EVALUATING THE EDOS-SHIELD MITIGATION TECHNIQUE USING AN EXPERIMENTAL TESTBED

BY

Saeed Omar Alsowail

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 **Saeed Omar Alsowail** under the direction of 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. Mohammed H. Sqalli
(Advisor)

for

Dr. Basem M. Al-Madani
Department Chairman

Marwan

Dr. Marwan H. Abu-Amara
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Khaled H. Salah
(Member)

19/3/14

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

I dedicate this work to my parents, my wife, my son Ahmed, my brothers and sisters.

Thank you for supporting me along the way.

Without your help, I could not have completed this work.

ACKNOWLEDGMENTS

All thanks and praises to ALLAH, the Almighty, for giving me the patience throughout this work, and the ability to complete it.

I would like to express my deep appreciation to my research supervisor, **Dr. Mohammed Sqalli**, for guiding me throughout the research. I also would like to thank the members of my thesis committee, **Dr. Marwan Abu-Amara**, and **Dr. Khaled Salah**, for their help in accomplishing this work. Thanks also to **Dr. Zubair Baig** for giving me nice comments.

I would like to thank King Fahd University of Petroleum and Mineral, and Hadhramout Establishment for Human Development for giving me the opportunity to complete my post graduate studies.

Finally, I would like to say thanks to my family and my friend for their continuous support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Full Name	**:**	**Saeed Omar Saeed Alsowail**
Thesis Title	**:**	**Evaluating The EDoS-Shield Mitigation Technique Using an Experimental Testbed**
Major Field	**:**	**Computer Networks**
Date of Degree	**:**	**December, 2013**

Cloud computing is recently considered as one of the most significant IT trends. Many large organizations are interested in cloud computing because of its elasticity, pay per use, and other benefits that it provides. However, even with all of its great advantages, the security of cloud computing is still in its infancy. Many new attacks have been developed especially for the cloud, and the Economic Denial of Sustainability (EDoS) attack is one of them. EDoS attacks target the bill of the cloud solution adopter to cause economic loss. In this work, we first present a taxonomy of the attacks that target cloud computing. Then, we provide a survey for the different types of attacks that can result in an EDoS attack. We also propose a comprehensive taxonomy of the EDoS attacks. Finally, we study the EDoS-Shield mitigation technique and evaluate its effectiveness in blocking EDoS attacks using an experimental testbed, which is the major contribution of this work.

ملخص الرسالة

الاسم الكامل: سعيد عمر سعيد الصويل

عنوان الرسالة: تقييم أسلوب الحماية EDoS-Shield عن طريق الاختبارات العملية

التخصص: شبكات الحاسوب

تاريخ الدرجة العلمية: ديسمبر 2013

تعد الحوسبة السحابية من مواضيع تكنولوجيا المعلومات الأكثر رواجاً في الآونة الأخيرة. أثارت الحوسبة السحابية اهتمام كبرى الشركات لمرونتها، وميزة الدفع المالي بحسب الاستخدام، اضافةً الى المميزات الأخرى التي تمتاز بها الحوسبة السحابية. ولكن على الرغم من كل مزايا الحوسبة السحابية فإن أمنها يعتبر في مراحله المبكرة. لقد صُممت الكثير من الهجمات الالكترونية خصيصاً لانتهاك أمن الحوسبة السحابية. نركز في هذا البحث على أحد أنواع هذه الهجمات، والذي يستهدف استنزاف الجانب المالي للمستفيد من الحوسبة السحابية، وهو ما يعرف بهجوم "Economic Denial of Sustainability (EDoS)". في هذا البحث، نقوم أولاً بتصنيف الهجمات الالكترونية المستهدِفة للحوسبة السحابية، ثم نقدم دراسةً عن الهجمات التي قد تتسبب في الخسارة المالية (EDoS) للمستفيد من خدمات السحابة الالكترونية. وفي هذه الدراسة أيضاً تصنيف لأنواع هجمات EDoS. وفي الأخير، نقوم بدراسة أحد أساليب الحماية من هجمات EDoS، والذي يعرف بـ EDoS-Shield، وذلك بتطبيقه واجراء اختبارات عملية عليه لمعرفة مدى فعاليته في صد هجمات EDoS، وهو الهدف الأساسي لهذا البحث.

CHAPTER 1

INTRODUCTION

Cloud computing is considered one the most significant IT topics today. Many large organizations are interested in cloud computing because of its elasticity, pay per use, and other benefits that it provides. However, before getting the full benefit of the cloud, there are some issues that have to be resolved first [1]. According to the International Data Corporation (IDC), security is considered the greatest challenge of cloud computing [2]. Hence, cloud computing security has become a major field of study [3, 4, 5].

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are two well-known security threats in current networks. They intend to make a service unavailable to end users by exhausting its computing or network resources. Christofer Hoff defined a new threat that can affect the cloud by transforming a conventional DDoS attack to an Economic Denial of Sustainability (EDoS) attack in the cloud [6]. In this case, the EDoS attack can be achieved by sending a large amount of undesired traffic towards the cloud to exploit its elasticity. The cloud adopter will allocate resources to process this undesired traffic. As a result, the adopter will be charged for processing this undesired traffic. This will lead to large-scale service withdrawal or bankruptcy.

In this work, we study the EDoS-Shield which is a mitigation technique used to block the EDoS attack targeting cloud computing [15]. The main contribution of this work is to evaluate the effectiveness of EDoS-Shield mitigation technique by implementing it using an experimental cloud computing testbed. In addition, we provide a taxonomy for the attacks that can target the cloud and a comprehensive survey of the different types of attacks that can result in EDoS attacks when applied to cloud computing. Moreover, we present a taxonomy for DDoS attacks, since they are considered the main form of attacks that can result in EDoS attacks. We also provide a comprehensive taxonomy for the EDoS attacks in cloud computing.

1.1 Research Objectives

Regardless of its great advantages, the security of cloud computing is still in its infancy. Many new attacks have been developed especially for the cloud. The Economic Denial of Sustainability (EDoS) attack is one in which the attacker targets the bill of the cloud solution adopter to cause economic loss. The ultimate objective of this research is to implement the EDoS-Shield mitigation technique and test its effectiveness to prevent EDoS attacks in clouds. Through this research, a comprehensive taxonomy of EDoS attacks will also be proposed. The primary objectives of this research are:

- Study the EDoS attacks and explore their effect on clouds.
- Explore the existing mitigation techniques used to block EDoS attacks on clouds.
- Propose a comprehensive taxonomy for the EDoS attacks.
- Setup an experimental testbed for a cloud.
- Implement the EDoS-Shield mitigation technique on the testbed.
- Test the effectiveness of the EDoS-Shield in mitigating EDoS attacks.

1.2 Main Contribution

The main contributions of this work are the following:

- A taxonomy of attacks in cloud computing.
- A comprehensive survey of the attacks that can result in EDoS attacks when applied to cloud computing.
- A comprehensive taxonomy for the EDoS attacks.
- Implementation of the EDoS-Shield mitigation technique and evaluation of its effectiveness.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we review the work achieved in the literature to cover the security issues of the cloud computing. Next, we study the EDoS attack and provide a taxonomy of its different type in Chapter 3. In Chapter 4, the testbed setup and the steps followed to perform the experiments are discussed. The results of the experiments are presented and discussed in Chapter 5. In Chapter 6, we modify the testbed to make it close to real-life. The same experiments are repeated using this testbed, and the results are discussed. Finally, the work presented in the thesis is concluded and the future work is discussed in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

According to the International Data Corporation (IDC), security is considered the greatest challenge of cloud computing [2]. Gartner, an information technology research and advisory company, listed a number of security risks of cloud computing that an organization should consider when moving to a cloud computing solution [23]. The Cloud Security Alliance (CSA) also published a report that lists their view of the top threads to cloud computing [24]. The security risks mentioned in both reports were taken seriously by many researchers.

Che et al. [25] surveyed the well-known security models of the cloud computing, including the cloud multi-tenancy model of the National Institute of Standards and Technology (NIST), the cloud risk accumulation model of CSA, Jerico Formu’s cloud cube model, and the mapping model of cloud security and compliance. They also studied the security strategies to protect the cloud from the perspectives of the customer, the service provider, and the government.

Gruschka and Jensen [26] proposed a taxonomy for the attacks on the services of the cloud. In their taxonomy, they classified the attacks with respect to the notion of the surfaces of the attack of the participants of the cloud computing. They found that there can be six attack surfaces in cloud computing: service-to-user, user-to-service, cloud-to-service, service-to-cloud, cloud-to-user, and user-to-cloud. They gave real-world examples to prove the efficiency of their classification.

Khorshed et al. [27] surveyed the literature for the concerns about the security of cloud computing. They investigated the most critical threats and their suggested solutions in the literature. They also discussed the different challenges in implementing solutions to those threats.

Grobauer et al. [28] gave a definition for a cloud specific vulnerability. Based on the definition that they propose, they provided a survey about the cloud-specific vulnerabilities.

Subashini and Kavitha [29] surveyed the security issues of the cloud computing based on the service delivery models. They reviewed the security issues in the Software as a Service (SaaS) model, the Platform as a Service (PaaS) model, and the Infrastructure as a Service (IaaS) model.

Jensen et al. [30] gave in depth explanation for the various technical security issues in cloud computing. They provided real-world examples about the security problems in the cloud. They also discussed the threats that can target the cloud, and discussed some of the possible countermeasures.

Bhadauria and Sanyal [21] conducted a survey about the security threats in the different levels of the cloud architecture. They also discussed the security issues in the cloud deployment models. Further, they compared the strengths and limitations of several existing security schemes.

Jangra and Bala [22] also surveyed the literature for the vulnerabilities, attacks, and security challenges in the cloud computing environment.

Vaquero et al. [8] analyzed the risks involved with multitenancy in cloud computing. They reviewed the literature for related risks and the proposed solutions to these risks. They also grouped the main attacks in relevant to the threats presented by [24].

In the literature, there are a small number of researches that focus on the EDoS attack and attempted to find a mitigation technique for it.

In the next section, we present a taxonomy of the attacks in cloud computing. Then, we discuss the DDoS attack and its relationship with the EDoS attack.

2.1 Taxonomy of Security Attacks in Cloud Computing

Cloud computing security is one of the major challenges that prevent large business organizations from adopting the cloud solution for their businesses. In addition to the attacks that are specific to the cloud, almost all the attacks that apply to any regular network can be applicable to the cloud [7].

The attacks on the cloud can be classified based on service delivery models (SaaS, Paas, and IaaS), but many attacks can fall in more than one category. So, in addition to the classification of the cloud attacks using the service delivery models, we classify them based on the cloud hierarchy level targeted by the attack. In Figure 2.1, we classify the attacks that target the cloud security into three categories: virtualization level attacks, application level attacks, and network level attacks. Table 2.1 shows a classification for the most popular attacks based on the categories illustrated in Figure 2.1. Not all of these attacks are specific to cloud computing. Most of them are applicable to both regular computer networks and the cloud computing environment.

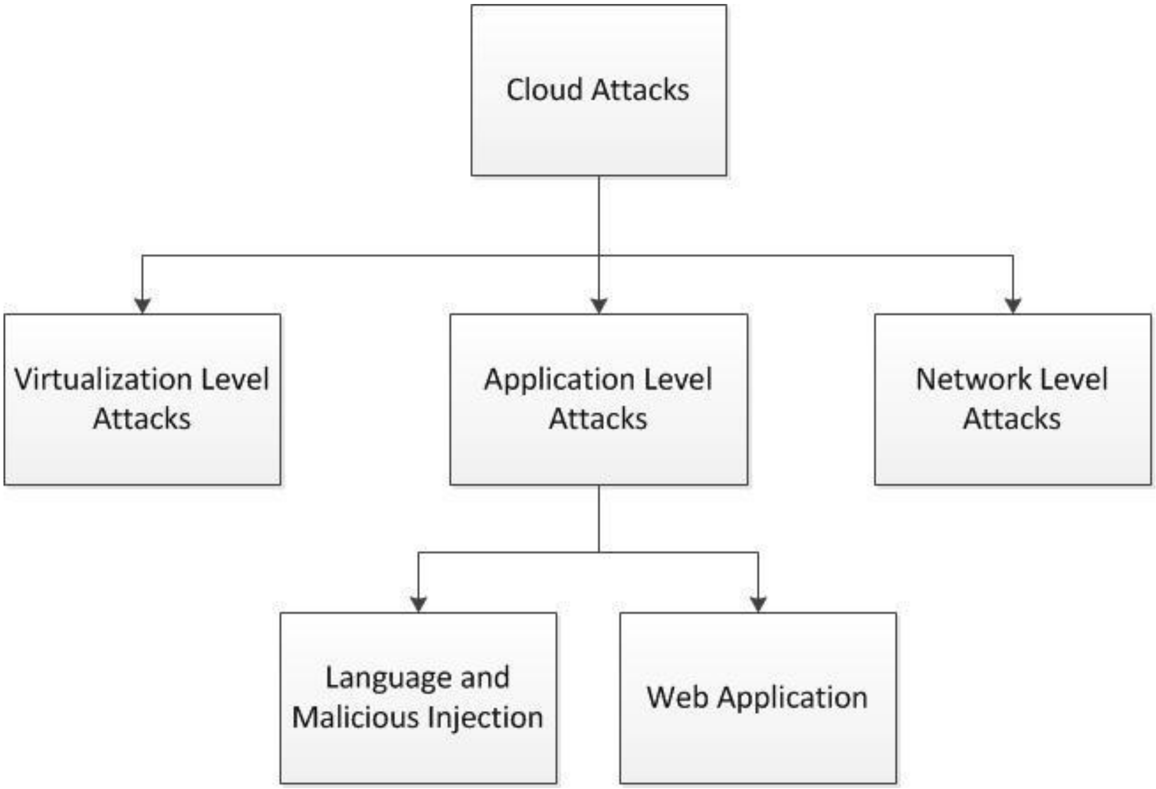

Cloud Attacks
Virtualization Level Attacks
Application Level Attacks
Network Level Attacks
Language and Malicious Injection
Web Application

Figure 2.1: A Taxonomy for The Cloud Security Attacks

Table 2.1: Classification of Popular Security Attacks Based On the Cloud Attacks Taxonomy

	Virtualization and Infrastructure Level Attacks	**Application Level Attacks**		**Network Level Attacks**
		Language and Malicious Injection	Web Application Attacks	
IaaS	Side channel attack. Timing channel attack. Cross-VMs attack*. Indirect Denial of Service attack. Covert Channel Attacks.	-	-	Eavesdropping MITM Attack. Replay Attack. Impersonation Attack.* DNS Cache Poisoning Attack. Sniffer Attacks. Byzantine Failure.* BGP Prefix hijacking. IP Address Reuse Attack.
PaaS	Cross-VMs attack*. Blue Pill attack.	Buffer Overflow Attack.* Backdoor and Debug Options.*	-	DDoS Sybil Attack. Impersonation Attack.* Byzantine Failure.*
SaaS	-	Buffer Overflow Attack.* XML Signature Wrapping Attack. Trojan horse / Malware. Backdoor and Debug Options.* Hidden Field Manipulation Attack. Metadata Spoofing Attacks.	SQL injection Attack. Cross-Site-Scripting (XSS): Stored or Reflected. Cookie Poisoning. CAPTCHA Breaking. DDoS URL Guessing Attack. Phishing Attack.	-

** : Attacks classified under more than one delivery model.*

2.1.1 Virtualization and Infrastructure Level Attacks

In cloud computing, the security of the hypervisor, which is also called the virtual machine monitor (VMM), is very critical. A hacker who could compromise the hypervisor will have the privileges that would enable him to control all the virtual machines that reside on this hypervisor. In addition to compromising the hypervisor, an attacker can also use a malicious virtual machine to attack and compromise virtual machines from this layer, i.e., hypervisor. Below are the well-known attacks that an attacker can use in this layer.

1) Covert Channel Attacks: A covert channel attack refers to any attack that establishes a communication between two processes which are not supposed to communicate at all. An attacker may use covert channels to enable his virtual machine to communicate with a legitimate machine in unauthorized way [8].

2) Side Channel Attack: A side channel attack is defined as any attack that uses the information of the physical implementation of the security algorithm. In side channel attacks, the attacker monitors the behavior of the physical characteristics of the security system, such as the power consumption and the timing information [8]. An attacker may benefit from the fact that different parts of the secret key will have different CPU timing. Based on this timing information, the attacker might be able to reconstruct the secret key. The side channel attacks that use the timing information are usually referred to as “timing channel attacks”.

3) Cross-VM Attacks: In cloud computing, virtual machines of different users may reside on the same physical host in order to maximize the utilization of the physical resources. The coexistence of virtual machines on the same physical host can allow an attacker who has access to one of these virtual machines to gain information from the other virtual machines. Ristenpart et al. [9] showed how this attack can be performed.

4) Blue Pill Attack: Blue Pill attack is a rootkit that creates a thin hypervisor between the original hypervisor and the guest operating system. This hypervisor will intercept anything coming from the guest OS and will respond to these requests using

fake replies. Rutkowska [10], the designer of this rootkit, claims that the guest OS has no way to detect this rootkit.

5) Indirect DoS Attacks: Jensen et al. [11] showed that the distributed denial of service attacks on a virtual machine in the cloud may result in indirect effect to the other untargeted virtual machines that reside on the same server. This is because large distributed denial of service attacks can consume much of the cloud resources.

2.1.2 Application Level Attacks

Unlike the virtualization and infrastructure level attacks, application level attacks target the applications used in cloud computing. Since the cloud services are accessed through the web, almost all the attacks that are used in regular web applications are applicable to cloud computing. Application level attacks can be further classified to language and malicious injection attacks, and web application attacks. Language and malicious injection attacks target the weaknesses in the programming languages and protocols. Web application attacks target the weaknesses of the web services. Application level attacks may either target the end user of the cloud services, or target the cloud solution adopter itself. A brief description is given below for buffer overflow, back door and debugs options, XML signature wrapping, and SQL injection attacks since they are quite popular attacks. The details of these attacks and the other attacks in this category can be found in [21] and [22].

1) Buffer Overflow Attack: in this attack, the attacker can cause the web application of the cloud adopter to execute arbitrary code by sending to it some crafted input. For

example, buffer overflow can be used to crash a program by putting it into an infinite loop, which will consume many resources [33].

