

5-2016

## Impact of virtualization on cloud network security

Koushicaa Sundar

*The University of Texas Rio Grande Valley*

Follow this and additional works at: <https://scholarworks.utrgv.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

---

### Recommended Citation

Sundar, Koushicaa, "Impact of virtualization on cloud network security" (2016). *Theses and Dissertations*. 93.

<https://scholarworks.utrgv.edu/etd/93>

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact [justin.white@utrgv.edu](mailto:justin.white@utrgv.edu), [william.flores01@utrgv.edu](mailto:william.flores01@utrgv.edu).

IMPACT OF VIRTUALIZATION ON CLOUD NETWORK SECURITY

A Thesis

by

KOUSHICAA SUNDAR

Submitted to the Graduate College of  
The University of Texas Rio Grande Valley  
In partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

May 2016

Major Subject: Electrical Engineering



# IMPACT OF VIRTUALIZATION ON CLOUD NETWORK SECURITY

A Thesis  
by  
KOUSHICAA SUNDAR

## COMMITTEE MEMBERS

Dr. Sanjeev Kumar  
Chair of Committee

Dr. Jun Peng  
Committee Member

Dr. Wenjie Dong  
Committee Member

May 2016



Copyright 2016 Koushicaa Sundar  
All Rights Reserved

## ABSTRACT

Sundar, Koushicaa, Impact of Virtualization on Cloud Network Security. Master of Science (MS), May, 2016, 121 pp., 75 figures, 66 references.

In this thesis, experimental evaluation of the effect of virtualization on the availability of servers has been performed under Distributed Denial of Service (DDoS) attacks for popular server Operating Systems such as Windows Server 2008 R2, Windows Server 2012 R2 virtualized using Hyper-V. A comparative evaluation of the performance of the servers before and after virtualization under DDoS attacks indicates that after virtualization there is a considerable increase in the vulnerability to attacks and a decline in the performance of the virtualized server compared to when the server is not virtualized.





## DEDICATION

The completion of my Master's would not have been possible without the blessings of Lord Vishnu. I would like to dedicate my work to my parents, Shri P.Sundar and Smt. S.Vasanthi, and my brother P.S.Agnideven, for their love, constant support and belief in my abilities.



## ACKNOWLEDGEMENTS

I would like to formally thank:

Dr. Sanjeev Kumar, my Advisor and Committee Chair, for his motivation, continued support and unwavering belief in my abilities. I had the privilege of learning so much from you, I owe my knowledge in the field of Network Security to you and without your able guidance this thesis would not have been possible.

Dr. Jun Peng and Dr. Wenjie Dong for their willingness to serve as committee members. Thank you for devoting your time. Thank you for the support and guidance.

Dr. Heinrich D. Foltz, for the support and encouragement he has provided. Thank you for giving me an opportunity to prove my teaching skills.

My fellow Graduate students in the NRL Lab, Rodolfo Baez Jr, David Leal, Edni Del Rosal and Ganesh Reddy for the technical discussions and knowledge-sharing. Thank you so much for your support.



## TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES.....	ix
CHAPTER I. INTRODUCTION.....	1
1.1 Problem Statement.....	2
1.2 Distributed Denial of Service Attacks.....	3
1.2.1 Ping Flood Attack.....	5
1.2.2 Smurf Attack.....	6
1.2.3 TCP-SYN Flood Attack.....	7
1.2.4 UDP Flood Attack.....	9
1.3 Cloud Computing and Virtualization.....	11
1.4 Thesis Outline.....	12
CHAPTER II. EVALUATION OF THE IMPACT OF DDoS ATTACKS ON WINDOWS SERVER 2012 R2 BEFORE VIRTUALIZATION.....	14
2.1 Experimental Setup.....	16
2.2 Parameters of Performance Evaluation.....	19

2.3 Results and Discussion.....	22
2.3.1 Ping Flood Attack.....	22
2.3.2 Smurf Attack.....	30
2.3.3 TCP-SYN Flood Attack.....	37
2.3.3.1 Blue Screen of Death (BSoD) under TCP SYN Attack.....	44
2.3.4 UDP Flood Attack.....	48
2.4 Chapter Summary.....	57
 CHAPTER III. EVALUATION OF THE EFFECT OF VIRTUALIZATION ON THE AVAILABILITY OF SERVERS UNDER DDoS ATTACKS .....	 58
3.1 Experimental Setup.....	59
3.2 Parameters of Performance Evaluation.....	64
3.3 Results and Discussion.....	67
3.3.1 Ping Flood Attack.....	67
3.3.2 Smurf Attack.....	75
3.3.3 TCP-SYN Flood Attack.....	81
3.3.4 UDP Flood Attack.....	86
3.4 Chapter Summary.....	93
 CHAPTER IV. EVALUATION OF THE IMPACT OF DDoS ATTACKS IN A MULTI-VM ENVIRONMENT .....	 94
4.1 Experimental Setup.....	95
4.2 Parameters of Performance Evaluation.....	98
4.3 Results and Discussion.....	99

4.3.1 Ping Flood Attack.....	99
4.3.2 Smurf Attack.....	100
4.3.3 TCP-SYN Flood Attack.....	101
4.3.4 UDP Flood Attack.....	102
4.3.5 Effect of the number of cores allocated on the Performance of a Virtual Machine under DDoS Attacks.....	103
4.3.6 Effect of the number of Virtual Machines on the Hyper-V Host under DDoS Attacks.....	110
4.4 Chapter Summary.....	112
CHAPTER V. CONCLUSIONS AND FUTURE WORK.....	114
REFERENCES.....	116
BIOGRAPHICAL SKETCH.....	121





## LIST OF FIGURES

	Page
Figure 1.1: DDoS Attack launched on a Victim Web Server.....	5
Figure 1.2: Victim server being attacked by botnets with Smurf Attack Traffic.....	6
Figure 1.3: Legitimate three-way handshake.....	7
Figure 1.4: Victim Server under the impact of TCP/SYN Flood Attack.....	8
Figure 2.1: Experimental Setup of the server.....	18
Figure 2.2: Average Processor Utilization of the Victim Web Server under Ping Attack.....	23
Figure 2.3: Core Utilization of the Victim Web Server under Ping Flood Attack.....	24
Figure 2.4: TCP Close sequence between a Web Server and a Client.....	25
Figure 2.5: Nonpaged Pool Allocations in the Server under Ping Flood Attack.....	27
Figure 2.6: HTTP Connections established between Clients and the Victim Server under Ping Flood Attack.....	28
Figure 2.7: HTTP Connection Latency of the Victim Server under Ping Flood Attack.....	29
Figure 2.8: Average Processor Utilization of the victim server under Smurf Attack.....	30
Figure 2.9: Core Utilization and Average Processor Utilization of the Victim Server under Smurf Attack.....	32
Figure 2.10: Nonpaged Pool allocation in the victim server under Smurf Attack.....	32
Figure 2.11: HTTP Connections Established by the victim server under Smurf Attack.....	33
Figure 2.12: Connection Latency when Smurf attack traffic is sent to two ports.....	35
Figure 2.13: Connection Latency when Smurf attack traffic is sent to four ports.....	36

Figure 2.14: Connection Latency when Smurf attack traffic is sent to two ports.....	37
Figure 2.15: Average Processor Utilization of server under TCP-SYN Flood Attack.....	38
Figure 2.16: Core Utilization and Average Processor Utilization under TCP-SYN Flood Attack.....	39
Figure 2.17: Nonpaged Pool Allocation of the victim server under TCP-SYN Flood Attack.....	40
Figure 2.18: Number of legitimate Connections Established per second under TCP-SYN Flood Attack.....	41
Figure 2.19: Connection Latency under TCP-SYN Flood Attack.....	42
Figure 2.20: Connection Latency when TCP-SYN Flood Attack traffic is sent to six ports.....	43
Figure 2.21: HTTP Connection Establishment of the victim server under TCP-SYN Flood Attack.....	45
Figure 2.22: Number of HTTP connections handled by the server under TCP-SYN Flood Attack.....	45
Figure 2.23: Blue Screen of Death (BSoD) displayed before the server crashed under 3.1 Gbps Attack Traffic.....	46
Figure 2.24: Duration of time the server is able to withstand the TCP-SYN Flood Attack Traffic before crashing.....	47
Figure 2.25: Average Processor Utilization of the server under UDP Flood Attack.....	49
Figure 2.26: Core Utilization and Average Processor Utilization under UDP Flood Attack.....	50
Figure 2.27: Nonpaged Pool Allocation in the server under UDP Flood Attack.....	51
Figure 2.28: HTTP Connections established per second under UDP Flood Attack.....	52
Figure 2.29: Connection Latency of the server under UDP Flood Attack.....	53
Figure 2.30: Connection Latency of the server when UDP Flood Attack traffic is sent to six ports .....	54

Figure 2.31: Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on Average Processor Utilization.....	54
Figure 2.32: Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on Nonpaged Pool Allocation.....	56
Figure 2.33: Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on HTTP connection establishment.....	56
Figure 3.1: Experimental Setup.....	63
Figure 3.2: Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack.....	68
Figure 3.3: Processor Utilization of the Virtual Machine under Ping Flood Attack.....	70
Figure 3.4: Processor Utilization of the Non-Virtualized Server under Ping Flood Attack.....	71
Figure 3.5: Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack.....	72
Figure 3.6: Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack.....	74
Figure 3.7: HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack.....	75
Figure 3.8: Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under Smurf Attack.....	76
Figure 3.9: Processor Utilization of the Virtual Machine under Smurf Attack.....	77
Figure 3.10: Processor Utilization of the Non-Virtualized Server under Smurf Attack.....	78
Figure 3.11: Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under Smurf attack.....	78

Figure 3.12: Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under Smurf Attack.....	79
Figure 3.13: HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under Smurf Attack.....	80
Figure 3.14: Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack.....	82
Figure 3.15: Processor Utilization of the Virtual Machine under TCP-SYN Flood Attack.....	82
Figure 3.16: Processor Utilization of the Non-Virtualized Server under TCP-SYN Flood Attack.....	83
Figure 3.17: Number of Nonpaged allocations in the Virtual Machine and Non-Virtualized Server under TCP-SYN Flood Attack.....	84
Figure 3.18: Number of HTTP Connections established by the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack.....	85
Figure 3.19: HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack.....	85
Figure 3.20: Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack.....	87
Figure 3.21: Processor Utilization of the Virtual Machine under UDP Flood Attack.....	88
Figure 3.22: Processor Utilization of the Non-Virtualized Server under UDP Flood Attack.....	89
Figure 3.23: Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack.....	90
Figure 3.24: Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack.....	91
Figure 3.25: HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack.....	92

Figure 4.1: Experimental Setup.....	97
Figure 4.2: Comparison of Processor Utilization of the Virtual Machines under Ping Flood Attack.....	100
Figure 4.3: Comparison of Processor Utilization of Virtual Machines under Smurf Attack...	101
Figure 4.4: Comparison of Processor Utilization of Virtual Machines under TCP-SYN Flood Attack.....	102
Figure 4.5: Comparison of Processor Utilization of Virtual Machines under UDP Flood Attack.....	103
Figure 4.6: Comparison of VM with 4 cores and 1 core under the effect of Ping Flood Attack.....	105
Figure 4.7: Comparison of VM with 4 cores and 1 core under the effect of Smurf Attack.....	106
Figure 4.8: Comparison of the Processor Utilization of a VM with different number of cores under the effect of TCP-SYN Flood Attack.....	107
Figure 4.9: Blue Screen of Death in the Virtual Machine running 2012 Windows Server Operating System under 220 Mbps TCP-SYN Flood attack.....	108
Figure 4.10: Hyper-V manager loses control over the VM which is under the impact of a DDoS Attack.....	109
Figure 4.11: Comparison of VM with 4 cores and 1 core under the effect of UDP Flood Attack.....	110
Figure 4.12: Hyper-V Host with six Virtual Machines.....	111
Figure 4.13: Processor Utilization of the Hyper-V Host when the attack traffic is sent to Virtual Machines installed in the Host.....	112



## CHAPTER I

### INTRODUCTION

Cloud Computing has rapidly evolved over the years and has been adapted by a vast majority of organizations because of features such as flexibility, scalability, ubiquity and pay-per-use model. Due to the myriad of benefits provided by cloud computing to organizations of all kind, the technology has enjoyed widespread acceptance and rapid implementation. Nearly every technology invented has had some drawbacks, some of these shortcomings could have a very huge impact on the organization. Unfortunately, Cloud Computing has one such major shortcoming, it can pose a great threat to the security of the organization if not secured against network security attacks. Due to the many advantages offered by Cloud Computing, several organizations moved to the cloud when the technology was still in its nascent stages. Owing to the fact that zero-day attacks started being launched at a time when organizations were still getting accustomed to Cloud Computing, many zero-day attacks are proving to be very dangerous to the increasingly popular Cloud infrastructure. These days attackers have become as sophisticated and smart and are sometimes unfortunately a step ahead of the white hat community.

In a Cloud Infrastructure, the information of the Cloud Consumers handled by the Cloud Service Providers (CSPs) has to be provided security of three components known as CIA triad. Confidentiality, Integrity and Availability (CIA) of information is very important to maintain the reliability and trustworthiness of Cloud Service Providers. The Economic Denial of Sustainability attack is one such vicious attack in which the target Cloud Service Provider (CSP) is attacked such that they are unable to support any of the legitimate clients by exhausting their resources [1]. This kind of attack would permanently undermine the much needed trust between the Cloud Service Provider and their clients.

In today's Internet-dominant world we wake to news of hacking into organizations and theft of security numbers, credit card credentials and identity theft. Sadly, even to this day, organizations consider network security as a choice instead of a necessity. Of late DDoS attacks are being launched with botnets available for a low cost [2], [3]. Now, it is more important than ever to protect our systems from any threat to the confidentiality, integrity and availability of stored data.

The first step towards protecting against attacks is to be aware of the vulnerabilities in our systems so that we could come up with ways to best handle the vulnerabilities in time to prevent attacks or to fortify our defense mechanisms to better handle attacks.

