EDoS-Shield adds overhead on the firewall to check the IP address of each request and the enhanced EDoS-Shield adds extra overhead on the firewall to check the IP address and its TTL value of each request. Then, these values are compared with blacklist and whitelist values. In our work, we avoid blocking the attack using IP addresses and only redirect the request if the auto-scaling feature and suspicion thresholds are triggered. In addition, no overhead on the firewall is added to check the IP address or the TTL values of the request for the purpose of comparing these values with blacklist and whitelist values.

VivinSandar et al. [36] explained the EDoS attack as a new breed of DDoS attacks and as one of the cloud specific attacks. They summarized the countermeasures of some approaches that tried to address the EDoS attack. The authors proposed a new protection technique for the EDoS attack. They conducted some experiments on AWS Amazon public cloud [37]. They concluded that their proposed protection technique will not eliminate completely the EDoS attack and more research is needed to prevent and mitigate the EDoS attack.

Kumar et al. [38] discussed and described the Economic Distributed Denial of Sustainability (EDDoS) in cloud computing. They proposed a mitigation technique for EDDoS attack. This mitigation technique consists of three modules, Packet filtering, Proof-of-work technique, and Egress filtering to avoid EDDoS attacks to the cloud. The authors focused on building an effective crypto puzzle [39], but this puzzle adds a computation overhead on the client side. In addition, in the cloud-computing environment, there are varieties of clients such as mobile clients, tablet clients, etc. These clients cannot handle the computation overhead to solve the crypto puzzle in order to gain access to cloud services. In addition, Gligor [40] stated that the client puzzles used as a proof of work are ineffective and unnecessary, as they impose a high overhead on legitimate client requests and only offer very weak guarantees.

Kumar et al. [41] proposed an EDDoS mitigation technique named Scrubber Service. This technique is used on-demand and is charged according to pay per use basis. This Scrubber service is responsible for the puzzle generation and verification, so there is no overhead on the cloud service. The proposed mitigation technique has two modes; the first one is the normal mode when the activity is normal on the cloud services. The

second one is the suspected mode when the activity is high on the cloud services. The change between these two modes depends on two threshold limits, resource consumption, and bandwidth consumption. The cloud services will forward all requests to the Scrubber service if it is in the suspected mode, so the scrubber can send a crypto puzzle to the client in order to prove its legitimacy for acquiring the service. The proposed mitigation has some limitations such as the disadvantage of the crypto puzzles that we mentioned earlier [40] and the threshold limits. The change of the modes could occur frequently by exceeding and returning under the threshold limit. The system could become unstable due to a fluctuation in the traffic activity around the threshold limit frequently.

Alosaimi et al. [42] proposed a new mitigation technique to encounter EDoS attack in a new cloud environment. This environment is where “Bring Your Own Device” (BYOD) policies in enterprises are defined. The attack is targeting the Identity and Access Management (IAM) vulnerabilities in the BYOD implementation in the organizations to gain access to the internal resources of the organizations and launch an EDoS attack. This attack is taking advantage of the missing of resource control and management of platforms of the BYOD devices. This attack could cause Direct DDoS to the organization itself or cause Indirect DDoS to other organizations that use the same cloud service provider. Their mitigation technique is called DDoS- Mitigation System (DDoS-MS). It investigates only two packets from the source of the request by using two types of testing, the Graphic Turning Test (GTT) and Crypto Puzzles. The two types of testing are used to authenticate the user and the packet. In addition, this technique is using the black and white lists based on IP addresses to control the access to the cloud services.

Alosaimi and Al-Begain [42] proposed an enhanced DDoS Mitigation technique that is an improvement of the previous mitigation technique [43]. The enhanced DDoS-MS is only investigating the first packet by using Graphic Turning Test (GTT). The enhanced DDoS-MS consists of a virtual firewall, a verifier node(s), a client puzzle server, an Intrusion Prevention System (IPS) device, and a Reverse proxy (RP) server. The virtual firewall has four different lists, the black list, the white list, the suspicious list, and the malicious list. The Intrusion Prevention System (IPS) is used to investigate the packet's content for malicious content such as malware. The Reverse proxy (RP) server is responsible for hiding the locations of cloud servers and balancing the load between these servers. In addition, it monitors the rate of traffic to detect the DDoS attacks. The client puzzle server is used as a reactive step to delay the requests of a user who tries to overwhelm the system.

These two mitigation techniques, DDoS-MS and Enhanced DDoS-MS, are based on an IP addresses lists approach that has many drawbacks such as blocking an entire NAT network if one of the public IP addresses is caught as an attacker and added to the black list. In addition, cloud services are accessible from everywhere, so it is difficult to recognize clients fingerprint and their TTL values; and some attack tools could change the value of TTL so this value is not always correct. In addition, they used the puzzle server that has many drawbacks discussed earlier. Finally, these two mitigation techniques add a huge overhead on the system because of their filters and lists. In our work, to make the system reliable and simple, we avoid blocking the attack using any list and only redirect the request if the auto-scaling feature and suspicion thresholds are triggered.

Since the DDoS attack is the main source of EDoS attack, we focus on the DDoS attacks on cloud computing environments and we discuss existing DDoS mitigation techniques. These techniques could help us improve the mitigation techniques for EDoS attacks.

According to the security report from NSFOCUS, 93.2% of the DDoS attacks in 2013 last for less than 30 minutes, similar to what was observed in 2012. In addition, the most popular methods that are used for attacking are TCP Flood and HTTP Flood, so the focus is on the application layer attack because it is less expensive and cause higher damage. “A typical application layer attack like HTTP Flood is popular among hackers because it specifically targets the consumption of CPU/storage/database resources, which can shut down a victim’s website without generating a large amount of network traffic. That being said, the traditional TCP Flood and UDP Flood will not disappear either, since they are still the most effective attacks against victims that are not protected by dedicated anti-DDoS mitigation equipment or service” [44], [45]. In our work, we concentrate on the application layer attacks and our proposed mitigation technique is based on the reactive mitigation strategy because most attacks last for a short time.

Chen et al. [46] surveyed the existing virtualization technologies, the benefits and advantages of implementing virtualization, and stated the security vulnerabilities of using virtualization. In addition, they presented two case studies, the first one is XEN hypervisor security analysis, and the second one is VMware hypervisor security analysis. The authors stated that the VMware VM and XEN are suffering from the DoS attack as one of the security issues [46]. Since the virtualization technology is the key feature of the cloud computing environment, then all security issues related to virtualization could affect the security of the cloud-computing environment.

Shea et al. [47] proposed a study of the impact of a DoS attack on four types of virtual machines (VMs). These virtual machines (VMs) are KVM, Xen, Open VZ, and Vanilla. The authors launched two types of attacks on the VMs, the first one is TCP DoS, and the second one is UDP flood. The authors described the degradation of performance of these VMs under attacks [47]. The combination of virtualization overhead and performance degradation in a DoS attack can lead to a 50% decrease in Web Server performance when compared to the non-virtualized Vanilla system using the same amount of resources. They stated that modern virtualization is more vulnerable under TCP SYN DoS attack. They stated that SYN-cookies and SYN-caches do not provide adequate protection for VMs. The DoS attack on these virtual machines could change to Economic Denial of Sustainability attack under the cloud-computing environment with auto scaling feature enabled.

The BitBucket is a code hosting web service offered by the public cloud Amazon [37]. This service was unavailable for more than 19 hours due to a DDoS attack; so many developers could not access their code projects hosted on the BitBucket. The attack is a massive-scale DDoS that contains massive flood of UDP packets coming into an Amazon IP, eating away all bandwidth. Amazon blocked the offending traffic, and service returned to normal after at least 16 hours from the time it was first reported. However, by the next morning, the problem returned, and another two hours passed before this second outage was reversed. The second attack used a flood of TCP SYN connection requests, rather than UDP packets [48].

Karnwal et al. [49] proposed a mitigation technique called filtering tree for the DoS attacks. They focus on REST, HTTP, and XML based DoS attacks because the cloud

computing users make their requests in XML then send these requests using HTTP protocol and build their system interface with REST protocol such as Amazon EC2 or Microsoft Azure. This technique acts as a service broker combined within a SOA model. This service converts the client requests in XML tree form and uses a virtual Cloud defender and its filters to investigate the client requests. The proposed architecture consists of three parts; the first part is named Embed SOAP Message, and is responsible for detecting a coercive parsing attack using SOAP signatures. The second part is named IP Trace-Back, and it is an IP address blacklist containing the blocked IP address (provided by the virtual Cloud defender). In addition, the third part is named virtual Cloud defender. The virtual Cloud defender contains five filters. The Sensor Filter monitors the high traffic and marks it as suspicious. The HOP Count Filter compares the Hop count of incoming traffic with the stored Hop Count value and if it does not match, then those messages are marked as having suspicious IP addresses. The IP Frequency Divergence detects the same frequency of IP messages and marks those messages as having suspicious IP addresses. The Puzzle Resolver Filter sends a puzzle to all suspicious IPs from previous filters, and if the suspicious IP address sends the correctly solved puzzle to the puzzle-resolver, then it means it is a legitimate client; otherwise, the puzzle resolver drops the request message and sends the suspicious IP address to the IP Trace-Back. Finally, the Double Signature Filter checks the xml message for open tag and drops the message if found, otherwise it sends the request to the cloud service. This mitigation technique has a large overhead due to the number of filters that the request must go through. In addition, this technique is based on the IP list and puzzle mechanisms and for which we have already stated the drawbacks.

As a summary, more investigation and exploration are needed to address many research challenges in the security of the cloud computing. The Economic Denial of Sustainability (EDoS) attack is a serious threat to the cloud computing environment due to its impacts on the economic side. I explored the exiting mitigation techniques for the EDoS attack, but these techniques are not sufficient to prevent the cloud computing environment from the EDoS attack. These mitigation techniques suffer from many drawbacks such as using IP address lists to block the attacker, using a puzzle to prove that the client commits to the resources, using firewalls or filters to process all the clients' requests causing an overhead. In our work, the cloud computing environment is operating in normal mode with no need to process the client requests. However, if the auto-scaling feature and the suspicion thresholds are triggered, then the cloud computing environment operates in the suspicious mode and redirects the client request to the EDoS Attack Defender. The EDoS Attack Defender will differentiate between legitimate users and automated attackers (Zombies). This component will differentiate between the traffic by sending Graphic Turing Tests [11], such as CAPTCHA [12]–[15], to the request generator. If the request generator could respond to the CAPTCHA, then it will be considered by the EDoS Attack Defender as a legitimate user, and its request will be directed to the cloud computing service. On the other hand, if the request generator could not respond to the CAPTCHA, then it will be considered by the EDoS Attack Defender as an automated attacker (Zombie) and its request will be blocked.

CHAPTER 3

EDoS ATTACK DEFENDER MITIGATION TECHNIQUE

In this chapter, we propose a novel solution, namely EDoS Attack Defender, to mitigate the Economic Denial of Sustainability (EDoS) attack in the cloud computing platforms. We design a discrete simulation experiment to evaluate its performance and cost metrics of this mitigation technique.

Cloud computing has an attractive feature namely auto-scaling. This feature allows for allocating new instances for the purpose of handling the increased demands on the services and releasing some existing instances when these demands decrease. Auto scaling is an automatic feature with minimal management effort. Auto scaling is triggered based on some parameters and thresholds. The cloud computing platform administrator based on the services that are provided chooses these parameters and thresholds. The administrator could select the average CPU utilization, memory, or response time as the auto scaling parameter with predefined threshold based on the specific application. The additional resources that are allocated are charged to the adopters of the cloud computing by “pay-as-you-use” model or utility computing. Therefore, the DDoS attack problem could create a huge financial impact when targeting the cloud computing environment. This new type of attack, namely Economic Denial of Suitability (EDoS) could drag the cloud adopter to a point that it could not pay the bill and could not sustain economically.

The main idea of the EDoS Attack Defender is to verify whether the requests are coming from legitimate users or generated by compromised machines (Zombies). Most of the

existing mitigating techniques are using filtering mechanisms that generate high overhead on the system or are using classifying techniques that use some parameters to classify the incoming requests, but that could block legitimate users or allow attack traffic. Unlike these mitigation techniques, the EDoS Attack Defender is only activated under suspicions conditions and starts investigating the traffic to distinguish between legitimate and attack traffic, and then only allows the legitimate users and drops all attack traffic.

DDoS is one of the major threats to many systems. DDoS is the main source of the EDoS attack. Therefore, we should get the benefit of all research and finding for the DDoS attack. The main scheme to prevent or mitigate the DDoS attacks is divided into two mitigation schemes. The first scheme is a reactive mitigation technique that has three phases. The first phase consists of distributed components to monitor the system for detecting any DDoS attacks. The second phase is triggered only if the first phase detected an on-going DDoS attack and it tries to locate the attack source. In the third phase, the DDoS attack is mitigated by deploying countermeasures. The second scheme is a proactive mitigation technique that takes appropriate actions and investigations before the attack hits the system [41], [50].

Our proposed technique is the EDoS Attack Defender and it is based on reactive mitigation schemes. The EDoS Attack Defender consists of three phases. In the first phase, we are monitoring the auto-scaling feature threshold to detect if there is an EDoS attack. The monitoring parameters are based on the auto-scaling parameters because the EDoS attack tackles the auto-scaling feature. In this phase, we monitor the average CPU utilization using an upper level threshold and a lower level threshold in order to avoid the first unnecessary auto scaling under the attack.

In the second phase, once an attack is detected, the cloud service will trigger the EDoS Attack Defender by forwarding all new requests to this component. The EDoS Attack Defender is responsible for differentiating between legitimate users and automated attackers (Zombies). This component will differentiate the traffic by sending Graphic Turing Tests [11], such as CAPTCHA [12]–[15], to the request generator.

In the third phase, the EDoS Attack Defender will drop all traffic that cannot respond to the CAPTCHA. However, the EDoS Attack Defender will forward all requests that pass the CAPTCHA validation to the cloud service.

The EDoS Attack Defender mitigation technique consists of two components, cloud service and EDoS Attack Defender see Figure 3.

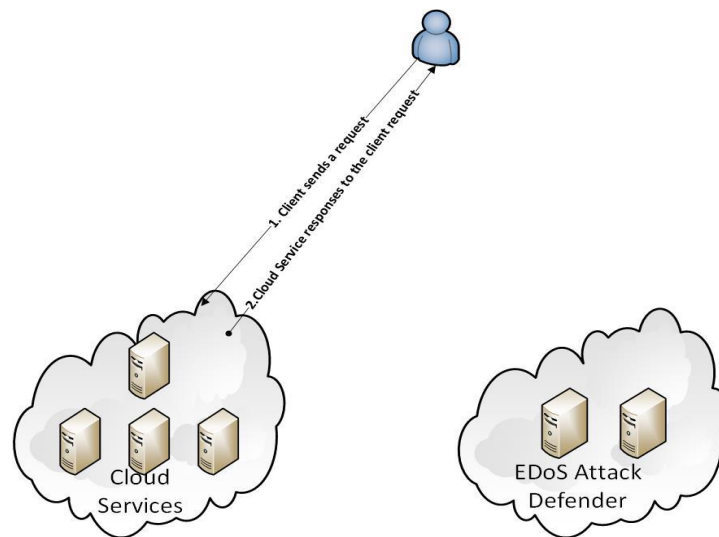


Figure 3 EDoS Attack Defender Components

The EDoS Attack Defender mitigation technique has four modes; Normal Mode, Suspicious Mode, Flash Crowd Mode, and Attack Mode see Figure 4.

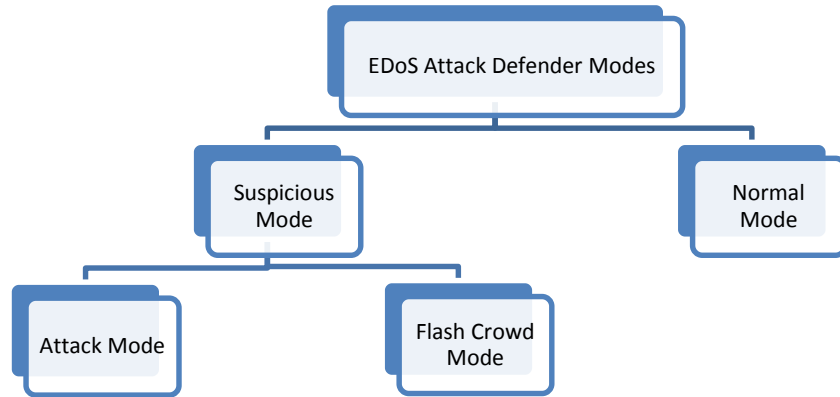


Figure 4 EDoS Attack Defender Modes

Since the EDoS attack is targeting the auto-scaling feature, we are also using the auto-scaling feature to detect the attack, and then prevent or mitigate this attack. In this mitigation technique, we have defined three thresholds, baseline utilization, the upper level utilization, and lower level utilization. The value of the baseline utilization is retrieved from the system historical behavior on an infrequent basis to track the system behavior.

The upper and lower utilization thresholds are defined based on the auto-scaling utilization threshold and baseline utilization. For example, if the auto-scaling utilization threshold is 80% then the upper level utilization threshold is 80%. The lower level utilization threshold is equal to the upper level utilization threshold minus 10%, which is 70%, in case the baseline utilization is less than 70%, otherwise the lower level utilization threshold will be equal to the baseline utilization. The upper and lower utilization thresholds are used to determine the suspicion mode region (See Figure 5).