2) Backdoor and Debug Options: A backdoor is used to allow an attacker to access a VM without authentication. Debug options are used to re-test the program and can be used by an attacker to access the VM without authentication.

3) XML Signature Wrapping: An attack in which the body of a SOAP message is moved to its header and a new malicious body is created. The attacker uses the new body to do malicious operations [11].

4) SQL Injection Attack: It is an attack in which harmful code is sent and executed in the database. The execution of this code can lead to serious problems like accessing sensitive information [21].

2.1.3 Network Level Attacks

Like any remote service, cloud computing is accessed using a network. Networks are vulnerable to many different types of attacks that may result in disastrous problems to the cloud adopter and/or the end user of the cloud. Replay attack and DNS cache poisoning are chosen as examples of the attacks of the network level category. More information about these attacks and the other network level attacks can be found in [21] and [22].

1) Replay Attack: An attack in which the attacker saves old messages sent to the victim and sends them again after a period of time [22]. These messages may include instructions that require much processing and hence require more computing resources.

2) DNS Cache Poisoning Attack: An attack in which DNS mapping is altered in a DNS server. This can harm the victim in different ways, including flooding it with large volumes of traffic that is intended to other servers.

After discussing the categories of the different types of attacks that can target the cloud at any level of the infrastructure hierarchy, the next section discusses the DDoS attacks and their relationship to EDoS attacks.

2.2 Distributed Denial of Service (DDoS) Attack

A Denial of Service (DoS) attack is used to deny legitimate users of a service from using that service [12]. Distributed Denial of Service (DDoS) is an attack that targets the availability of a system using multiple nodes controlled by the attack perpetrator [13]. A traditional DDoS attack is transformed to an EDoS attack when applied to the cloud [1, 6, 14, 15, 16]. DDoS attacks are considered the most popular EDoS attacks in cloud computing since DDoS attacks intend to consume as much resources as possible. Because of this tight relationship between EDoS and DDoS attacks, we reviewed the literature to cover the different types of DDoS attacks. Figure 2.2 presents a comprehensive taxonomy of the DDoS attacks. It covers the classifications proposed by [13, 17, 18, 19, 20]. This section gives a brief discussion for the higher levels in this taxonomy. More information can be found in [13, 17, 18, 19, 20].

1. Architecture

The architecture of a DDoS attack defines the type of machines used in the attack, how they are controlled by the attack perpetrator, how they communicate the attack commands, and how the actual attack is performed. Based on its architecture, the DDoS system can be classified into agent-handler, reflector, and IRC-based DDoS attack [18].

2. Degree of Automation

The degree of automation describes how interactive the attack perpetrator should be in order to compromise machines and to send the attack commands to these machines. Based on the degree of automation, DDoS attacks can be manual, semi-automatic, or automatic. The semi-automatic and the automatic attacks can further be classified based on the host scanning strategy, propagation mechanism, and vulnerability scanning strategy [17].

3. Attack Dynamics

Based on the attack dynamics, a DDoS attack can be continuous or variable. In continuous DDoS attacks, the attack rate is the same all the time. In variable DDoS attacks, the attack can start with a low rate, and then increase over time; or it can fluctuate from low to high and vice versa. The variable rate gives the attack more chances of not being discovered [17, 19].

4. Exploited Vulnerability

The DDoS attacks on a specific target may cause bandwidth depletion or resource depletion. The bandwidth depletion DDoS attacks consume all the available bandwidth of a target machine making it inaccessible by legitimate users. The resource depletion DDoS attacks consume the resources of the target machine so that they will be unavailable for legitimate users [13, 18, 20].

5. Persistence of Agent Set

DDoS attacks can be classified based on the persistence of agent set into constant set and variable set. In constant set attack, all the agents execute the attack simultaneously. All the agents attack at the same time and stop at the same time. In variable set DDoS attack, groups of agents will be activated to start the attack at the same time while the other groups are off. After a period of time, the attacking agent groups will be deactivated and the other groups will start over [17].

6. Impact on the Victim

Based on the impact on the victim, DDoS attacks can be classified into disruptive attacks and degrading attacks. Disruptive Attacks are those that cause the target machine to crash. Degrading Attacks consume resources of the target machine, making it unavailable to legitimate users or very slow in responding to them [17, 18, 19].

7. Source IP Address Validity

It is necessary for the attacking machine to change its IP in the source field in order to prevent any trace back operations. DDoS attacks can be classified based on the source IP address validity into valid source IP address attacks and spoofed IP address attacks [17].

8. Victim Type

Based on the type of the victim targeted by the attack, DDoS attacks can target an application, resource, host, infrastructure, or network.

9. Possibility of Characterization

DDoS attacks can be either characterizable or non-characterizable. Characterizable DDoS attacks target a specific protocol or application, and can be recognized using the IP address and the transport header values. Non-characterizable attacks use combinations of different protocols in the attack packets to consume the bandwidth of the target [17, 18].

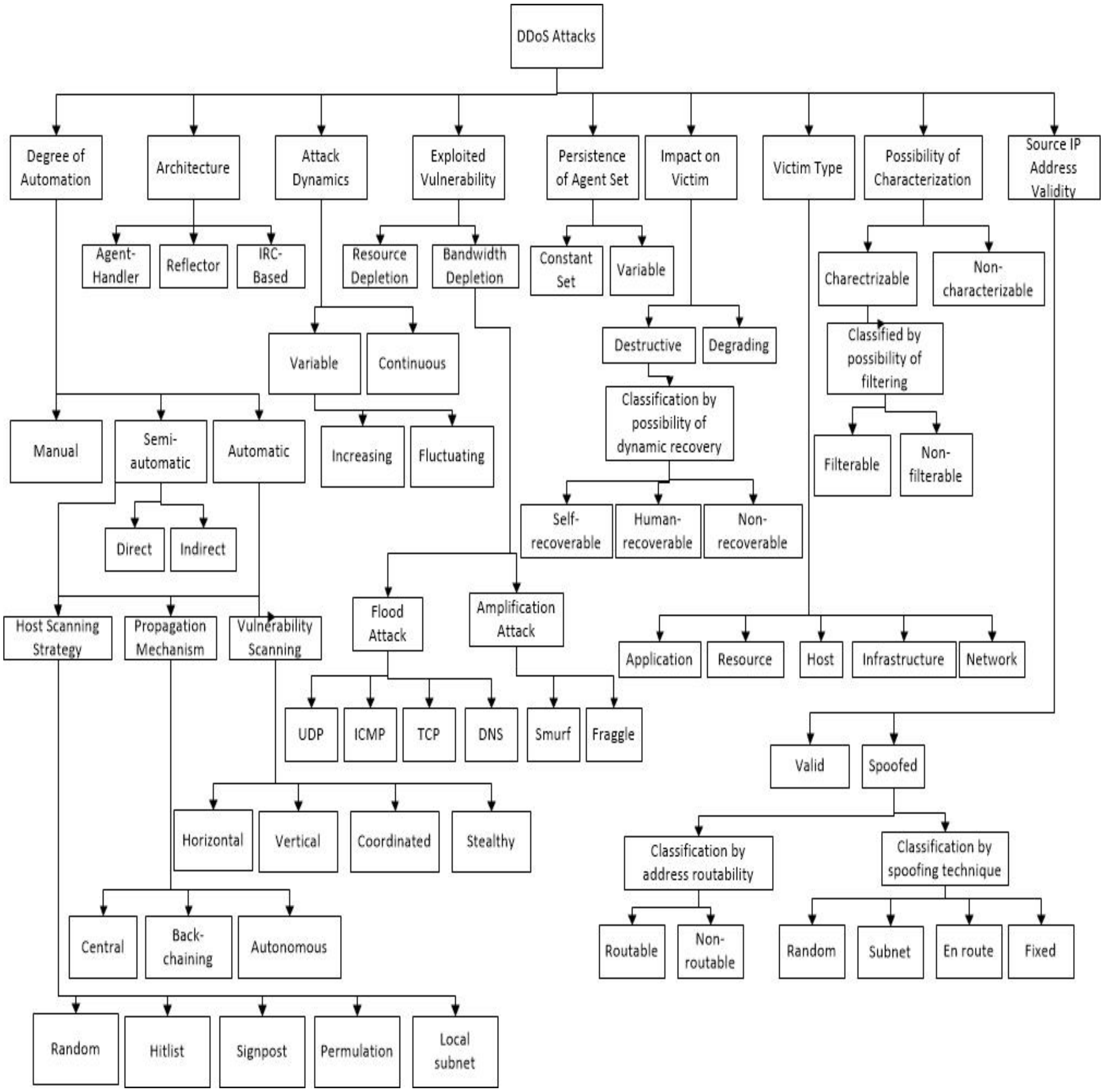


Figure 2.2: Taxonomy of DDoS Attacks

CHAPTER 3

Economic Denial of Sustainability (EDoS)

Small organizations, i.e., cloud adopters, tend to rent storage and computing resources as a service from a cloud computing provider. The reason for this is to reduce investments. An organization will sign a Service Level Agreement (SLA) with the cloud provider so that more resources will be allocated to this organization as needed. For some organizations, the upper boundary for the SLA is very high (very large portion of the resources of the cloud provider could be allocated to such an organization, if needed). An organization is billed based on its resource usage. An Economic Denial of Sustainability, or EDoS, attack is used by an attacker to cause economic loss to the cloud solution adopter. The resources of an EDoS attack's victim will expand in order to handle the requests of the attack due to the elasticity property of the cloud. The cloud adopter, i.e., the victim, will have to pay for all the resources that have been utilized by the attack.

EDoS attacks are only specific to cloud computing [14]. In this work, we assumed that any attack that targets the cloud adopter economically is considered an EDoS attack. However, DoS attacks and DDoS attacks are considered the most famous EDoS attacks when transformed from the conventional networks to the cloud computing environment [1, 6, 14, 15, 16]. These attacks are achieved by targeting the bandwidth of the victim's network or by targeting the victim's processing capacity. In conventional networks, these attacks aim to either exhaust the resources of the victim or crash them. In cloud computing, however, the elasticity property of the cloud will not allow the resources of the targeted adopter to be exhausted by the attack. Instead, it will allocate more resources, resulting in an EDoS attack. To further complicate the scenario, HTTP based DDoS attacks are the most challenging attacks because it would be difficult to filter the legitimate traffic from the attack traffic. A cloud adopter must distinguish legitimate traffic from malicious traffic or it will end up blocking traffic that comes from legitimate users. In addition to the DDoS attack, there are several attacks that can result in EDoS. In the next section, we provide a survey for the attacks that can result in an EDoS attack.

3.1 A Survey for EDoS Attacks

In Tables 3.1-3.4, we listed all the attacks mentioned in Table 2.1 and checked them to determine if they can result in an EDoS attack.

Table 3.1 shows the virtualization and infrastructure level attacks. The attacks that may result in an EDoS attack in this category include the covert channel, cross-VMs, and blue pill attacks.

Table 3.2 lists the language and malicious injection attacks subcategory of the application level attacks category. The attacks of this category that may cause an EDoS attack are the buffer overflow, XML signature wrapping, Trojan horse, and backdoor and debug options.

In Table 3.3, the web application attacks subcategory of the application level attacks category is presented. From this subcategory, the SQL injection attack is the only one that may result in an EDoS attack.

In Table 3.4, the attacks of the network level category are presented. Out of these, attacks that might cause EDoS are the MITM, replay attack, DNS cache poisoning, and BGP prefix hijacking.

Table 3.1: Virtualization and Infrastructure Level Attacks That May Result In EDoS

Attack	**Attack Description**	**EDoS?**	**Reason for Decision**
Covert channel attack	A prohibited communication between two processes which are not supposed to communicate [8].	**Yes**	If the attacker can perform this attack, then he will have the capability to send instructions to the victim's VM. Depending on the type of these instructions, the attacker can instruct the victim's VM to do operations that are resource extensive.
Side channel and Timing channel attacks	These attacks use the information of the physical implementation of the security algorithm to reconstruct the secret key [8].	No	Many VMs use the same hardware of a single host server. The attacker should find a way to recognize when his victim is using the hardware. The attacker has also to recognize that information gained from the hardware is related to his victim, and it is not of another VM, which is difficult in the cloud computing environment.
Cross-VMs attack	An attack in which the attacker is a VM that resides on the same cloud as the victim VM [9].	**Yes**	If the attacker could communicate with the victim's VM, then the attacker's VM can instruct the victim's VM to do operations that are resource extensive.
Blue Pill attack	An attack in which a thin rootkit hypervisor is implemented between the VMs and the hypervisor [10].	**Yes**	A hacker can use this rootkit to control all the VMs that run on the rootkit. He can instruct a VM to perform some tasks and the legitimate cloud adopter that owns this VM will have to pay for this.
Indirect DoS	Indirect DoS attack is caused when a neighbor VM is under DDoS attack. When more resources are needed, they will not be available since they are allocated to the VM under DDoS attack [11].	No	The victim VM of an indirect DoS attack will not consume any additional resources because they would be consumed by the VM that is under the DDoS attack.

Table 3.2: Application Level: Language and Malicious Injection Attacks That May Result in EDoS

Attack	Attack Description	EDoS?	Reason for Decision
Buffer overflow attack	The attacker can cause the web application of the cloud adopter to execute arbitrary code by sending it some crafted input. For example, it can be used to crash a program by making it run in an infinite loop which will consume many resources [33].	**Yes**	If a code that results in buffer overflow has been successfully injected and executed in the victim's VM (e.g., a program that is put in infinite loop), then this will result in consuming many resources.
XML signature wrapping	An attack in which the body of a SOAP message is moved to its header. The attacker uses the new body to do malicious operations [11].	**Yes**	The new instructions sent by the attacker in the body of the new packet might cause economic loss to the cloud adopter if they include operations that will result in high resource allocation.
Trojan horse	A malicious program that hides itself as a legitimate file.	**Yes**	Depending on its type, a Trojan horse can be used just to interrupt the work of a VM by instructing it to execute resource extensive operations. To execute these operations, the VM will be allocated more resources resulting in an EDoS attack.
Backdoor and debug options	A backdoor is used to access a VM without authentication. Debug options are used to re-test the program and can be used by an attacker to access the VM without authentication.	**Yes**	When an attacker gets an access to the victim's VM, the attacker can do anything, including performing resource extensive processing on behalf of the legitimate cloud adopter who will have to pay for performing this processing.
Hidden field manipulation attack	During a session, some of the data that are sent to the client are sent in hidden fields. This altered data will be displayed by the client instead of the original.	No	This attack is used to change the content of a web page to make it offensive, but it will not consume any resources and hence it will not result in an EDoS attack.
Meta data spoofing	A meta data file contains information about the mechanisms that will be followed during a session. It is sent before a session starts.	Indirect	This attack can be used as a first step to break the security of the VM. If succeeded, it will increase the chances of executing an EDoS attack. However, it is not an EDoS attack by itself.

Table 3.3: Application Level: Web Application Attacks That May Result in EDoS

Attack	Attack Description	EDoS?	Reason for Decision
SQL injection attack	It is an attack in which harmful code is sent and executed in the database [21].	**Yes**	The injection might result in sophisticated processing (e.g., very complicated SQL statements) which requires more resources.
Cross-Site-Scripting (XSS) attack	An attack in which malicious script is posted on a web page. The browser of a user exploring this page will execute the script and sensitive information can be stolen [34].	No	This attack is used mainly to target the end-users of the services provided by the cloud adopter.
Cookie poisoning	Modifying a cookie to impersonate a legitimate user and get an unauthorized access [21].	Indirect	If performed successfully, the attacker will get an unauthorized access by impersonating the cloud adopter. It is the first step to an EDoS attack, but not an EDoS attack by itself.
CAPTCHA breaking	Breaking the CAPTCHA will deny recognizing human from computers [21].	Indirect	CAPTCHA is used as a countermeasure for DDoS attacks. If it is broken, DDoS attacks will result in EDoS attacks in the cloud.
URL guessing	Discovering the URL address of a VM.	No	Knowing the URL of the VM will give a chance to the attacker to cause problems such as knowing the suitable attack that can be used to bypass the security of the VM. It is not an EDoS attack.
Phishing	A method used to collect user passwords or financial data by fooling the user using forged e-mails with fake websites [35].	No	This attack is applicable to the end-user of the services of the cloud adopter.

Table 3.4: Network Level Attacks That May Result in EDoS

Attack	**Attack Description**	**EDoS?**	**Reason for Decision**
Eavesdropping and Sniffer attacks.	Listening to communications between a client and a cloud adopter, or between a cloud adopter and a cloud provider.	Indirect	Eavesdropping and Sniffer attacks will not cause EDoS attack by themselves, but they will increase the chances of the attacker.
MITM	The attacker creates two SSL/TLS connections, one with the client and the other with the server. It acts as a proxy between them [36].	**Yes**	The attacker will impersonate the cloud adopter and will have the capability of instructing the VM belonging to the adopter to execute operations on behalf of the adopter which results in an EDoS attack.
Replay attack	An attack in which the attacker saves old messages sent to the victim and sends them again after a period of time [22].	**Yes**	The attacker sends old messages that he saved previously. These messages might include instructions that require very powerful computing resources. Without proper security rules for handling old messages, the victim will execute the requests in these messages again and again which will result in economic loss.
DNS cache poisoning	An attack in which DNS mapping is altered in a DNS server.	**Yes**	Causes large volumes of traffic to be forwarded to the victim which will result in EDoS.
BGP prefix hijacking	Modifying the BGP advertisements so that traffic is routed to unintended destinations [21].	**Yes**	This might cause large volumes of traffic to be routed to a targeted cloud adopter. This will result in an EDoS attack.
IP address reuse attack	The IP address reuse issue may result in forwarding traffic to unintended destinations.	No	From outside the cloud, the IP address of the cloud adopter is usually static. Since the IP address is not being changed frequently, the chances of being flooded because of the IP address reuse issue is limited.

This section was an introduction to the EDoS attack in which the idea of the attack was discussed and explained. The attacks presented in Table 2.1 were also studied to check if an attack among these can result in an EDoS attack. Tables 3.1-3.4 show these attacks and state whether an attack can result in an EDoS attack or not. A brief description for each attack and the reason for considering it leading to an EDoS attack or not are also presented in the Tables 3.1-3.4. In the next section, we provide a taxonomy for the EDoS attacks based on the way an attack, from Tables 3.1-3.4, can result in an EDoS attack, which will affect the adopter economically.