### **1.1 Problem Statement**

The primary goal of cloud computing is to maximize hardware utilization and efficiency, this is achieved through virtualization. This thesis aims to analyze the vulnerabilities that virtualization might have introduced to the security of Cloud Computing. In this thesis, the impact of Distributed Denial of Service (DDoS) attacks on virtualized systems is evaluated by



comparing the effect of DDoS attacks on a non-virtualized server and on a virtualized server hosting several virtual machines and to determine if virtualization makes a system more susceptible to crashing.

## **1.2 Distributed Denial of Service Attacks**

Distributed Denial of Service Attacks are one of the most infamous attacks well known for the crippling effect they have on their victim. As the name suggests, the Denial of Service attack affects the availability aspect of the CIA triad. The first step taken by the attacker before launching this attack is to gather an army of computers called botnets which can be controlled through his commands. Botnet is a blend of the words *robot* and *network*.

An attacker sends malware through emails which when downloaded by the unsuspecting user makes the system a botnet. The recruited botnets are controlled by the attacker through a command and control server in a Master-Slave fashion. Once the attacker has recruited enough number of botnets, he commands all the botnets to send attack traffic simultaneously to the targeted victim thus overwhelming the target and making it unavailable to legitimate users or clients. Since systems from various locations are used to launch the attack, this attack is said to be a Distributed Denial of Service (DDoS) attack.

Now days, huge and long-lasting DDoS attacks as high as 600 Gbps are being observed against organizations and are making headline news frequently [4]. DDoS attacks have far-reaching consequences and leave a lasting impact on the victim organization by affecting the trust of the customers, loss of data and loss of revenue. The attacks launched are becoming more and more sophisticated and vicious, such as the ransomware attack in which attackers demanded ransom to decrypt sensitive medical information which they had encrypted by exploiting an

unpatched vulnerability in an application server [5]. It has been predicted that the occurrence of such attacks could increase in 2016 [6]. Under such circumstances, it is very crucial that organizations take a closer look at the inherent vulnerabilities and the host-based defense mechanisms available in the servers that have been deployed in their offices, such measures would greatly decrease the chances of falling prey to attacks [7].

The impact of DDoS attacks on various client and server Operating Systems have been evaluated in [8] – [15], [46]-[51], the results of the publications indicate that over time there has been improvement in the defense strategies employed by Operating Systems to combat against DDoS attacks, but at a time when bots are available in the form of DDoS-for-hire services for renting for just \$38 an hour [2], [3] it is becoming increasingly difficult to protect systems against attacks.

The DDoS attacks are classified depending on the protocol that was used to launch the attack, there are different types of DDoS attacks.

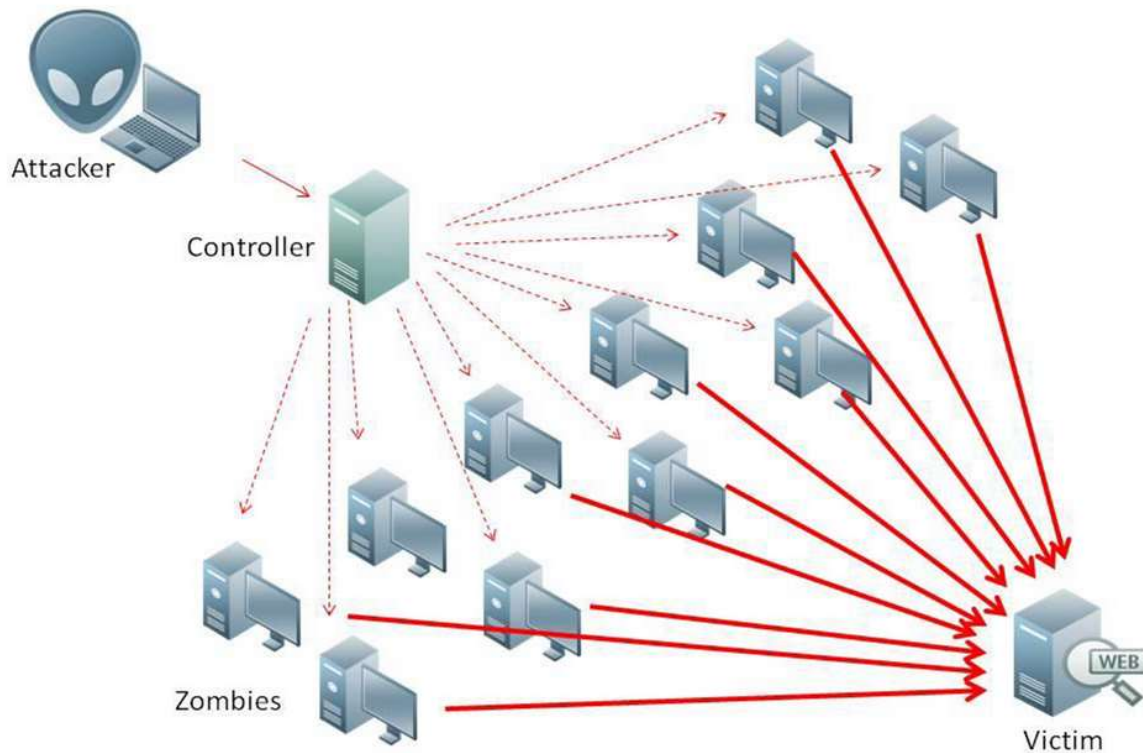


Figure 1.1. DDoS Attack launched on a Victim Web Server

### 1.2.1 Ping Flood Attack

The Ping flood attack is a layer-III DDoS attack which is carried out by manipulating ICMP echo request messages. The Internet Control Message Protocol (ICMP) is used to provide feedback about any problems that are experienced between the source and destination systems that could affect the quality of communication. Ping is one of the functions of the ICMP protocol, used for testing the connectivity between a pair of hosts.

The Ping utility is implemented using the echo request and reply messages. The RFC (Request For Comments) 792 requires that a system that receives an ICMP Echo request message must respond with an Echo reply [55]. Hence, after receiving an echo request, the receiver switches the source and destination addresses and changes the value of the type field from 8 to 0. When a pair of systems can exchange echo request and reply, it implies that a path

exists between the two hosts through which they can communicate. In a Ping attack, the attacker sends a lot of echo requests to the victim server, as a result, the victim is overwhelmed by all the echo requests. Since the victim server must respond to all the echo requests, the server will not be able to serve the requests of legitimate users.

### 1.2.2 Smurf Attack

The Smurf Attack is also carried out by exploiting the ping function of ICMP. The difference between the Ping attack and Smurf attack is that echo replies are used to bombard the victim instead of echo requests. To generate a flood of echo requests, the attacker directs his botnets to send spoofed echo request messages to a broadcast domain.

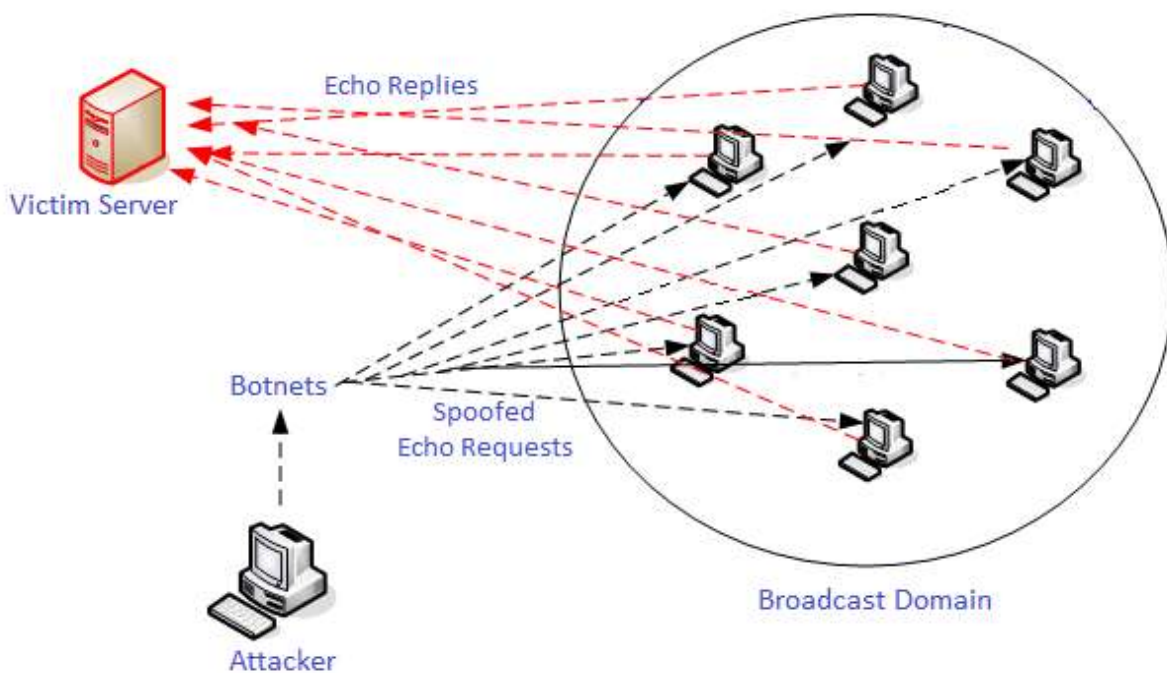


Figure 1.2 Victim server being attacked by botnets with Smurf Attack Traffic

The source IP address field of the spoofed echo requests contains the IP address of the victim, therefore the broadcast domain sends all its echo reply messages to the victim. Although, unlike in the case of Ping attack, the victim does not have to send any response, it has been

observed that Smurf attack is very detrimental to the functioning of the processor and affects the availability to a much greater extent than Ping attack. The Figure 1.2 shows a victim server under Smurf Attack.

### 1.2.3 TCP/SYN Flood Attack

The Transport Control Protocol (TCP) is one of the most important transport protocols used in the Internet. The Internet Protocol (layer III) was designed to provide best effort service hence it does not provide reliable data delivery. TCP was implemented to ensure reliable data delivery hence TCP forms an integral part of the Internet. Along with reliability, it also provides flow control and congestion control. The former is employed to prevent the sender from overwhelming the receiver by sending data at a rate higher than the receiver can accept. The latter is used to avoid overflowing the buffers located at the core of the Internet.

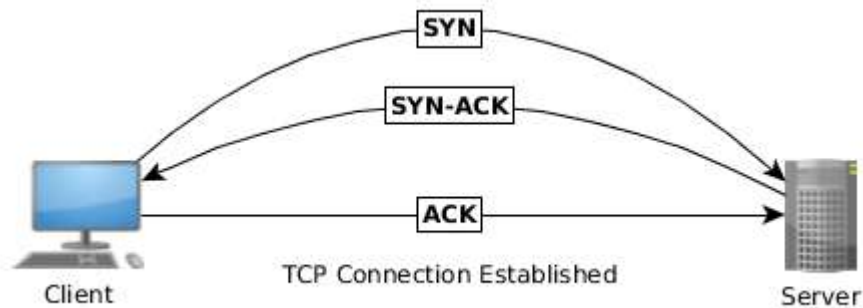


Figure 1.3. Legitimate three-way handshake

TCP ensures reliability by creating a connection between the sender and the receiver through a three-way handshake mechanism. Hence TCP is known as a connection-oriented protocol. In the first step of the three-way handshake, the sender sends a connection request by setting the SYN bit in the TCP packet to 1. The second step involves the receiver sending a SYN-ACK packet with both SYN and ACK bits set to 1 upon receiving the SYN request from the sender. In addition to sending SYN-ACK, the receiver also allocates buffers and variables for each TCP

connection. In the final step, the sender sends an ACK packet to the receiver and allocates buffers and variables following which the connection is set up [12].

The process of allocation of resources at the receiver before the completion of the three-way handshake is exploited in the TCP/SYN attack. To launch this attack, the attacker commands his botnets to send a volley of SYN packets to the victim. For each SYN packet the victim receives, the victim is forced to send a SYN/ACK packet and allocate resources. This keeps the victim server too occupied to be able to handle the connection requests of legitimate users leading to a denial of service.

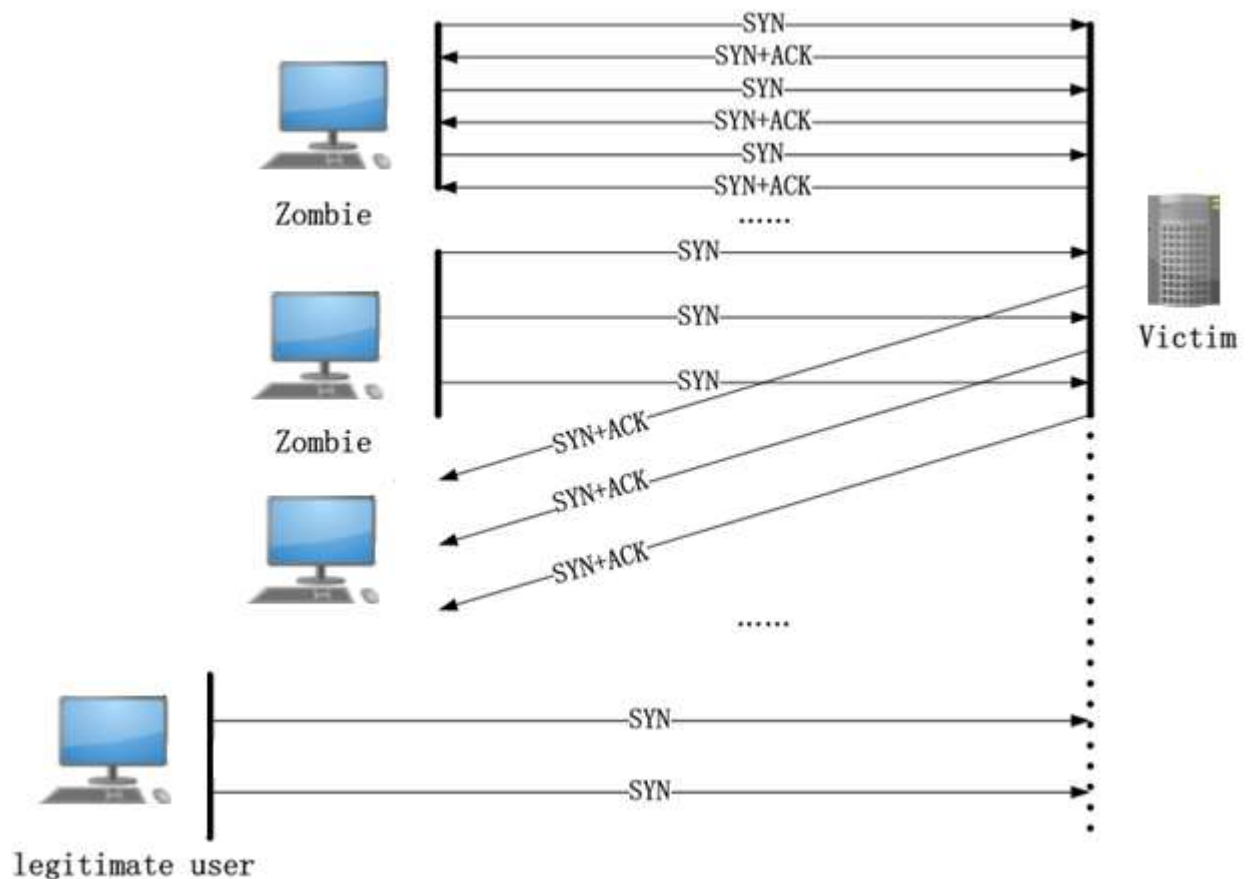


Figure 1.4. Victim Server under the impact of TCP/SYN Flood Attack

The inherent vulnerability due to incomplete TCP-SYN handshakes was identified as early as 1994 [41]. TCP SYN based DDoS attack is considered a common type of denial of service attacks [42] and many server platforms lack sufficient protection against this attack. Many schemes have been suggested to defend against this DDoS attack, however not many server platforms are automatically implementing effective protections against such attacks.