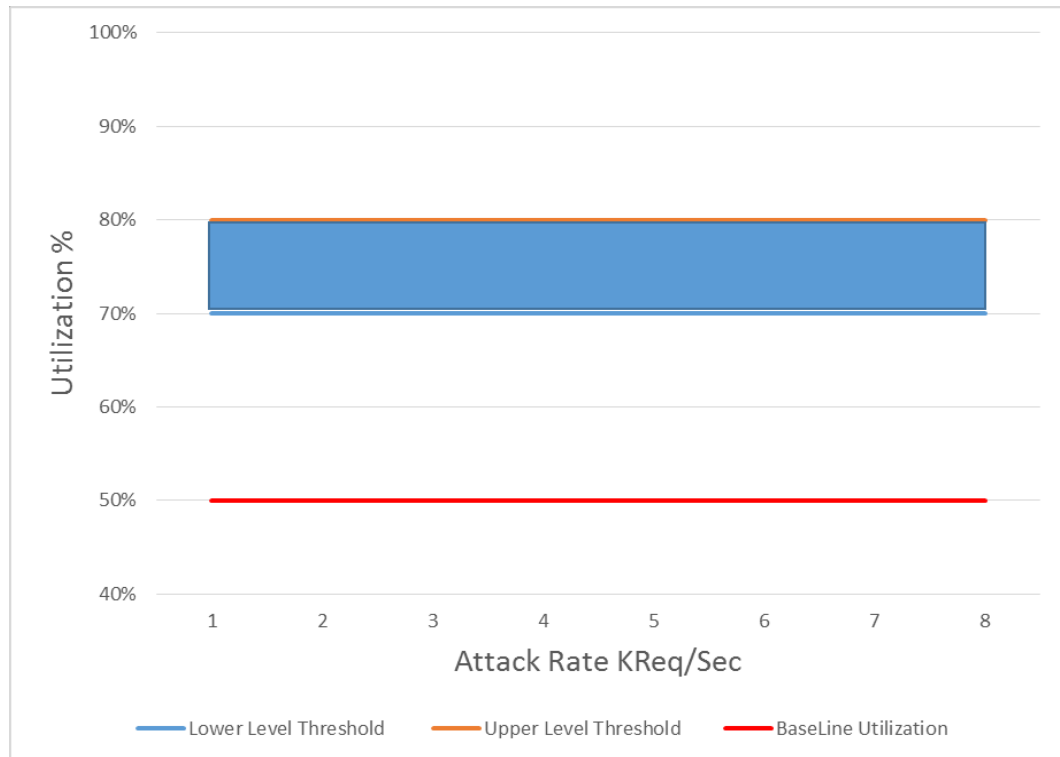


Figure 5 EDoS Attack Defender, Suspicion Mode Thresholds

In addition, we have defined a timer named the Attack Period Timer. This timer is used with the baseline utilization to state if the attack finished, by calculating the legitimate response percentage in this period.

The cloud service will start operating normally in the Normal Mode (No overhead or checking in this mode). We consider the cloud computing environment in the normal mode if the current system utilization is below the lower level utilization threshold, and the cloud services respond to the client requests as shown in Figure 6.

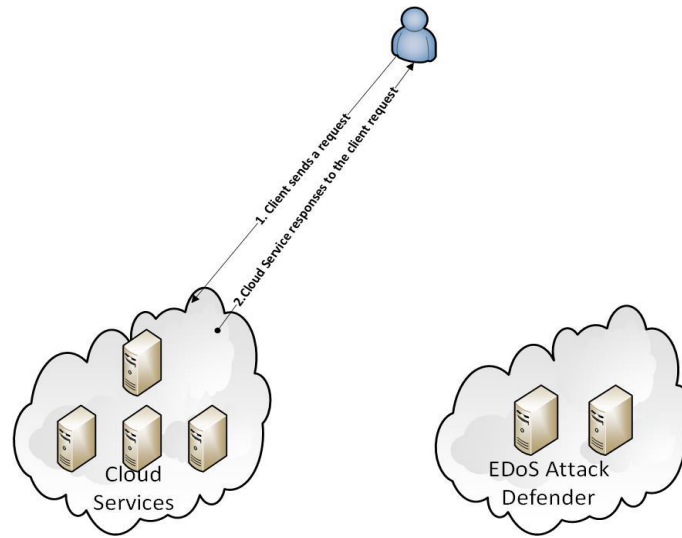


Figure 6 Normal Mode

The cloud service will monitor the lower and upper level utilization thresholds and if the system current utilization is inside the suspicion mode region, then the mode will change to Suspicious Mode. For example, if the lower and upper level utilization thresholds are 70% and 80% respectively, then the cloud service will change to Suspicious Mode if the system utilization value is between 70% and 80%.

In Suspicious Mode, all new incoming requests are forward directly to the EDoS Attack Defender component to check if it is a Flash Crowd Mode or Attack Mode. Then, the EDoS Attack Defender will differentiate between legitimate users and automated attackers (Zombies) by sending CAPTCHA to the request generator. After that, the EDoS Attack Defender will drop the request in case of a failure in responding to the CAPTCHA, and consider the request generator as an attacker as shown in Figure 7.

The EDoS Attack Defender will redirect the request to the cloud service in case of a success in responding to the CAPTCHA, and consider the request generator as a legitimate user. Then, the cloud services will respond to the client request as shown in Figure 8.

The EDoS Attack Defender component will keep sending CAPTCHA to all new incoming requests and record the number of legitimate responses to the CAPTCHA. Then, when the current system utilization is out of the suspicion region (above the upper level utilization or below the lower level utilization), we will calculate the percentage of legitimate responses by dividing the number of legitimate responses over the total number of requests. If this value is greater than 90%, then it is a Flash Crowd Mode; and if it is below 90%, then it is Attack Mode. We have chosen 90% as the success rate of the CAPTCHA to mimic the real life when some legitimate users (10%) could not solve the CAPTCHA for some reason. In addition, there are many types of CAPTCHA that are deployed with different success rates. For example, the reCAPTCHA [51] has an overall success rate of 96.1% based on more than 1 billion responses. However, this success rate is in the range of 92.6-96.9% for users of a native language that is not English [51]. Therefore, we select 90% to include all users. However, we could change this percentage based on the CAPTCHA type selected.

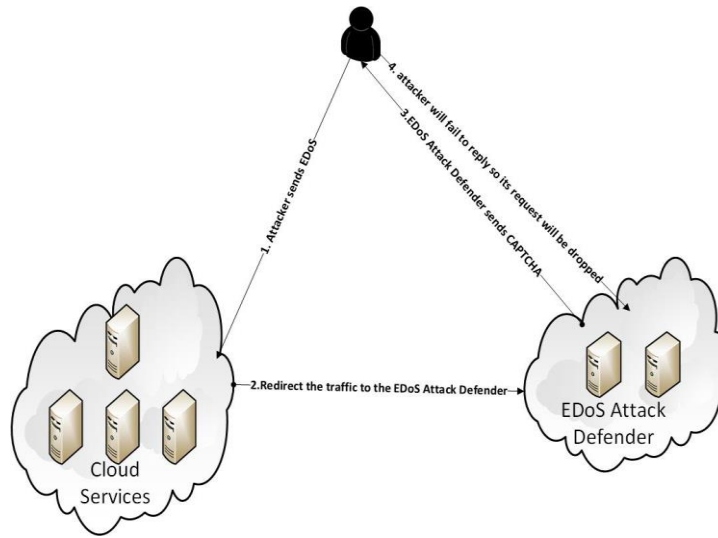


Figure 7 Suspicious and Attack Mode, Attacker Case

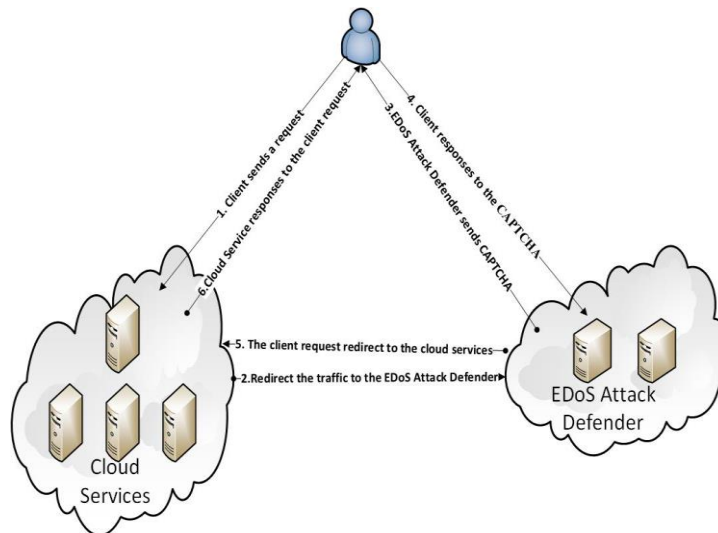


Figure 8 Suspicious and Attack Mode, Legitimate User Case

In the Flash Crowd Mode, the requests will be served directly from the cloud service as in the Normal Mode.

In the Attack Mode, the EDoS Attack Defender will continue to send the CAPTCHA to the Request Originator and record the number of legitimate responses.

The EDoS Attack Defender will drop all traffic that cannot respond to the CAPTCHA. However, the EDoS Attack Defender will forward all requests that pass the CAPTCHA validation to the cloud service.

The EDoS Attack Defender will keep sending CAPTCHA to all new incoming traffic until the system's current utilization is equal or smaller than the baseline utilization. Then, the EDoS Attack Defender will send the CAPCHA for a certain amount of time based on the Attack Period Timer; and after that, we would calculate the percentage of legitimate responses by dividing the number of the legitimate responses over the total number of requests in this period. If this value is greater than 90%, then the system returns back to the Normal Mode, and otherwise it stays in the Attack Mode.

3.1 Proposed Architecture and Algorithms

The following flowcharts of Figure 9 and Figure 10 describe the cloud service and EDoS Attack Defender components behavior.

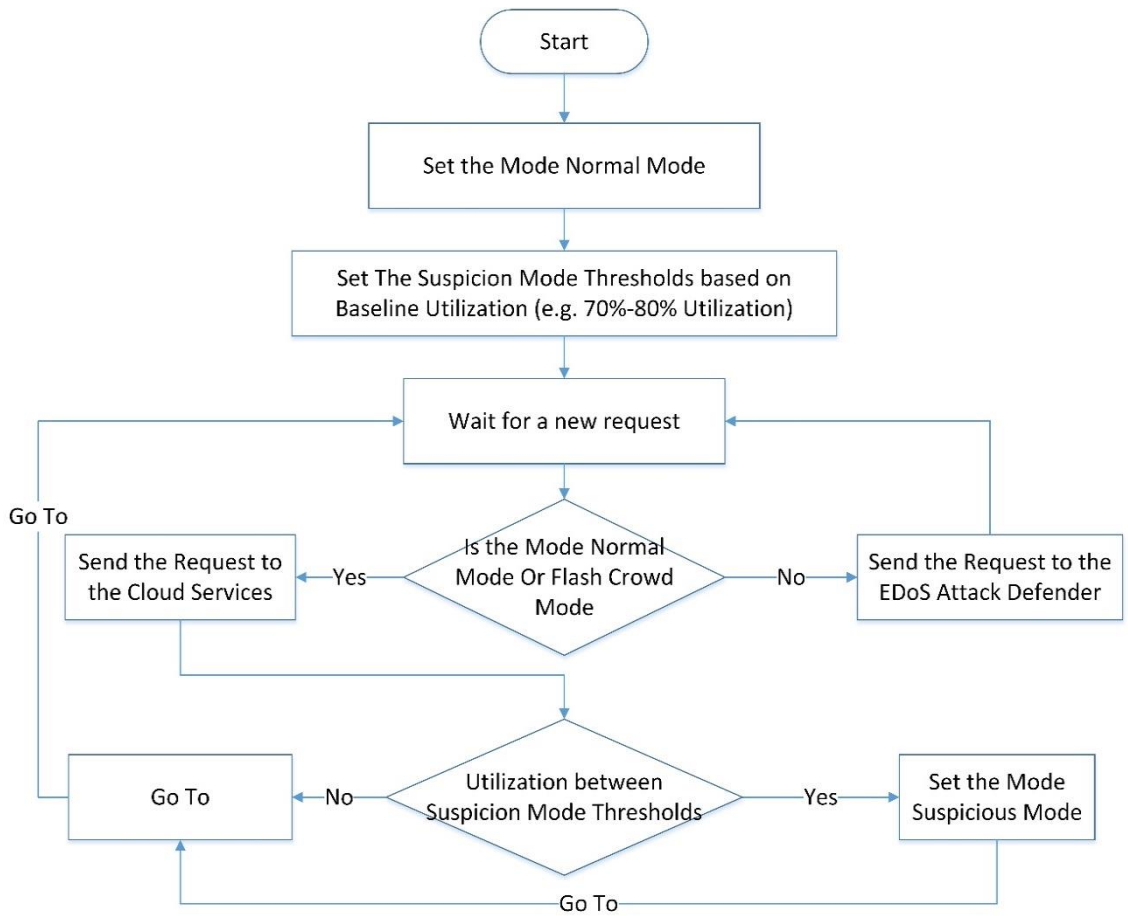


Figure 9 Cloud Service Component Flowchart

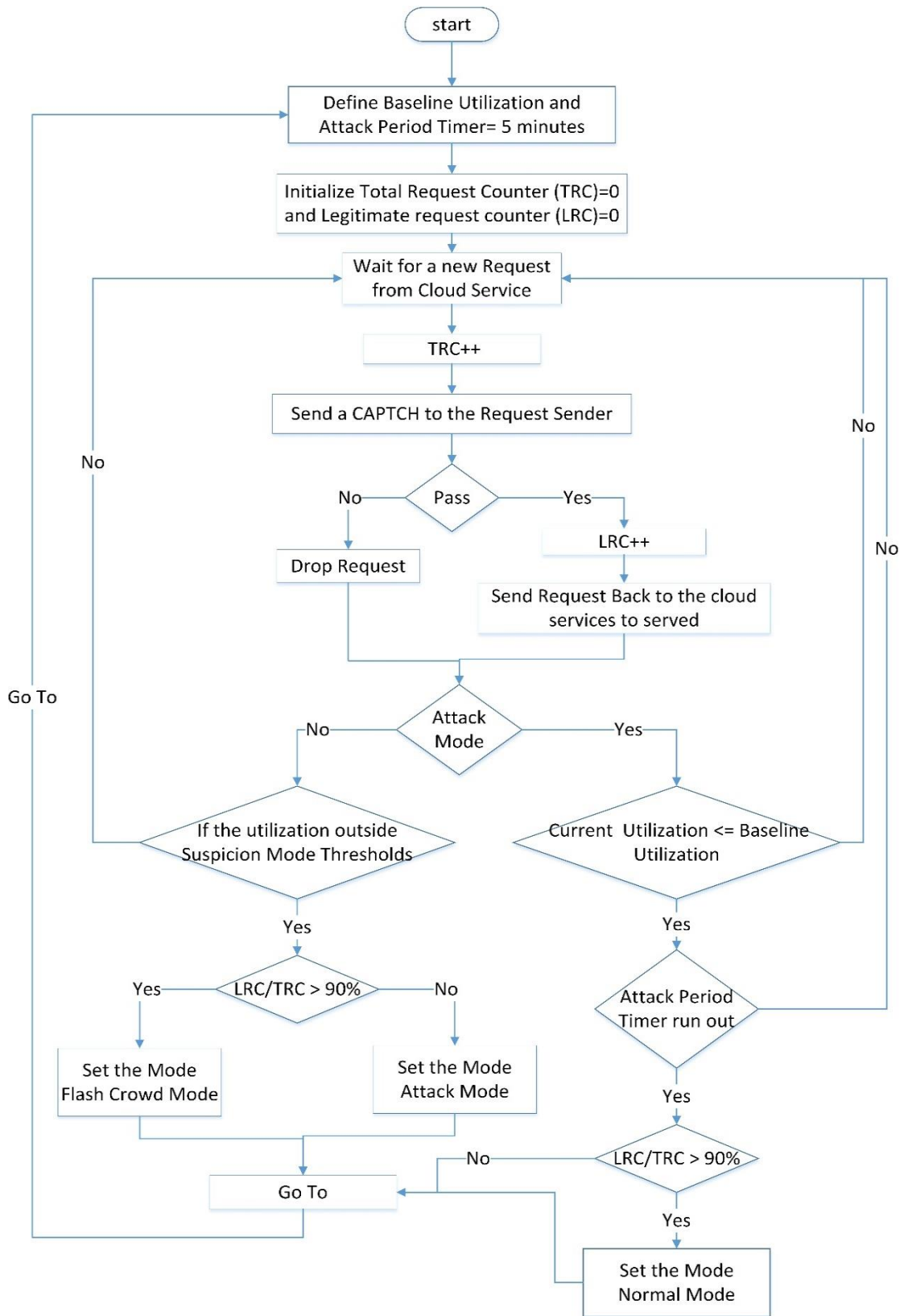


Figure 10 EDoS Defender Component Flowchart

Algorithm 3.1 describes the actions at the cloud service when receiving a request. At the cloud service, the actual platform is kept simple with negligible overhead and without major modification. A request will be forwarded directly to the cloud services instances if the current mode is Normal Mode or Flash Crowd Mode. Otherwise, the request will be forwarded to the EDoS Attack Defender for further investigation. After that, we will check if the system utilization is between the suspicion mode thresholds, then the current mode will change to Suspicious Mode.

Algorithm 3.1: Cloud Service

Input:

$R \leftarrow$ Request
 $CM \leftarrow$ Cloud Mode (Current Mode)
 $NM \leftarrow$ Normal Mode
 $SM \leftarrow$ Suspicious Mode
 $FCM \leftarrow$ Flash Crowd Mode
 $AM \leftarrow$ Attack Mode
 $SU \leftarrow$ System Utilization
 $SMT \leftarrow$ Suspicion Mode Thresholds

Begin:

$CM \leftarrow NM$
 Define SMT
While (New request from cloud service) {
 If ($M == NM \ || \ M == FCM$)
 {
 Forward R to Cloud Services
 If ($SU \in SMT$)
 $CM \leftarrow SM$
 }
 Else
 Forward R to EDoS Attack Defender
 }//END While
End

Algorithm 3.2 describes the actions at the EDoS Attack Defender when receiving a request from the cloud service for further investigation. At the EDoS Attack Defender, we will differentiate between Attack Mode and Flash Crowd Mode by sending CAPTCHA to all new incoming requests. The EDoS Attack Defender will drop the request in case of a failure in responding to the CAPTCHA and consider the request generator as an attacker. However, the EDoS Attack Defender will redirect the request to the cloud service in case of a success in responding to the CAPTCHA and consider the request generator as a legitimate user. Then, the cloud services will respond and serve the client request. In addition, it will check if the attack has finished by using the attack period timer and the percentage of the legitimate response to the CAPTCHA.