3.2 Taxonomy of EDoS Attacks

From Tables 2.1and 3.1-3.4, the EDoS attacks can be classified based on the cloud service delivery models and the cloud attacks taxonomy presented in Figure 2.1. However, we decided to provide a taxonomy for EDoS attacks based on the way an attack, from Tables 3.1-3.4, can result in an EDoS attack in cloud computing. The reason of classifying EDoS attacks in this way is to categorize EDoS attacks in a limited number of categories so that a single mitigation technique for an attack category can possibly be used to countermeasure other EDoS attacks that fall under that category. Following this way of classification, the EDoS attacks can be classified into 5 categories: Resource Extensive Requests Attacks, Malicious Code Attacks, Impersonation Attacks, Prohibited Access Attacks, and Flooding Attacks. Figure 3.1 shows the proposed taxonomy for the EDoS attacks based on the way an attack impacts or affects the cloud to cause economical loss to its adopter.

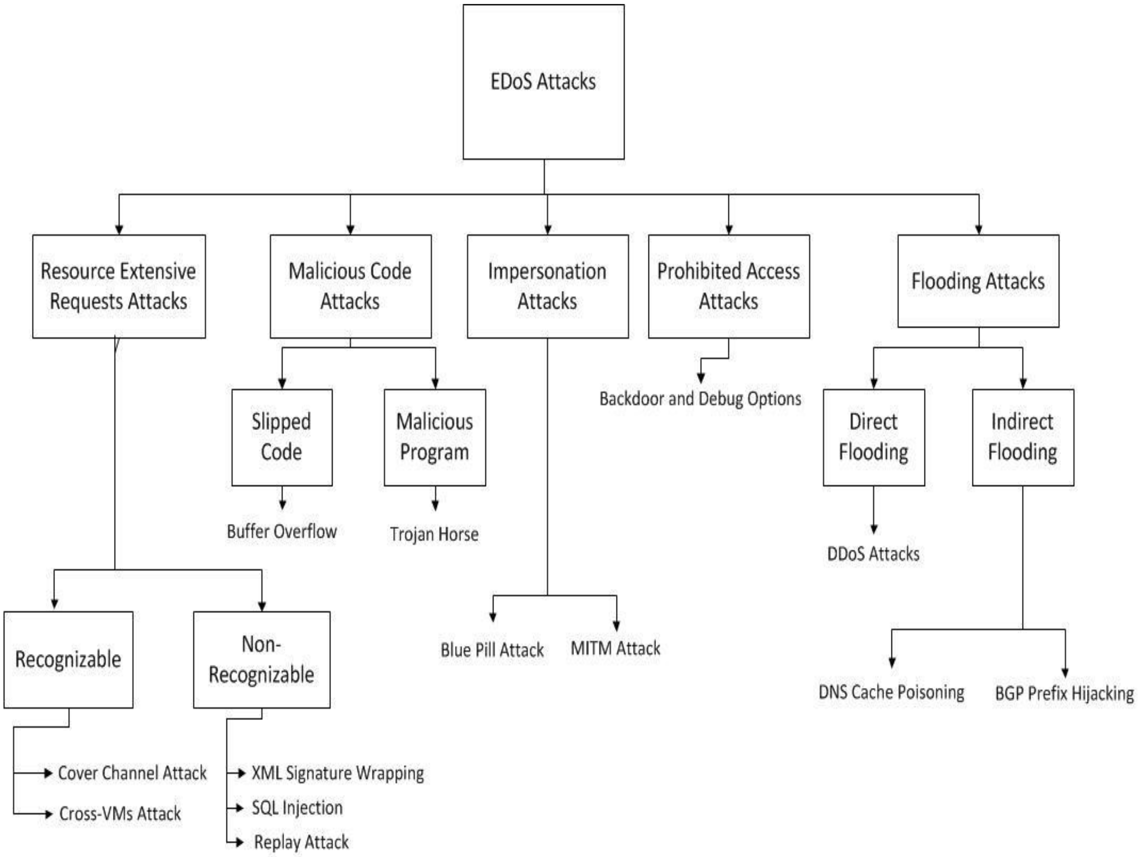


Figure 3.1: Taxonomy for EDoS Attacks

1. Resource Extensive Requests Attacks

In a resource extensive requests attack, the attacker sends requests to the victim’s VM that result in resource extensive operations which will force the victim to request more resources. The type of requests is different for different applications. For example, if an image processing application is hosted in the cloud, then the attacker can request very complicated operations on many large images. These operations require much more computing resources in order to be executed. As a result, additional computing resources will be allocated to respond to these complicated operations and the cloud solution adopter will have to pay for them at the end. The attacker might send these requests intentionally to cause economic loss to the cloud adopter.

Attacks under this category can further be classified into two subcategories: recognizable and non-recognizable. In recognizable attacks, the attack requests can be recognized because they are violating the security rules which prohibit communication between these two VMs. For example, if the attacker’s VM tries to send a request to a VM that resides on the same physical host as the attacker’s VM, then this would be detected because it is not allowed. The attacker has to find a way to break this rule before sending any requests. Attacks that are of this type are the covert channel attack and cross-VMs attack.

In non-recognizable attacks, the attacker sends requests that seem legitimate to the system. However, these requests are spoofed and are intended to cause economic loss to the cloud adopter. From Tables 3.1-3.4, the EDoS attacks that are of this type are XML signature wrapping attack, SQL injection attack, and replay attack.

2. Malicious Code Attacks

In malicious code attacks, a malicious code is executed in the victim's VM that results in high consumption of computing resources. Malicious code attacks are achieved by inserting instructions of malicious code as a complete malicious program or in a malicious way to a legitimate program that runs in the VM. When executing this code, it will consume many resources like in the case of making a program run in an infinite loop. The difference between this category and the resource extensive requests attacks is that in the case of malicious code attacks, programming instructions are executed inside a legitimate program (as in the case of buffer overflow), or as a malicious program (as in the case of Trojan horse). In resource extensive requests attacks, resources are allocated to respond to spoofed requests that ask executing a job that requires many resources.

Malicious code attacks have two subcategories: slipped code and malicious program attacks. In slipped code attacks, instructions are inserted in a malicious way to a program that is running in the victim's VM. These instructions might be used to execute resource extensive operations. From Tables 3.1-3.4, the attack that falls under this category is the buffer overflow attack. In malicious program attacks, a complete malicious program is inserted and executed somehow in the victim's VM. Trojan horse is an example of the attacks that fall under this category.

3. Impersonation attacks

A legitimate cloud adopter is charged for executing his jobs in the cloud. If the identity of the cloud adopter is spoofed somehow by an attacker, the attacker will use the resources of the legitimate adopter on behalf of this adopter. The legitimate adopter will have to pay for executing the jobs for that attacker. Attacks that are of this type are considered as impersonation attacks. From Tables 3.1-3.4, the EDoS attacks that come as a result of an impersonation attack are the blue pill attack and the man-in-the-middle (MITM) attack.

4. Prohibited Access Attacks

In prohibited access category attacks, the attacker accesses and controls the VM of the victim in a prohibited way. After accessing it, the attacker can use the VM for his own purposes. The attacker has full control on the attacked VM in this category and the legitimate adopter will have to pay for executing the jobs of the attacker. The attacks that are of this type from Tables 3.1-3.4 are the backdoor and debug options.

5. Flooding Attacks

Flooding attacks are the most common type of EDoS attacks in which large volumes of traffic are sent to the victim's VM which results in requesting more computing resources to respond to. Flooding attacks can be classified further into two subcategories: direct flooding and indirect flooding attacks. In direct flooding attacks, the attacker directly floods the VM of the victim using any DDoS technique from those explained in the previous section. This subcategory includes all the DDoS attacks. In indirect flooding attacks, the attacker will not send traffic directly to the victim. Instead, he will perform a malicious action that will result in rerouting large volumes of traffic to the VM of the victim. From Tables 3.1-3.4, the attacks that fall under this subcategory are the DNS cache poisoning and the BGP prefix hijacking.

3.3 Existing Mitigation Techniques

In the literature, researchers that attempted to address the EDoS attack are very few. Some mitigation techniques have been developed to block EDoS attacks. This indicates that more research is required to protect the cloud computing from EDoS attacks.

As a mitigation technique for EDoS, sPoW is used and it requires a proof of work from the clients before completing the interaction with the server [31]. However, sPoW has a number of disadvantages discussed by Sqalli et al. [15].

VivinSandar and Shenai [14] showed how a DDoS attack is transformed to an EDoS attack in the cloud. They also surveyed the literature for mitigation techniques against EDoS and DDoS attacks in the cloud. Finally, they proposed a security framework for EDoS attack protection. However, Modi et al. [32] pointed that this mitigation technique is inefficient because it is based on the traditional firewall only.

Kumar et al. [37] proposed a mitigation technique for the EDoS attack using in-cloud scrubber service. Their solution is provided as a service by the cloud service provider. The solution uses two modes of operation, normal mode and suspected mode. When the web server is working as expected, then the system will work in the normal mode. But when the service provider notices that the traffic that targets the web server exceeds an acceptable threshold, then the operation will be switched to the suspected mode. In the suspected mode, the requests will be sent to a scrubber server which will send puzzles to the clients to distinguish legitimate requests from bot requests. Their proposed solution also attempts to detect low-rate DDoS attacks.

Sqalli et al. [15] proposed a solution called the EDoS-Shield to mitigate the EDoS attack. The EDoS-Shield classifies the requests to whitelisted and blacklisted based on the source of the request, legitimate or bot. This is achieved using a verifier node which creates the whitelist and blacklist. A virtual firewall is used to block all the requests that come from the blacklisted sources. This work was expanded by Al-Haidari et al. [16] to mitigate the attack in case the attacker uses spoofed IP addresses. The following section is allocated to explain the EDoS-Shield in more details as it is the main topic of this work.

3.4 The EDoS-Shield Mitigation Technique

Figure 3.2 shows the architecture of the EDoS-Shield mitigation technique. The main components of the EDoS-Shield mitigation technique are the virtual firewall (VF) and the verifier node (V-Node). The virtual firewall has two lists of IP addresses, whitelist and blacklist. The whitelist consists of those source IP addresses which are considered legitimate. All the requests that come from those sources are allowed to pass the firewall to the cloud adopter servers. On the other hand, all the IP addresses that are contained in the blacklist are considered malicious, and hence all the traffic that comes from these IPs is blocked by the firewall.

When there is a request from an unknown source, i.e., its IP is not included in the firewall's lists, the request is forwarded to the V-Node. The V-Node sends a graphical Turing test to the source of this request. If the request has been issued by a human, the human will be able to pass the test, i.e., respond to the test. Then, the V-Node will add the IP address of the source of the request to the whitelist of the firewall. Any following requests from this source will be allowed to pass the firewall. However, if the request has been generated by a machine, e.g., bot, the machine will fail to solve the test. In this case, the V-Node will add the IP address of the source of the request to the blacklist of the firewall. Any following requests from this source will be blocked by the firewall.

From the discussion above, it is clear that the EDoS-Shield mitigation technique is capable of blocking the direct flooding type of EDoS attacks presented in the taxonomy of Figure 3.1. The EDoS-Shield might not be suitable to mitigate indirect flooding attacks. The reason for this is the fact that the attack perpetrator may intend to make the attack packets be forwarded to the cloud using devices or servers that are supposed to be legitimate. If the IP addresses of these servers or devices are listed in the whitelist of the firewall of the EDoS-Shield, then all the traffic forwarded from these devices will be accepted, which will result in an EDoS attack. Hence, the EDoS-Shield in its basic form is only capable of blocking the direct flooding type of EDoS attacks presented in the taxonomy.

In this work, the EDoS-Shield mitigation technique has been implemented using an experimental testbed. Chapter 4 discusses the testbed setup and the steps followed to evaluate the effectiveness of the EDoS-Shield in mitigating the EDoS attacks. Then, in Chapter 5, the results collected from the testbed are presented, discussed, and compared with those reported in Sqalli et al. [15].

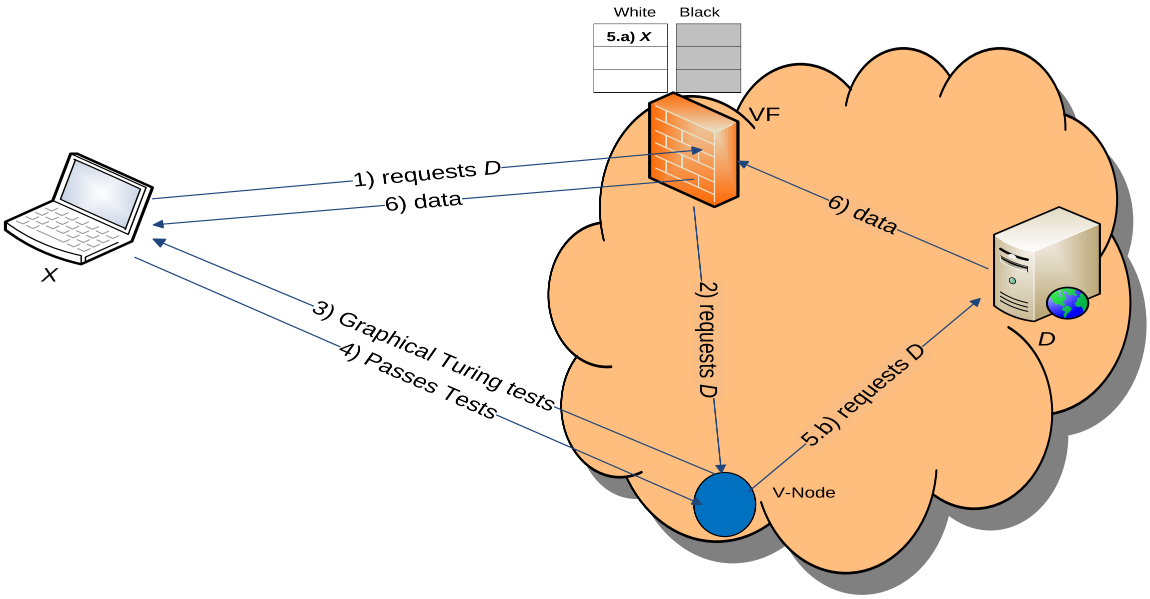


Figure 3.2: The EDoS-Shield Architecture

CHAPTER 4

TESTBED SETUP

In this chapter, we will discuss the testbed setup and how the experiments are performed.

4.1 Testbed Architecture

This section discusses how the EDoS-Shield mitigation technique has been implemented and evaluated using an experimental testbed in the lab. Since the main objective of this work is to compare the results obtained from the experimental testbed to those obtained from the simulation in [15], we prepared the testbed to be very close to the assumptions made in the simulation. First, the testbed has been designed without implementing the mitigation technique in order to study the effect of the EDoS attack on the cloud before adding the mitigation technique. Figure 4.1 shows the testbed before implementing the EDoS-Shield mitigation technique. Next, the EDoS-Shield mitigation technique was implemented in the testbed, and its effectiveness in blocking the EDoS attack was evaluated. Figure 4.2 shows the testbed after implementing the EDoS-Shield mitigation technique. The results obtained from the testbed are compared to those obtained from the simulation in Chapter 5 for both cases. The main components of the testbed for each case will be discussed next.