The first TCP-SYN attack, also known as SYN flood attack was reported in 1996 [43]. Since then, network system security has been improved to a great extent through the development of technologies such as Intrusion Detection and Intrusion Prevention Systems, Firewalls, Proxies and through the implementation of several strategies such as SYN cookies [40], [44], packet filtering based on sender IP addresses, reducing the SYN-RECEIVED timer, recycling the oldest half-open Transmission Control Block (TCB), SYN cache [45] to name a few.

#### **1.2.4 UDP Flood Attack**

The UDP Flood Attack exploits the User Datagram Protocol (UDP). UDP is a connectionless transport layer protocol which gained popularity since it enabled a user to send data without waiting for any kind of acknowledgement from the receiver providing best effort service like the Internet Protocol. Although this meant that the data that was sent would not be guaranteed to reach the destination, making UDP a less reliable transport protocol than TCP, thanks to the continuous evolution of the internet, more often than not, the data gets delivered to the destination.

The main advantage in using UDP is that unlike in the case of the TCP, the sender does not have to wait for an acknowledgement from the receiver. As a result, UDP gave complete control

over the time at which data can be sent to the destination. Thus, UDP has been widely used as a transport protocol. The lack of reliability in UDP was compensated by incorporating methods to check the reliability in the application level protocol that was transported by UDP. The Distributed Denial of Service (DDoS) attacks are well known for exploiting a protocol to overwhelm a targeted victim. DDoS attacks have always been effective because they use those protocols that would definitely elicit a standard response from the target that was defined by the standards, the UDP flood attack is no different. It has a high impact on the victim server which is nearly as severe as the effect that TCP-SYN flood attack has on the target.

Several methods have been developed to detect Distributed Denial of Service attacks by employing a wide range of mechanisms and intelligent intrusion detection systems. Packet- analysis has been one of the most used methods of detection where the patterns of previous occurrences of DDoS attacks are compared with the current network traffic to detect any match in the pattern [56]. Entropy-based methodologies or chaos theory have also been proved to be effective in the detection of attack traffic some of which could be implemented in cloud environment [57-59]. In [60], attacks is detected based on the anomalous nature of the DDoS attack traffic compared to legitimate or regular client traffic, analysis and handling of outliers. Some of the less common strategies that could be used in detection have been listed in [61]-[63].Some of these methods have been claimed to be accurate in their detection more than 90 percent of the time [61]-[62].



### 1.3 Cloud Computing and Virtualization

Cloud Computing is one of the few technologies which has risen to prominence and gained widespread acceptance in a very short span of time. This is due to the innumerable benefits such as scalability, lower maintenance, accessibility, minimized expenses due to the pay per use model and tailor-made services such as IaaS, PaaS, SaaS, etc. offered by the technology to organizations of all kind. The National Institute of Standards and Technology (NIST) defines Cloud Computing as follows: Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [36].

Cloud Computing services are provided in various types depending on the specific needs of the cloud consumer. The different types of services offered are IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). Depending on the size-based needs of the cloud consumer, the cloud infrastructure is classified as Private, Public and Hybrid cloud [8]. IaaS provides users with access to physical resources, networks, bandwidth, and storage. Using PaaS, Cloud Consumers can access the Operating Systems and platforms necessary to build and develop applications. SaaS enables end users to access applications and also allows them to make limited modifications to the applications.

Virtualization is at the core of Cloud Computing contributing to the simplicity, rapid provisioning and elasticity of cloud computing technology. The primary goal of cloud computing is maximized resource utilization, this is accomplished through virtualization which allows consolidating multiple servers with different roles as separate virtual machines on a single physical server hardware. Various components of a typical system in the cloud are: Hardware,

the Virtual Machine Manager (VMM) or Hypervisor and the Virtual Machines or instances. The VMM maps an image of the hardware to the Virtual Machines running on top of it, giving each Virtual Machine an impression that it has its own hardware (Memory, Processing Power and Input/Output devices). Virtual Machines are also known as instances and comprise of a guest Operating System and Application, each VM is unaware that other VMs are sharing the same hardware. The three most commonly used types of architecture are Full Virtualization, Paravirtualization and OS-based Virtualization.

Despite the many advantages that cloud computing offers to an organization, one major drawback that nearly outweighs all the benefits is the inability to offer best-in-class security. The virtualization platform chosen for this thesis is Microsoft's Hyper-V. Microsoft servers are one of the most widely used in the Information Technology field. Hyper-V is the virtualization platform introduced by Microsoft. It is offered as a stand-alone server and as a role in the Windows Server. Since most of the organizations already use Windows server, it is considered to be more cost effective to use the Hyper-V role in the server for their virtualization and cloud computing needs. Like other virtualization platforms, Hyper-V also supports running multiple Operating Systems, Windows, Linux, in parallel, on a single server.

#### **1.4 Thesis Outline**

The main goal in the thesis is to analyze the impact of virtualization on the performance of a server when it is under Distributed Denial of Service attacks. The attacks are measured for different transmission rates starting with 10 Mbps till 6000 Mbps or 6 Gbps. To determine the impact of virtualization, it was important to study the effect of the attacks on the server before it was virtualized and then the same attacks were launched on the server after it was virtualized.

This thesis is organized into five chapters. Chapter I is an introduction to virtualization and the description of the different Distributed Denial of Service attacks that have been used in this thesis. The chapter II presents the results of the performance of the server before virtualization under the impact of different DDoS attacks. In chapter III, the server is virtualized and one virtual machine is installed. All the cores in the server hardware are allocated to the virtual machine and the same DDoS attacks are launched on the VM. Two more virtual machines are installed in the hardware and all the three virtual machines are allocated one core each. Each virtual machine is then tested for performance under DDoS attacks. In order to study the effect of processing power allocated to a Virtual Machine on the performance of the VM, the VM performance has been studied for different core allocation. Followed by this, a total of six virtual machines are installed in the Hyper-V host and the effect of the virtual machines on the host has been studied. All these results are presented in the chapter IV of the thesis. In chapter V, the thesis is concluded and some future work has been suggested to further investigate the impact of virtualization and expand to different virtualization platforms in addition to Microsoft's Hyper-V.

## CHAPTER II

### EVALUATION OF THE IMPACT OF DDoS ATTACKS ON WINDOWS SERVER 2012 R2 BEFORE VIRTUALIZATION

Microsoft Windows Server Operating Systems, known for their friendly Graphical User Interface (GUI), are one of the most widely deployed in the industry today. This chapter analyzes the impact of Distributed Denial of Service (DDoS) attacks on Windows Server 2012 R2 which is the latest server OS from Microsoft. In order to understand the effect of virtualization on the OS, the immunity of the server OS before virtualization was evaluated initially. All the experiments were conducted in under a controlled environment at the Network Research Lab at the University of Texas Rio Grande Valley. Four of the most common DDoS attacks, Ping, Smurf, TCP SYN flood and UDP flood attacks were launched on the Microsoft server. Being the most used server OS, Windows Server 2012 R2 is expected to be fortified against such well known DDoS attacks.

Ever since the internet came into existence it has been growing at an exponential pace. Slowly, the internet diversified to cater to many other demands such as banking, retail, social networking and entertainment, in addition to being used for communication. Gradually, as internet became an integral part of people's lives, a new phenomenon called E-commerce emerged which enabled an internet user to shop anywhere at any time.

Today, the Electronic commerce or e-commerce industry has evolved into a billion dollar industry with China holding the first position in E-Commerce spending nearly \$560 billion with 600 million internet users followed by the United States where half of the nation's population buys online contributing to a revenue of approximately \$350 billion dollars [24] and this trend is projected to steadily grow in 2016 [64]. Needless to say, web servers are the backbone of the E-Commerce industry.

The success of an E-Commerce organization depends to a great extent on the speed at which its web servers serve its clients requests. Studies conducted by Amazon indicate that if there is a slowdown of one second in loading a page, Amazon could lose \$1.6 billion in sales each year. Google conducted a similar study which showed that if the search results are displayed even with a delay of only four tenths of a second, the number of searches per day would be decreased by eight million, which in turn would lower the number of online advertisements that Google can advertise [65]. This delay caused by a web server in responding to a client request is known as latency, as a result, configuring Web servers that serve clients with the minimum or no latency is of utmost importance to an E-commerce company.

It is evident that E-Commerce organizations such as Amazon, Google, eBay and the like cannot afford to have web servers that could be slow in responding to the myriad requests that they receive every second. In addition to latency, bandwidth is another factor that would affect the connection speed between the clients and a server. The Distributed Denial of Service (DDoS) attacks are known to be bandwidth intensive and processor intensive in addition to the many other negative impacts that they have on their victim. Therefore, it is crucial for web servers to have the ability to defend from a DDoS attack without causing a significant increase in the latency. This raised the question- Are Web Servers, especially Virtualized Web Servers capable

of protecting against DDoS attacks without any significant degradation in performance? Does Virtualization affect the performance of Web Servers, if so to what extent? To answer these questions, it is necessary to evaluate the performance of Non-Virtualized Web Servers against DDoS attacks and compare their performance with that of Virtualized Web Servers.

The impact of DDoS attacks on various client and server Operating Systems have been evaluated in [1] – [7], [36], [46]-[51], the results in the publications indicate that over time there has been improvement in the defense strategies employed by Operating Systems to combat against DDoS attacks, but at a time when bots are available in the form of DDoS-for-hire services for renting for just \$38 an hour [8], [9] it is becoming increasingly difficult to protect systems against such attacks.

## **2.1 Experimental Setup**

The Windows Server 2012 R2 Standard operating system is installed in a Dell PowerEdge T320 [37] hardware with Intel Xeon E5-2407 v2 quad core 2.4 GHz processor [38] and 8 GB RAM. The built-in firewall of the server was enabled with the default settings throughout all the experiments. The victim server was set up as a Web server, as a result, the latest version of Internet Information Services (IIS 8.0) service was installed in the server OS following the instructions in [39]. The experimental set up is shown in figure 2.1. The attack traffic was simulated in a controlled environment at the Network Research Lab at the University of Texas Rio Grande Valley (UTRGV).

A sample webpage called index.html is created in the victim web server which is running the IIS service. Once the sample web page is created, the IIS manager could be accessed from the Server Manager console. The IIS manager provides users with the option of adding a new

webpage to the web server. Under the server name, in the tab called web pages, the newly created sample webpage “index” is added. Then it is verified from the application pools tab if an application pool was created for index. Now another sub category called index would appear below webpages. Under the webpage settings, an option for a default document icon is chosen. In the default document icon, the entry index.html is moved up so that it can be accessed through an HTTP request.

Whenever the server receives a Hyper Text Transfer Protocol (HTTP) request from a client requesting the webpage index.html, the server responds to the request by sending the webpage through an HTTP reply. In order to recreate a typical web server environment, the HTTP requests were sent by means of simulating the users or clients in the lab. For the remainder of the thesis, the terms legitimate traffic or client traffic are also used to refer to HTTP requests. The most well-known DDoS attacks, Ping, Smurf, TCP SYN flood and UDP flood attacks, were launched on the target server. The server hardware consists of six network adapters, each connected to a Gigabit Ethernet link. All the four attacks were launched on the server in three different scenarios. These scenarios are classified based on the number of ports or network adapters in the server to which the attack traffic was sent. In all the three different experimental setups, the legitimate or client traffic is sent at the rate of 2500 HTTP requests per second to the server.