Algorithm 3.2: EDoS Attack Defender

Input:

R ← Request
 RS ← Request Sender
 CM ← Cloud Mode (Current Mode)
 NM ← Normal Mode
 SM ← Suspicious Mode
 FCM ← Flash Crowd Mode
 AM ← Attack Mode
 SU ← System Utilization
 SMT ← Suspicion Mode Thresholds
 BU ← Baseline Utilization
 APT ← Attack Period Timer
 TRC ← Total Request Counter
 LRC ← Legitimate Request Counter

Begin:

Read BU
 TRC=0, LRC=0

While (New request from cloud service) {
 TRC++
 Send to RS a graphic Turing test
If (Turing test passes) {
 LRC++
 Forward R to Cloud Services
 }
Else

Drop R

```

If (CM == AM) //Attack Mode
    If ( SU <= BU ) {
        If ( APT run out )
            If ( LRC/TRC >= 90% ) {
                CM←NM
                return
            }
            Else
                return
        }
    Else //Suspicious Mode {
        If ( SU ∉ SMT )
            If ( LRC/TRC >= 90% ) {
                CM←FCM
                return
            }
            Else {
                CM←AM
                return
            }
        }
    }
} //END While

END

```

Our proposed mitigation technique will be compared to the EDoS-Shield technique. The EDoS-shield is one of the mitigation techniques that are designed to prevent and mitigate the EDoS attacks. The EDoS-Shield architecture consists of two main components. The first component is a Virtual Firewall (VF) and it works as a filter mechanism based on a whitelist and a blacklist that hold IP addresses of the originating clients. The second component is a Verifier Cloud Node (V-Node) and it uses the graphic Turing tests such as CAPTCHA to verify legitimate requests. Then, it updates the whitelist and blacklist based on the results of the verification process [34], [35].

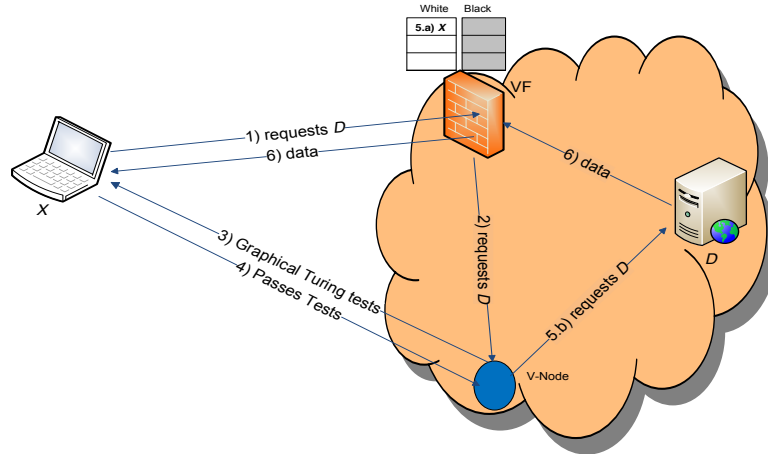


Figure 11 the EDoS-Shield Architecture [34]

Our proposed mitigation technique, the EDoS Attack Defender, will be triggered only if the auto-scaling feature is activated, instead of monitoring all incoming traffic and classifying it into a blacklist and a whitelist as in the EDoS-Shield technique. The EDoS-Shield generates an overhead on the VF to check the incoming traffic. In addition, the EDoS-Shield requires communication between the VF and the V-Node, which causes a degradation on the performance of the cloud computing environment. On the other hand, our proposed technique only redirects the requests to the EDoS Attack Defender in case of a suspicious mode. The EDoS-Shield classified the traffic based on the IP address, and that has many drawbacks such as blocking an entire NAT network if one of the public IP addresses is caught as an attacker and added to the blacklist. The EDoS Attack Defender does not classify the incoming traffic and it triggers only in case of high traffic. Similarly, the EDoS Attack Defender uses CAPTCHA to differentiate between legitimate users and automated attackers (Zombies) similar to what the EDoS-Shield does.

In our work, we will conduct simulation experiments for the EDoS attacks and our proposed mitigation technique: EDoS Attack Defender. The simulation will evaluate the

EDoS Attack Defender performance and then compare these results with the EDoS-Shield simulation results.

We conducted different simulation experiments on our proposed mitigation technique to verify the effectiveness of the EDoS Attack Defender. In addition, the comparison between our proposed mitigation technique and the EDoS-Shield will verify its effectiveness.

CHAPTER 4

SIMULATOR

This chapter discusses the implementation of the simulation to evaluate the proposed mitigation technique. We also verify and validate our simulation by comparing the obtained results with the EDoS-Shield mitigation technique results [34].

4.1 Simulator's Design

The simulation is implemented using C# programming language. A user-friendly graphical user interface is used to monitor all simulations. A discrete-event simulation model is used. The simulation is strictly based on the guidelines given by the book of Law and Kelton and the simulation setup of the EDoS-Shield mitigation technique paper [34], including the use of initial seeds that were ten million apart, and avoiding any overlapping in the random number streams during the simulation. As recommended in [52], PMMLCG (prime modulus multiplicative linear congruential generator) is used in the simulation for generating random numbers. The PMMLCG is an efficient generator and one of the most popular methods for generating random numbers [52].

Our simulation model has two types of events including the ARRIVAL and DEPARTURE events. An ARRIVAL event occurs when a new request arrives at the system. A DEPARTURE event occurs when a request is completely processed by the

system. The two events are generated independently such that each event has its own seed and random-number stream (See Figure 12).

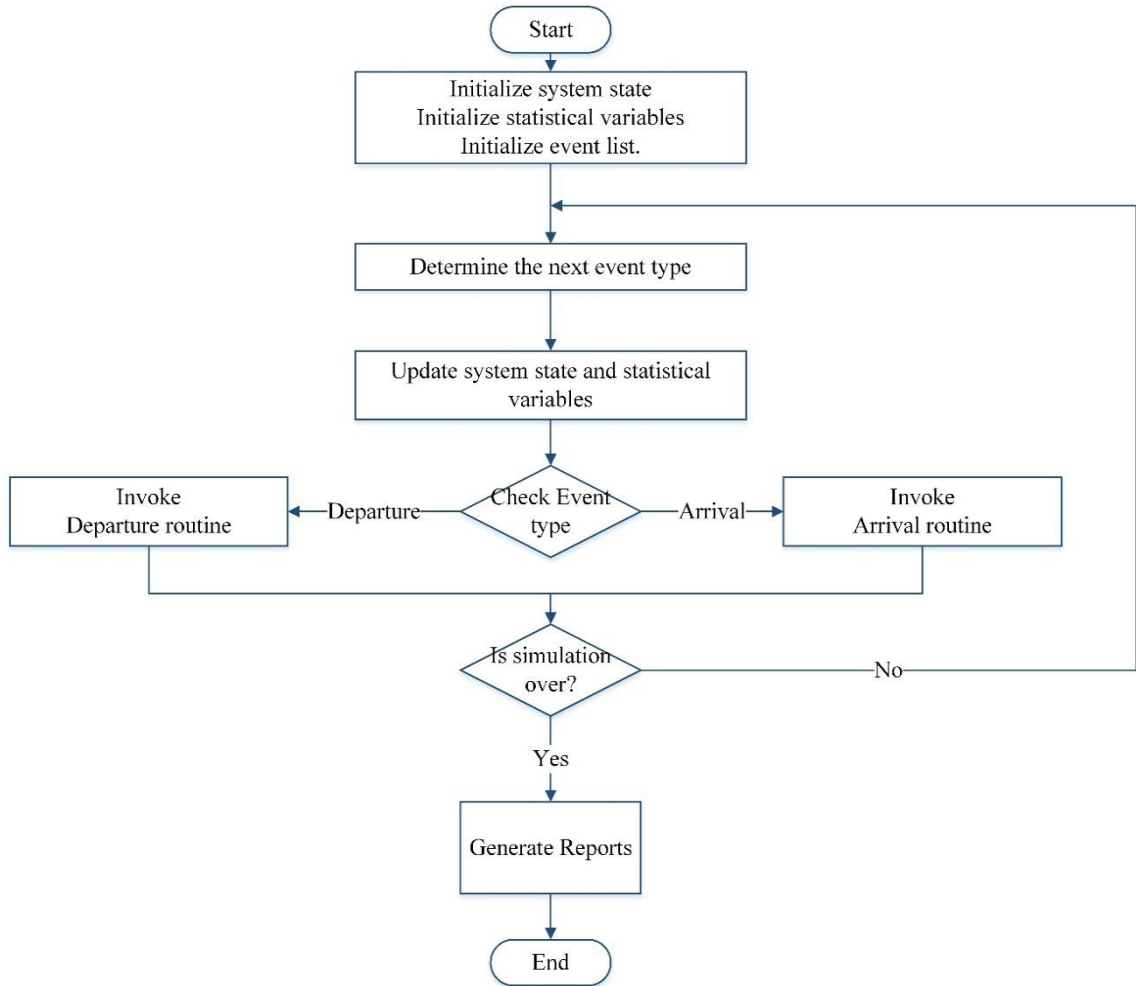


Figure 12 Simulation Model Flowchart

A discrete-event simulation experiment was conducted to evaluate the performance of the cloud service under the EDoS attack in terms of key performance indicators including end-to-end response time, computing resources utilization, and throughput. Since the EDoS attack is mainly targeted towards the cloud adopter, we have also evaluated the

cost associated with the computing resources and bandwidth allocations at the cloud service side.

In the simulation experiment, we have considered the same setup as that of the queuing model presented in Figure 13. The input to the simulation is a combination of aggregated traffic from different sources including the attackers' traffic. We have considered the Poisson nature of the incoming traffic. We have assumed a fixed input rate of 400 Request /sec representing the rate of the legitimate requests coming from clients and a variable input rate ranging from 400 Request/sec to 8000 Request/sec representing the rate of the attack traffic.

We have assumed the parameters in the simulation experiments as follows.

1. The capacity of an instance is of 100 requests per second.
2. The number of initial running instances is 5 instances.
3. The upper utilization threshold is of 80%.
4. The scaling size is variable and it can be easily changed through the GUI of the simulation.
5. The price of the small instance and the large instance is \$0.115 and \$0.46 per hour, respectively.
6. The bandwidth price is of \$0.01 per GB in/out data transferred.

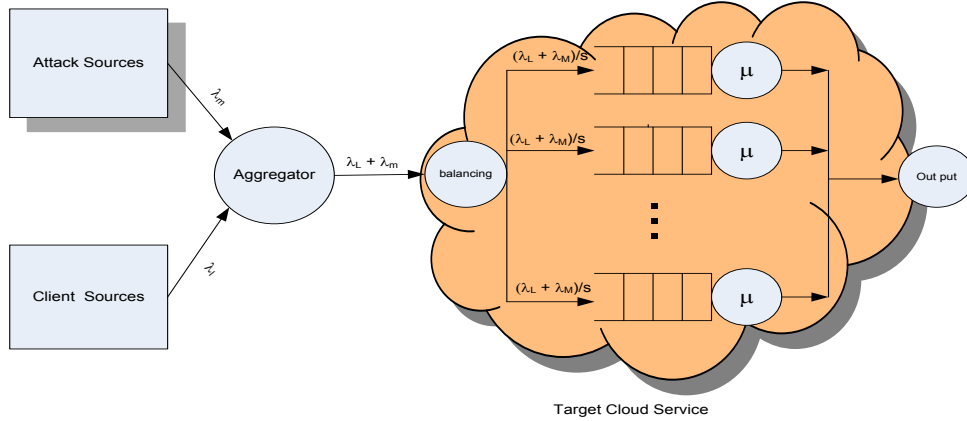


Figure 13 Queuing model for EDoS attack against a cloud service [34]

The Poisson arrival is assumed as DDoS attack [53], [54] since the DDoS attacks are the major source of EDoS attacks. Therefore, we have assumed that the attack traffic input to our simulation is Poisson arrival.

4.1.1 Simulator's Assumptions

1. The cloud computing services have a well-planned initial capacity.
2. The attacker cannot respond to the CAPTCHA.
3. The legitimate users would respond to 90% of the CAPTCHA requests.

4.2 Simulation Measures

4.2.1 Response Time

Since the response time is an important requirement in most of the SLAs, Table 3.1 shows the measured response time considering $M/M/s$, $M/M/1$, and parallel $M/M/1$ queues. It shows different response time measurements based on the input load.

Table 1 Response time Equations for different queuing model [55]

	<i>Approximated M/M/s</i>	<i>M/M/1 with $S\mu$</i>	<i>S M/M/1 each with μ</i>
Low load	$\frac{S}{S\mu - \lambda}$	$\frac{1}{S\mu - \lambda}$	$\frac{S}{S\mu - \lambda}$
High load	$\frac{1}{S\mu - \lambda}$	$\frac{1}{S\mu - \lambda}$	$\frac{S}{S\mu - \lambda}$

S is the number of instances, μ is the service time, and λ is the arrival rate.

For calculating the end-to-end response time, we use the decomposition method. First, the network is divided into subsystems including the Load balancers, EDoS Attack Defender, and the web servers (Cloud Service). Then, for each individual subsystem, we get the average delay considering their related input and capacity. Finally, the end-to-end response time is calculated by aggregating all the delays along paths from the source (clients) to the web server (Cloud Service) as a destination. Figure 14 below shows the considered subsystems and paths from the sources to the target cloud service.

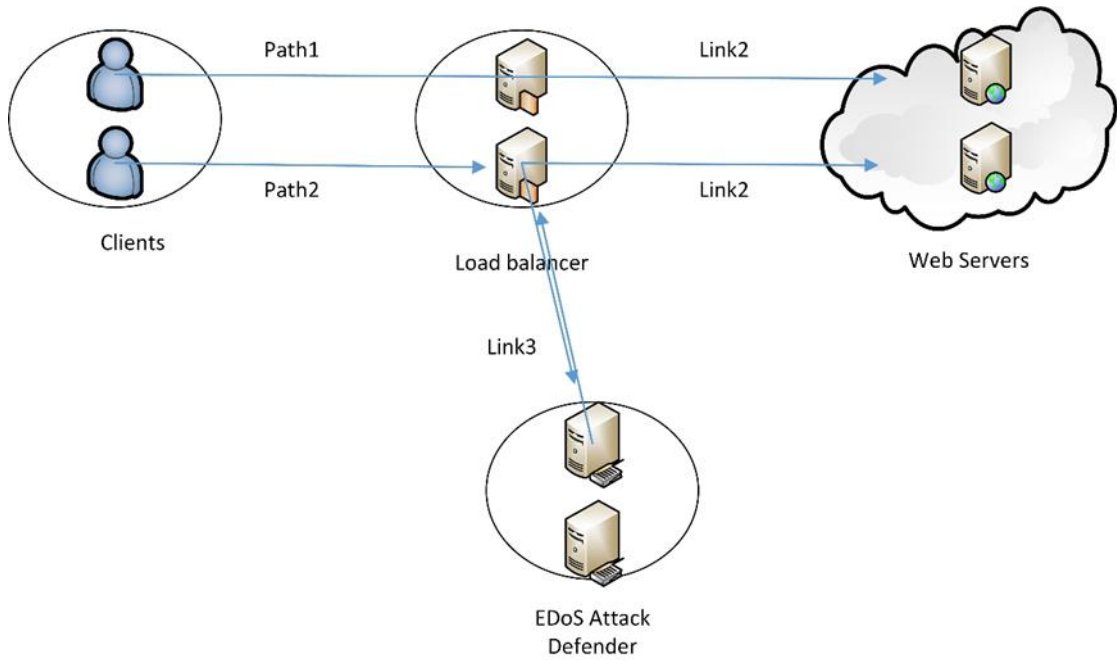


Figure 14 Traffic flow paths from sources to a destination

The delay for Path1 is calculated by using the following equation:

Mean Response Time = Delay of Load balancer + Dealy_link2 + Delay Cloud instances (web servers)

The delay for Path2 is calculated by using the following equation:

Mean Response Time = Delay of Load balancer + Dealy_link3 + Delay of EDoS Defender + Delay Cloud instances (web servers)

We have ignored the delay encountered because of the link between the source (clients) and the load balancer since we focused on the performance of the mitigation technique that starts at the load balancer.

The end-to-end response time for the cloud instances (web servers) is calculated by estimating the waiting time in the queue for every request and adding the service time.

The average Cloud instances delay for the requests is defined as:

Cloud instances delay = (total delay time in queue / total requests) + service time.

4.2.2 Computing Resources Utilization

The CPU utilization is one of the main parameters of the auto-scaling conditions. The CPU utilization is being used in our proposed mitigation technique to prove the mitigation concept. However, we could use the link utilization or other auto-scaling parameters by simply making small changes such as altering the thresholds.

Assuming that all the computing instances have the same capacity of computing power, $\mu_i = \mu$, and the arrival rate $\lambda_i = \lambda/S$ at each instance is , the mean computing utilization U is calculated as follows:

$$U = \frac{\sum_i^S \frac{\lambda_i}{S\mu}}{S} = \frac{\lambda}{S\mu}$$

However, the CPU utilization of every instance in the simulation is calculated by aggregating the instance busy period divided by the total time of simulation. The CPU utilization of the running instances is calculated by taking the average of all running instances.