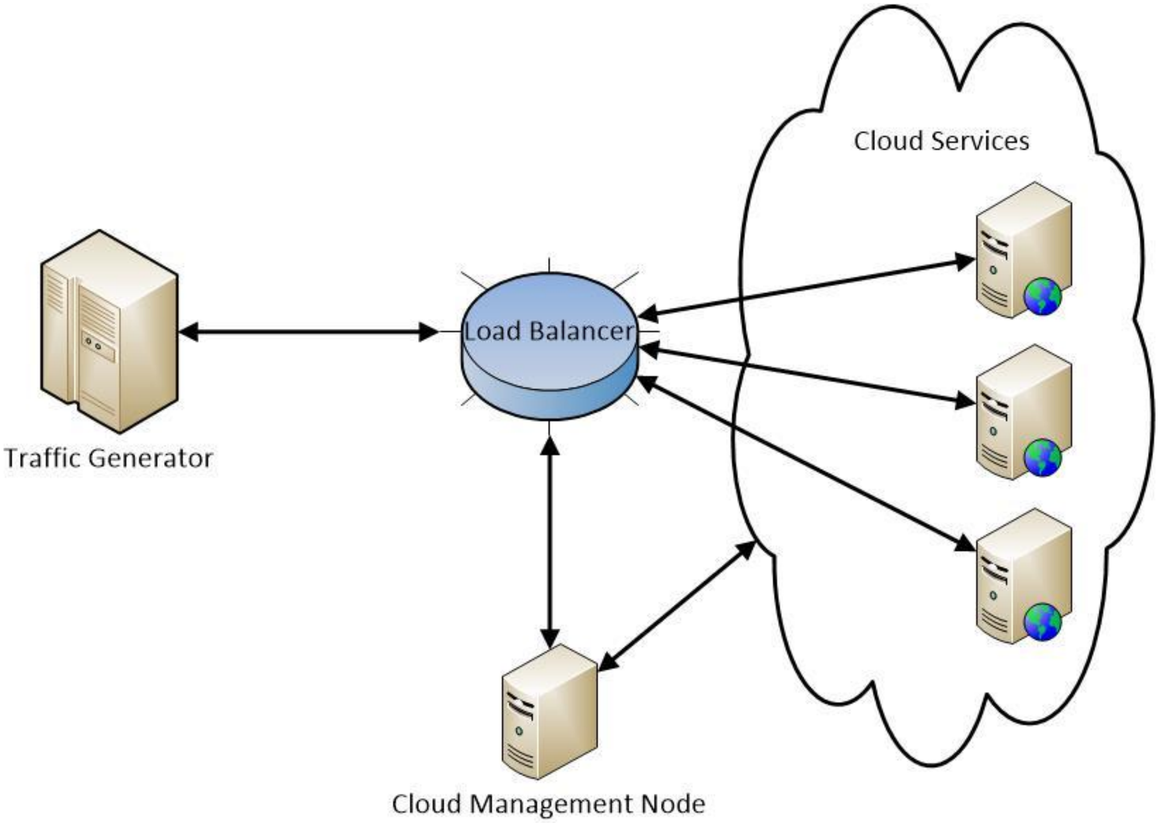


Figure 4.1: The Testbed without the EDoS-Shield Mitigation Technique

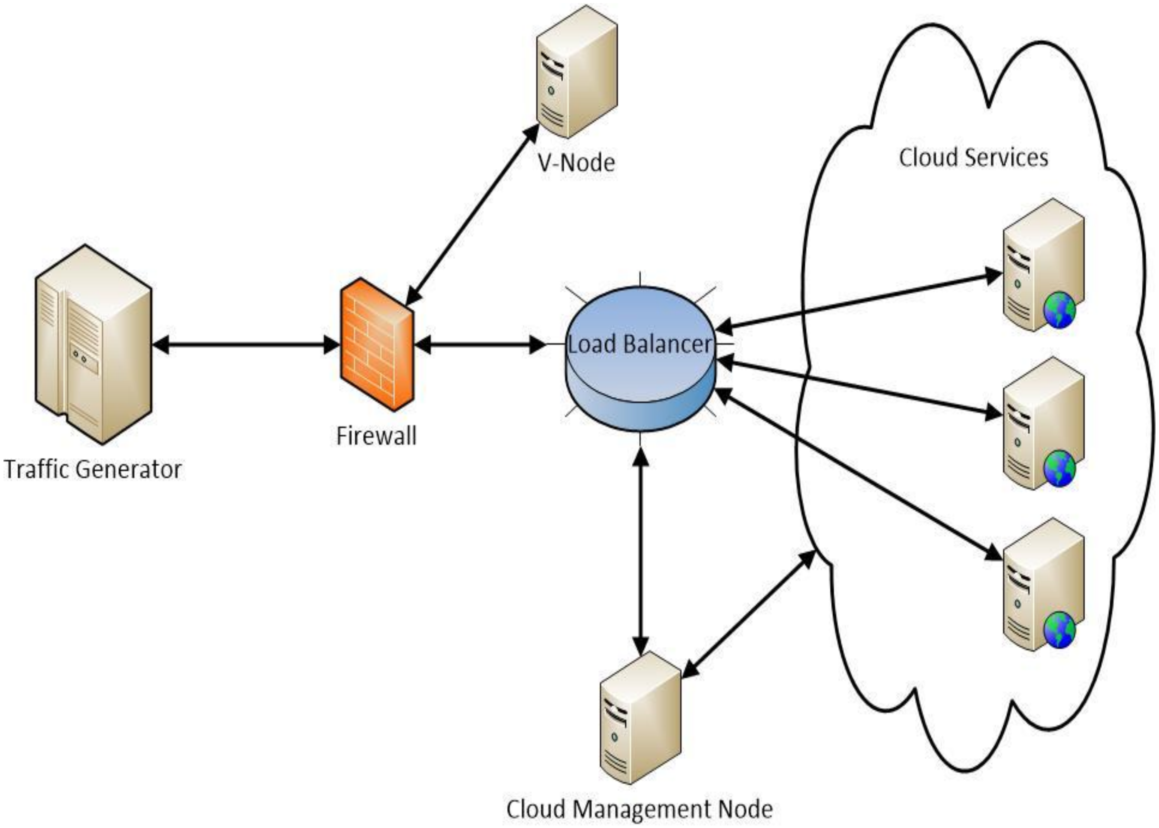


Figure 4.2: The Testbed with the EDoS-Shield Mitigation Technique Implemented

4.1.1 Components of the Testbed before Adding the EDoS-Shield

To study the effect of the EDoS attack on the cloud, a testbed was prepared without implementing the EDoS-Shield, like in Figure 4.1. The main component of this testbed will be discussed in this section.

Cloud Services

The main component of our testbed is the cloud. Citrix's CloudPlatform [38] and XenServer [39] were used to deploy the cloud. The CloudPlatform is a cloud management software which is responsible for managing the cloud and its resources. A single physical server was used as a management node on which the CloudPlatform was installed. Three physical servers were used as compute nodes on which the hypervisor, i.e., XenServer, was installed. The virtual machines (VMs), or the instances, on which the services provided by the cloud are deployed, run on these compute nodes. All the VMs are identical small instances that were created from a single template. This template contains a simple web server configured on CentOS Linux operating system [40]. Apache Server was used as the web server [41]. More details about the template configuration will be provided in the following sections.

Load Balancer

The load balancer is used to load the traffic among the VMs of the cloud. Our testbed uses Citrix's NetScaler VPX (200) [42] as a load balancer. NetScaler VPX is a virtual appliance that is installed on XenServer, on a separate physical server. NetScaler is configured and managed through the CloudPlatform. It is the entry point to the cloud services and hence all the traffic that comes to the cloud, or goes out of the cloud, passes through it. The dashboard of NetScaler is used for monitoring during performing the experiments as will be indicated later.

Traffic Generator

The traffic generator is used to simulate legitimate and malicious HTTP requests during experiments. We used Apache JMeter as the HTTP traffic generator [43]. In addition to the basic features that come with JMeter by default, we added the standard set of plugins [44], so that we can add more features to JMeter. We installed JMeter on 8 VMs running on XenServer, which is installed on a separate physical server. Then, we performed a set of experiments to generate traffic for different numbers of VMs. We found that changing the number of attacking VMs, while generating the attack traffic at the same rate each time, will not affect the results of the experiments. In our experiments, we used 8 attacking VMs to simulate 8 users.

Three JMeter plugins were used to generate and control the traffic in addition to monitoring it. These plugins are the ultimate thread group, the throughput shaping timer, and the hits per second listener. We also used the HTTP request sampler to format the HTTP requests. Below is a brief description of each one of these components and how it has been used in the experiments.

The ultimate thread group plugin is used to create the threads that simulate real users. The maximum HTTP request rate that a JMeter VM sends in our experiments is 1000 Request/Second. To achieve this rate, the ultimate thread group was configured to create 1100 threads. The additional 100 threads are used to guarantee that the HTTP requests are always more than the HTTP requests rate that is targeted. This in turn will guarantee that the targeted HTTP requests rate is always achieved. The ultimate thread group was configured to create the threads within 30 seconds.

Since the traffic created by the 1100 threads is always more than the targeted HTTP requests rate, the throughput shaping timer is used to specify and send the exact rate that is targeted in an experiment. The targeted rate is achieved in two steps. First, the throughput shaping timer starts with 1 Req/Sec rate and keeps increasing the rate until the targeted rate is achieved. The throughput shaping timer was configured to complete this step in 30 seconds. After achieving the targeted rate, the throughput shaping timer will keep sending HTTP requests at that rate as a fixed rate in the second step. The throughput shaping timer is configured to keep using the targeted rate for 3600 seconds. These requests are sent to the IP address specified in the HTTP request sampler, which is used in the experiments only to specify the destination IP address to which the traffic will be forwarded.

To make sure that the JMeter VM sends HTTP requests at the targeted rate, the hits per second listener plugin is used. Figure 4.3 shows a snapshot of this plugin. The targeted rate in the figure is 400 Req/sec. The figure shows that the throughput shaping timer keeps increasing the rate in the first 30 seconds. Then, it keeps sending the traffic at a fixed rate.

Finally, to make sure that the aggregated traffic created from all the JMeter VMs is at the targeted rate of an experiment, the dashboard of NetScaler is used. Figure 4.4 shows a snapshot for the NetScaler dashboard. The targeted rate in the figure is 1200 Req/Sec.

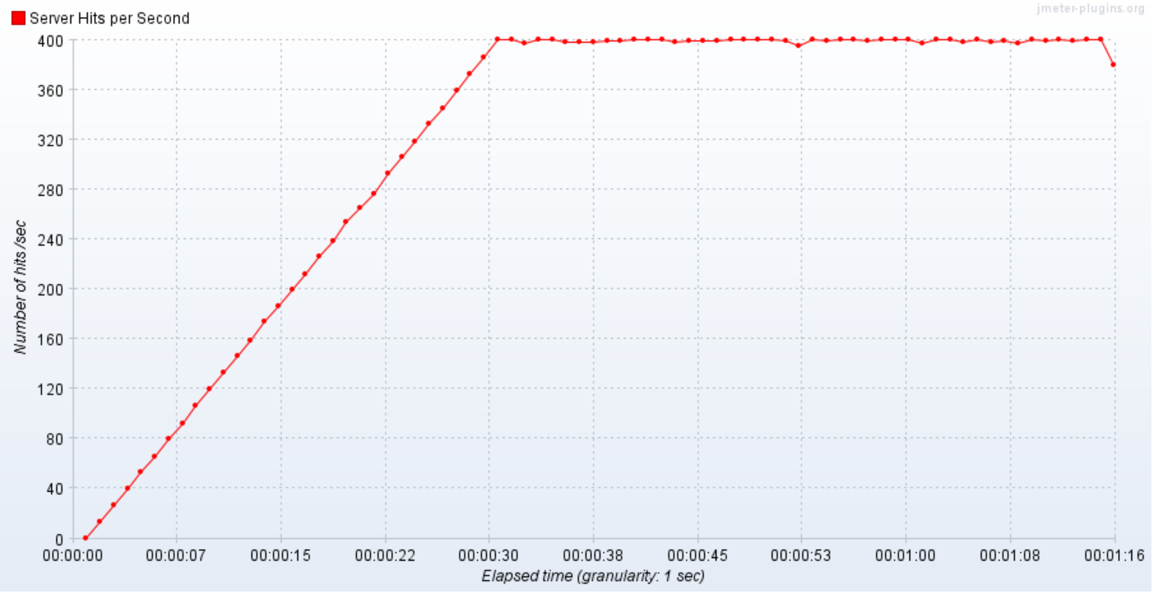


Figure 4.3: A Snapshot of the Hits Per Second Plugin of JMeter

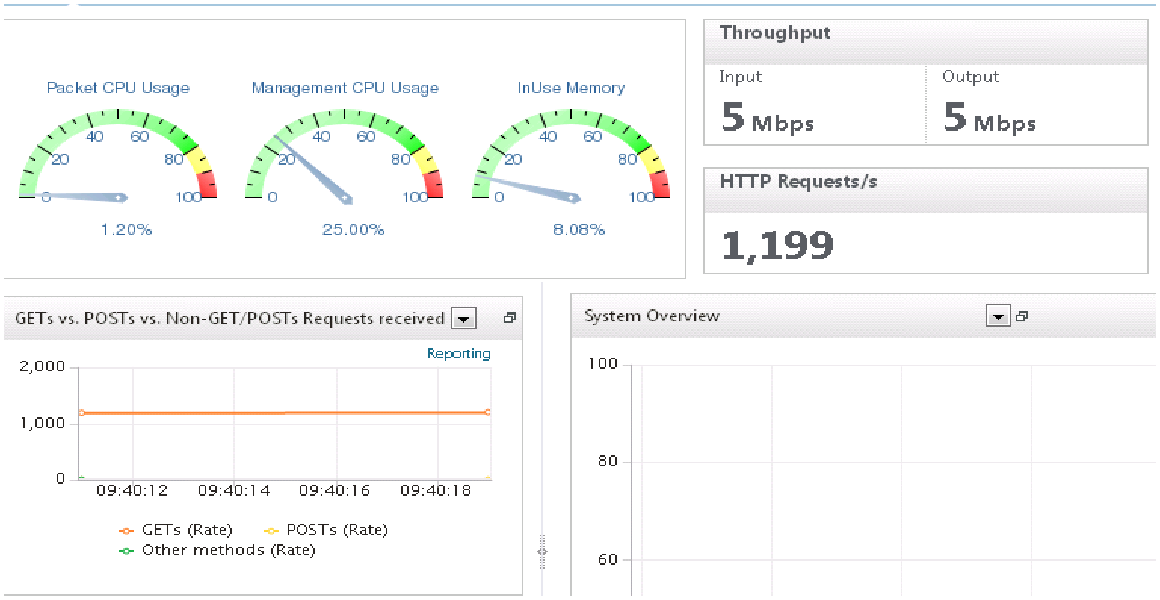


Figure 4.4: A Snapshot for the Dashboard of NetScaler

4.1.2 Components of the Testbed after Adding the EDoS-Shield

In addition to the components discussed in the previous section, the firewall and the verifier node were added to the testbed to build the EDoS-Shield mitigation technique.

Firewall

The firewall is used to filter all the traffic that comes to the cloud. The traffic that comes from the whitelisted sources is allowed to access the cloud services, while the traffic that comes from blacklisted sources is dropped. To achieve this, we used Linux's iptables firewall [45]. The iptables firewall on a CentOS Linux was configured to forward the traffic from unknown sources, i.e., traffic which the IP address of its source is not listed in the firewall lists, to the V-Node. The iptables forwards the traffic of a whitelisted source to the load balancer, and it drops the traffic that comes from a blacklisted source. The lists of the firewall are updated by the verifier node.

Verifier Node (V-Node)

The verifier node (V-Node) is responsible for updating the whitelist and blacklist on the firewall. It is a web server that sends Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) to the clients and updates the lists of the firewall based on the response of the client. We implemented the V-Node using the WampServer [46] installed on a Windows VM running on a separate physical server. The CAPTCHA was implemented using the code in [47].

Figure 4.5 shows the physical network topology of the testbed. Two VLANs were used to separate the traffic of the experiments from the other traffic in the lab. The firewall, verifier node, VMs running on the XenServer servers, and the JMeter server are all on a separate VLAN. This allows the experiments to be executed using a single physical switch without the need of a router. All the other devices, including the physical XenServer Servers, are connected to another VLAN. The Network Attached Storage (NAS) is used by the CloudPlatform to store the data of the VMs. This configuration enables the live migration of VMs from one XenServer host to another automatically when needed. The JMeter server is connected to the network using 2 Gigabit network interface cards. From the JMeter VMs, 4 are connected to the network using one of these network cards. The other 4 VMs are connected using the other network card.

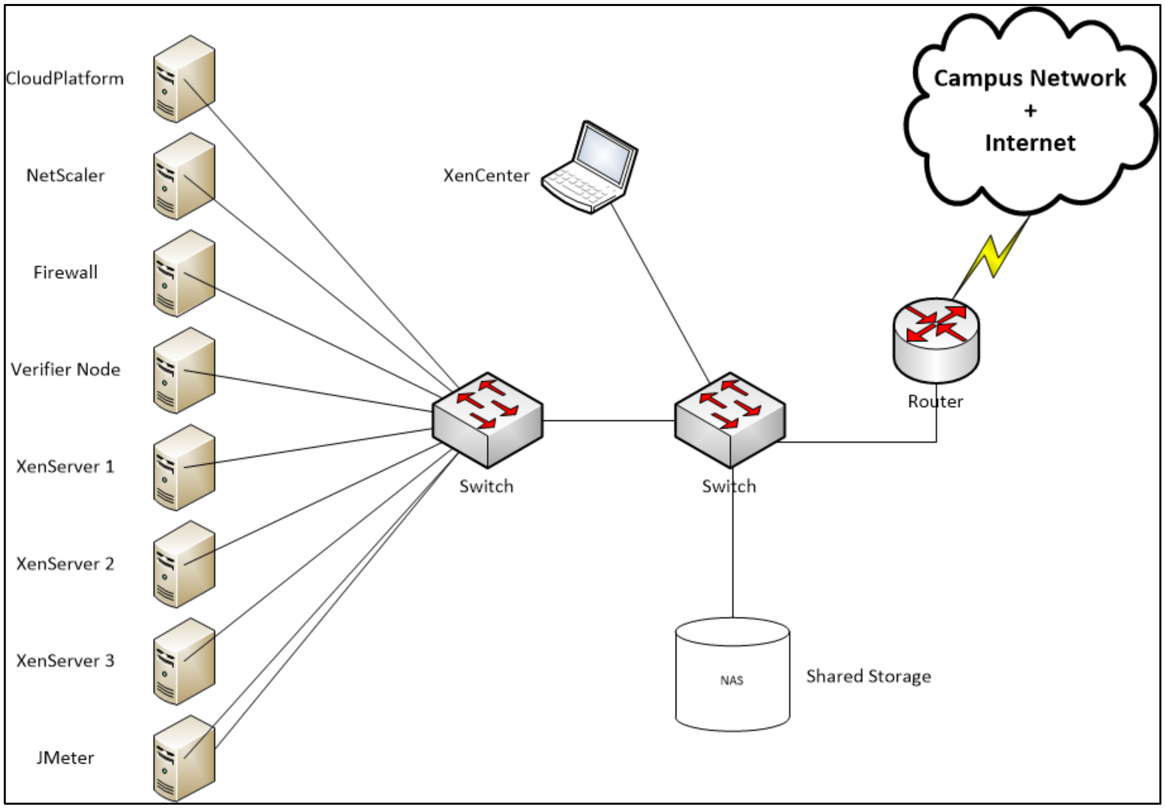


Figure 4.5: The Physical Network Topology of the Testbed

4.2 Experiments Execution Steps

This section describes the steps followed to perform the experiments. In the following subsection we provide the details of performing the experiments to study the effect of the EDoS attack on the cloud before using the EDoS-Shield mitigation technique. After that, we discuss the experiments executions steps to evaluate the effectiveness of the EDoS-Shield in blocking the EDoS attack.

4.2.1 Studying the Effect of EDoS Attack on Cloud Computing

In order to study the effect of the EDoS attack on cloud computing, we performed a set of experiments without using the EDoS-Shield. The results of these experiments are used to study the effect of the EDoS attacks in terms of CPU utilization and the response time. These results are also compared to the simulation results obtained in [15]. Figure 4.1 illustrates the testbed architecture used in these experiments. In these experiments, the traffic generator component sends the traffic directly to the load balancer. The load balancer sends the traffic to the instances (VMs) of the cloud on which a simple web page is hosted. This web page was designed to make the web application on an instance to cause the same CPU utilization like that of the simulation. Hence, the web servers on the cloud instances have the following properties:

1- Each instance has the capability to handle 100 HTTP Request/Second (Req/Sec).

2- The packet size of the response is 580 bytes.

Then, following the same assumptions of the simulation, we assumed the upper threshold that will trigger autoscaling is 80% CPU utilization. This means that a new instance should be created and assigned to the load balancer if the total CPU utilization for all the instances exceeds 80%.

The maximum attack rate that has been used in the simulation is 8000 Req/Sec, and the maximum number of instances is 106. We executed half of the experiments of the simulation because of the limited resources in the testbed. Hence, the maximum attack rate that we used in the experiments is 4000 Req/Sec, and the maximum number of instances that we used is 56.Before starting an experiment, we make sure that all the cloud instances that will be used in the experiment are connected to the load balancer from the dashboard of NetScaler. The number of the instances that are used in an experiment depends on the rate of the attack in that experiment. To ensure that the incoming traffic to the cloud will not use more than 80% of the processing resources, we increase the number of instances following the same approach used in simulation. Hence the number of the required instances will be calculated as follows:

$$\frac{\lambda}{S\mu} \leq 0.8 \text{. Thus, } S = \lceil 1.25 \times \lambda / \mu + 1 \rceil \tag{4.1}$$

Where S is the required number of instances, λ is the traffic arrival rate, and μ is the service rate.