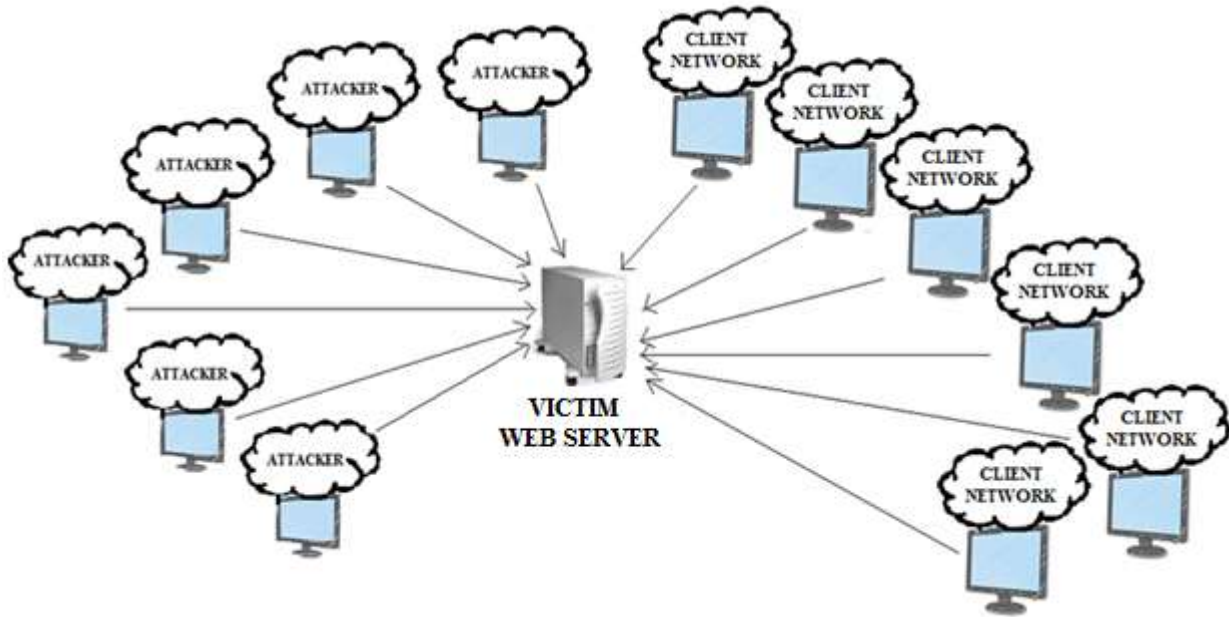


Figure 2.1. Experimental Setup of the server

In the first scenario, the attack traffic was sent to two out of the six ports in the server. Initially, only the legitimate HTTP connections are sent to the server in order to establish the baseline behavior of the victim server in the absence of any attack traffic. The legitimate HTTP connections requests were sent to the server for five minutes.

Once the baseline of the server was established, the attack traffic is injected into the network along with the legitimate traffic. The attack traffic magnitude sent to each port or network adapter ranged from 100 Mbps to 1000 Mbps or 1 Gbps. The attack traffic load sent to each port is increased in increments of 100 Mbps for every five minutes and is sent to the server simultaneously along with the HTTP client requests.

Thus, once the baseline behavior is captured, then an attack traffic load of 100 Mbps is introduced to the server with the legitimate traffic, then after five minutes, the attack traffic load was changed to 200 Mbps and sent along with the client traffic to each port. Each attack traffic load is sent to the server for a uniform duration of five minutes. This process is repeated until an



attack load of 1000 Mbps is sent to each port. Hence, the total duration of the experiment is fifty five minutes, five minutes for obtaining the baseline and five minutes for each increased attack traffic load.

In the second scenario of the experiment, the attack traffic was sent to four ports in the server instead of two ports. Finally in the third set up, the attack traffic was sent to all the six network adapters in the target server simultaneously with the client traffic. Various parameters of the web server were monitored and recorded to enable evaluation of the server.

## **2.2 Parameters of Performance Evaluation**

The parameters that were monitored during the experiment are the Average Processor Utilization of the web server, Core Utilization of the web server, Nonpaged Pool Allocations, the number of HTTP (Hyper Text Transfer Protocol) connections handled by the server and the Connection Latency experienced by the clients. All these five parameters were measured throughout the duration of the experiment starting from the baseline behavior. These parameters were recorded by using the Data Collector Sets available in the performance monitor of the Windows Server 2012 R2 and the counters available in the client systems.

The Processor Utilization of a computer is analogous to the heartbeat and has a strong influence on the performance of the server. The name of the counter that is used to monitor processor utilization is known as \Processor(\_Total)\% Processor Time and is defined as “The percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then subtracting that value from 100%. (Each processor has an idle thread that consumes cycles when no other threads are ready to run)” [16]. The Total Processor Utilization is the average of core

utilization of all the cores in a server, in this case the server has four cores. The Central Processing Unit (CPU) has to be functional at all times for the server to be able to deliver its most efficient performance. Monitoring the processor utilization enables a person to accurately observe the effect that an attack has on the server. Whenever the CPU utilization exceeds its optimal value, it will start impacting the efficiency of the server.

The second parameter that was recorded during the course of the experiments was the Core Utilization. The core utilization counter, available under the % Processor Time, can be used to determine the processor consumption of each individual core present in the system. The counter `\Processor(_Total)\% Processor Time` is the average of the core utilization of all the cores in a processor. The processor used for the thesis consists of four cores hence, the counters that were used to monitor the core utilization of the server are `\Processor(0)\%Processor Time`, `\Processor(1)\%Processor Time`, `\Processor(2)\%Processor Time` and `\Processor(3)\%Processor Time` [54]. Although the term Total processor utilization might be misunderstood as the sum total of core utilization of all the cores in a server, Total Processor Utilization actually refers to the average of the average of the core utilizations in a server.

If processor utilization can be used to observe the effect of an attack on the server, the change in the memory usage of the server will throw light on the root cause of the issue, impact of DDoS attacks on servers. Analyzing the memory utilization of the server will help explain the reason behind why an attack has such a huge impact on the performance of the server. The Non-Paged pool and the Paged pool are the two memory resources used by an Operating System and its device drivers for storing data structures. The Non-Paged pool in the memory can only be allocated in the physical memory and not in the virtual memory unlike in the case of Paged pool [17], [52]-[53]. As a result, the number of non-paged allocations is considered to be a

representation of the memory utilization and is monitored throughout the experiment. The performance counter used is called \Memory\Pool Nonpaged Allocs.

Depending on the type of services installed in a server, the performance efficiency of the server can be analyzed using different parameters. The role installed in the Windows Server 2012 R2 Server OS used in this thesis is the Internet Information Services (IIS) Manager. The IIS Manager is used for the configuration and management of the services offered by a web server. Therefore, the Number of HTTP connections that the server is able to establish with the clients is chosen as one of the parameters to measure the performance of the server. The web server hosts a URL (Uniform Resource Locator) for the web page called index.html. The simulated client network records the statistics of the communication with the server. The clients monitor the number of positive HTTP responses received from the server, which in turn can be used to judge the efficiency of the web server as the attack progresses.

In today's internet replete with tech-savvy consumers, the speed at which responses are received are as important as the response itself. Hence it is expected of a web server to not only respond to client requests but do so within a few milliseconds. As a result, the delay caused in responding to an HTTP request, also known as Connection Latency, is considered as one of the deciding factors to determine the efficiency and quality of a web server. Therefore, the connection latency is also monitored to analyze the strain that the attack causes to the server and how it affects the speed of response. The Connection Latency is defined as "the average time elapsed between the time the client sends a SYN packet and the time it receives the SYN/ACK" Connection latency is measured in microseconds in the counter available in the client. In this thesis, the connection latency is represented in milliseconds.

## 2.3 Results and Discussion

### 2.3.1 Ping Flood Attack

Ping attack is launched by sending a barrage of ICMP Echo request messages to a chosen victim server. It misuses the Ping utility that is used to check the connectivity between two computers, due to the infamy of the Ping and Smurf attacks, some organizations configure their systems to block ping messages. The Ping attack was launched on the server based on all the three different experimental setups. The Total or Average Processor Utilization of the server for all the setups is shown below in figure 2.2. When there is no attack traffic, the baseline behavior of the server is the same in all the three cases since only the legitimate traffic, 2500 requests per second, is sent to the server. The baseline processor utilization is nearly seven percent.

When the attack traffic load is increased further in steps of ten percent, the processor utilization of the server also keeps increasing proportionately. In the third setup, when 100 percent of the attack traffic, or 6000 Mbps, is sent to the server, the processor utilization reaches 96 percent. The efficiency of the server and its availability to the clients is inversely proportional to the processor utilization. Hence, the continuous increase in the processor utilization incapacitates the web server from establishing connections with its clients.

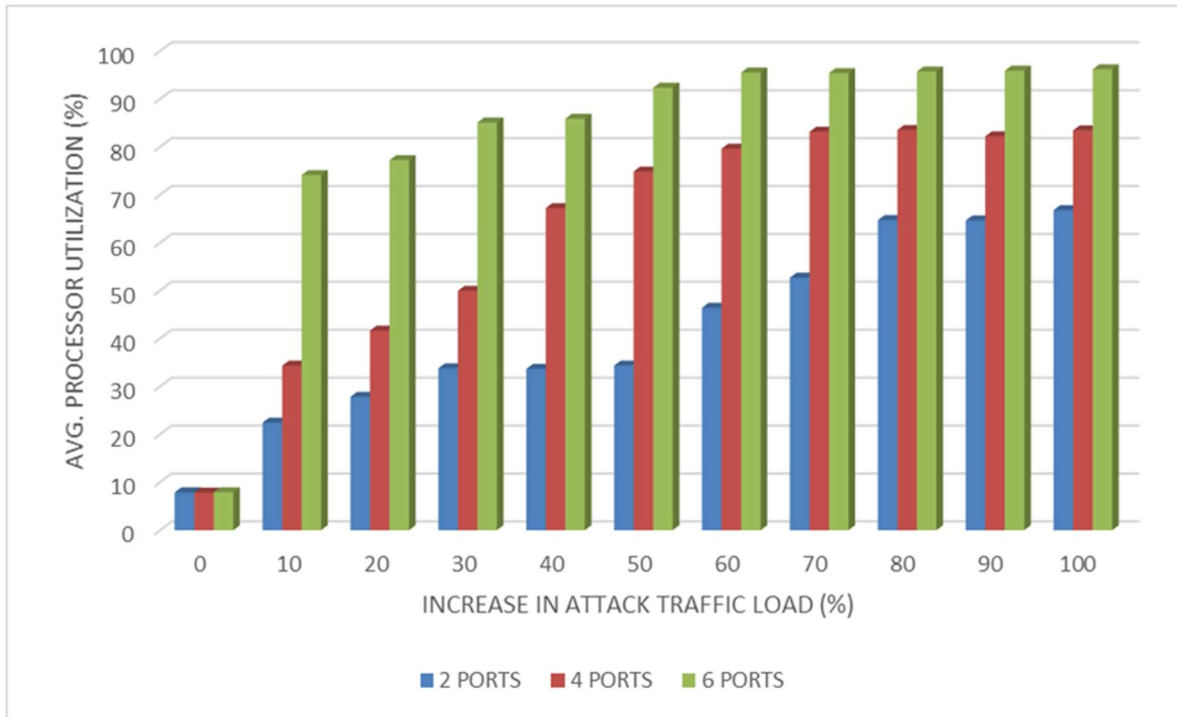


Figure 2.2. Average Processor Utilization of the Victim Web Server under Ping Attack

As mentioned earlier, the server consists of four cores, the core utilization of the server is also monitored in order to better understand the change or lack of change in the allocation of cores after virtualization. Figure 2.3 shows the core utilization of the server under Ping attack, when the attack traffic is sent in the range of 600 Mbps to 6000 Mbps. The light blue line indicates the average processor utilization of the server for the same attack traffic range. From the graph, it is very noticeable that the processor utilization of the server is shared equally among all the four server cores.

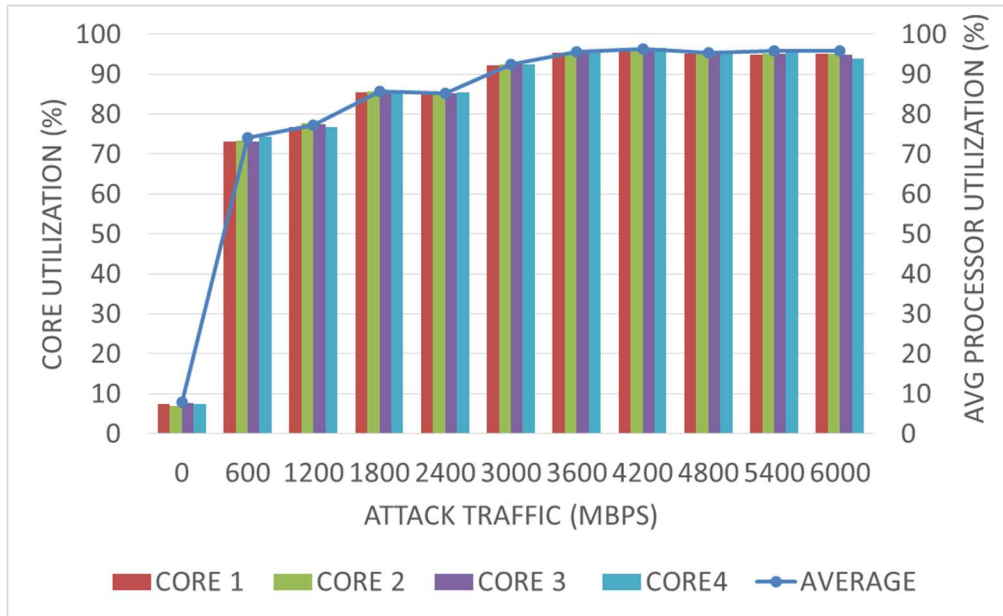


Figure 2.3 Core Utilization of the Victim Web Server under Ping Flood Attack

The figure 2.5 displays the number of Nonpaged pool allocations in the server throughout the duration of the experiment. The memory from the Nonpaged pool is allocated for any object which represents threads and processes. For each HTTP request received from the client, the server schedules a thread as soon as the connection is established. The total number of connections that a server can establish simultaneously can be configured by modifying the value of MaxClients [19]. If a server receives more number of requests than the value configured in MaxClients, then the additional requests will wait until a web server thread becomes available.