4.2.3 Throughput

The throughput of the cloud system is calculated using Little's formula. The throughput of M/M/1 queuing system is $\mu \times \rho = \mu \times (\lambda/S\mu) = \lambda/S$, where μ is the service rate and ρ is the utilization for computing resources. Therefore, the average throughput for all instances is λ .

Similarly, the average throughput is calculated by multiplying the utilization of every instance calculated earlier by the arrival rate for every instance and then dividing by the total number of instances.

4.2.4 Cost

Several pricing models can be used by the cloud computing systems. Many service providers and cloud adopters provide the flexibility to optimize the customers' costs. Amazon EC2 provides customers with three different purchasing models that give them the flexibility to optimize their costs. On-Demand Instances allow customers to pay a fixed rate by the hour with no commitment. For the Reserved Instances model, customers pay a low, one-time fee and in turn receive a significant discount on the hourly charge for that instance. The Spot Instances model enables customers to bid whatever price they want for instance capacity, resulting in even greater savings if their applications have flexible start and end times.

A Cloud user has to pay for the computing resources, the network traffic volume, and for the storage service, if required. In our work, we consider the cost related to both computing usage and bandwidth usage.

The cost has been calculated based on the following equation as used by Amazon [56].

Cost of bandwidth = bandwidth price for GB \times arrival rate measured in GB/s \times Time

Cost of instances = instances price \times the number of running instances \times the average utilization running instances \times Time

Total cost = Cost of bandwidth + Cost of instances

The cost of an instance is set to \$0.115 as it is recently reported in Amazon for small on-demand instances running on the Windows operating system [56]. Regarding the cost associated with the bandwidth allocation, we have used a base price of \$0.01 per GB in/out data transferred based on the reported prices of Internet data transfer "in" and "out" of Amazon EC2 [56].

4.3 Simulator's Validation

The simulation is validated by comparing obtained simulation results with the EDoS-Shield simulation results for similar scenarios. In the first scenario, the EDoS-Shield work has considered different attack rates to show the impact of the attack on the targeted cloud service. The second scenario is for the optimal case where there is no attack targeting the cloud service. Each of the following subsections will discuss and compare the results obtained with the results presented in the EDoS-Shield work [34], [35]. In every scenario, we used different number of instances based on the attack rates. The number of instances is $6+5k$, $k = [0-20]$ based on the EDoS-Shield work [34], [35]. The EDoS-Shield work proposed the following equation to calculate the required instances:

$$S_{required} = \left\lceil \frac{\lambda_l + \lambda_m}{\mu} + 1 \right\rceil$$

Where $S_{required}$ is the required instances, μ is the utilization, λ_m is the EDoS attack rate, and λ_l is the legitimate traffic rate.

The number of instances is displayed in every graph.

4.3.1 Response Time

Figure 15 shows the comparison of the two results, the EDoS shield work and our implemented simulation. It is clear from the graph that both simulations have similar results for the response time with small differences due the randomness of the simulation.

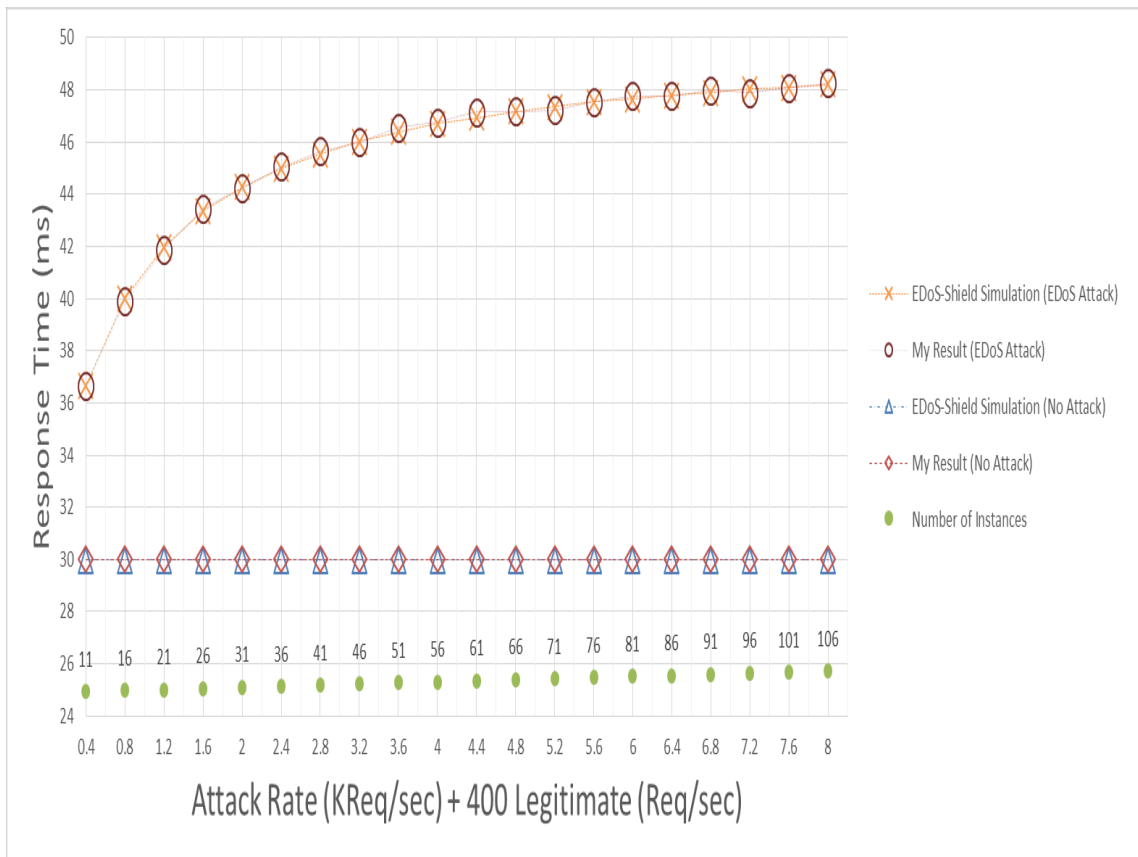


Figure 15 simulation results for EDoS Shield and our simulations, Response Time

4.3.2 Computing Resources Utilization

For the computing resources utilization, the outputs are identical as shown in Figure 16

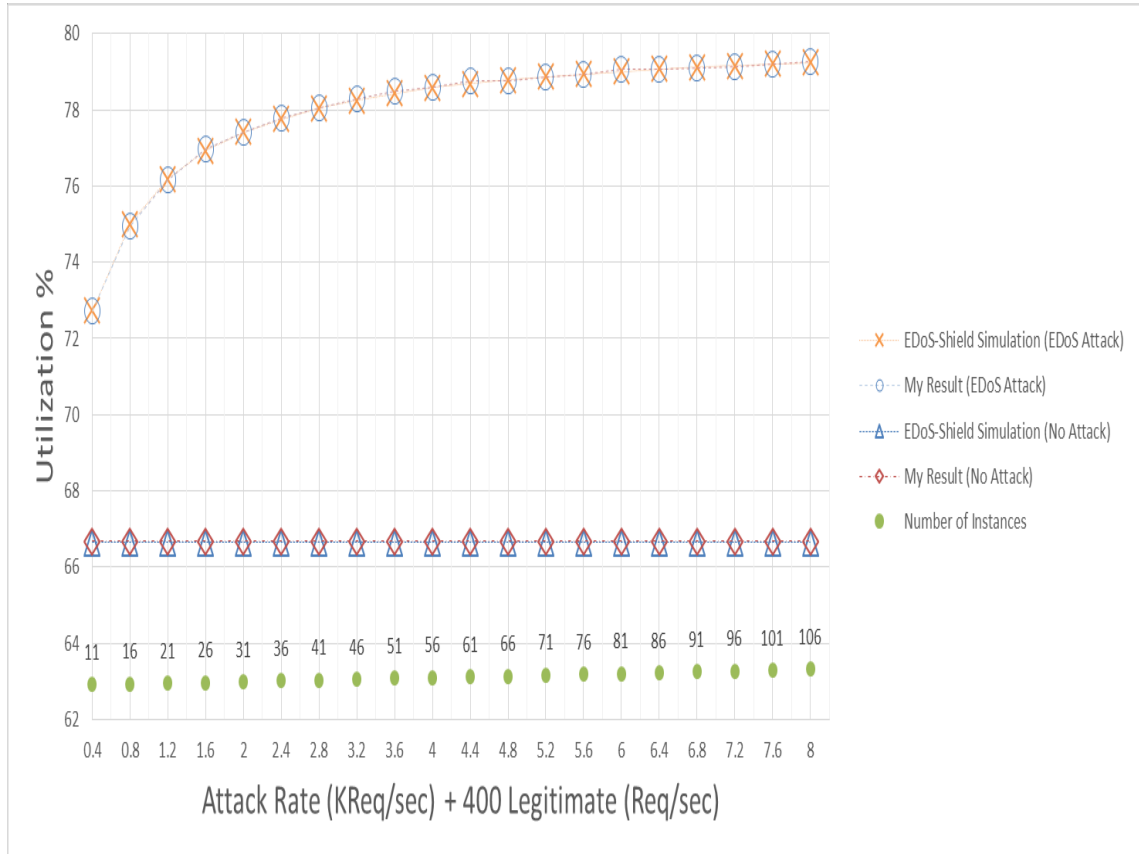


Figure 16 simulation results for EDoS Shield and our simulations, Utilization

4.3.3 Cost

Figure 17 shows the computing resources and bandwidth cost for EDoS shield simulation. Our cost results are identical to the EDoS shield simulation results.

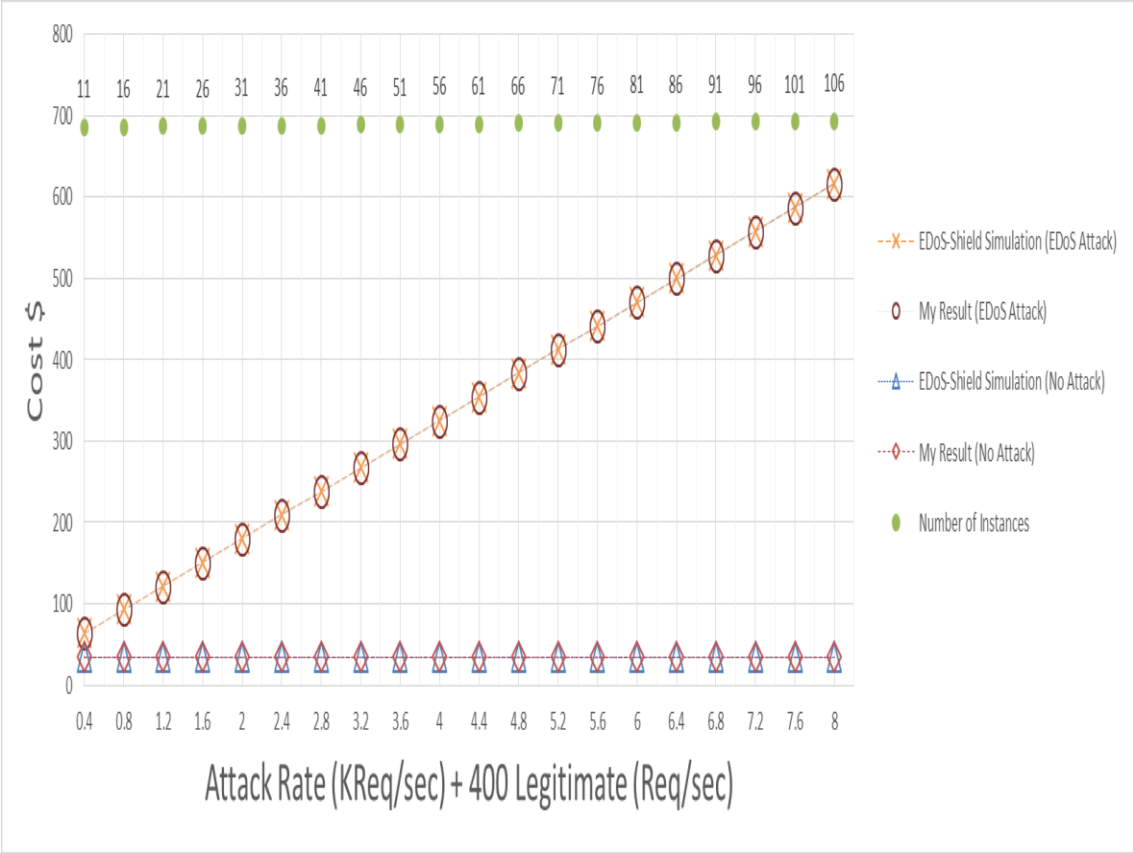


Figure 17 simulation results for EDoS Shield and our simulations, Cost

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

This chapter presents the simulation results for the proposed mitigation technique. We have conducted three experimental scenarios using the simulation model discussed in the previous chapter. In the first scenario, we have considered different arrival rates to show the overhead (if any) of the proposed mitigation technique. The second scenario is for the flash crowd mode where there is high traffic coming to the cloud service. In the third scenario, we have considered different attack rates to show the effectiveness of our proposed mitigation technique. In addition, the output of the proposed mitigation technique is compared with the EDoS-Shield simulation results in the attack mode.

5.1 Normal Mode Results

In this scenario, we used different legitimate arrival rates ranging from 400 to 8000 requests/second to state the overhead (if any) of the proposed mitigation technique. We used different number of instances based on the legitimate arrival rates. The number of instances is $8 + 6k$, $k = [0-19]$. This selection of the number of instances is based on the EDoS-Shield work [34], [35]. The EDoS-Shield work proposed the calculation of the required instances to achieve 80% utilization under the EDoS attack. In the Normal and Flash Crowd modes, we calculated the required instances to achieve 66% utilization to examine the system under normal activity. The number of instances is displayed in every graph. In this scenario, there is no attack targeting the cloud services.

5.1.1 Response Time Evaluation

Figure 18 shows the response time for the implemented mitigation technique: EDoS-Defender in the normal mode. It is compared with the response time without a mitigation technique. The results are identical because no overhead or extra processing is required by the EDoS-Defender.

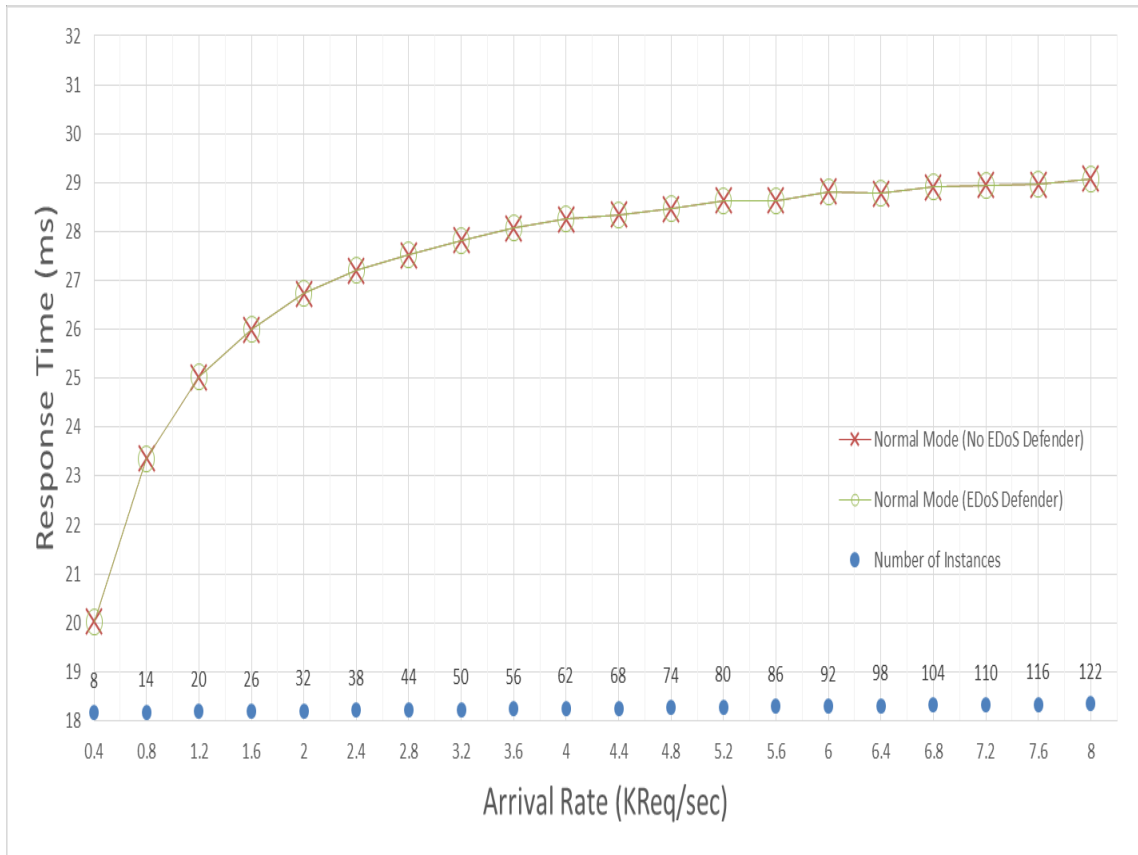


Figure 18 Response Time evaluation Normal Mode

5.1.2 Resources Utilization Evaluation

Figure 19 shows the resources utilization for the implemented mitigation technique: EDoS-Defender in the normal mode for different arrival rates. It is compared with

resources utilization with and without a mitigation technique. The results are identical because no overhead or extra processing is required by the EDoS-Defender.