In addition to the EDoS attack rate, there is a 400 Req/Sec fixed rate of the legitimate traffic. This rate is added to the EDoS attack rate in all the experiments. Figure 4.6Figure 4.6 shows the number of required instances for each experiment based on equation 4.1. The service rate of each instance is 100 Req/Sec as discussed previously.

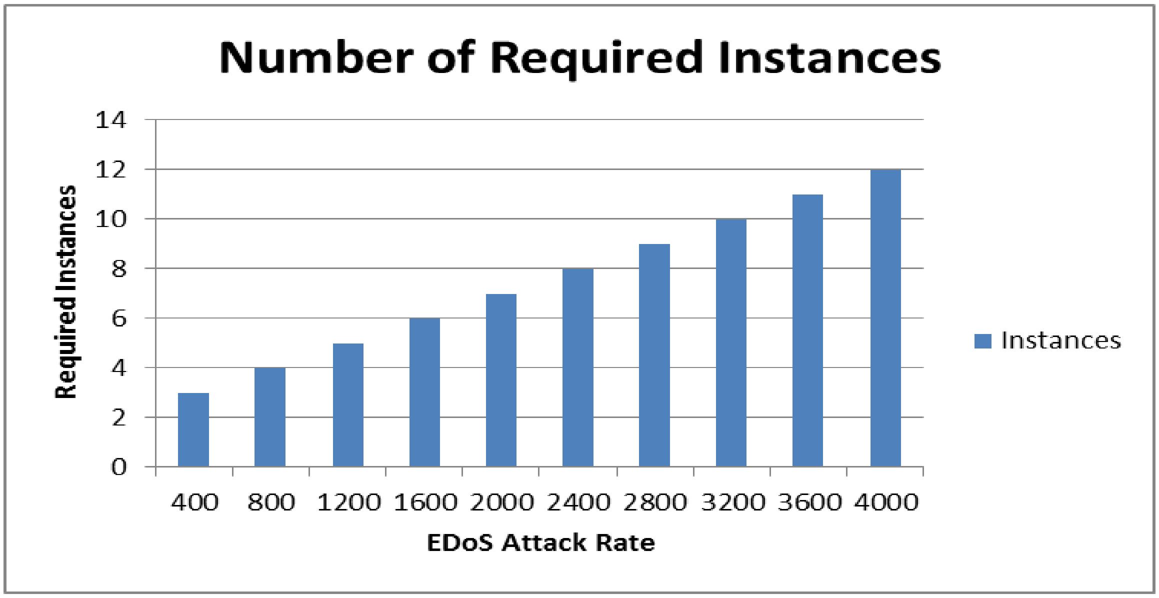


Figure 4.6: Number of Required Instances before Using the EDoS-Shield

After making sure that the appropriate number of instances has been assigned to the load balancer, we start the experiment by running the traffic generation on JMeter. We make sure that the HTTP requests are sent in the targeted rate using the hits per second plugin of JMeter and through the dashboard of NetScaler, as discussed earlier.

We keep monitoring the CPU utilization through Citrix's XenCenter [46], which is installed on a laptop to collect the results. When the CPU utilization of the instances reaches the steady state, the CPU utilization of each instance is collected separately, and then the average CPU utilization is calculated. The response time is measured using an add-on installed on the Firefox web browser called Firebug [47]. For each experiment, the response time is collected 30 times, and then the average is calculated.

For each rate of the EDoS attack, experiment is repeated 10 times. Each time the CPU utilization and the response time are collected. After collecting the results for all the 10 repetitions, the average CPU utilization and the average response time are calculated.

This section explained the steps followed when performing the experiments of studying the effect of the EDoS attack on cloud computing. The next section discusses the steps followed when performing the experiments of evaluating the EDoS-Shield mitigation technique.

4.2.2 Evaluating the EDoS-Shield Mitigation Technique

In this section, the experiments performed to evaluate the effectiveness of the EDoS-Shield in mitigating the EDoS attack are discussed. Most of the steps are the same as described in the previous section. The new change in these experiments is the introduction of the firewall and the V-Node, which are the components of the EDoS-

Shield. Figure 4.2 shows the architecture of the testbed after implementing the EDoS-Shield.

In this set of experiments, the firewall is the entry point to the cloud instead of the load balancer. All the traffic that comes to the cloud, or goes out of the cloud passes the firewall. JMeter on the 8 traffic generator VMs is configured to send the traffic to the firewall. We assumed the following for the traffic generator VMs when performing the experiments:

1- From the 8 traffic generator VMs, 2 will simulate the legitimate traffic, while the other 6 VMs will simulate the malicious traffic.

2- The CAPTCHA will be entered correctly for the legitimate traffic, and incorrectly for the malicious traffic. There is no timeout or false positives.

3- The first request from a VM will be sent using its web browser. The CAPTCHA will be answered correctly for the legitimate VMs, and incorrectly for the malicious VMs.

In all the experiments, the dashboard of NetScaler shows that only the legitimate traffic arrives to the cloud. Since the legitimate traffic is only 400 Req/Sec, then the number of cloud instances that are used in all the experiments is 6, as the equation 4.1 indicates. This is illustrated in Figure 4.7.

The experiment for each of the EDoS rates is repeated 10 times. The results are collected and calculated the same way as explained in the previous section.

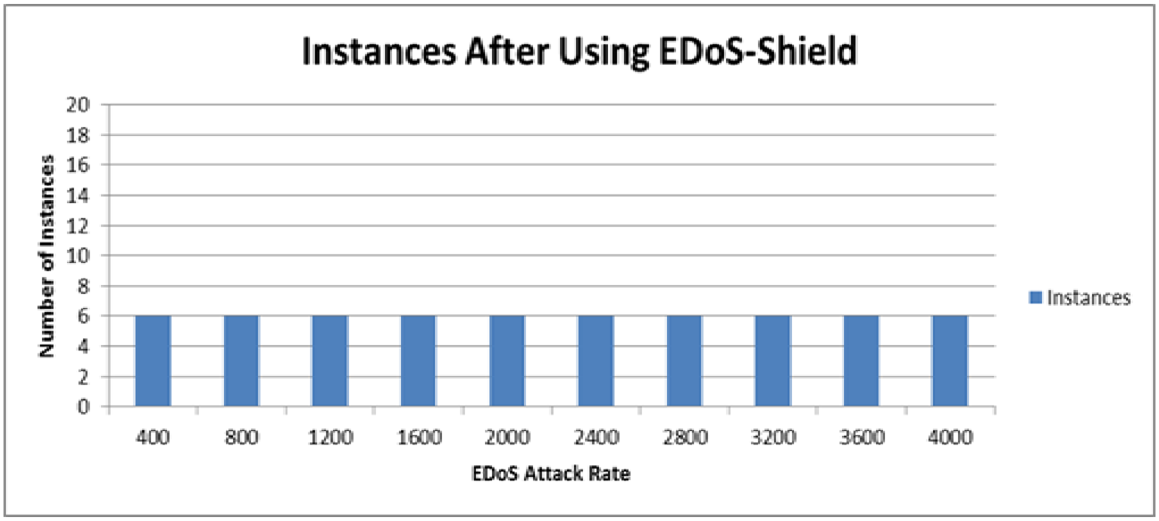


Figure 4.7: Number of Required Instances When Using the EDoS-Shield

This chapter discussed the testbed setup and the steps followed when performing the experiments using the experimental testbed. In Chapter 5, the results of the experiments will be presented and discussed.

CHAPTER 5

RESULTS AND DISCUSSION

In Chapter 4, the steps of performing the experiments were explained. In this chapter, the results of the experiments are presented and discussed. In section 5.1, the results of the experiments that study the effect of the EDoS attack on cloud computing are discussed. The results of the experiments studying the EDoS-Shield mitigation technique are presented and discussed in section 5.2.

5.1 Studying the Effect of EDoS Attack on Cloud Computing

The first set of experiments was performed to study the impact of the EDoS attack on cloud computing before using the EDoS-Shield mitigation technique. The steps followed when performing these experiments were discussed in section 4.2.1. The results obtained from these experiments are discussed in this section.

Each of the experiment has been repeated 10 times. Figure 5.1 shows the standard deviation for the CPU utilization results collected from each experiment. Figure 5.1 shows that the standard deviation for the CPU utilization results is very small. This indicates that there are no major differences in the results collected for each experiment.

Figure 5.2 compares the average CPU utilization results of the testbed to those of the simulation. The CPU utilization results of the testbed are very close to the results of the simulation. Both results show that when the rate the EDoS attack increases, the CPU utilization increases. But the CPU utilization will not exceed the threshold of 80% since more instances will be added to the cloud as the attack rate increases. Both results show that when the rate the EDoS attack increases, the CPU utilization increases. More instances will be added to the cloud as the attack rate increases. The addition of the new instances for handling the attack requests will result in a severe economic loss for the cloud adopter.

Figure 5.3 shows the relative error percentage for the CPU utilization comparison of Figure 5.2. Figure 5.2 illustrates that the results obtained from the testbed are very close to the results obtained from the simulation in terms of CPU utilization. The difference between the results of the testbed and the simulation is always below 5%.

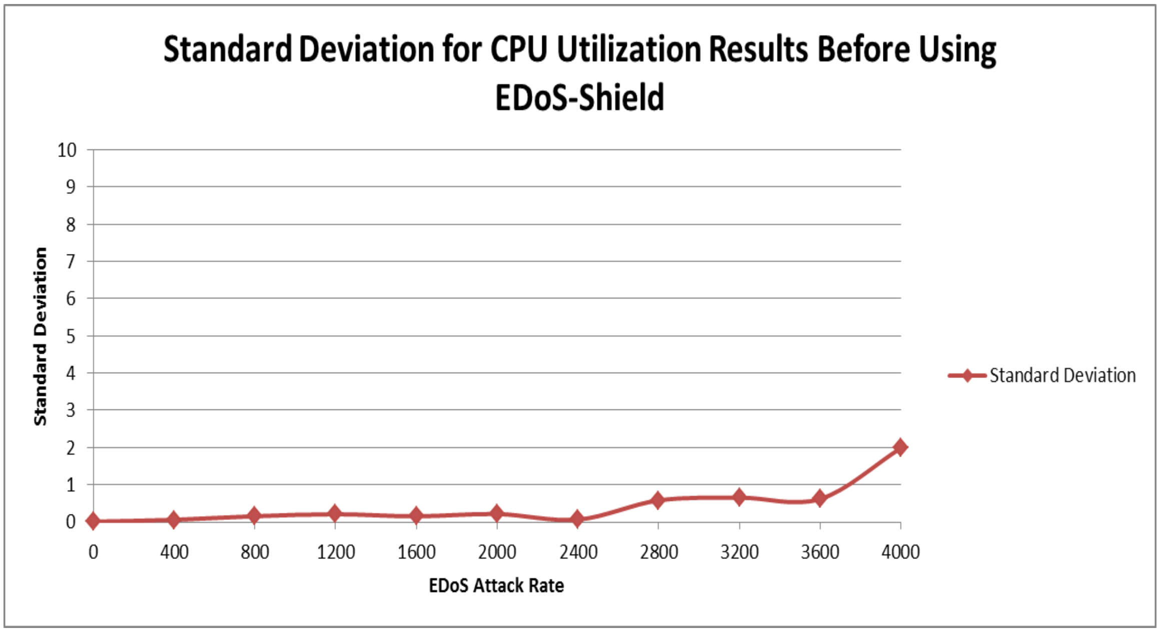


Figure 5.1: Standard Deviation for the CPU Utilization Results

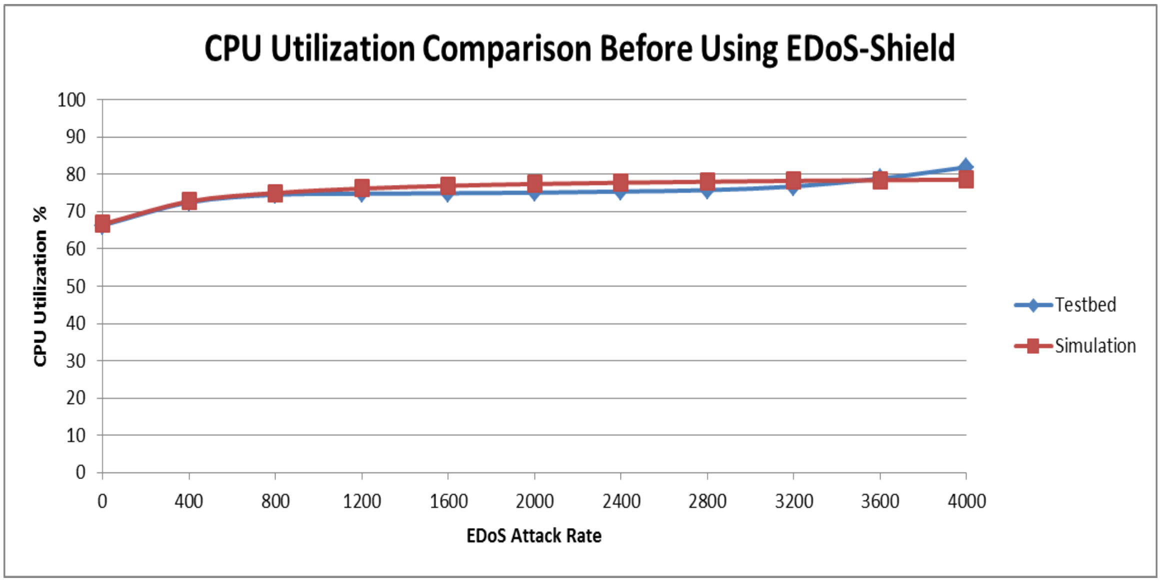


Figure 5.2: CPU Utilization Comparison before Using the EDoS-Shield

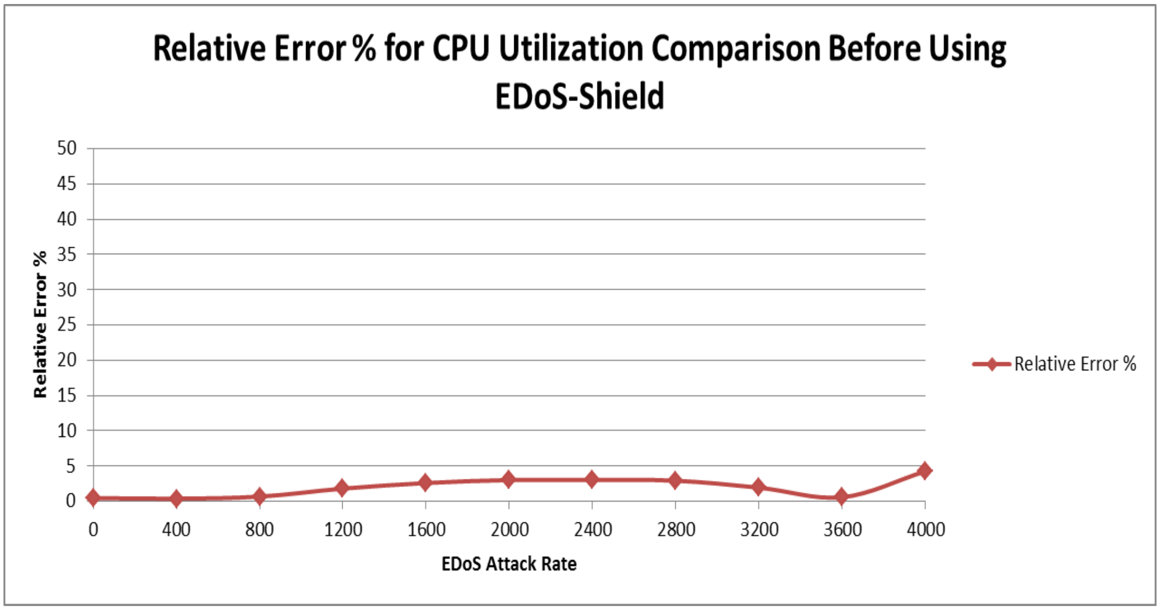


Figure 5.3: Relative Error Percentage for the CPU Utilization Comparison before Using the EDoS-Shield

Figure 5.4 shows the standard deviation for the response time results before using the EDoS-Shield. The standard deviation in Figure 5.4 illustrates that the results collected for the response time for each experiment are close to each other.

Figure 5.5 shows a comparison between the results obtained for the response time in both the testbed and simulation. Figure 5.5 shows that the response time results from the testbed are close to those of the simulation. The results for both the testbed and simulation in Figure 5.5 clearly show the effect of the EDoS attack on cloud computing in terms of response time. Figure 5.5 shows that when the rate of the EDoS attack increases, the response time will be greatly affected too. In addition to the delay that the users of the service provided by the cloud adopter will experience, more instances will be allocated to the cloud adopter if the autoscaling policy is based on the response time.

Figure 5.6 shows the relative error percentage for the response time comparison reported in Figure 5.5. Although the figure shows that the difference between the results of some of the experiments is around 16%, the maximum difference between the response time results of the testbed and the simulation is around 5 milliseconds.

In the next section, the results obtained after implementing the EDoS-Shield mitigation technique are discussed.