Once the server decides to shut down a TCP connection with a client, the server sends a TCP packet with the FIN bit set to the client and enters the FIN\_WAIT1 state, during this state, the scheduled web server thread can be used for a maximum of two seconds. Figure 2.4 shows the close sequence between the client and the web server. After receiving the TCP packet from the server, the client sends an acknowledgement and enters the CLOSE\_WAIT state and the same web server thread is continued to be used by that particular connection. Upon receiving the

acknowledgement from the client, the server enters the FIN\_WAIT2 state and will not send any further data to the client [18]. In this state, the server will continue to use the thread for another two seconds while it is waiting for the FIN from the client. If the client responds with a FIN, the web server would send an ACK to the client, if not the server would enter the TIME\_WAIT state and wait for a duration equal to the Maximum Segment Lifetime (MSL) after which the server would close the connection. The web server thread is not utilized by the particular connection when the server is in the TIME\_WAIT state.

The maximum number of web server threads that can be simultaneously utilized depend on the sum of fixed startup cost of memory for each thread and the maximum runtime memory

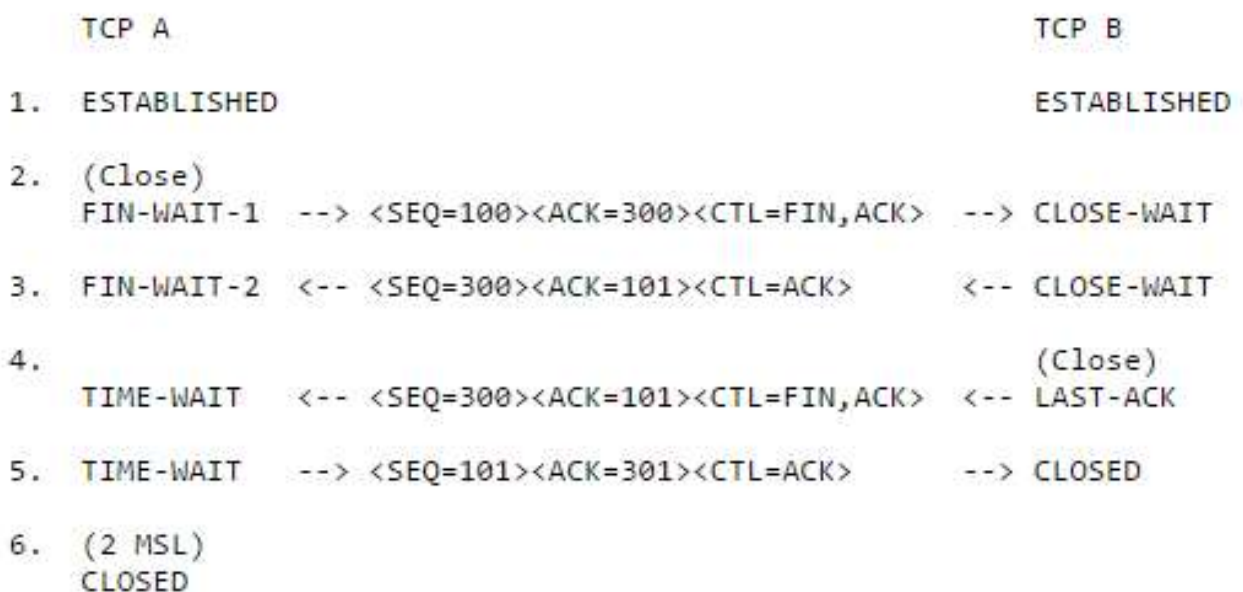


Figure 2.4 TCP Close sequence between the Web Server and a Client [20]

usage per thread. The magnitude of both these parameters depend on the workload and configuration [19]. Since threads require memory to be allocated in the nonpaged pool, the legitimate connection requests and the attack traffic would require the allocation of memory from the nonpaged pool which in turn uses the RAM. Similarly, since all the four types of DDoS

attack traffic are nothing but standard protocols, that are sent in large numbers, the server would allocate nonpaged memory upon receiving ICMP echo packets, TCP-SYN packets and UDP datagrams. Since these attacks target the memory of their victim, the impact on the performance of the server is said to be due to memory leak. [21]-[23] describe some similar vulnerabilities that could be exploited to launch Denial of Service attacks on victims.

During the baseline determination, the number of nonpaged pool allocations in the server was 52540. After the attack traffic was introduced, the number of allocations kept increasing linearly with increasing Ping (ICMP echo request) attack traffic. From the figure 2.5 it can be observed that when the attack traffic is sent to more number of ports, the number of nonpaged allocations also increases proportionately.

When the server is receiving 6000 Mbps attack traffic, there are nearly 1000000 nonpaged pool allocations. Since the number of nonpaged allocations represents the memory utilization of the server, it is evident that there is a direct correlation between the number of nonpaged pool allocations and the number of HTTP connections established by the server.



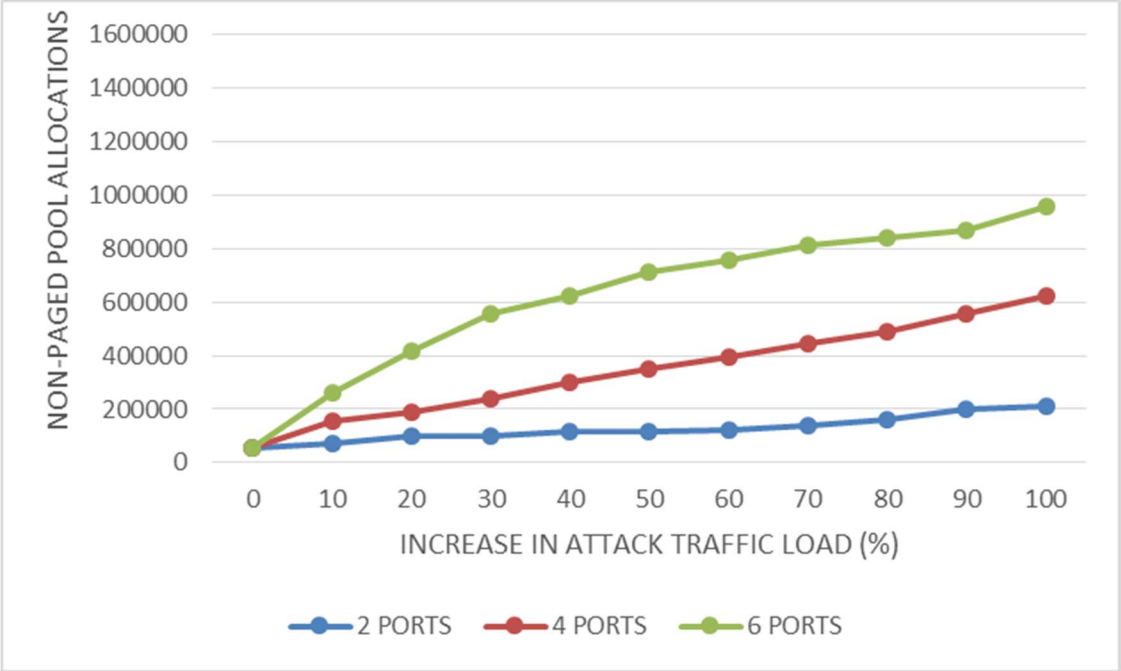


Figure 2.5 Nonpaged Pool Allocations in the Server under Ping Flood Attack

With increasing number of allocations, the number of connection establishments decreases. As mentioned earlier, both the attack traffic and the legitimate traffic need to use the physical memory and not the virtual memory, being higher in number the flood of ICMP requests accelerates the nonpaged pool memory allocation which deprives the legitimate connections of the physical memory. This trend can be observed in the figure 2.6 which shows the connection establishment behavior of the server in all the three scenarios.

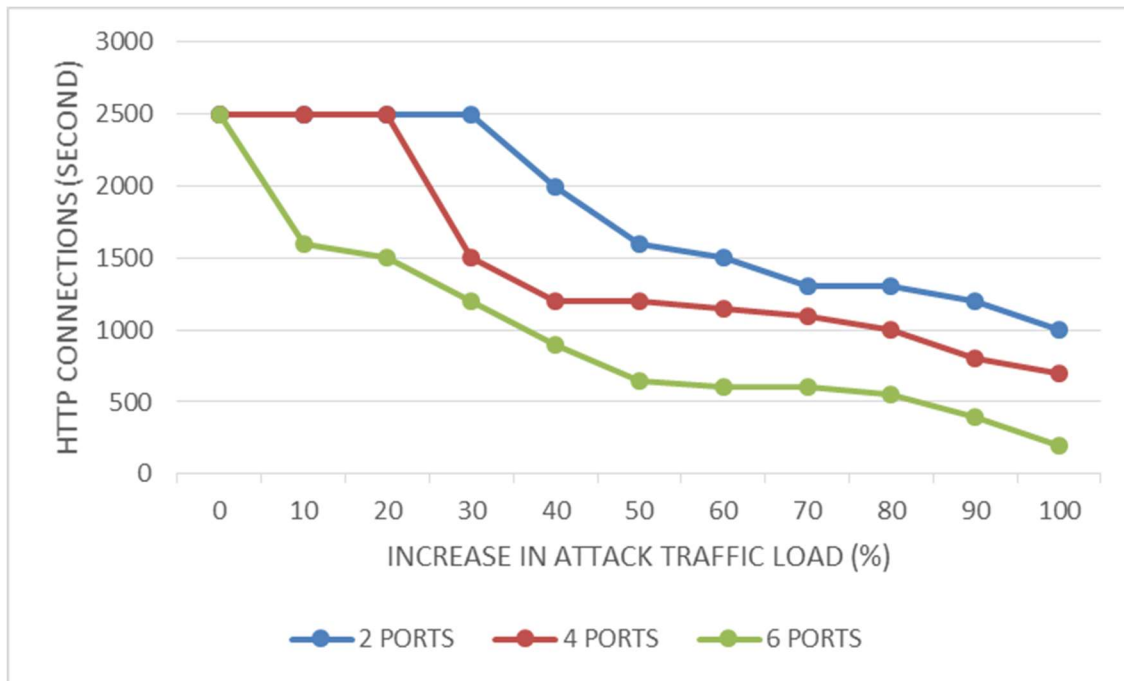


Figure 2.6. HTTP Connections established between Clients and the Victim Server under Ping Flood Attack

Initially, in order to establish the baseline, only the legitimate traffic is sent to the web server for five minutes at the rate of 2500 connections per second. As the attack traffic load is increased, the number of connections that the server is able to establish with the clients keeps on decreasing. When the attack traffic is sent only to two ports, the number of connections does not start to decrease until thirty percent of the attack load or 600 Mbps attack traffic is sent to the server. This indicates that the server is able to handle the Ping attack traffic effectively up to a magnitude of 600 Mbps. As the magnitude of attack traffic is increased further, the number of connections that the server is able to establish keeps decreasing. In the second scenario, the number of HTTP connections the server is able to establish at 100 percent attack traffic load, 4000 Mbps, is approximately 740 connections per second, which is almost thirty percent of the number of connections the server is able to handle in the absence of attack traffic.

The effect of the Ping attack on the target server is much worse in the third setup when the attack traffic is sent to all the six ports of the server. In this case, when the highest magnitude of attack traffic, 6000 Mbps is sent to the server, it is only able to handle 250 connections per second. From the results, it is evident that the increase in processor utilization degrades the performance of the server to such an extent that the server is only able to maintain one tenth of the number of connections it was able to maintain in the baseline.

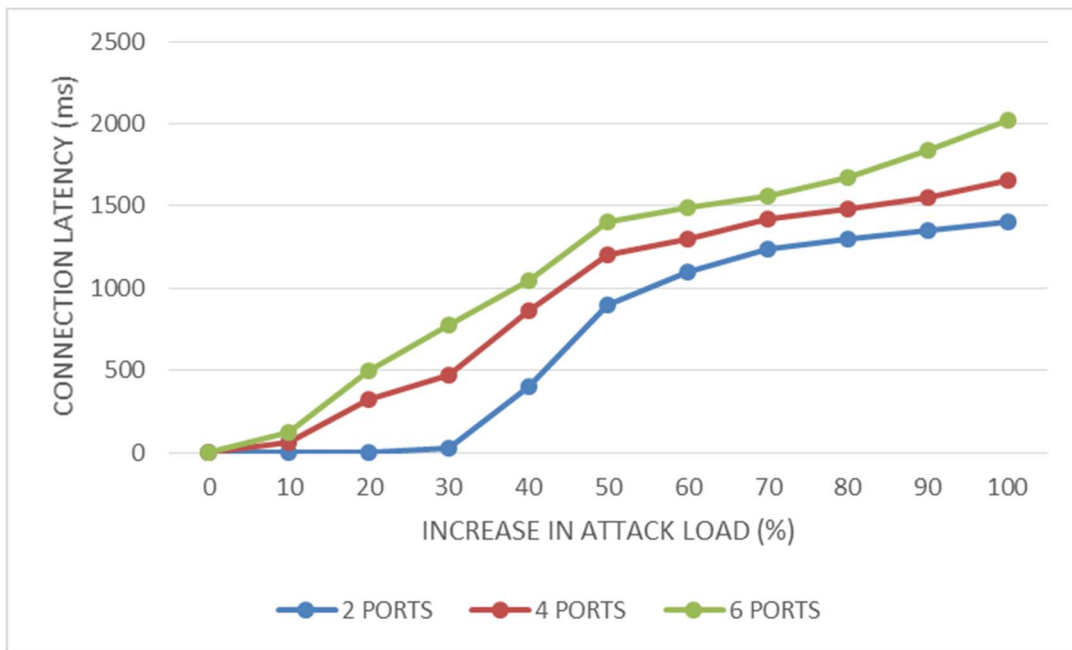


Figure 2.7. HTTP Connection Latency of the Victim Server under Ping Flood Attack

The connection latency of the server is shown in figure 2.7. The connection latency is an important parameter that can be used to evaluate the ability of the server to handle an attack. The latency keeps increasing with increasing attack traffic load and increasing processor utilization. When a Ping attack traffic of 6 Gbps is sent to the server, it leads to a connection latency of 2000 milliseconds.

### 2.3.2 Smurf Attack

The Smurf attack is launched by sending a flood of ICMP reply packets to a system, this is considered to be one of the most notorious DDoS attacks since it has a very detrimental effect on its victim. The Smurf attack is launched on the Windows server 2012 R2 to evaluate the extent to which the server is able to withstand the attack without any noticeable effect on the efficiency of the web server. Figure 2.8 displays the average processor utilization of the server under different attack scenarios. From the graph it can be definitely inferred that Smurf attack has an almost crippling effect on the server.