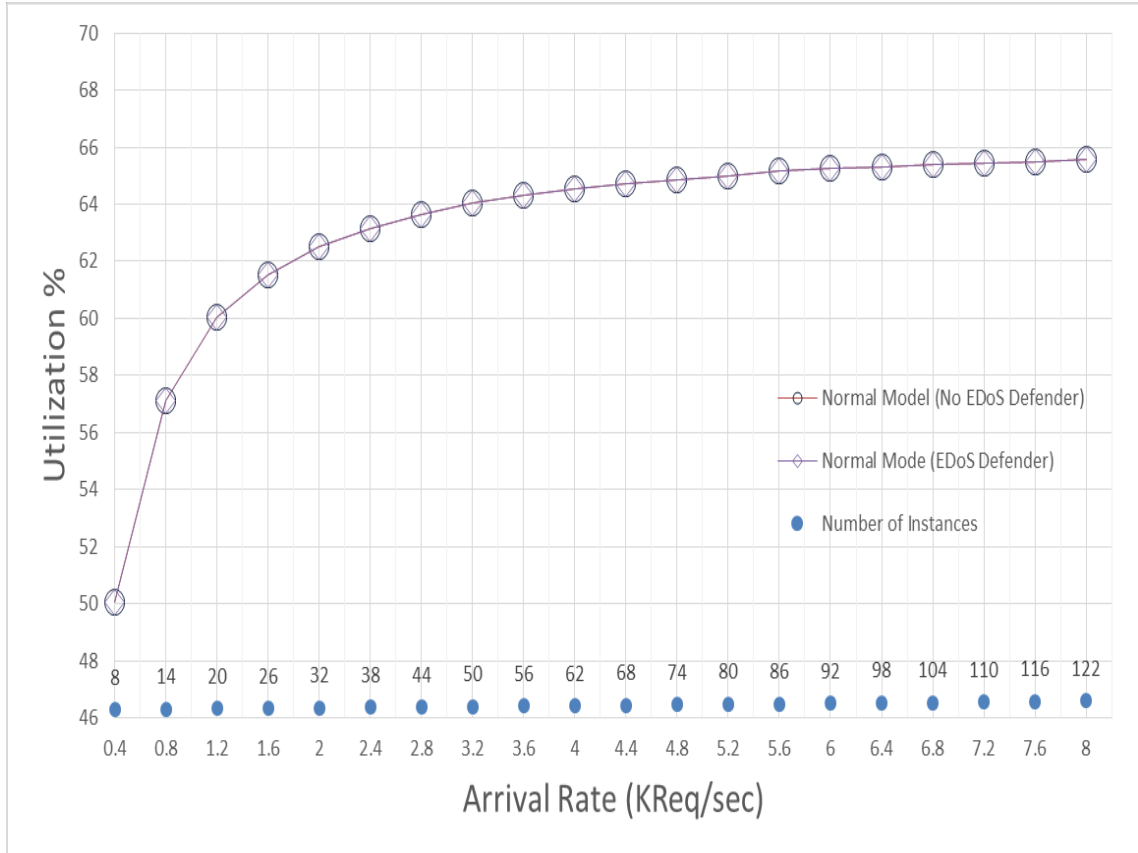


Figure 19 Resources Utilization evaluation Normal Mode

5.1.3 Cost Evaluation

Figure 20 shows the Cost evaluation for the implemented mitigation technique: EDoS-Defender in the normal mode for different arrival rates. It shows a comparison with the cost of resources without a mitigation technique. In the EDoS-Defender, the cost is a little bit higher because of the addition of the price of the large instance that is used as an EDoS-Defender instance.

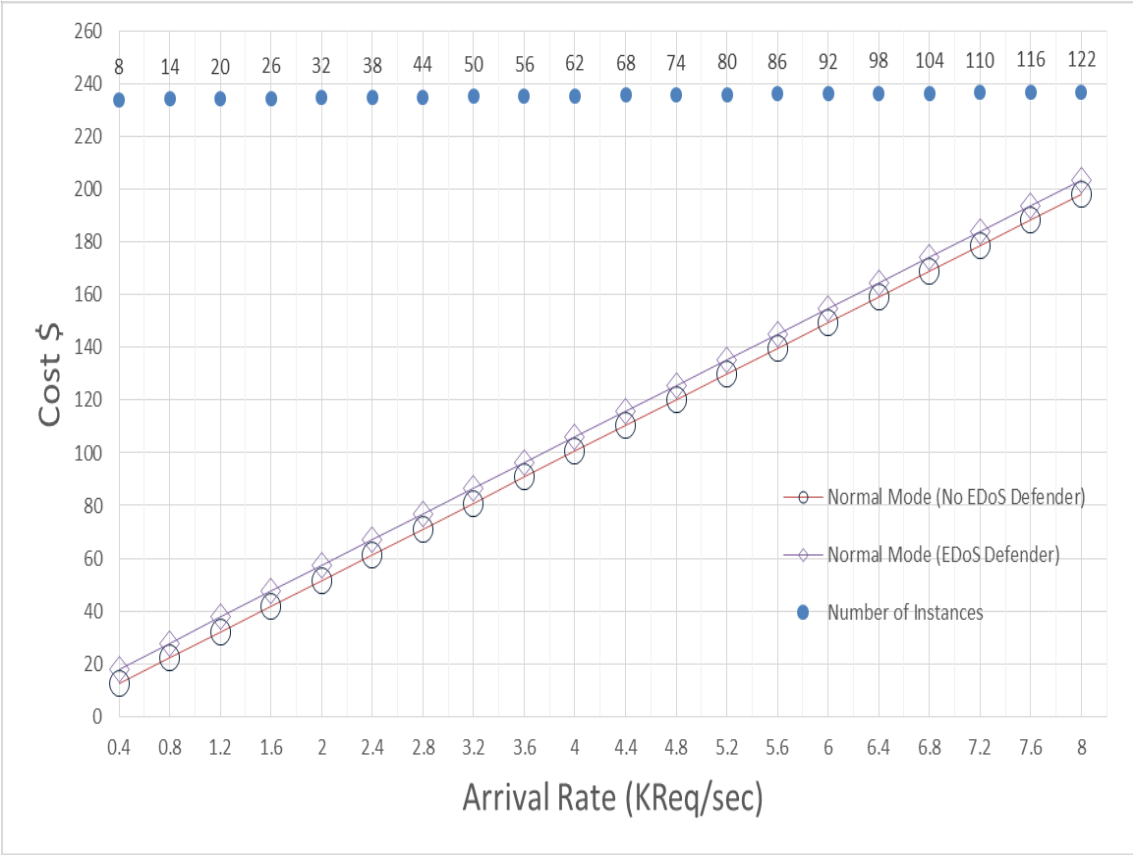


Figure 20 Cost evaluation Normal Mode

5.1.4 Throughput Evaluation

Figure 21 shows the throughput evaluation for the implemented mitigation technique: EDoS-Defender in the normal mode for different arrival rates. It is compared with the throughput of resources without a mitigation technique. The results are identical because the EDoS Attack Defender will not affect the throughput rate.

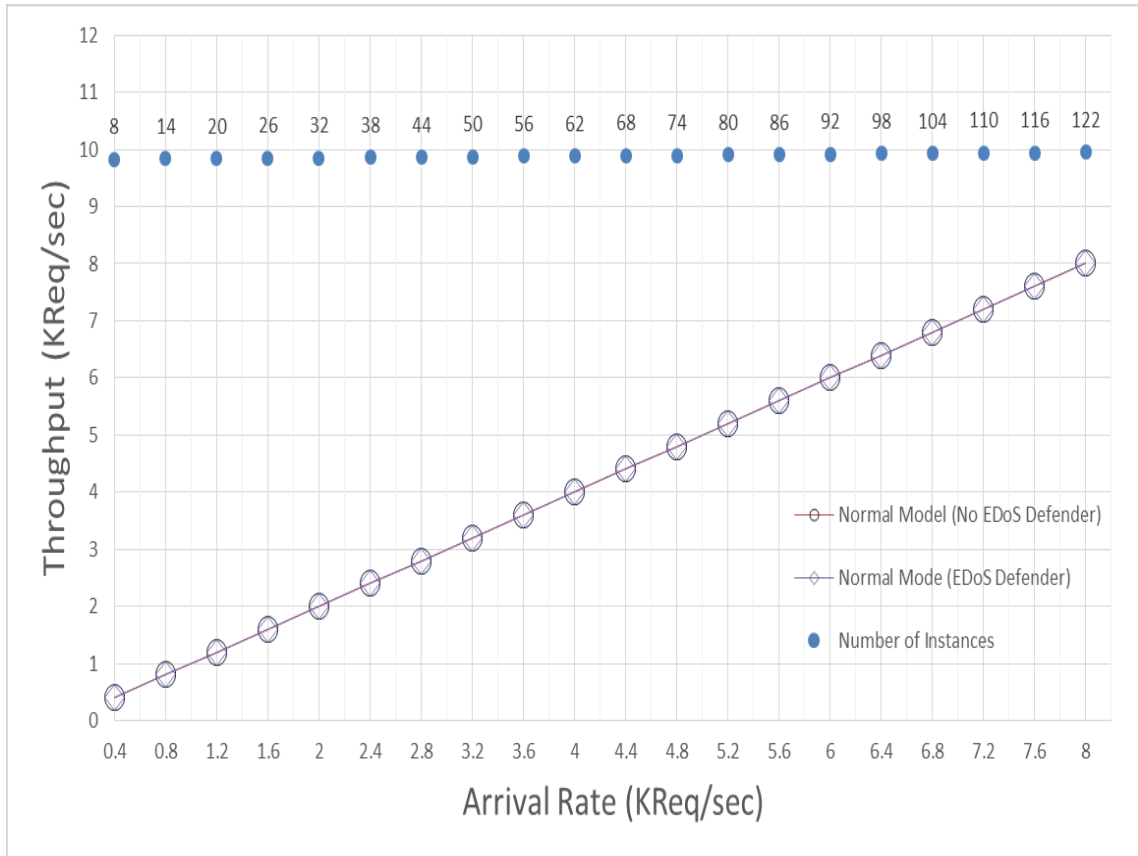


Figure 21 Throughput evaluation Normal Mode

5.2 Flash Crowd Mode Results

In this scenario, we used different legitimate arrival rates ranging from 400 to 8000 requests/second to state the overhead (if any) of the proposed mitigation technique. This mode will occur only if there is a legitimate high traffic (Flash crowd). The system will be in suspicious mode if the utilization is in the validation window, i.e., 70% to 80%. In this case, the cloud service will redirect all incoming requests to the EDoS Defender for a validation period. In this validation period, the EDoS defender will send a CAPTCHA to the clients to differentiate between Flash Crowd mode and Attack mode. In this scenario, there is no attack targeting the cloud services. In every scenario, we used a fixed number

of instances that is five instances, and then let the system auto-scale as. This auto-scaling is based on three instances per scaling up event. The number of instances is displayed in every graph.

5.2.1 Response Time Evaluation

Figure 22 shows the response time for the implemented mitigation technique: EDoS-Defender in the Flash Crowd mode. It is compared with the response time without a mitigation technique. There is a little difference because of the CAPTCHA that is sent during the validation period from the EDoS-Defender to the legitimate users. This validation period is triggered to differentiate between Flash Crowd mode and Attack mode.

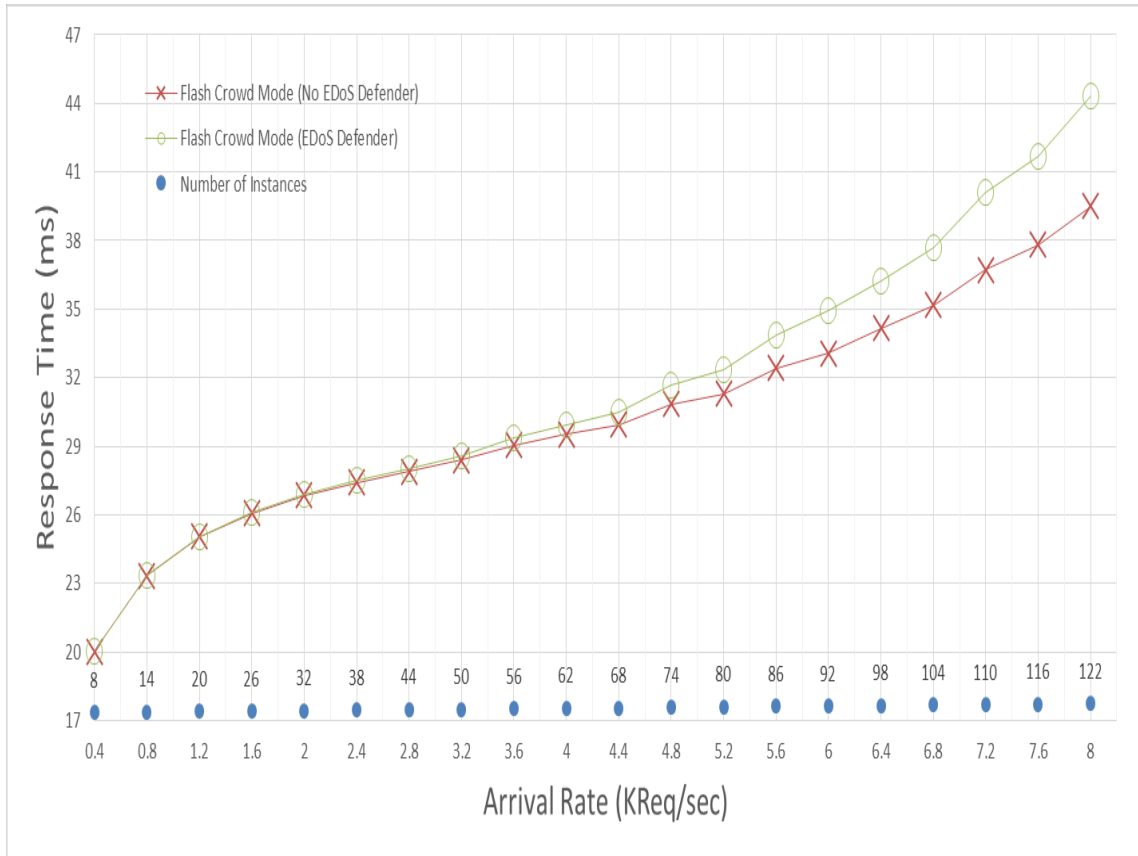


Figure 22 Response Time evaluation Flash Crowd Mode

5.2.2 Resources Utilization Evaluation

Figure 23 shows the resources utilization for the implemented mitigation technique: EDoS-Defender in the Flash Crowd mode. It is compared with resources utilization without a mitigation technique. The results are identical because the cloud service utilization is not affected by our proposed EDoS Defender mitigation technique since there are no attacks in this mode.

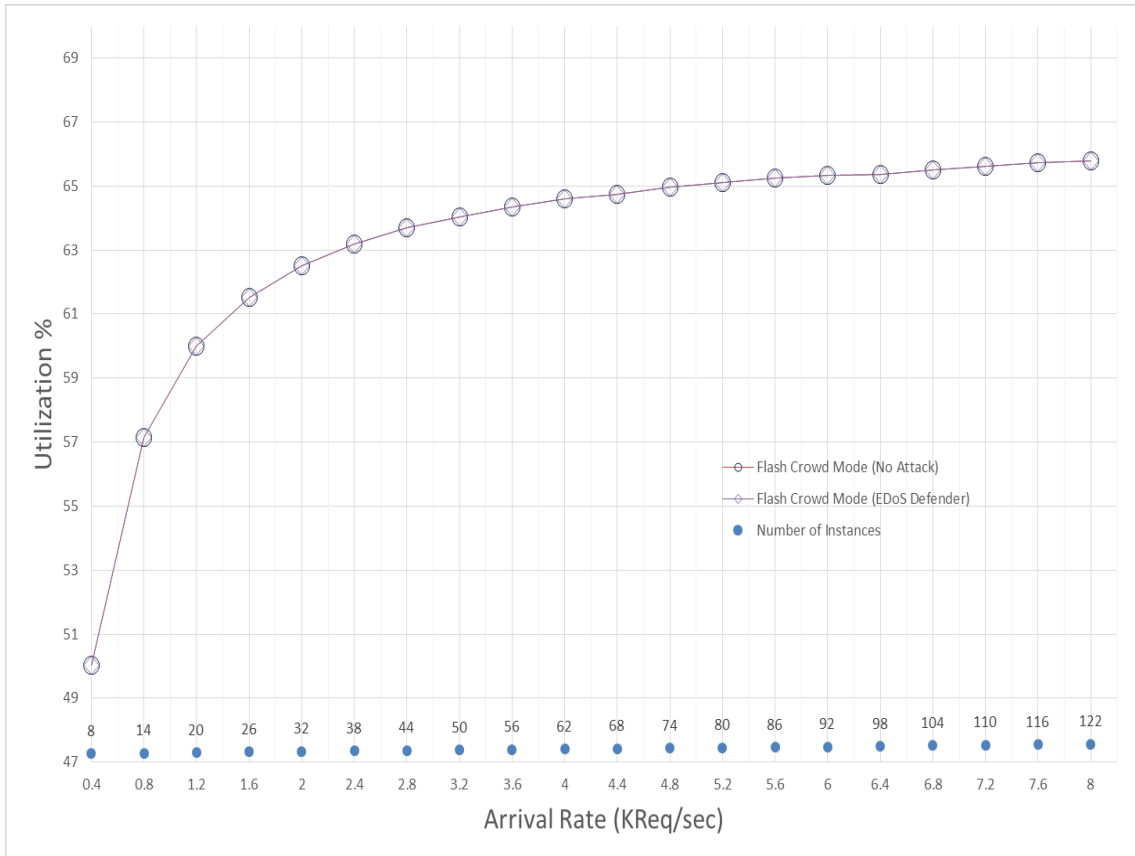


Figure 23 Resources Utilization evaluation Flash Crowd Mode

5.2.3 Cost Evaluation

Figure 24 shows the Cost evaluation for the implemented mitigation technique: EDoS-Defender in the Flash Crowd mode for different legitimate arrival rates. It is compared with the cost of resources without a mitigation technique. In the EDoS-Defender, the cost is a little bit higher because of the addition of the price of the large instance that is used as an EDoS-Defender instance.