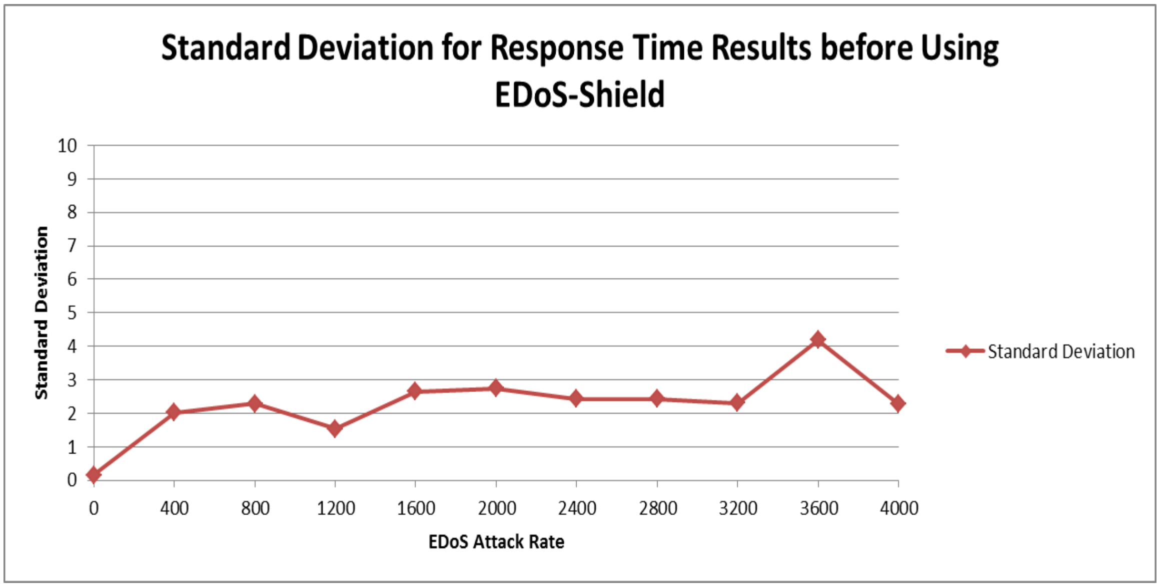


Figure 5.4: Standard Deviation for the Response Time Results Before Using the EDoS-Shield

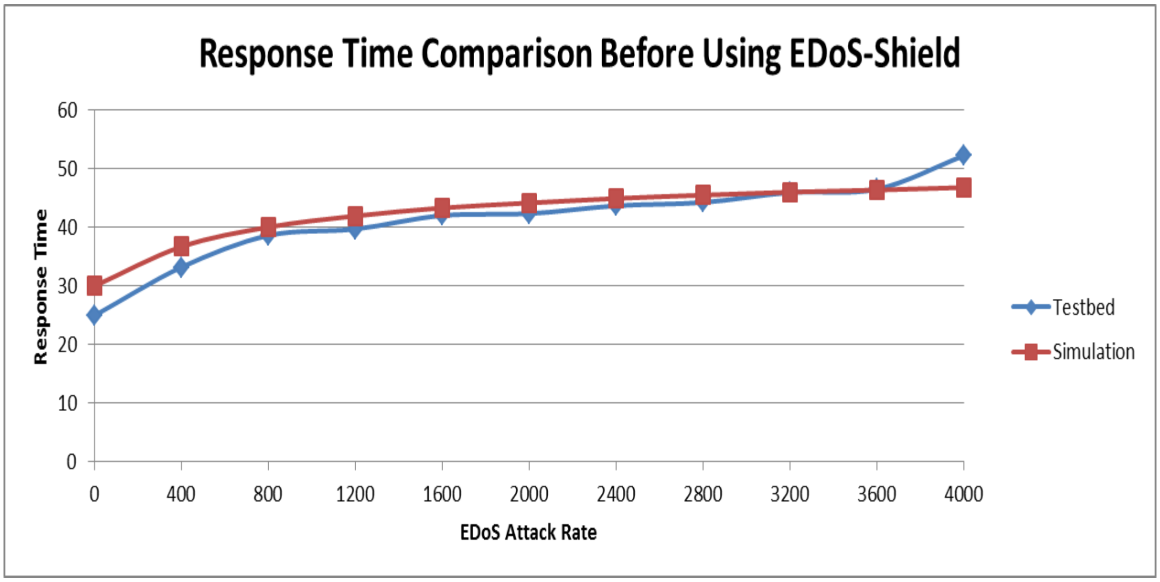


Figure 5.5: Response Time Comparison Before Using the EDoS-Shield

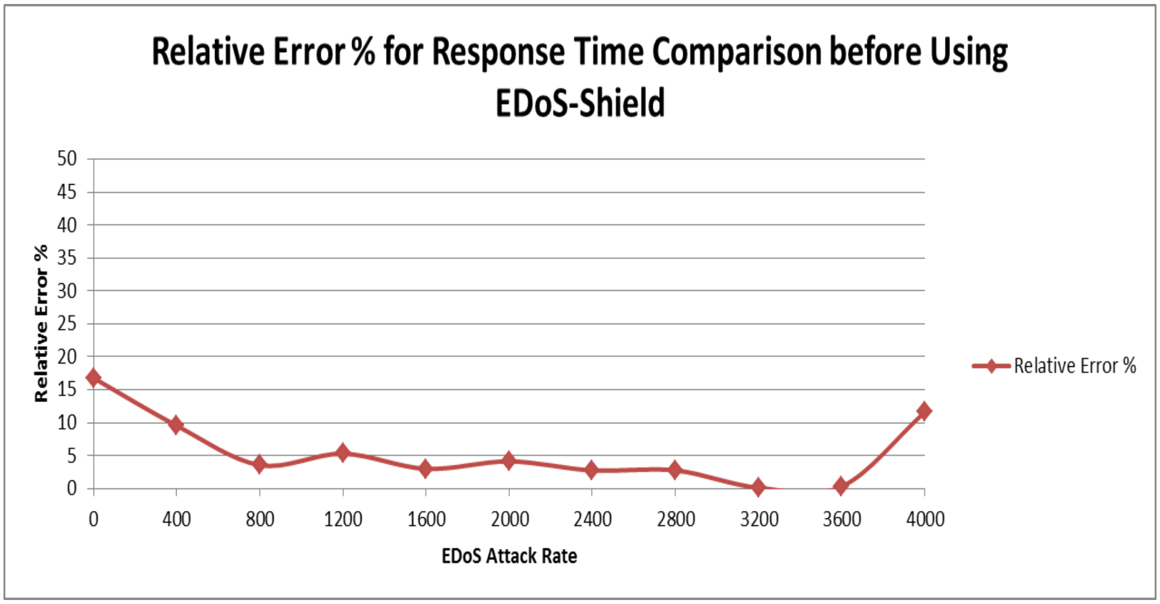


Figure 5.6: Relative Error Percentage for the Response Time Comparison Before Using the EDoS-Shield

5.2 Studying the EDoS-Shield Mitigation Technique

After studying the effect of the EDoS attack on cloud computing, the results of adding the EDoS-Shield mitigation technique will be discussed in this section.

After implementing and using the EDoS-Shield mitigation technique on the testbed, the dashboard of NetScaler showed that only the traffic of the legitimate traffic generator VMs is allowed to arrive to the cloud instances. The HTTP requests rate on the dashboard of NetScaler is always 400 Req/Sec, which is the rate of the legitimate traffic. All the traffic that comes from the attacking VMs that are blacklisted is dropped by the firewall. For this reason, the CPU utilization for both the testbed and simulation is almost fixed during all the experiments, as shown in Figure 5.7. Both the results of the testbed and the simulation in Figure 5.7 show that the EDoS-Shield is capable of eliminating the effect of the EDoS attack on the CPU utilization.

The relative error percentage for the CPU utilization comparison when using the EDoS-Shield is calculated and presented in Figure 5.8. As shown in the figure, the relative error percentage is always below 1%.

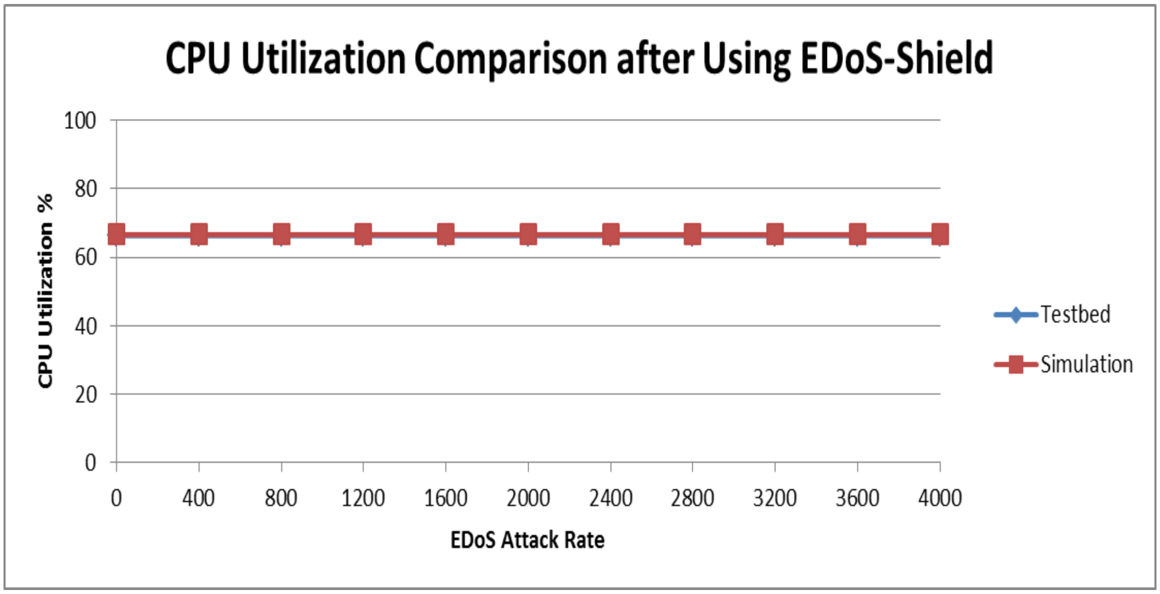


Figure 5.7: CPU Utilization Comparison After Using the EDoS-Shield

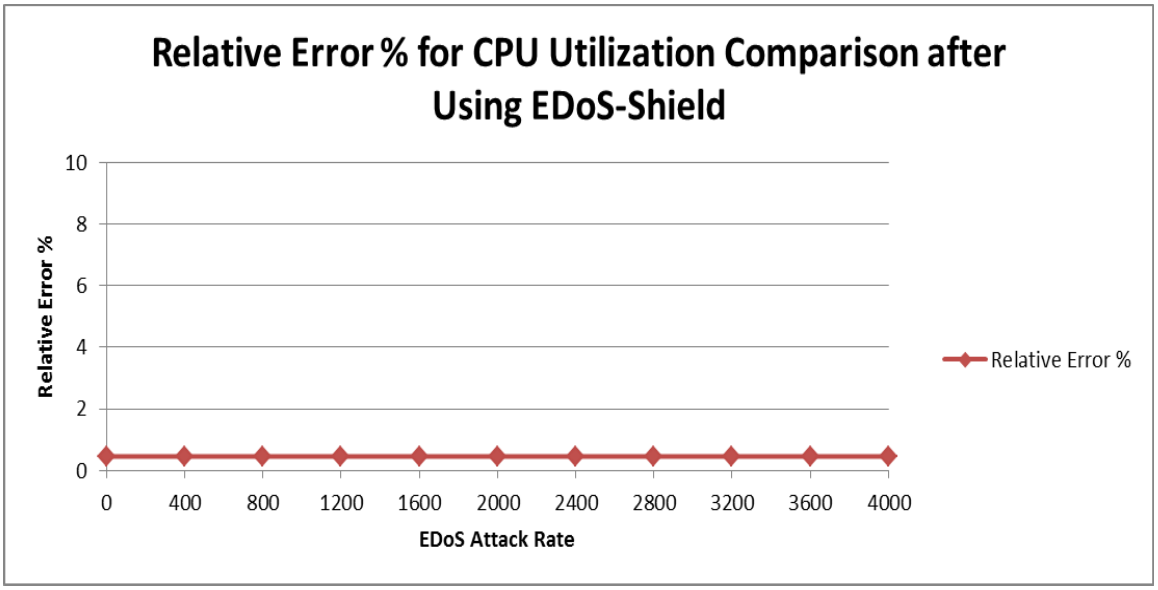


Figure 5.8: Relative Error Percentage for the CPU Utilization Comparison After Using the EDoS-Shield

The standard deviation for the response time results in Figure 5.9 illustrates that the response time results collected are close to each other. The figure shows no much difference in the collected results for the response time of each experiment.

Figure 5.10 shows a comparison between the values of the response time for both the testbed and simulation when using the EDoS-Shield mitigation technique. This shows that there is a small difference between the response time results of the testbed and the simulation when the EDoS-Shield is used. The slight increase in the values of response time as the EDoS attack rate increases, which is clear in the results of the testbed, is due to the packet processing time at the firewall. Figure 5.10 shows that the EDoS-Shield has significantly decreased the effect of the EDoS attack on the response time.

Figure 5.11 shows the relative error percentage for the response time comparison between the testbed and the simulation when the EDoS-Shield mitigation technique is used. The maximum difference is around 5 milliseconds in the first experiment.

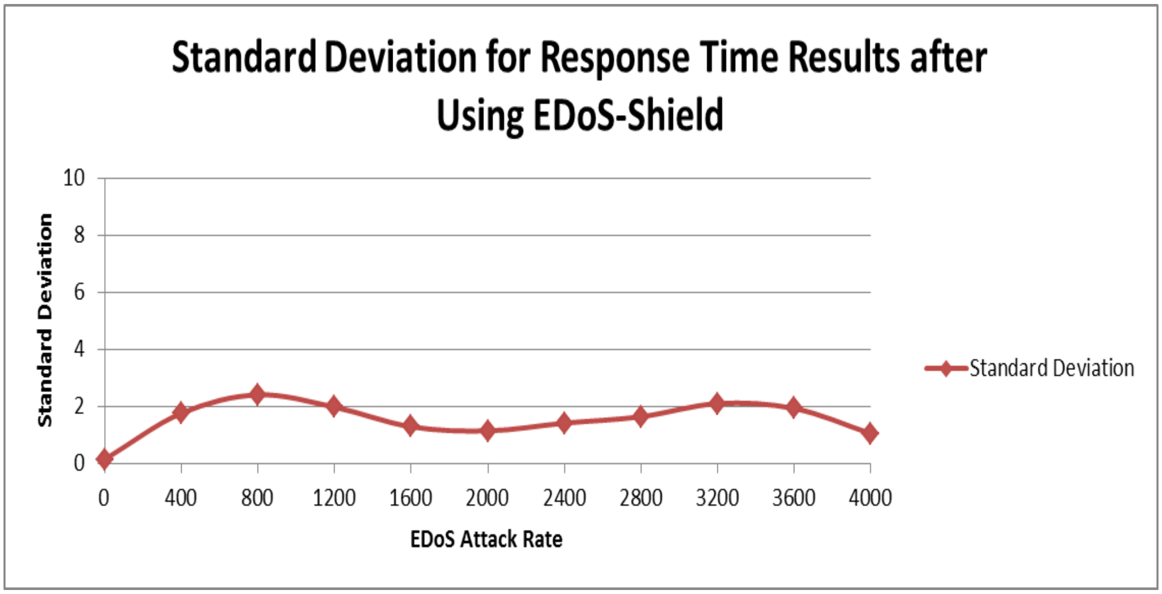


Figure 5.9: Standard Deviation for the Response Time Results After Using the EDoS-Shield

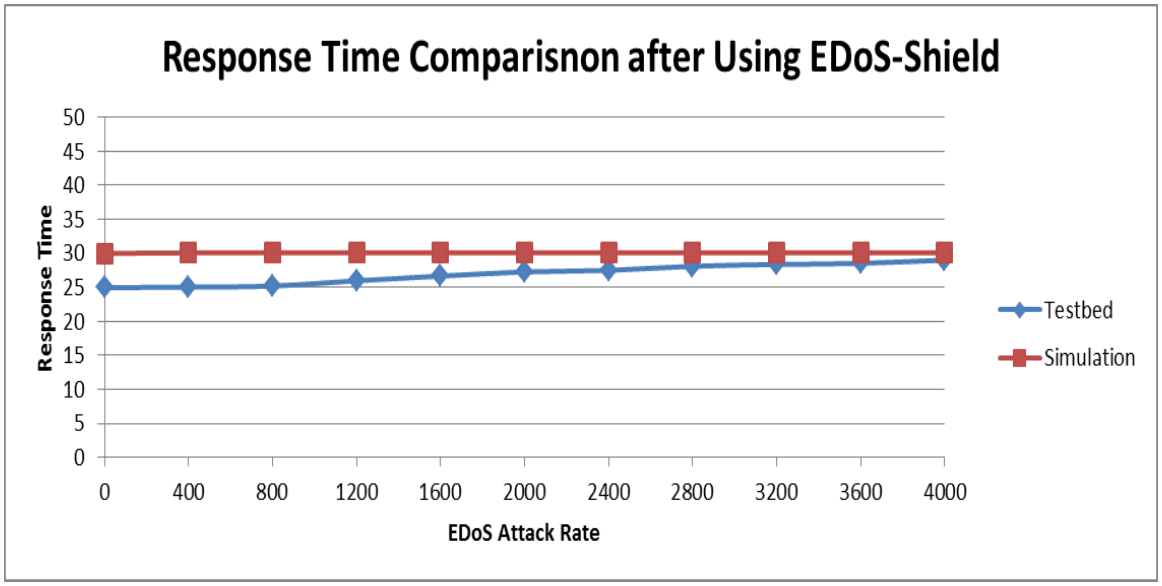


Figure 5.10: Response Time Comparison after Using the EDoS-Shield

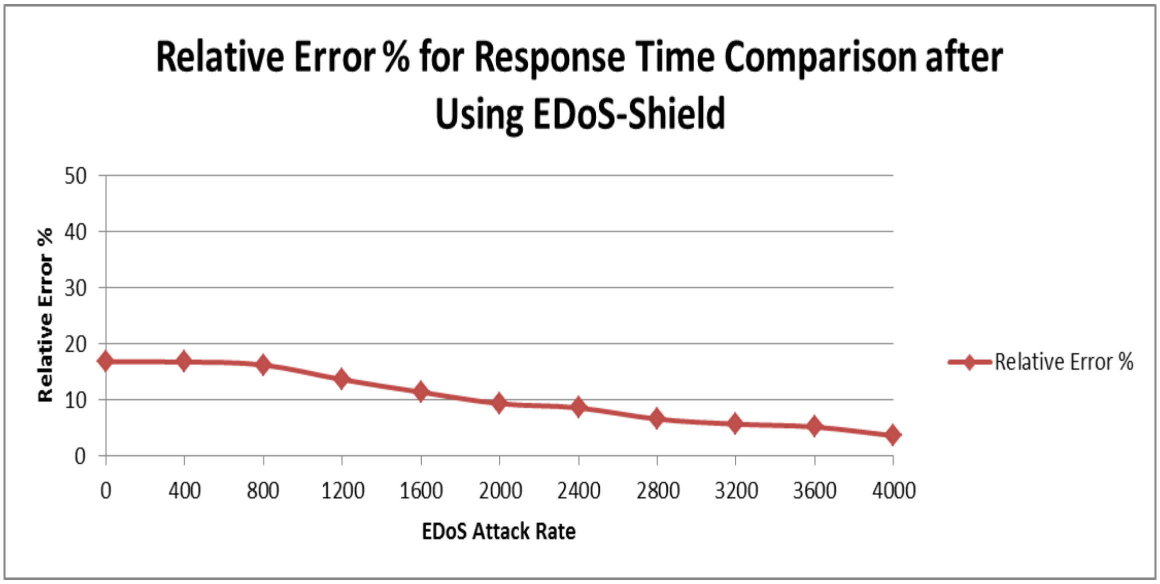


Figure 5.11: Relative Error Percentage for the Response Time Comparison after Using the EDoS-Shield

5.3 Challenges and Limitations

Below are some challenges and limitations of this work:

- Executing the experiments and collecting the results are performed manually. For instance, the attack is launched from JMeter instances manually, the CPU utilization results are collected manually via XenCenter, and the response time results are collected manually via Firebug. This process takes much time.
- XenCenter gives the CPU utilization results in integer numbers. The results of the simulation are given in real numbers, which makes them more precise. The current results of the testbed are not severely affected by this; however, we will be able to obtain more accurate values if another way is used to collect the results of the CPU utilization in real numbers, as in the case of simulation.
- The number of physical servers used in the testbed is limited. If more servers could be added to the testbed, then more experiments can be executed.
- In the current testbed, the malicious user is assumed to enter the CAPTCHA incorrectly in order to be added to the blacklist of the firewall. In real life scenario, the malicious user will not respond to CAPTCHA at all.