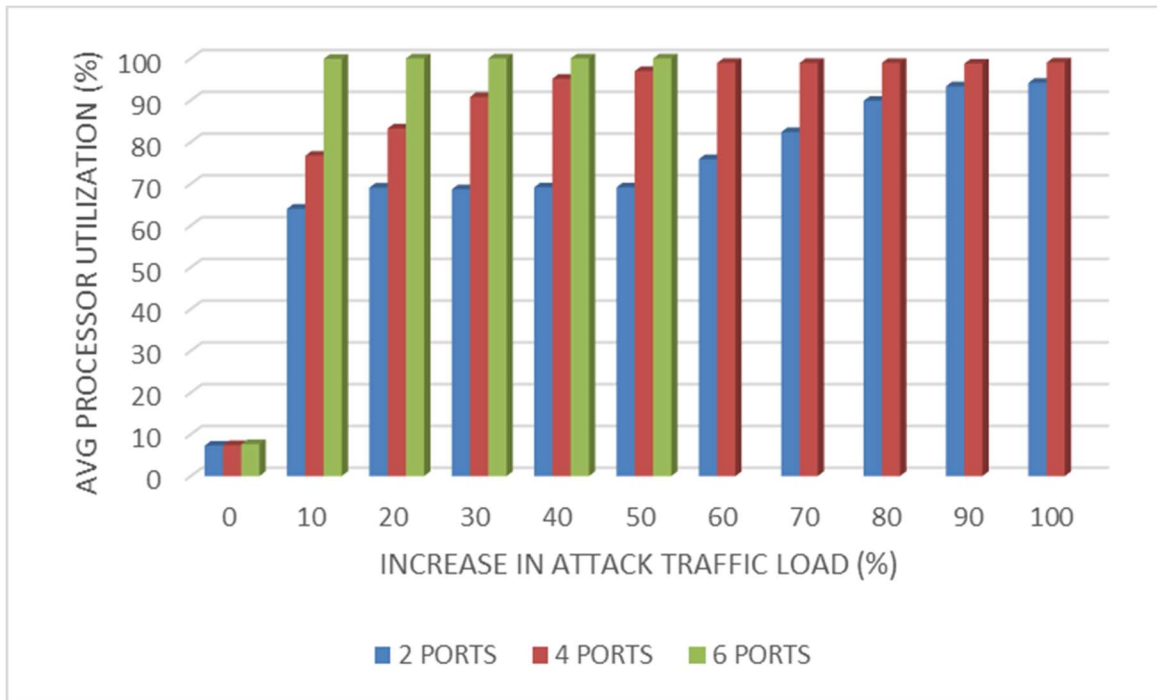


Figure 2.8. Average Processor Utilization of the victim server under Smurf Attack

It can be inferred that the Smurf attack has much greater impact on the server than the Ping attack. The processor utilization keeps on increasing proportionately with increasing load of attack traffic. The same trend can be observed for the processor utilization under the second experimental setup in which case, the attack traffic load is sent to four ports and the attack traffic

ranges from 400 Mbps to 4000 Mbps. When the attack traffic load is increased to 4000 Mbps, the processor utilization reaches a magnitude of 92 percent.

In the third and final attack scenario, the processor utilization shot to 100 percent when 1200 Mbps, only twenty percent of the attack traffic load (6000 Mbps) was sent to the server. It can be observed from the graph that there is no value entered for processor utilization after the attack traffic load is increased to 60 percent attack traffic load or 3600 Mbps. The average processor utilization continued to remain the same for the next twenty minutes until the attack load was increased to fifty percent of the total attack traffic load after which the performance monitor was stopped by the server. By the time the attack load was increased to 3600 Mbps, the processor utilization had already been 100 percent for 20 minutes and so the processor could not expend any more of its processing power and continue to run the performance monitor. Hence, the performance monitor stopped recording data after an attack traffic of magnitude 3000 Mbps was sent to the victim server.

Figure 2.9 shows the core utilization of the server in the first attack scenario where the attack traffic starts from 200 Mbps and is increased by 200 Mbps every five minutes until a maximum attack traffic of 2000 Mbps is sent to the server. All the four cores have been equally used throughout the experiment, this behavior is similar to that observed when Ping attack was launched on the server.

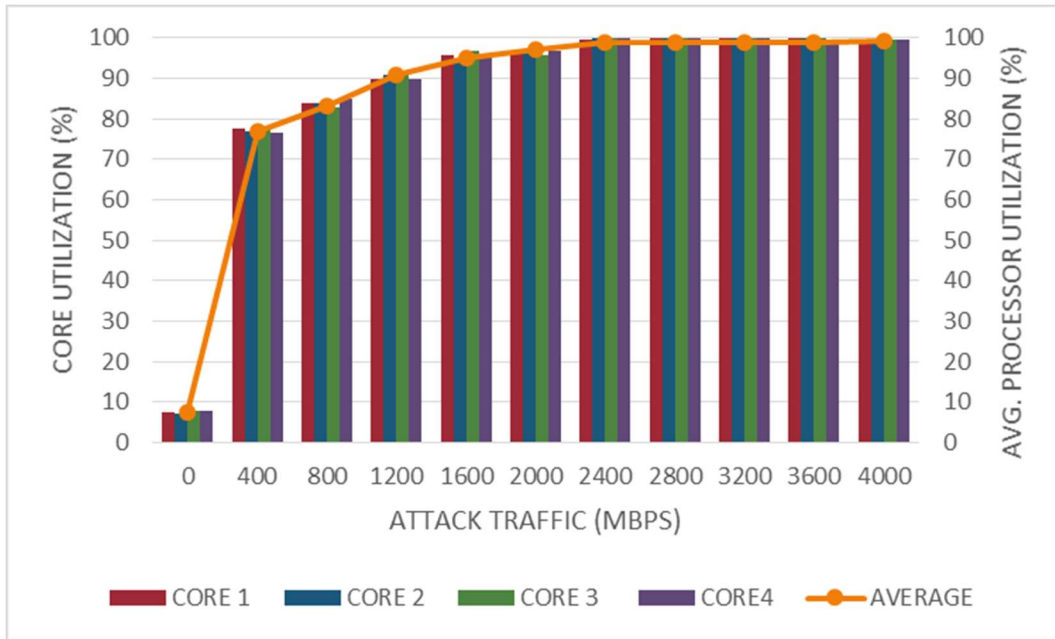


Figure 2.9. Core Utilization and Average Processor Utilization of the Victim Server under Smurf Attack

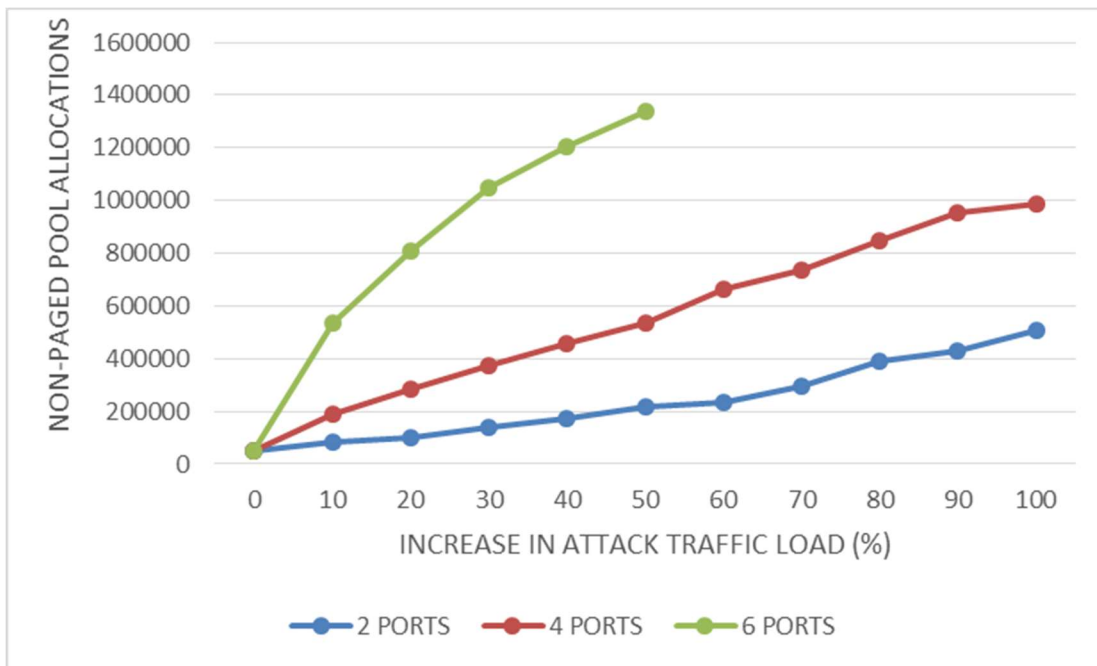


Figure 2.10 Nonpaged Pool allocation in the victim server under Smurf Attack

The memory utilization of the server under Smurf attack is shown in figure 2.10. Since the performance counter was stopped after 50 percent of the attack traffic load was sent during the

third experiment, the number of nonpaged pool allocations were only gathered up to 3000 Mbps. In the case of Ping attack, the maximum number of nonpaged pool allocations was 1000000 which was observed when the server was receiving an attack traffic of 6000 Mbps. But in the case of Smurf attack, even though the number of nonpaged pool allocations for 6000 Mbps is unknown, the number of nonpaged pool allocations reaches nearly 14000000 when the attack traffic magnitude is only 3000 Mbps. This explains the reason why Smurf attack has a very negative impact on the server compared to Ping Flood attack.

The figure. 2.11 shows the HTTP connections established by the server per second for the three attack scenarios. For each experiment, only the legitimate client traffic was sent to the server at the rate of 2500 connections per second for the first five minutes and then ten percent of the total attack traffic load was introduced to the server.

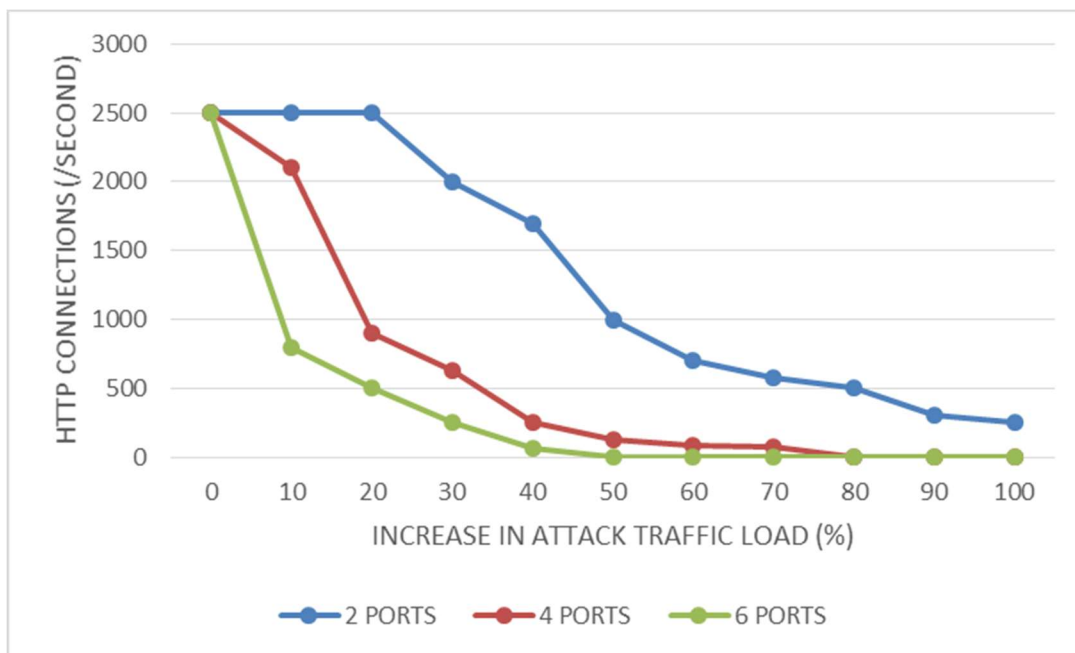


Figure 2.11. HTTP Connections Established by the victim server under Smurf Attack

In Ping attack, the lowest number of connections the server was able to maintain was 250 connections per second for the maximum attack traffic of magnitude 6000 Mbps. In Smurf attack, the server could not establish more than 250 connections per second for a relatively low attack traffic of magnitude 2000 Mbps. In the second experimental setup for the Smurf attack, as soon as the attack was launched on the server, the number of client HTTP connections that the server could establish decreased to 2100 connections per second. With increasing attack traffic load, the HTTP connections declined rapidly until the server was receiving 70 percent of the attack load, or 3200 Mbps. When the attack traffic load was further increased, the server was unable to establish even a single connection with the clients.

The performance of the server worsened in the third scenario when the attack traffic was sent to all the six ports of the server. Since the performance monitor was stopped after fifty percent of the attack traffic load, 3000 Mbps, was sent to the server, the average processor utilization and core utilization of the victim server could not be continuously monitored. But as the HTTP connection establishment and latency was monitored from the client, it was possible to determine the efficiency of the server through the number of connections established by the server.

From Figure.2.11 it can be observed that the number of legitimate connections kept on decreasing continuously until it was not able to support any more client connection requests at fifty percent of the attack load. At the same time that the server stopped responding to client requests, the performance monitor was also stopped. Hence at 3000 Mbps, the Smurf attack had completely ceased all the activities of the server.

Before the server stopped responding to client requests, the server established comparatively much fewer number of connections than that observed during the baseline, in



addition to that, the server was not able to respond to the limited of client requests without a considerable increase in latency. This can be observed from the connection latency of the server shown in Figures 2.12, 2.13 and 2.14. Since the server stopped responding to connection requests at very different times during the attack for different scenarios, unlike in the case of Ping attack, the connection latencies of server for the three different setups are shown separately.