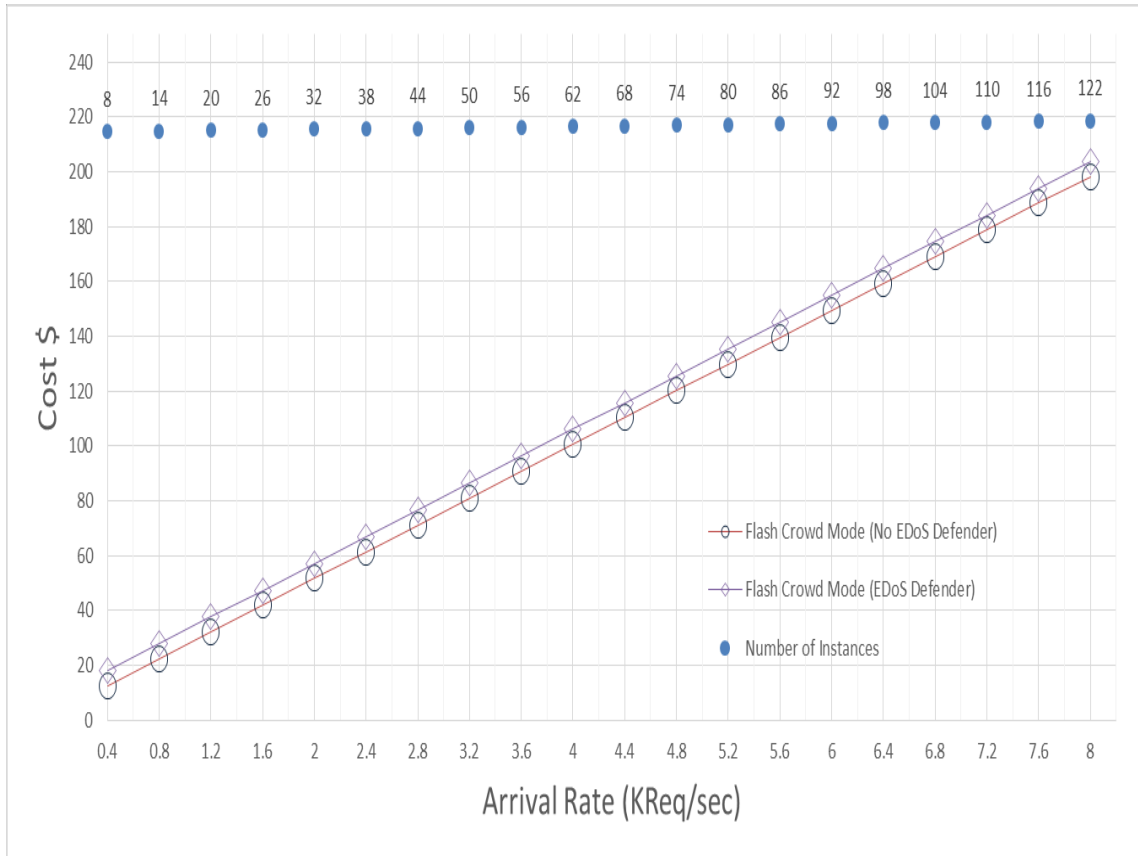


Figure 24 Cost evaluation Flash Crowd Mode

5.2.4 Throughput Evaluation

Figure 25 shows the throughput evaluation for the implemented mitigation technique: EDoS-Defender in the Flash Crowd mode for different legitimate arrival rates. It is compared with the throughput of resources without a mitigation technique. The results are identical because the EDoS Attack Defender will not affect the Throughput rate.

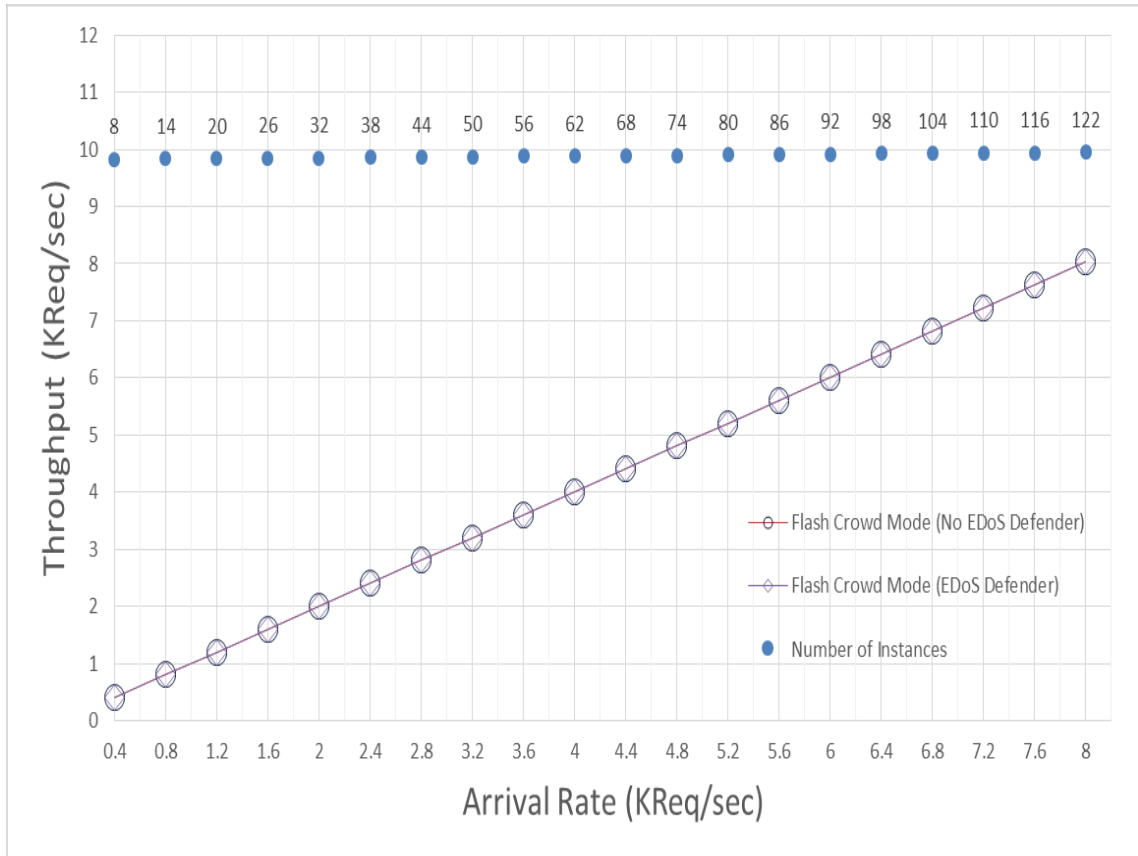


Figure 25 Throughput evaluation Flash Crowd Mode

5.3 Attack Mode Results and EDoS Shield Comparison

In this scenario, we used different attack rates ranging from 400 to 8000 requests/second to evaluate the proposed mitigation technique: EDoS-Defender under the EDoS attack. The arrival rate is assumed to follow the Poisson distribution. This mode will occur only if there is some attack traffic. The system will be in suspicious mode if the current system utilization is in the range of suspicion mode thresholds, i.e., 70%-80%. In this case, the cloud service will redirect all incoming requests to the EDoS Defender for a validation

period. In this validation period, the EDoS defender will send a CAPTCHA to the clients to differentiate between Flash Crowd mode and Attack mode. If the clients failed to respond to the CAPTCHA in this validation period, then it is considered as Attack mode. In this section, we compare the EDoS-Defender results with the results for the EDoS attack without a mitigation technique, the EDoS-Shield optimal case, the EDoS-Shield whitelist case, and the EDoS-Shield blacklist case. The EDoS-Shield optimal case refers to the EDoS-Shield when there is no spoofing IP address, i.e., all legitimate clients' IP addresses are in the whitelist and all attacker IP addresses are in the blacklist. The EDoS-Shield whitelist case refers to when the attacker spoofed an IP address that is already in the whitelist. In addition, this can occur when the attacker compromises machines behind a NAT, for which the exposed public IP addresses are already in the whitelist. Therefore, the EDoS-Shield will consider these attackers as legitimate clients. The EDoS-Shield blacklist case is used to describe the EDoS-Shield when an attacker successfully carries out IP address spoofing. This IP address was not used before to access the cloud services and as result, the IP is added to the blacklist. Therefore, the legitimate clients cannot access the cloud service because their IP addresses are already in the blacklist. Also, the attacker could compromise machines that are behind NAT protocol and add their public IP addresses to the blacklist, thus preventing the legitimate users from accessing the cloud services with these public IP addresses. Therefore, the EDoS-Shield will consider these legitimate clients as attackers. The EDoS-Shield blacklist case results are not shown in the following figures because all requests will be dropped and treated as attack requests.

5.3.1 Response Time Evaluation

Figure 26 shows the response time for the implemented mitigation technique: EDoS-Defender in the Attack mode. It is compared with the response time for the EDoS attack without a mitigation technique, the EDoS-Shield optimal case, and the EDoS-Shield whitelist case. The EDoS-Shield optimal case shows good results in comparison with the EDoS-Defender. The EDoS-Shield whitelist case shows results similar to the results of the case without a mitigation technique. The response time of the EDoS-Defender under attack starts with a low response time when the attack rate is 400 requests/sec and 400 legitimate requests/sec, because a high number of legitimate users have to answer the CAPTCHA during the validation window. The response time of the EDoS-Defender increases when the attack rates increase because of higher attack rates.

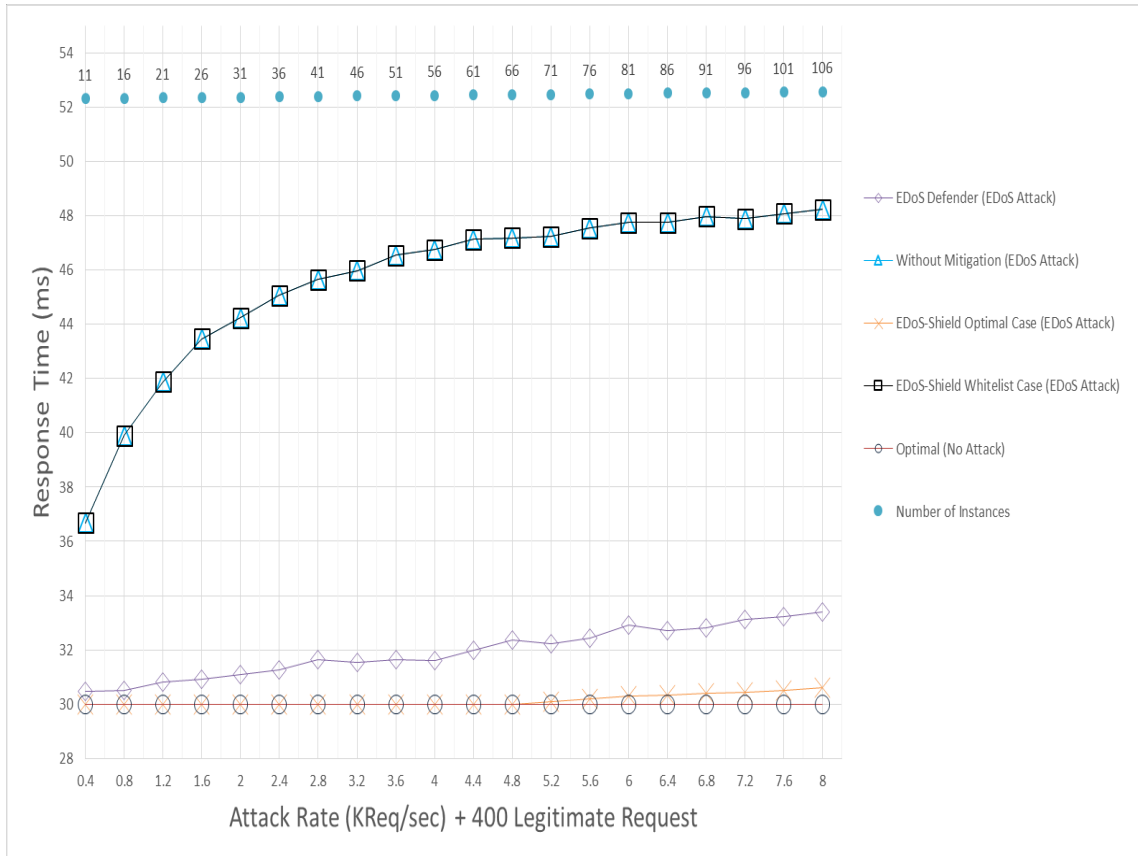


Figure 26 Response Time evaluation Attack Mode

5.3.2 Resources Utilization Evaluation

Figure 27 shows the resource utilization for the implemented mitigation technique: EDoS-Defender in the Attack mode. It is compared with resource utilization for the EDoS attack without a mitigation technique, the EDoS-Shield optimal case, and the EDoS-Shield whitelist case. The EDoS-Shield optimal case shows identical results in comparison with the EDoS-Defender, in which all attack traffic will be blocked. The EDoS-Shield whitelist case shows results similar to the results of the case without a mitigation technique. The EDoS Attack Defender’s results are close to the optimal case.

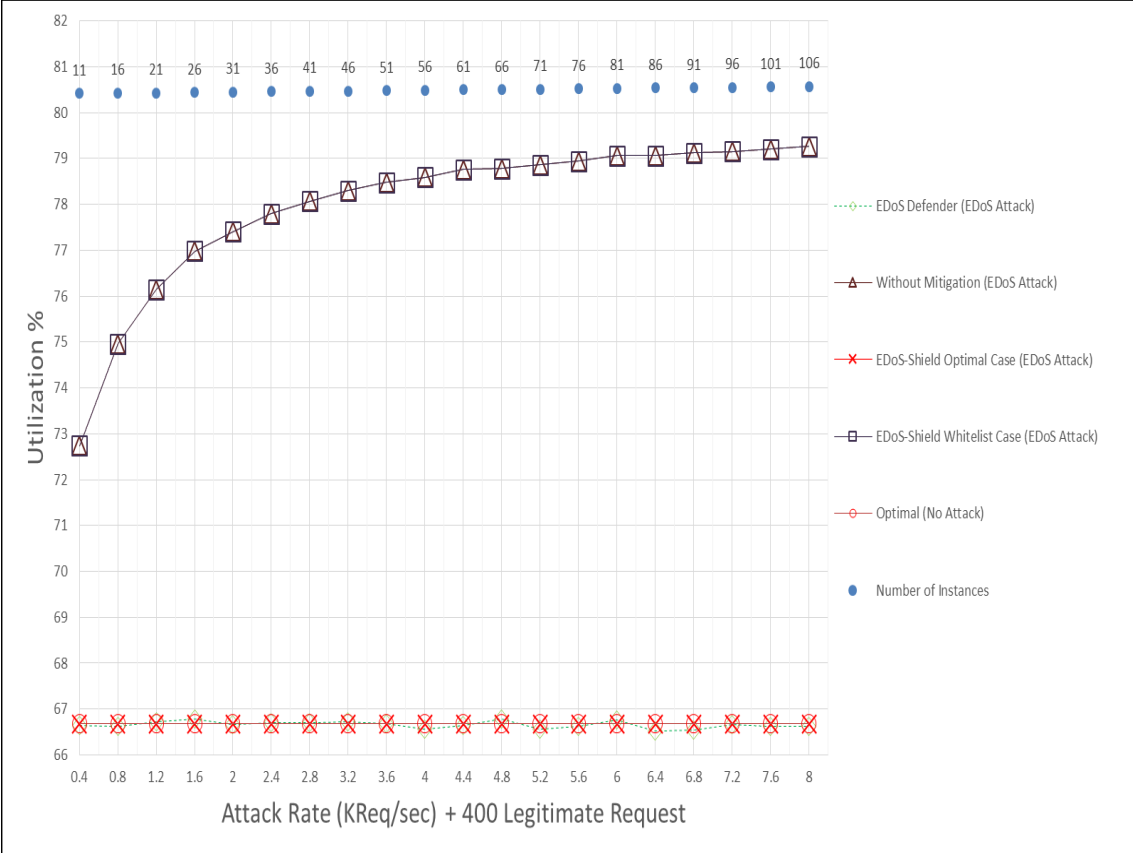


Figure 27 Resources Utilization evaluation Attack Mode

5.3.3 Cost Evaluation

Figure 28 shows the cost evaluation for the implemented mitigation technique: EDoS-Defender in the Attack mode. It is compared with the cost evaluation for the EDoS attack without a mitigation technique, and the EDoS-Shield optimal case. The EDoS-Defender shows the best cost evaluation because it uses only one large instance for the EDoS-Defender. However, the EDoS-Shield uses two large instances for the Virtual Firewall (VF) and one small instance for the Verifier Node (V-Node).