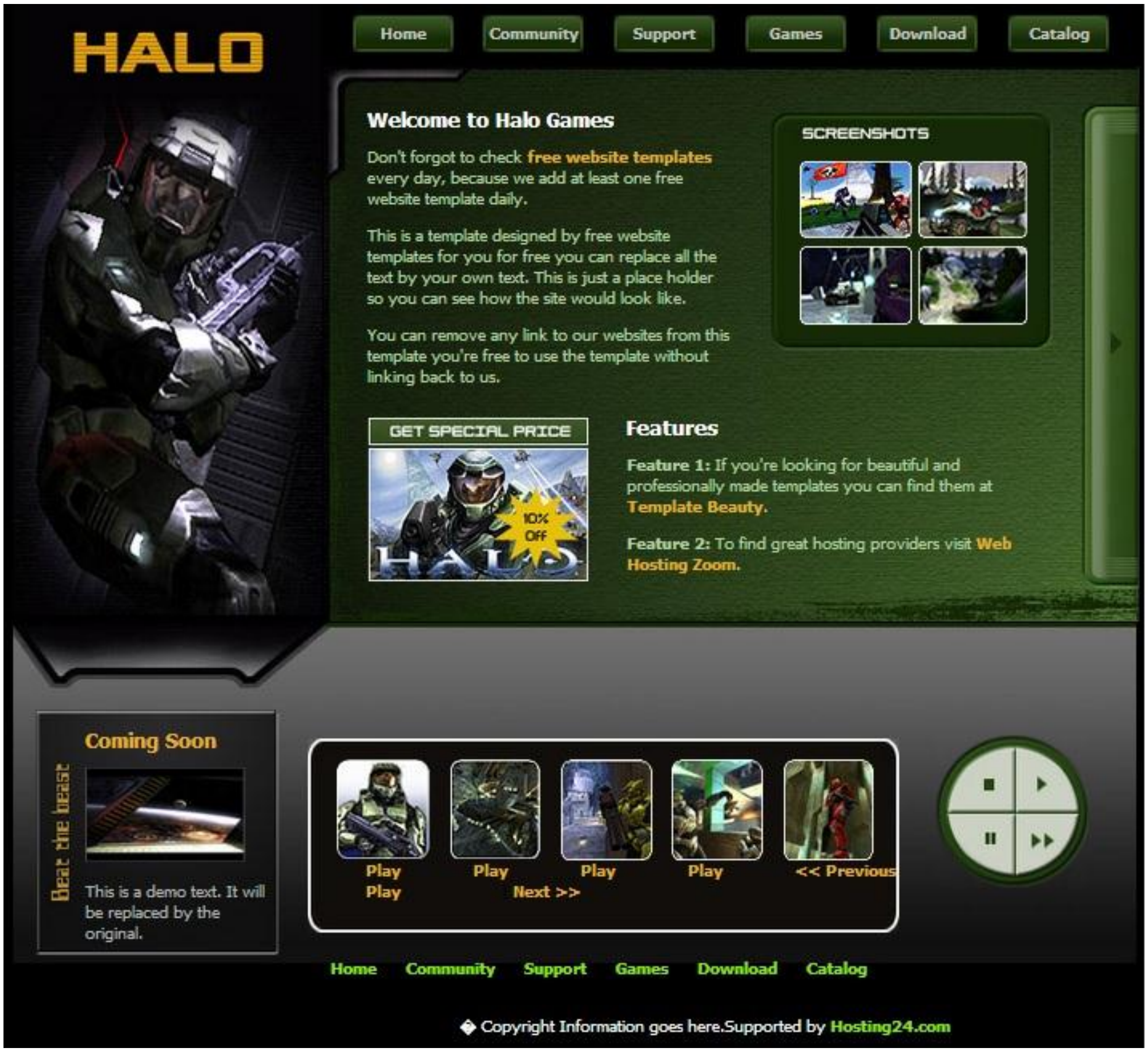
CHAPTER 6

EXPERIMENTS IN A REAL-LIFE ENVIRONMENT

The previous two chapters were dedicated to the validation of the simulation results. In this chapter, we study the effect of the EDoS attack in an environment that is very close to real-life. We also study the effect of using the EDoS-Shield in such environment. The major changes in the testbed are explained in the next section. The following two sections present and discuss the results of the experiments.

6.1 Testbed Setup Changes

The testbed setup used for these experiments is the same as that used to validate the simulation results. The only difference is the replacement of the template. The new template has a real website that has an index page with text and many pictures [50]. The index page has 24 elements that are downloaded to the browser of the client, with a total size of 507.4 KB. We added an additional picture to the index page to achieve this size and make it close to the size of that of modern websites like Yahoo. Figure 6.1 shows a snapshot of the index page.



Welcome to Halo Games

Don't forgot to check **free website templates** every day, because we add at least one free website template daily.

This is a template designed by free website templates for you for free you can replace all the text by your own text. This is just a place holder so you can see how the site would look like.

You can remove any link to our websites from this template you're free to use the template without linking back to us.

Features

Feature 1: If you're looking for beautiful and professionally made templates you can find them at **Template Beauty.**

Feature 2: To find great hosting providers visit **Web Hosting Zoom.**

Coming Soon

This is a demo text. It will be replaced by the original.

◆ Copyright Information goes here.Supported by **Hosting24.com**

Figure 6.1: A Snapshot of the Index Page of the New Website Template

We used the small size service offering for the VMs hosting this website. We found that it is better to set to 40% the upper CPU utilization threshold that indicates the need of creating additional instances. The reason behind this decision is the fact that we found that the 40% CPU utilization is achieved by sending around 1600 HTTP requests per second. This means that a rate of 3200 Req/Sec is needed to achieve the 80% CPU utilization. However, we found that NetScaler will not allow more than approximately 2100 HTTP Req/Sec to pass through it in these experiments. At rates higher than this, the throughput will exceed the limit permitted by the current license of NetScaler, i.e., 200 Mbps. This is due to the size of the new website which is 507.4 KB as opposed to 580 Bytes used in the previous experiments.

Since the maximum HTTP requests rate that can be used is these experiments is around 2100 Rec/Sec, we performed the experiments using the EDoS attack rates of 0, 400, 800, 1200, and 1600. In addition, 400 Req/Sec rate is used as legitimate traffic. Hence, the maximum traffic rate that is used in the experiments is 2000 Req/Sec. The maximum number of instances used in the experiments before using the EDoS-Shield is 2, while the number of instances is always equal to 1 when using the EDoS-Shield. This is shown in Figure 6.2 and Figure 6.3 respectively.

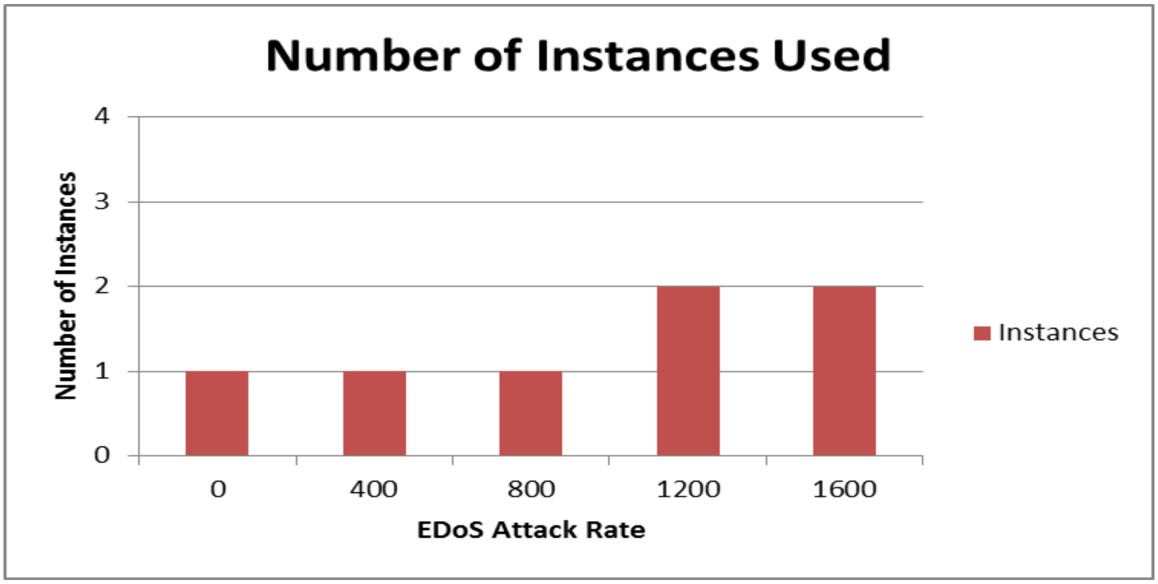


Figure 6.2: Number of Instances Used before Using the EDoS-Shield

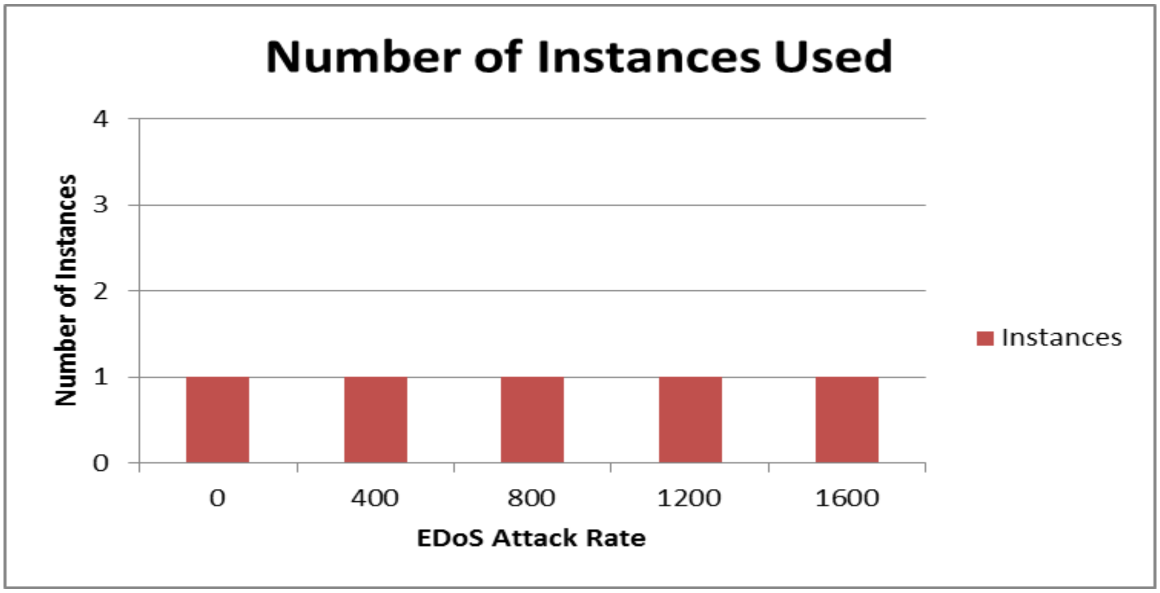


Figure 6.3: Number of Instances Used after Using the EDoS-Shield

6.2 The Effect of the EDoS Attack on the Cloud

Figure 6.4 shows the CPU utilization before using the EDoS-Shield. It is clear that the CPU utilization increases in the first three experiments as the attack rate increases. Then, it gets low again at the attack rate of 1200 Req/Sec. At this attack rate, the total traffic rate is 1600, when adding the legitimate traffic. This will results in around 40% CPU utilization. As a result, a new instance is created, and the CPU utilization will decrease by nearly a half, as shown in Figure 6.4. The CPU utilization increases again when increasing the EDoS attack rate to 1600 Req/Sec. This behavior is the same as that of the results reported in Chapter 5 in illustrating that more computing resources will be allocated to the cloud as the EDoS attack rate increases. This is because the CPU utilization increases as the EDoS attack rate increases.

Figure 6.5 shows the response time results before using the EDoS-Shield. As shown in the figure, the EDoS attack has a severe effect on the response time if no mitigation technique is used. This behavior is the same as discussed previously in Chapter 5.

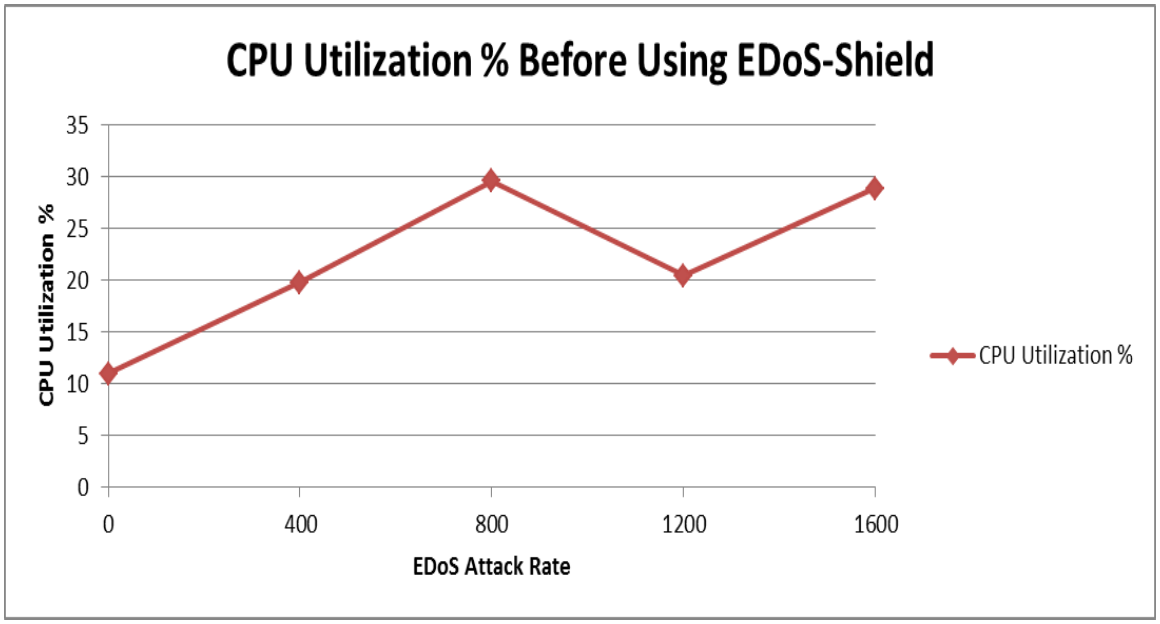


Figure 6.4: CPU Utilization before Using the EDoS-Shield

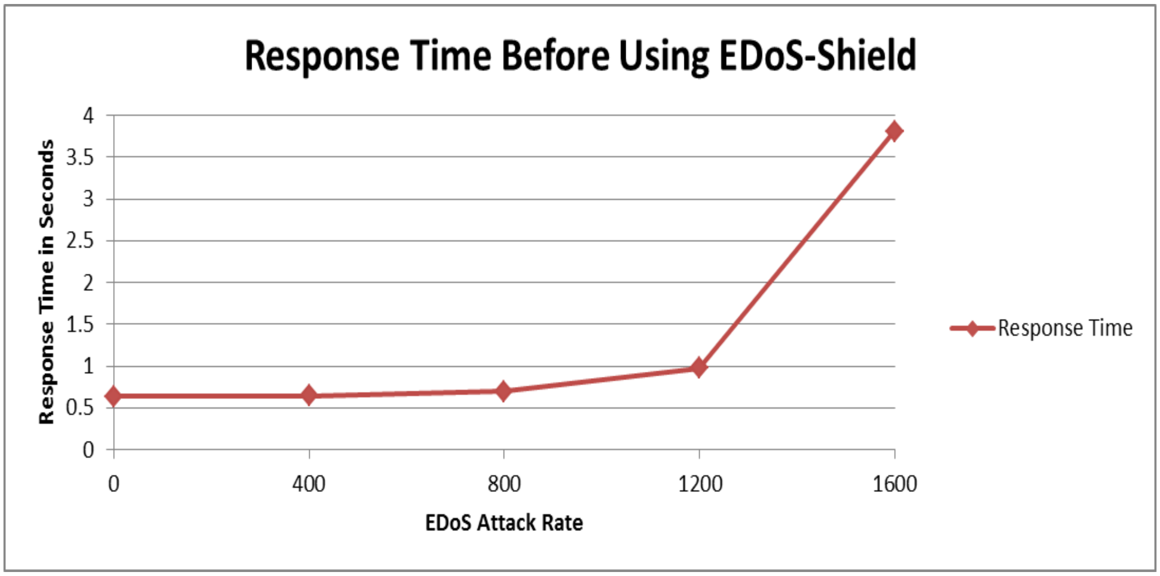


Figure 6.5: The Response Time before Using the EDoS-Shield

6.3 The Effect of Using the EDoS-Shield

Figure 6.6 presents the results of the CPU utilization after using the EDoS-Shield. The CPU utilization is almost fixed, and the number of instances is always 1. The dashboard of NetScaler shows that only the legitimate traffic can access the cloud services. This behavior is the same as that reported in Figure 5.7.