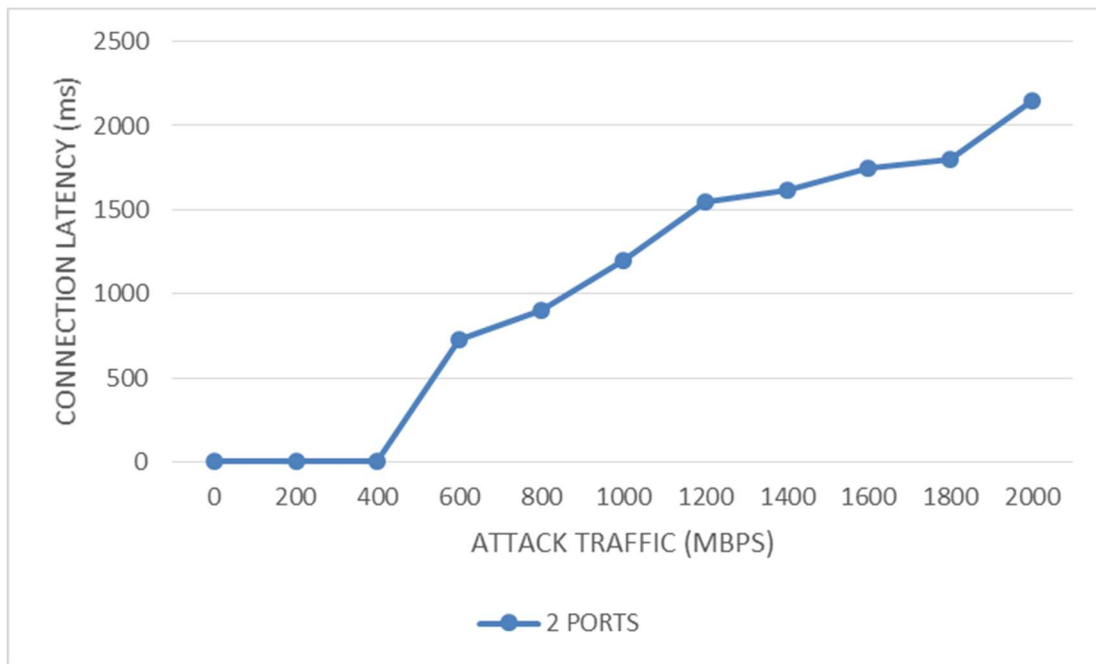


Figure 2.12 Connection Latency when Smurf attack traffic is sent to two ports

The Connection Latency is captured up to the point of time the server continues to establish connections, after the server stops responding to client requests, the connection latency keeps increasing exponentially since the server does not establish any connection with the clients. When the Smurf attack traffic is sent to only two ports of the server, the server manages to continue to establish 250 connections per second when 100 percent of the attack traffic load (2000 Mbps) was sent to the server, but it took 2150 milliseconds to establish 250 connections.

In the second experimental set up when the attack traffic was sent to four ports, the maximum attack traffic load up to which the server was able to establish connections was 2800 Mbps. At this point, the server was capable of establishing 40 connections per second as opposed to the 2500 connections it could establish per second in the absence of attack traffic. Even though the server is only handling nearly one fiftieth of the connections it could handle earlier, the server took 3200 milliseconds as opposed to the 0 milliseconds it took earlier.

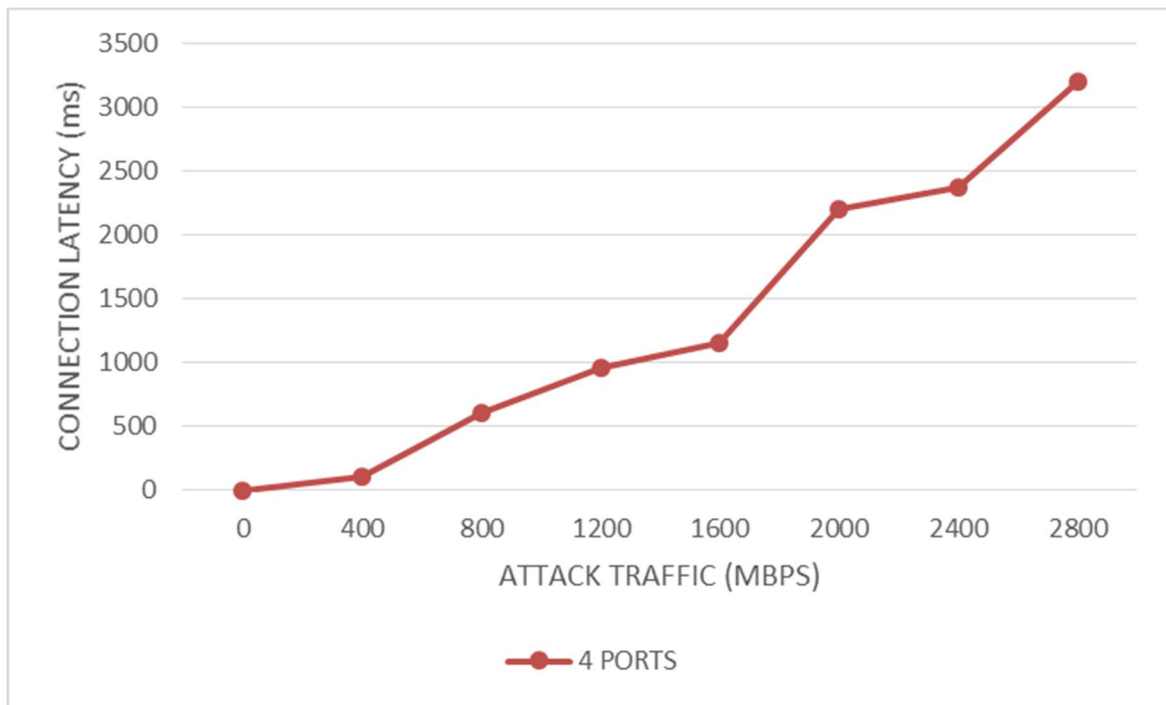


Figure 2.13 Connection Latency when Smurf attack traffic is sent to four ports

The connection latency observed in the third experimental setup was very similar to the previous scenario. The server could handle 50 connections per second under 2400 Mbps attack traffic but took 3825 milliseconds to establish the connections.



Figure 2.14 The Connection Latency when Smurf attack traffic is sent to all the six ports

### 2.3.3 TCP-SYN Flood Attack

The TCP-SYN flood attack is one of the most devastating of all the DDoS attacks. One of the reasons it is much feared is because it is nearly impossible to defend against this attack. This is because unlike in the case of Ping and Smurf, where the ICMP echo request or reply could be blocked, blocking the external TCP connections from a server is not a wise strategy to prevent the attack since that it is the equivalent of launching a self-imposed Denial of Service attack on the server.

In the first experimental setup, the processor utilization increased linearly with increase in attack traffic load. When hundred percent of the attack traffic load, 2000 Mbps, was sent to the server, the processor utilization was 74 percent. This indicates that the TCP-SYN flood attack is relatively less damaging to the server than the Smurf attack.

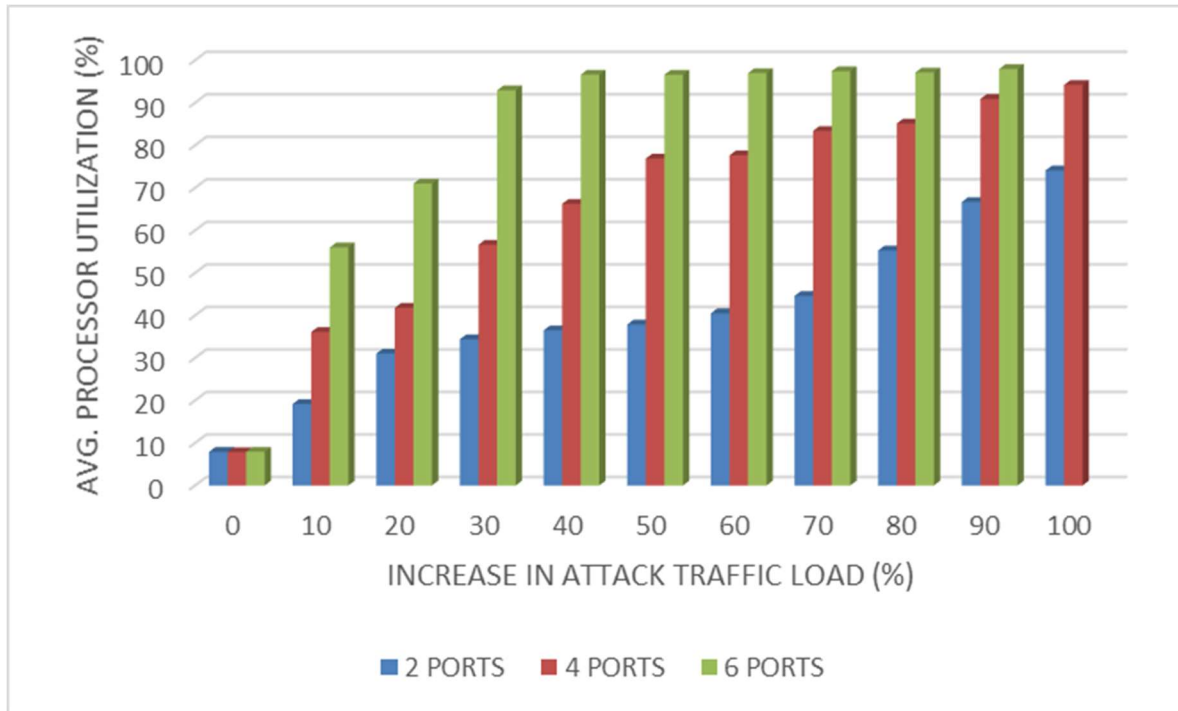


Figure 2.15 Average Processor Utilization of server under TCP-SYN flood attack

The second experimental setup gave similar results as the first one, but overall the magnitude of the processor utilization had increased by nearly 15 percent from the first attack scenario as shown in Figure.2.15. By the end of the second experiment, where the 1000 Mbps attack traffic was sent to each of the four ports, the processor utilization had reached 94 percent. Although this attack is less effective than Smurf, it has more impact than Ping attack. This can be inferred from the fact that the processor utilization of the server had reached 96 percent when the server was under Ping attack only when the attack traffic was sent at the rate of 6000 Mbps, but with TCP-SYN flood attack, the processor utilization has already reached 94 percent when a 4000 Mbps- magnitude attack traffic was sent to the server.

In the third and final experiment, the attack traffic is sent to all the six ports of the server in which the attack traffic sent to each of the six ports is increased in steps of 100 Mbps every five

minutes until the total traffic sent to all the ports together was 6000 Mbps. The processor usage reached 56 percent when the attack traffic was introduced and continued to increase with increasing attack traffic load. The highest processor utilization recorded for the TCP-SYN flood attack was 97.94 percent when an attack traffic of magnitude 5600 Mbps, 90 percent of the total attack load, was sent to the server. When the total load of attack traffic, 6000 Mbps, was launched on the server, the processor utilization could not be determined since the server stopped the performance monitor. As a result, the processor utilization of the server during the final increase in attack traffic load could not be determined accurately.

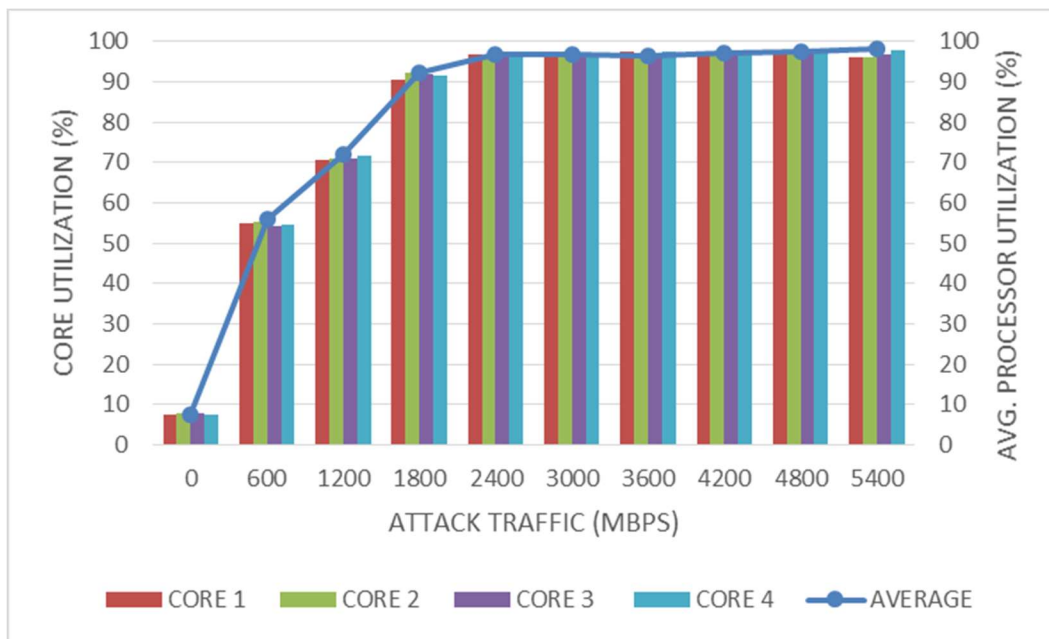


Figure 2.16 Core Utilization and Average Processor Utilization under TCP-SYN Flood Attack

The Figure.2.16 displays the core utilization and the average processor utilization of the server recorded in the third experimental set up. The plot indicates that throughout the duration of the experiment, both during the baseline measurement and after the attack traffic was started, the threads have been scheduled such that all the four cores of the server were utilized equally.