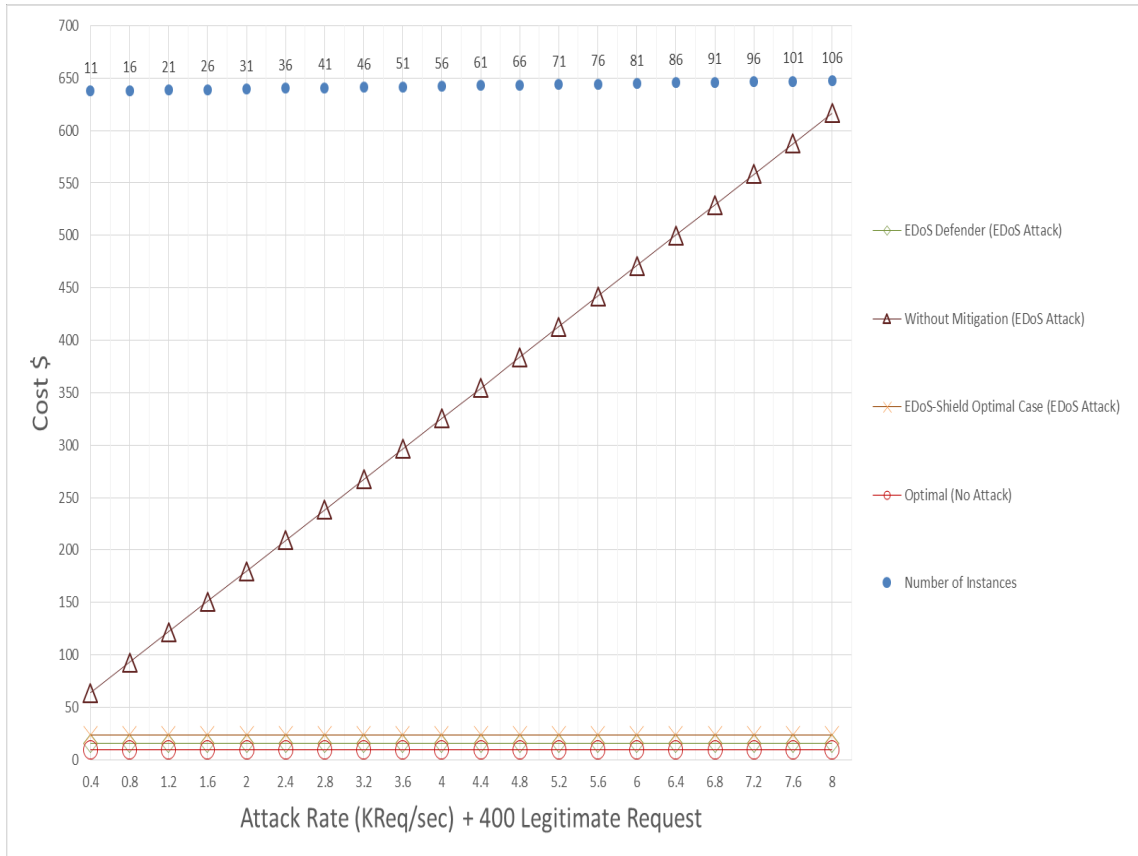


Figure 28 Cost evaluation Attack Mode

5.3.4 Throughput Evaluation

Figure 29 shows the throughput of the legitimate requests for the implemented mitigation technique: EDoS-Defender in the Attack mode. It is compared with the throughput of the legitimate requests for the EDoS attack without a mitigation technique, the EDoS-Shield optimal case, and the EDoS-Shield whitelist case. Regarding the throughput evaluation, it is expected that the throughput of the legitimate requests will not be affected by the attack rate even without applying the mitigation technique. This is because the targeted cloud service is an on-demand cloud-based. According to the nature of the cloud computing system, i.e., scalability nature, we are assuming that there are enough on-

demand cloud resources to be provisioned to the cloud instances executing the service. As a result, there is no degradation of the throughput rate of the legitimate requests in all scenarios.

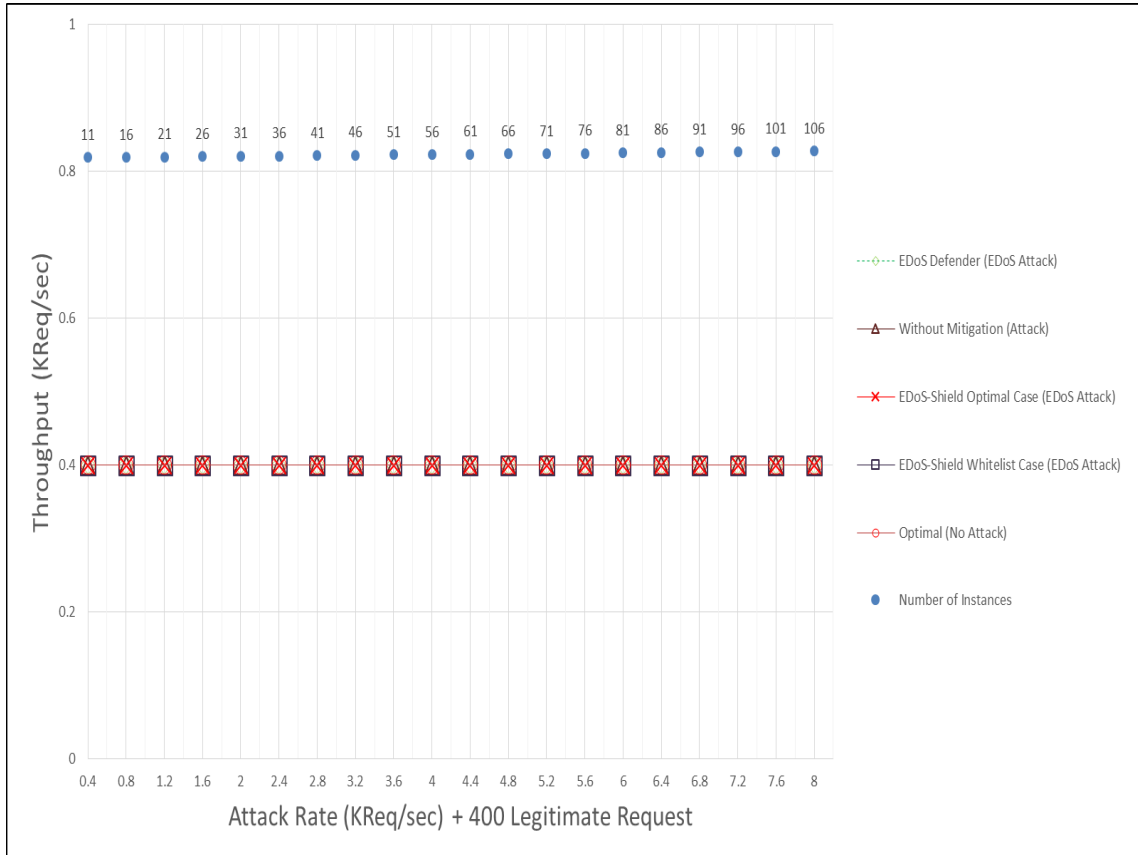


Figure 29 Throughput evaluation Attack Mode

5.4 Detailed Results

In this scenario, we used different results representation to show the effect of the EDoS Attack Defender using different parameters such as utilization versus the simulation time to see how the system behaves at different stages of the simulation time.

5.4.1 Resources Utilization Evaluation

Figure 30 shows the resources utilization for the implemented mitigation technique: EDoS-Defender in the Attack mode with 8000 requests attack rate and 400 requests legitimate rate. The system has 9 instances. At the beginning of the attack, the utilization rises to almost 100% because of the high attack rate. However, this increase is instantaneous so the auto-scaling feature is not triggered but the EDoS Attack Defender is triggered and investigates the traffic and keeps sending the Graphic Turing Test (CAPTCHA) to all new incoming requests including the legitimate and the attacker requests. The attacker will not respond to the CAPTCHA so all attack requests are dropped and the utilization is dropped as well to the baseline utilization, which is in this case 45%. Then, the EDoS Attack Defender uses the Attack Period Timer to detect if the attack has finished. The EDoS Attack Defender checks the percentage of legitimate responses to the Graphic Turing Test (CAPTCHA) every time the Attack Period Timer runs out. If the percentage is greater than 90%, then the attack has finished and the current mode will change to Normal Mode. Otherwise, the attack has not finished and the EDoS Attack Defender keeps sending CAPTCHA to all new incoming requests.

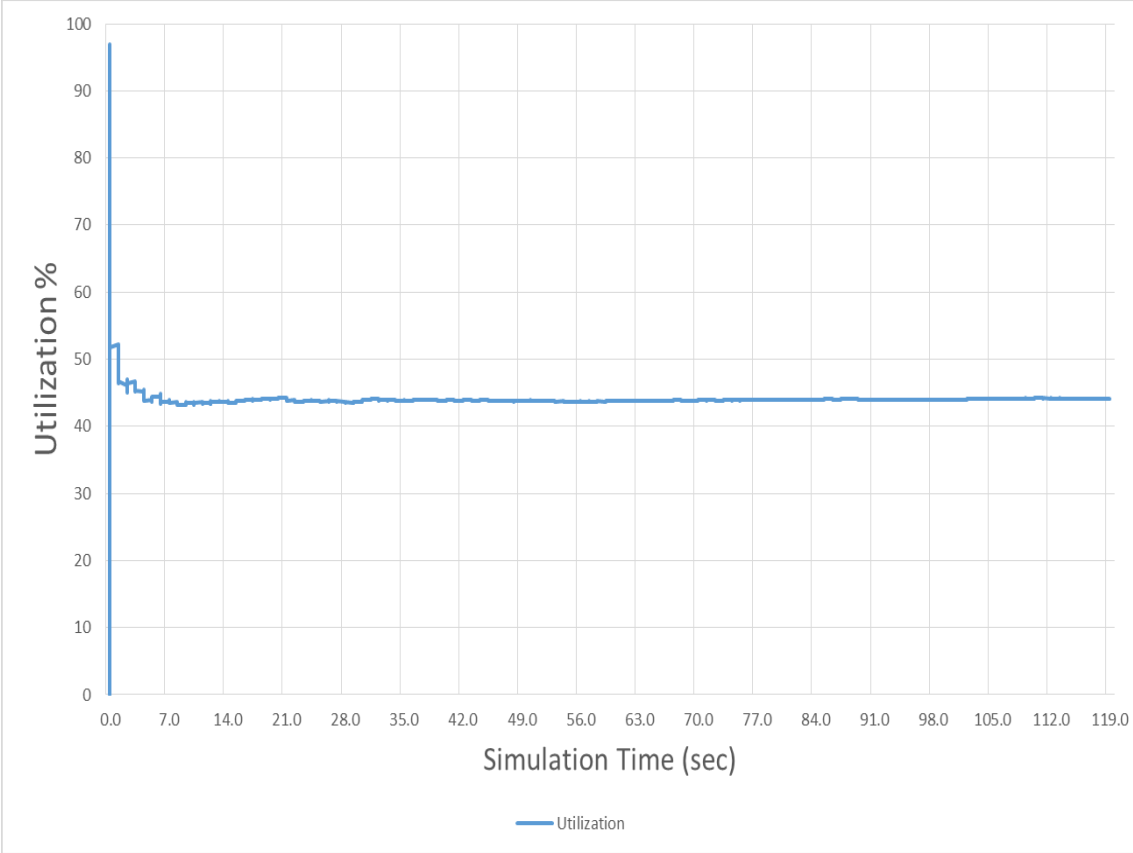


Figure 30 Resources Utilization evaluation Attack Mode

5.4.2 Response Time Evaluation Including Load Balancer Delay

In this scenario, we included the load balancer delay and link delay in the end-to-end response time result. Liu and Wee [57] reported that an Amazon EC2 instance can handle 800 Mbps when used as a load balancer. "Because the load balancer does not process the traffic, but rather, only forwards the packets, we expect the results to hold for other web applications." [57]. We assume requests with an average size of 580 bytes [58], by simple calculations, $(580 \times 8) / (800 \times 10^6)$, the average processing time (forwarding time) for a request in the load balancer is about 5.8 μ s or less. Therefore, we use a load balancer instance with 5.8 μ s service time in our experiments. Thorsten Von Eicken [59] proposed a benchmark for the load balancer on the amazon cloud environment. This test includes different existing load balancers such as HAproxy, Zeus, aiCache, and Amazon's Elastic Load Balancing service. They focused on how many requests per second the load balancer is able to handle. The result shows that Amazon Elastic load balancer could process more than 20k requests per second. The capacity of the links in the cloud infrastructure is set to 10 Gbps so as to calculate the delay regarding the link, $(580 \times 8) / (10 \times 10^9)$. The link delay is approximately equal to 0.464 μ s.

Figure 31 shows the response time for the implemented mitigation technique: EDoS-Defender in the Attack mode with the load balancer delay and the link delay.

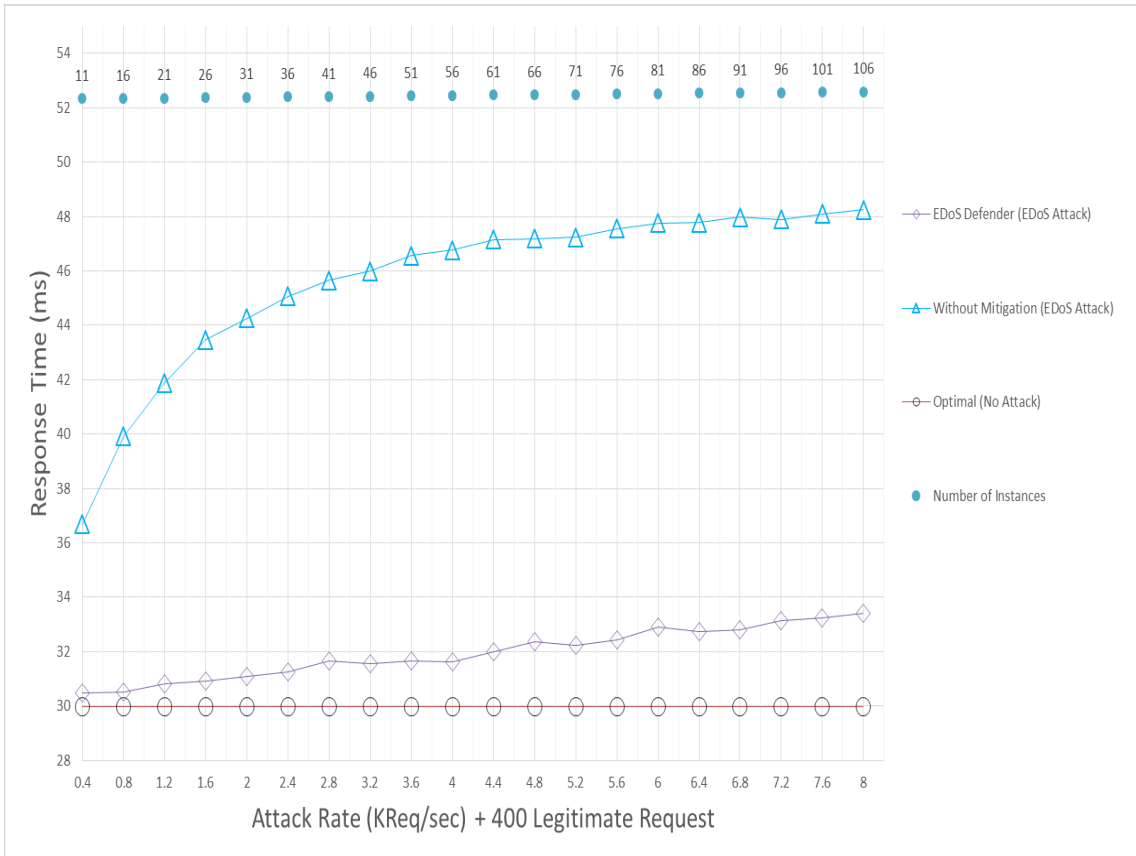


Figure 31 Response Time with Load Balancer Delay evaluation Attack Mode

5.4.3 Response Time Evaluation in Flash Crowd

In this scenario, we used five instances with a scaled up parameter instead of three instances. Figure 32 shows an improvement in the response time with a higher auto-scaling size. However, with higher auto-scaling size, we could allocate unnecessary instances that will increase the cost.

Therefore, in this thesis, the auto-scaling size is set to three and it gives reasonable CPU utilization and response time.

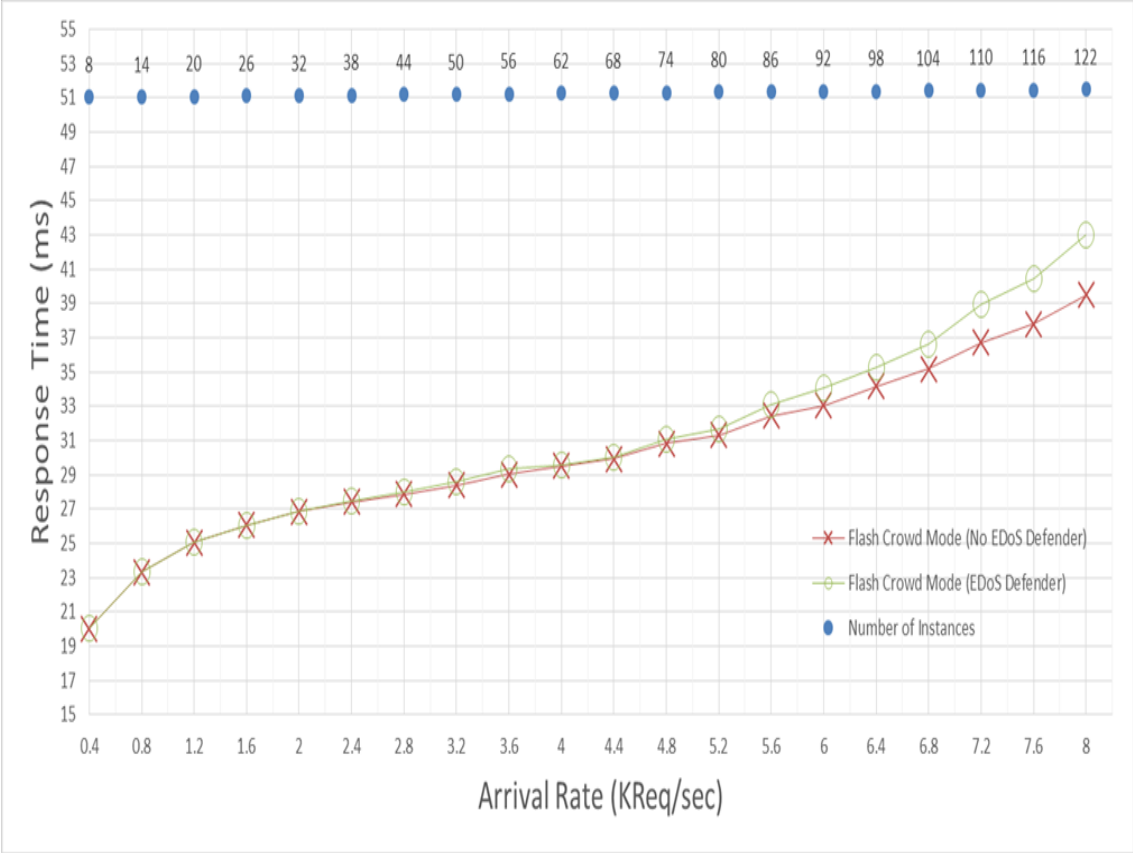


Figure 32 Response Time using five instances for scaling up Flash Crowd Mode

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter summarizes the major contribution and findings in this thesis. The main objective of this research is to design and implement a mitigation technique capable of preventing or mitigating the impact of the Economic Denial of Suitability (EDoS) attack on the cloud computing environment. In addition, this chapter states the limitations of the proposed work with possible improvements as future work.