In Figure 6.7, the results of the response time after using the EDoS-Shield are presented. The response time increases slightly as the attack rate increases because of the packet processing at the firewall. However, this increase is significantly below the results of the response time of Figure 6.5, when no mitigation technique is used at all. This is clearly illustrated in Figure 6.8, which shows the comparison between the results of the response time before and after using the EDoS-Shield.

The results of this chapter confirm the results of Chapter 4 in illustrating the severe effect of the EDoS attack on the CPU utilization and the response time. The results of both chapters also confirm that the use of the EDoS-Shield can significantly minimize the effect of the EDoS attack on cloud computing.

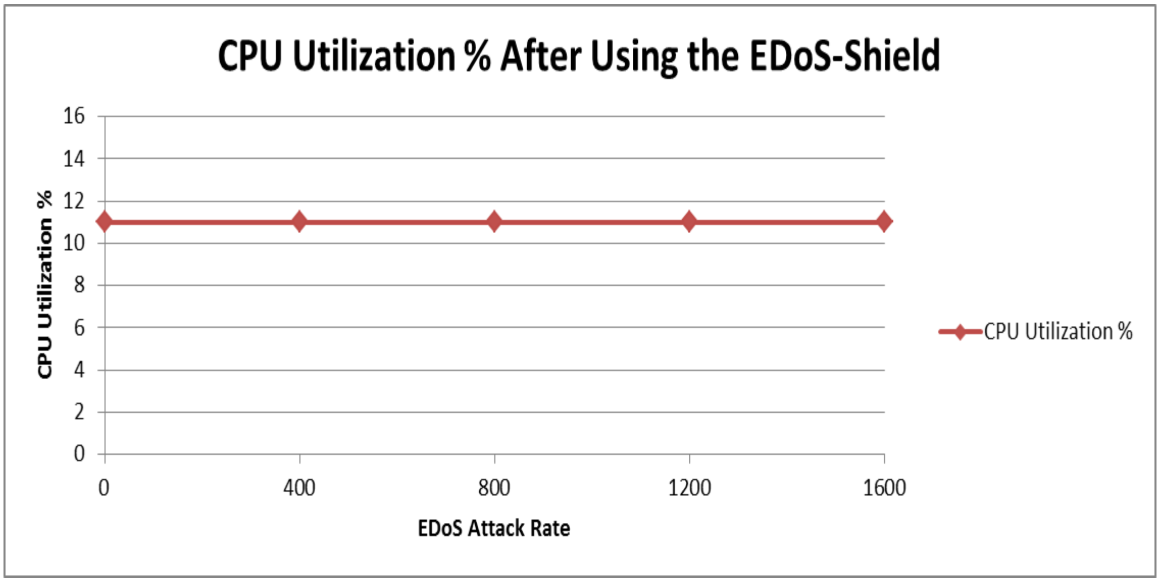


Figure 6.6: CPU Utilization after Using the EDoS-Shield

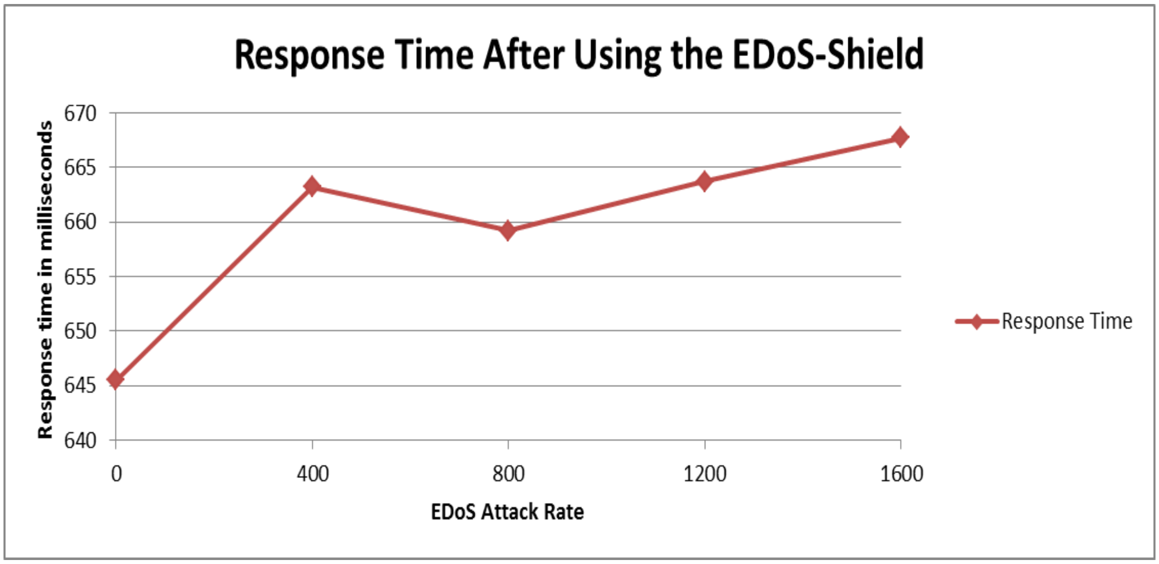


Figure 6.7: Response Time after Using the EDoS-Shield

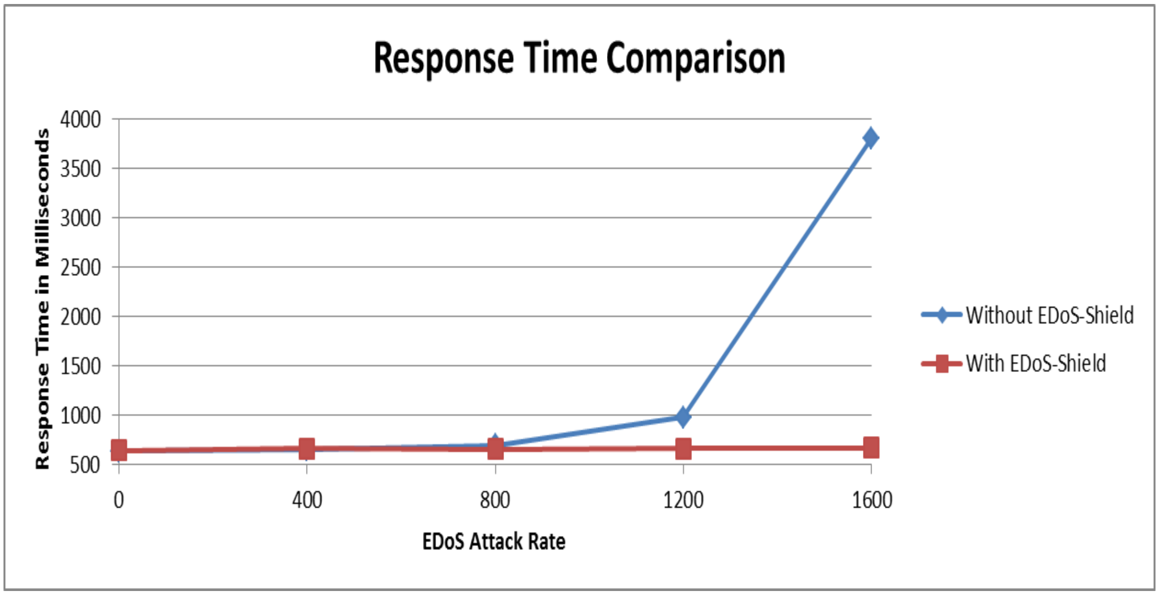


Figure 6.8: Response Time Comparison before and after Using the EDoS-Shield

CHAPTER 7

CONCLUSION AND FUTURE WORK

Cloud computing is considered one of the hottest IT topics today. Many large organizations are interested in cloud computing because of its elasticity, pay per use, and other benefits that it provides. But regardless of its great advantages, the security of cloud computing is still in its infancy. Many new attacks have been developed especially for the cloud.

In this work, we reviewed the literature to study the security of cloud computing. We surveyed the literature for the attacks that target the cloud computing, and proposed a taxonomy for these attacks based on cloud hierarchy level targeted by the attack.

The DDoS attack was also studied in this work since it is the major cause of the EDoS attack, when transformed from conventional networks to cloud computing. A multi-dimensional taxonomy for the DDoS attacks was also proposed in this work.

After that, the EDoS attack was studied in detail. We surveyed the literature for the attacks that may result in an EDoS attack. We also provided a taxonomy for the EDoS attacks based on the way an attack may cause an EDoS attack. This taxonomy was proposed this may so that a single mitigation technique for an EDoS attack category may be used to block all the attacks under that category. The existing mitigation techniques for the EDoS attack were also reviewed.

The major part of this work is the design and implementation of an experimental testbed that was used to evaluate the effectiveness of the EDoS-Shield mitigation technique in blocking the EDoS attack.

The experimental results of the testbed showed that the EDoS attack has a severe effect on the cloud, in terms of computing resources utilization and response time, if no mitigation technique is used. The results of the testbed confirmed those of the simulation in that as the EDoS attack rate increases, both the computing resources utilization and response time will increase too. The fluctuation in the response time increases as the EDoS attack rate increases.

The EDoS-Shield was implemented on the testbed to evaluate its effectiveness. The results of the testbed show that when the EDoS-Shield is used, the EDoS attack traffic will be dropped by the firewall. The EDoS-Shield showed that it can greatly reduce the computing resources that will be allocated due to an attack. This in turn will save the cloud adopter from paying for computing resources reserved for an attack. The EDoS-Shield also improved the response time significantly.

Both the results of the testbed and the simulation are confirming each other in illustrating that the EDoS-Shield is capable of blocking the EDoS attack and provide significant economic savings to the cloud adopter.

The ultimate completing of this work in the future is to implement the enhanced EDoS-Shield in an experimental testbed and evaluate it. The enhanced EDoS-Shield is capable of detecting spoofed IP addresses that are used by malicious users pretending to be legitimate.

References

[1] Mohammed H. Sqalli, “Modeling and Mitigation of Economic Denial of Sustainability (EDoS) Attacks in Cloud Computing”, NSTIP Research Proposal, KFUPM.

[2] International Data Corporation, "New IDC IT Cloud Services Survey: Top Benefits and Challenges", IDC, December 2009. http://blogs.idc.com/ie/wp-content/uploads /2009/ 12/idc_cloud_challenges_2009.jpg

[3] Mansfield-Devine, S., “Danger in the clouds”. Network Security, December 2008, pp. 9-11.

[4] Kaufman, L. M., “Data Security in the World of Cloud Computing”, IEEE Security & Privacy, July 2009, pp. 61-64.

[5] T. Mather, S. Kumaraswamy, and S. Latif,"Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, Inc., 2009.

[6] C. Hoff, ”Cloud Computing Security: From DDoS (Distributed Denial Of Service) to EDoS (Economic Denial of Sustainability)”, Retrieved November 27, 2008, from http://rationalsecurity.typepad.com/blog/2008/11/cloud-computing-security-from-ddos-distributed-denial-ofservice-to-edos-economic-denial-of-sustaina.html

[7] Gruschka, N., and M. Jensen. “Attack Surfaces: A Taxonomy for Attacks on Cloud Services.” In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference On, 276–279, 2010.

[8] Vaquero, L. M., L. Rodero-Merino, and D. Morán. “Locking the Sky: a Survey on IaaS Cloud Security.” Computing 91, no. 1 (2011): 93–118.

[9] Ristenpart, T., E. Tromer, H. Shacham, and S. Savage. “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds.” In Proceedings of the 16th ACM Conference on Computer and Communications Security, 199–212, 2009.

[10] J. Rutkowska Introducing Blue Pill, [online] Available:

http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html

[11] Jensen, M.; Schwenk, J.; Gruschka, N.; Iacono, L.L.; , "On Technical Security Issues in Cloud Computing," Cloud Computing, 2009. CLOUD '09. IEEE International Conference on , vol., no., pp.109-116, 21-25 Sept. 2009.

[12] CERT CC. Denial of Service Attacks, http://www.cert.org/tech_tips/denial_of_service .html

[13] Kumar, P.A.R.; Selvakumar, S.; , "Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms," Advance Computing Conference, 2009. IACC 2009. IEEE International , vol., no., pp.1275-1280, 6-7 March 2009.

[14] VivinSandar, S., and S. Shenai. “Economic Denial of Sustainability (EDoS) in Cloud Services Using HTTP and XML Based DDoS Attacks.” International Journal of Computer Applications 41, no. 20 (2012): 11–16.

[15] Sqalli, M. H., F. Al-Haidari, and K. Salah. “EDoS-Shield-A Two-Steps Mitigation Technique Against EDoS Attacks in Cloud Computing.” In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference On, 49–56, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6123480.

[16] Al-Haidari, F., M. H. 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, 1167–1174, 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6296109.

[17] Mirkovic, J., and P. Reiher. “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms.” ACM SIGCOMM Computer Communication Review 34, no. 2 (2004): 39–53.

[18] Asosheh Dr, A., and N. Ramezani. “A Comprehensive Taxonomy of DDOS Attacks and Defense Mechanism Applying in a Smart Classification.” WSEAS Transactions on Computers 7, no. 4 (2008): 281–290.

[19] Douligeris, C., and A. Mitrokotsa. “DDoS Attacks and Defense Mechanisms: Classification and State-of-the-art.” Computer Networks 44, no. 5 (2004): 643–666.

[20] Specht, S. M., and R. B. Lee. “Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures.” In Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems, 543–550, 2004. http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.

[21] Bhadauria, R., and S. Sanyal. “Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques.” arXiv Preprint arXiv:1204.0764 (2012). http://arxiv.org/abs/1204.0764.

[22] Jangra, A., and R. Bala. “A Survey on Various Possible Vulnerabilities and Attacks in Cloud Computing Environment” (2012). http://researchmanuscripts.com/IJCBRJanuary2012/4.pdf.

[23] Brodkin, J. “Gartner: Seven Cloud-computing Security Risks.” Infoworld (2008): 1–3.

[24] Top Threats to Cloud Computing. Cloud Security Alliance

[25] Che, J., Y. Duan, T. Zhang, and J. Fan. “Study on the Security Models and Strategies of Cloud Computing.” Procedia Engineering 23 (2011): 586–593.

[26] Gruschka, N., and M. Jensen. “Attack Surfaces: A Taxonomy for Attacks on Cloud Services.” In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference On, 276–279, 2010. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5557984.

[27] Khorshed, M. T., A. B. M. Ali, and S. A. Wasimi. “A Survey on Gaps, Threat Remediation Challenges and Some Thoughts for Proactive Attack Detection in Cloud Computing.” Future Generation Computer Systems (2012). http://www.sciencedirect.com/science/article/pii/S0167739X12000180.

[28] Grobauer, B., T. Walloschek, and E. Stocker. “Understanding Cloud Computing Vulnerabilities.” Security & Privacy, IEEE 9, no. 2 (2011): 50–57.

[29] Subashini, S., and V. Kavitha. “A Survey on Security Issues in Service Delivery Models of Cloud Computing.” Journal of Network and Computer Applications 34, no. 1 (2011): 1–11.

[30] Jensen, M., J. Schwenk, N. Gruschka, and L. L. Iacono. “On Technical Security Issues in Cloud Computing.” In Cloud Computing, 2009. CLOUD’09. IEEE International Conference On, 109–116, 2009. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5284165.

[31] S. HinKhor and A. Nakao, "sPoW: On-Demand Cloud-based eDDoS Mitigation Mechanism", HotDep (Fifth Workshop on Hot Topics in System Dependability), Lisbon, Portugal, 2009.

[32] Modi, C., D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan. “A Survey of Intrusion Detection Techniques in Cloud.” Journal of Network and Computer

Applications (2012). http://www.sciencedirect.com/science/article/pii/S1084804512001178.

[33] The Open Web Application Security Project (OWASP), “Buffer Overflow”, February 2009. https://www.owasp.org/index.php/Buffer_Overflow

[34] Kieyzun, A.; Guo, P.J.; Jayaraman, K.; Ernst, M.D.; , "Automatic creation of SQL Injection and cross-site scripting attacks," Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on , vol., no., pp.199-209, 16-24 May 2009

[35] Juan Chen; Chuanxiong Guo; , "Online Detection and Prevention of Phishing Attacks," Communications and Networking in China, 2006. ChinaCom '06. First International Conference on , vol., no., pp.1-7, 25-27 Oct. 2006

[36] Oppliger, R.; Rytz, R.; Holderegger, T.; , "Internet Banking: Client-Side Attacks and Protection Mechanisms," Computer , vol.42, no.6, pp.27-33, June 2009.

[37] Naresh Kumar, M.; Sujatha, P.; Kalva, V.; Nagori, R.; Katukojwala, A.K.; Kumar, M., "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service," Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on , vol., no., pp.535,539, 3-5 Nov. 2012

[38] CloudPlatform, http://www.citrix.com/products/cloudplatform/overview.html, 2013

[39] XenServer, http://www.citrix.com/products/xenserver/overview.html, 2013

[40] CentOS, http://www.centos.org/, 2013

[41] Apache HTTP Server, http://httpd.apache.org/, 2013

[42] NetScaler, http://www.citrix.com/products/netscaler-application-delivery-controller/overview.html, 2013

[43] Apache JMeter, http://jmeter.apache.org/, 2013

[44] JMeter Plugins, http://jmeter-plugins.org/, 2013

[45] iptables, http://www.netfilter.org/, 2013

[46] WampServer, http://www.wampserver.com/en/, 2013

[47] SecurImage PHP CAPTCHA, http://www.phpcaptcha.org/, 2013

[48] XenCenter, http://www.xenserver.org/overview-xenserver-open-source-virtualization/download.html, 2013

[49] Firebug, http://getfirebug.com/, 2013

[50] 000Webhost.com, http://www.000webhost.com/templates/Games/template_45, 2014

Vitae

- Name: Saeed Omar Saeed Alsowail
- Nationality: Yemeni
- Date of Birth: 10/20/1984
- Email: Alsowail.kfupm@gmail.com
- Address: Mukalla, Yemen
- Phone: 00966558818916
- Received B.Eng. in Information Technology from University of Aden, Aden, Yemen.
- Worked in Cisco Networking Academy as a Lab Instructor from Feb, 2010 to Aug, 2010.
- Completed M.S in Computer Networks from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in December 2013.