6.1 Conclusion

Cloud computing is a promising technology. However, the security of cloud computing must be investigated deeply. EDoS attack is one of the major threats targeted towards the cloud computing environments, and which needs to be considered. In this thesis, a novel solution is presented to mitigate or prevent such attacks. The mitigation technique, namely EDoS Attack Defender, is based on reactive mitigation schemes. It is only triggered when there is suspicious traffic coming to the cloud platform. Therefore, all incoming traffic is directed to the EDoS Attack Defender instance for investigation and to validate that the traffic intensity does not exceed a defined threshold. The EDoS Attack Defender uses the Graphic Turing Test such as CAPTCHA to differentiate between legitimate and attack traffic. The performance measures show the effectiveness of our proposed mitigation technique. In addition, we compare our proposed mitigation technique with the EDoS-Shield in three cases, the optimal case, the whitelist case, the blacklist case. The comparison results show that our proposed mitigation technique is

better than the EDoS-Shield in the whitelist case and the blacklist case in terms of performance metrics. In the whitelist case, the EDoS-Shield could still allow some attackers to launch the EDoS attack. And, in the blacklist case, the EDoS-Shield could still block some legitimate users and prevent them from using the cloud services. The comparison results between the EDoS-Shield in the optimal case and our proposed mitigation technique show that the EDoS Defender is better in terms of cost by 44.3% and the EDoS-Shield is better in terms of response time by 8.80% in the worst case (8000 attack requests/second). However, the EDoS Defender will not block or prevent any legitimate users from the cloud services or allow any attackers to achieve their objective by launching an EDoS attack. Overall, the proposed mitigation technique shows promising results.

6.2 Future Work

The future work improvements will look into the following three aspects:

1. Using a smarter method to detect if the attack has finished, and considering the down-scaling feature under the attack. The first improvement will consider better ways to detect if the attack has finished, rather than using the Attack Period Timer. The second improvement will consider different types of attacks other than the DDoS attacks that could lead to EDoS attacks. The third improvement will consider the down-scaling feature while the attack is undergoing and will run some scenarios to confirm if the EDoS Attack Defender will take care of such a scenario or whether it needs further improvement.

2. To enhance the simulation, we could add different distributions of the attack and simulate different types of cloud services.
3. In addition, one of the future directions for this work is to conduct experimental implementation of the proposed mitigation technique using a test bed of a private or public cloud and compare the obtained results with our simulation results.

References

- [1] Ahmed, Mohiuddin, A. S. M. Chowdhury, M. Ahmed, and M. H. Rafee, "An Advanced Survey on Cloud Computing and State-of-the-art Research Issues.," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 201–207, 2012.
- [2] F. Gens, "New idc it cloud services survey: top benefits and challenges," *IDC Exch.*, 2009.
- [3] S. Mansfield-Devine, "Danger in the clouds," *Netw. Secur.*, vol. 2008, no. 12, pp. 9–11, 2008.
- [4] L. M. Kaufman, "Data security in the world of cloud computing," *Secur. Privacy, IEEE*, vol. 7, no. 4, pp. 61–64, 2009.
- [5] S. SUDHA and V. VISWANATHAM, "ADDRESSING SECURITY AND PRIVACY ISSUES IN CLOUD COMPUTING," *J. Theor. Appl. Inf. Technol.*, vol. 48, no. 2, pp. 708–719, 2013.
- [6] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. "O'Reilly Media, Inc.," 2009.
- [7] M. Jensen and J. Schwenk, "The accountability problem of flooding attacks in service-oriented architectures," in *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, 2009, pp. 25–32.
- [8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Futur. Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
- [9] C. Hoff, "Cloud Computing Security: From DDoS (Distributed Denial Of Service) to EDoS (Economic Denial of Sustainability)." [Online]. Available: <http://rationalsecurity.typepad.com/blog/2008/11/cloud-computing-security-from-ddos-distributed-denial-of-service-to-edos-economic-denial-of-sustaina.html>.
- [10] C. Hoff, "A Couple Of Follow-Ups On The EDoS (Economic Denial Of Sustainability) Concept." [Online]. Available: <http://rationalsecurity.typepad.com/blog/edos/>.
- [11] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, and D. Rubenstein, "Using graphic turing tests to counter automated DDoS attacks against web servers," *Proc. 10th ACM Conf. Comput. Commun. Secur. - CCS '03*, p. 8, 2003.

- [12] M. Mehra, M. Agarwal, R. Pawar, and D. Shah, "Mitigating denial of service attack using CAPTCHA mechanism," *Proc. Int. Conf. Work. Emerg. Trends Technol. - ICWET '11*, no. Icwet, p. 284, 2011.
- [13] Carnegie Mellon University, "CAPTCHA: Telling Humans and Computers Apart Automatically." [Online]. Available: <http://www.captcha.net/>.
- [14] J. Yan and A. El Ahmad, "CAPTCHA security: a case study," *Secur. Privacy, IEEE*, no. August, 2009.
- [15] L. Von Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," *Adv. Cryptology— ...*, pp. 294–311, 2003.
- [16] P. Mell and T. Grance, "The NIST definition of cloud computing," *Commun. ACM*, vol. 53, no. 6, p. 50, 2010.
- [17] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, Apr. 2010.
- [18] H. Chen and S. Li, "A queueing-based model for performance management on cloud," in *Advanced Information Management and Service (IMS), 2010 6th International Conference on*, 2010, pp. 83–88.
- [19] J. Walraevens, S. Wittevrongel, and H. Bruneel, "Performance analysis of a priority queue with session-based arrivals and its application to E-commerce web servers," *Int. J. Adv. Internet Technol.*, vol. 2, no. 1, pp. 46–57, 2009.
- [20] A. Khosravani, B. Nicholson, and T. Wood-Harper, "A case study analysis of risk, trust and control in cloud computing," in *Science and Information Conference (SAI), 2013*, 2013, pp. 879–887.
- [21] U. Shah, M. Sonar, and H. Desai, "A Concise Study on Issues Related To Security, Privacy and Trust in Cloud Services," *researchgate.net*, vol. 2013, 2013.
- [22] D. Catteddu and G. Hogben, "Cloud Computing: Benefits, Risks and Recommendations for Information Security," *European Network and Information Security Agency (ENISA)*, 2009. [Online]. Available: www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport.
- [23] S. Yu, R. Doss, W. Zhou, and S. Guo, "A general cloud firewall framework with dynamic resource allocation," in *Communications (ICC), 2013 IEEE International Conference on*, 2013, pp. 1941–1945.
- [24] M. LEMOUDDEN, N. BOUAZZA, B. EL OUAHIDI, and D. BOURGET, "A SURVEY OF CLOUD COMPUTING SECURITY OVERVIEW OF ATTACK

VECTORS AND DEFENSE MECHANISMS.,” *J. Theor. Appl. Inf. Technol.*, vol. 53, no. 2, 2013.

- [25] R. Buyya, R. N. Calheiros, and X. Li, “Autonomic cloud computing: Open challenges and architectural elements,” in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, 2012, pp. 3–10.
- [26] S. Yu, Y. Tian, S. Guo, and D. Wu, “Can We Beat DDoS Attacks in Clouds?,” pp. 1–11, 2013.
- [27] Z. Xiao and Y. Xiao, “Security and privacy in cloud computing,” *Commun. Surv. Tutorials, IEEE*, vol. 15, no. 2, pp. 843–859, 2013.
- [28] G. Mateescu and M. Vlădescu, “Identity Management Approach for Software as a Service,” in *ICSNC 2013, The Eighth International Conference on Systems and Networks Communications*, 2013, no. c, pp. 148–153.
- [29] M. Okuhara, Masayuki and Suzuki, Takuya and Shiozaki, Tetsuo and Hattori, “Fujitsu ’ s Approach to Cloud-related Information Security,” *FUJITSU Sci. Tech. J*, vol. 47, pp. 459–465, 2011.
- [30] N. Brender and I. Markov, “Risk perception and risk management in cloud computing: Results from a case study of Swiss companies,” *Int. J. Inf. Manage.*, vol. 33, no. 5, pp. 726–733, Oct. 2013.
- [31] V. K. Reddy and D. L. S. S. Reddy, “Security Architecture of Cloud Computing,” *Int. J. Eng.*, 2011.
- [32] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in Cloud,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.
- [33] S. Khor and A. Nakao, “spow: On-demand cloud-based eddos mitigation mechanism,” in *HotDep (Fifth Workshop on Hot Topics in System Dependability)*, 2009.
- [34] M. H. Sqalli, F. Al-Haidari, and K. Salah, “EDoS-Shield - A Two-Steps Mitigation Technique against EDoS Attacks in Cloud Computing,” *2011 Fourth IEEE Int. Conf. Util. Cloud Comput.*, pp. 49–56, Dec. 2011.
- [35] F. Al-Haidari, M. Sqalli, and K. Salah, “Enhanced EDoS-Shield for Mitigating EDoS Attacks Originating from Spoofed IP Addresses,” in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, 2012, pp. 1167–1174.

- [36] S. VivinSandar and S. Shenai, "Economic denial of sustainability (edos) in cloud services using http and xml based ddos attacks," *Int. J. Comput. Appl.*, vol. 41, no. 20, pp. 11–16, 2012.
- [37] Amazon, "AWS, Amazon Public Cloud." [Online]. Available: <http://aws.amazon.com/>.
- [38] M. Kumar, R. Korra, P. Sujatha, and M. Kumar, "Mitigation of Economic Distributed Denial of Sustainability (EDDoS) in cloud computing," *researchgate.net*.
- [39] X. Wang and M. Reiter, "Mitigating bandwidth-exhaustion attacks using congestion puzzles," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 257–267.
- [40] V. D. Gligor, "Guaranteeing access in spite of distributed service-flooding attacks," in *Security Protocols*, 2005, pp. 80–96.
- [41] M. Naresh Kumar, P. Sujatha, V. Kalva, R. Nagori, a. K. Katukojwala, and M. Kumar, "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service," *2012 Fourth Int. Conf. Comput. Intell. Commun. Networks*, pp. 535–539, Nov. 2012.
- [42] W. Alosaimi and K. Al-Begain, "A New Method to Mitigate the Impacts of the Economical Denial of Sustainability Attacks Against the Cloud," in *Proceedings of the 14th Annual Post Graduates Symposium on the convergence of Telecommunication, Networking and Broadcasting (PGNet)*, 2013, pp. 116–121.
- [43] W. Alosaimi and K. Al-Begain, "An Enhanced Economical Denial of Sustainability Mitigation System for the Cloud," in *2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies*, 2013, pp. 19–25.
- [44] NSFOCUS, "NSFOCUS Mid-Year DDoS Threat Report," 2013.
- [45] B. Prince, "DDoS Attacks Occur on Average Every 2 Minutes, Security Firm Finds," *Secutiy Week*. [Online]. Available: <http://www.securityweek.com/ddos-attacks-occur-average-every-2-minutes-security-firm-finds>.
- [46] Q. Chen, R. Mehrotra, and A. Dubeyy, *On state of the art in virtual machine security*. 2012.
- [47] R. Shea and J. Liu, "Understanding the impact of denial of service attacks on virtual machines," in *Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*, 2012, p. 27.

- [48] C. Metz, "DDoS attack rains down on Amazon cloud," *Regist. Online Artic.* http://www.theregister.co.uk/2009/10/05/amazon_bitbucket_outage, 2009.
- [49] T. Karnwal, T. Sivakumar, and G. Aghila, "A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack," *2012 IEEE Students' Conf. Electr. Electron. Comput. Sci.*, pp. 1–5, Mar. 2012.
- [50] H. Beitollahi and G. Deconinck, "FOSeL: Filtering by Helping an Overlay Security Layer to Mitigate DoS Attacks," in *2008 Seventh IEEE International Symposium on Network Computing and Applications*, 2008, pp. 19–28.
- [51] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: human-based character recognition via Web security measures.," *Science*, vol. 321, no. 5895, pp. 1465–8, Sep. 2008.
- [52] W. D. Kelton and A. M. Law, *Simulation modeling and analysis*. McGraw Hill Boston, MA, 2000.
- [53] N. Singh, S. P. Ghreera, and P. Chaudhuri, "Denial of Service Attack: Analysis of Network Traffic Anomaly using Queuing Theory," *arXiv Prepr. arXiv1006.2807*, 2010.
- [54] H. Liu, "A new form of DOS attack in a cloud and its avoidance mechanism," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, 2010, pp. 65–76.
- [55] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of queueing theory*. John Wiley & Sons, 2013.
- [56] AWS, "Amazon EC2 Pricing." [Online]. Available: <http://aws.amazon.com/ec2/pricing/>. [Accessed: 01-Dec-2013].
- [57] H. Liu and S. Wee, "Web server farm in the cloud: Performance evaluation and dynamic architecture," in *Cloud Computing*, Springer, 2009, pp. 369–380.
- [58] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings of INET*, 1998, vol. 98, pp. 21–24.
- [59] T. Von Eicken, "Benchmarking Load Balancers in the Cloud," *Cloud Management Blog*, 2010. [Online]. Available: <http://www.rightscale.com/blog/cloud-management-best-practices/benchmarking-load-balancers-cloud>. [Accessed: 01-Dec-2013].

Vitae

Name : Mohammed Yahya Alkaff
Nationality : Yemeni
Date of Birth : 9/16/1980
Email : myalkaff@hotmail.com
Address : Yemen Hadramout Mukalla
Academic Background : Computer Engineer

Educational & Qualifications

- ❖ Jan2011- Jan2014: M.Sc. in Computer Networks, KFUPM, Dhahran – KSA.
GPA: 3.750/4.0 (Excellent).
- ❖ Sep2000-Jun2005: B.Sc. in Computer Engineering, Yarmouk University –Jordan.
GPA: 3.176/4.0 (Very Good).

Experience

- ❖ Sep2012-Jan2014: Microsoft Student Partner (KSA).
- ❖ Jan2011-Jan2014: Graduate Student in KFUPM (KSA).
- ❖ Jul2006-Oct2010: Field Engineer (LWD & MWD Engineer) in Schlumberger (Yemen).
- ❖ Mar2006-Jun2006: Intelligent Network Engineer in HUAWEI (Yemen).
- ❖ Sep2004-Feb2005: Internship - Web Developer in Fact Applied Computer Technology (Jordan).

Training Courses

- ❖ Entrepreneurship Institute located in KFUPM Dhahran (Saudi Arabia)
Venture Concept Competition, Date 19/11/2012 – 21/4/2013.
- ❖ Cisco Networking Academy located in KFUPM Dhahran (Saudi Arabia)
CCNA Exploration: Network Fundamentals Course, Date 17/9/2012 - 14/11/2012.
- ❖ Schlumberger Mussafah Abu Dhabi (UAE)
Drilling Optimization course, Date 31/3/2009: 2/4/2009.
- ❖ Schlumberger Training Centre SLC (USA)
Advance Interpretation course, Date 25/2/2008: 9/3/2008.
- ❖ Schlumberger Training Centre UTC (UK)
Engineer1 course, Date 25/9/2006: 2/12/2006.
- ❖ HUAWEI Training Center MENA located in Cairo (Egypt)
Intelligent Network Basic Architecture course, Date 19/3/2006 - 20/4/2006.
- ❖ IT Institute located in Amman
RHCSA (Red Hat Certified System Administrator) courses, Date 3/7/2005 - 9/8/2005.
- ❖ Microsoft IT Academy Program located in Yarmouk University (Jordan)
MCSE (Microsoft Certified System Engineer) courses, Date 28/6/2004 - 11/4/2005.

Certifications

- ❖ Microsoft Certified Professional (MCP) in Configuring Windows 8.
- ❖ Cisco Certified Network Associate (CCNA) Exploration: Network Fundamentals.
- ❖ Drilling Optimization from Schlumberger.
- ❖ Advance Interpretation from Schlumberger.
- ❖ Eng1 certified from Schlumberger.
- ❖ Intelligent Network Basic Architecture (HUAWEI Company).
- ❖ Microsoft Certified Professional (MCP) in Microsoft Windows XP Professional.
- ❖ Microsoft Certified Professional (MCP) in Microsoft Windows 2003 Server.
- ❖ Microsoft Certified Professional (MCP) in Installing, Configuring, and Administering Microsoft ISA Server 2000.

Published Application

- ❖ 18 Windows phone applications
<http://www.windowsphone.com/en-us/store/search?q=alkaff>

Research Publication

- ❖ Mohammed Yahya Alkaff and Mohammed H. Sqalli, “Design and Deployment of a Cloud Computing Platform for Testing Attacks,” The Fourth Scientific Conference for Students of Higher Education in the K.S.A. (SSC4), Makkah, Saudi Arabia, April 29- May 2, 2013. <http://www.youtube.com/watch?v=4mmsg1CN2do>

Achievement

- ❖ Third winner in the Venture Concept Competition and awarded by the Rector of KFUPM, Date 21/4/2013.