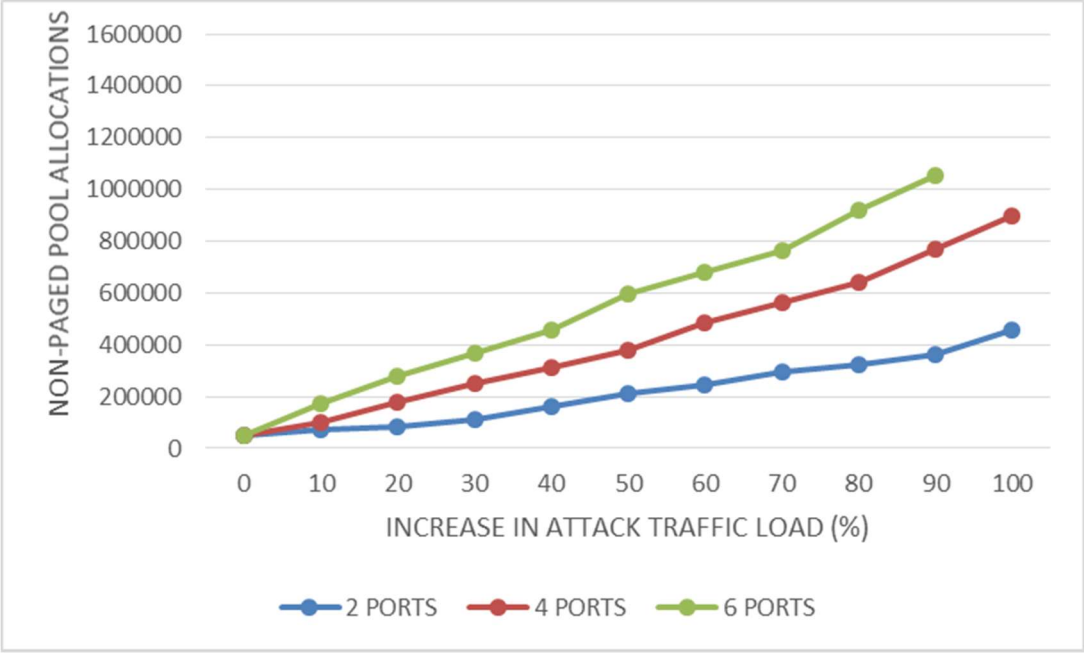


Figure 2.17 Nonpaged Pool Allocation of the victim server under TCP-SYN Flood Attack

The number of nonpaged pool allocations increase with increase in attack traffic magnitude as shown in figure 2.17. Similar to the Smurf attack, since the performance monitor was stopped before the end of the third experiment, the number of allocations when the server is receiving TCP-SYN attack traffic of magnitude 6000 Mbps could not be determined. When the server is receiving 90 percent of the total attack traffic load, 5400 Mbps, there are approximately 1050000 nonpaged pool allocations in the server which is less than the number of allocations during Smurf attack but more than that during the Ping flood attack.

One of the ways to measure the efficiency of a web server during an attack is to determine the number of client connections that the server is able to establish successfully compared to the connections it was able to make during the absence of any attack traffic. Initially, 2500 HTTP connections were sent to the web server to determine the baseline, the server was able to respond to all the HTTP requests and establish connections with the clients. Once the attack traffic was introduced, the number of legitimate connections gradually started decreasing with increasing

attack traffic load and increasing processor utilization. This is shown in Figure.2.18. Since the HTTP connections were not measured using the performance monitor but were monitored from the client, it was possible to determine the number of connections that the server was able to establish irrespective of whether the performance monitor was running or stopped. Therefore, it was possible to determine the number of HTTP connections that the server established when the full attack traffic load, 6000 Mbps, was sent to the server. Since the server was completely under the influence of the attack, it was not able to establish even a single connection with the client when the server was targeted with TCP-SYN attack traffic with a magnitude of 6 Gbps.

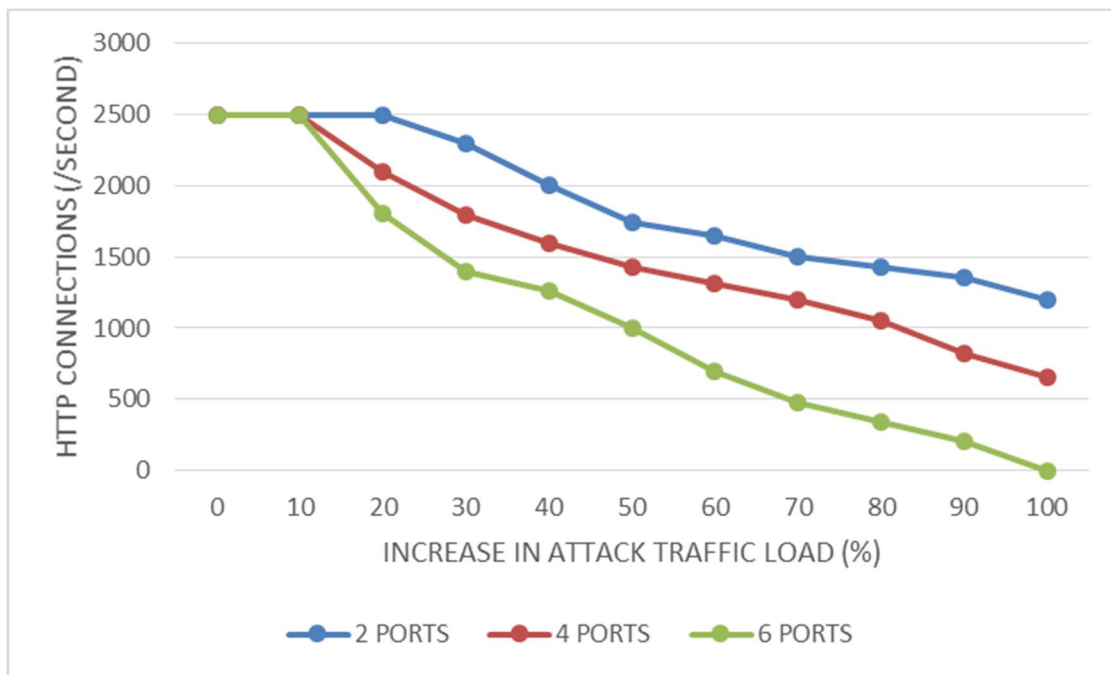


Figure 2.18 Number of legitimate Connections Established per second under TCP-SYN Flood Attack

Another parameter that could be used to determine the impact of the attack on the Windows server is the Connection Latency. For the Windows Server 2012 R2, acting as a web server, connection latency and the number of HTTP connections the server is capable of establishing is the key to determine the ability of the server to withstand the attacks. Figure 2.19 shows the

connection latencies for the first and the second experimental set up. The connection latency had been increasing proportional to the increase in the processor utilization.

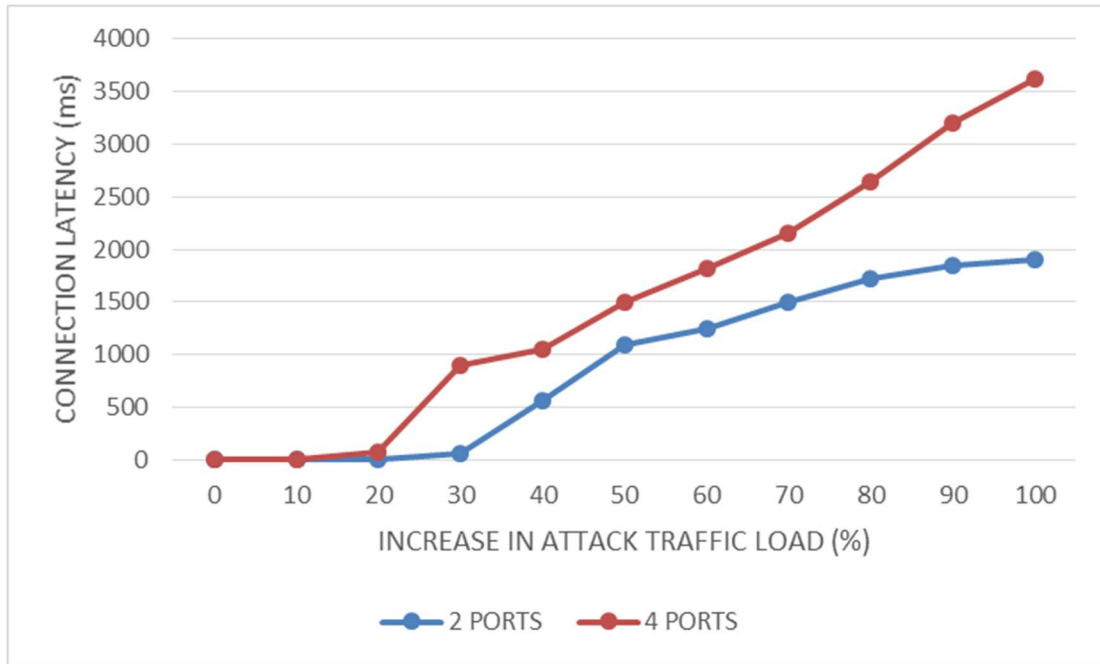


Figure 2.19 Connection Latency under TCP-SYN Flood Attack

In the absence of any attack traffic, the server establishes 2500 connections per second with nil latency. But after the attack traffic is introduced to the server, even though the server handles lesser number of connections than it did previously, the server is slow to respond to the connection requests causing an increase in the connection latency. During the last increase in the attack traffic load of the first and second experimental setups, the server took 1900 milliseconds and 3615 milliseconds respectively to establish connection with the clients.



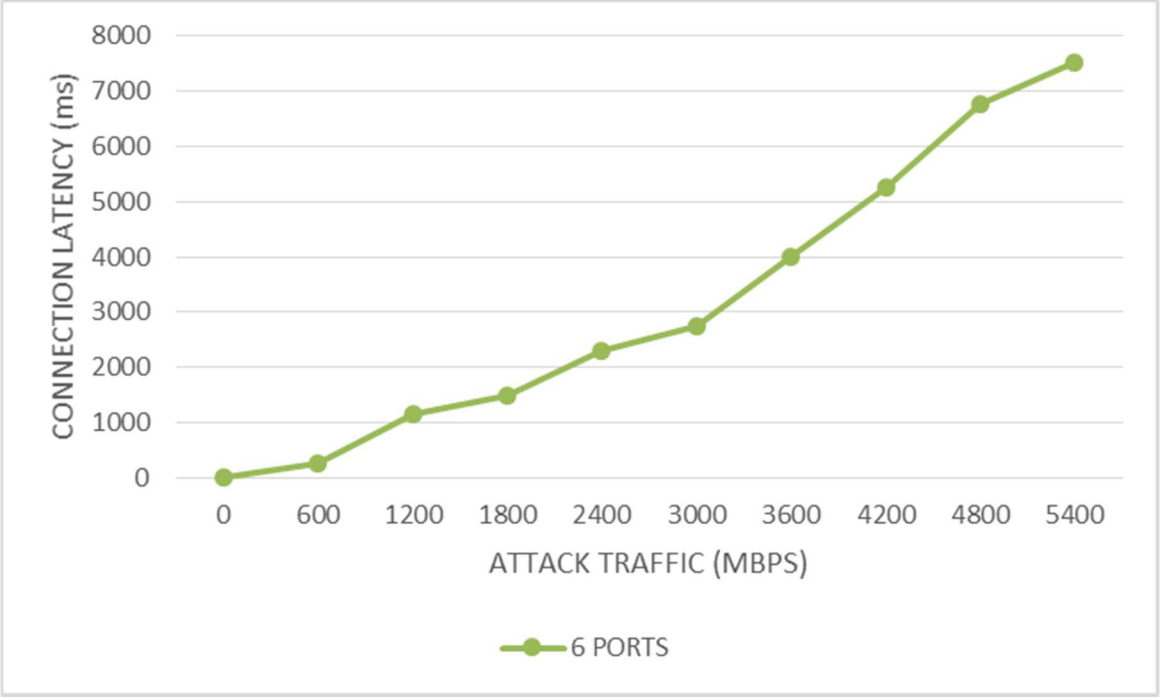


Figure 2.20 Connection Latency when TCP-SYN Flood Attack traffic is sent to six ports

For the third attack scenario, since the server does not establish any client connections when 6000 Mbps attack traffic is sent, the connection latency is displayed only up to 90 percent of the attack traffic load or 5400 Mbps. This is shown in Figure. 2.20. The connection latency observed for the penultimate step of the third experiment, is the highest observed so far, 7000 milliseconds. As mentioned earlier, the delay in connection establishment is as important as the number of connections that the server is able to maintain while under the impact of a DDoS attack. If the server takes nearly seven seconds to establish a connection with the client, although it might not seem like a long time, in the world of e-Commerce, it could prove disastrous to an organization. Hence it does not suffice if a web server handles more number of connections, it must also be able to do so with zero delay. The TCP-SYN attack is determined to be very much detrimental to the server in this respect.

**2.3.3.1 Blue Screen of Death (BSoD) due to TCP-SYN Flood Attack.** Considering the huge impact that the TCP-SYN flood attack had on the performance of the web server measured in terms of number of HTTP connections established and more importantly the connection latency, it was further analyzed using a slightly modified experimental setup.

Initially, the legitimate traffic was sent to the server for five minutes, then attack traffic was sent to the server along with the legitimate traffic. In order to observe the impact of the increase in volume of attack traffic, the number of HTTP connections that the server can handle has been recorded for different magnitudes of attack traffic ranging from 1 Gbps to 6 Gbps. Each increase in the attack traffic load along with the legitimate traffic was sent to the server for a duration of five minutes.

The baseline or nominal performance of the server in the absence of attack traffic was first determined by sending HTTP connection requests from legitimate clients at the rate of 60000 connections per second for five minutes. When there was no attack traffic, the server could successfully handle all the client connection requests that were sent to it. Once the baseline had been established, the attack traffic was introduced into the network along with the client traffic. Now, the attack traffic and the client traffic are sent to the server at the rate of 1 Gbps for 5 minutes. Due to the attack traffic, the number of legitimate connections that the server can handle decreased to 32000 connections as opposed to the 60000 connections established by the server in the absence of attack traffic as shown in figure.4. Then the attack traffic magnitude was increased and two Gbps attack traffic was sent to the server along with client traffic for the same duration, five minutes, this time the number of HTTP connections dropped to 25000 connections per second.

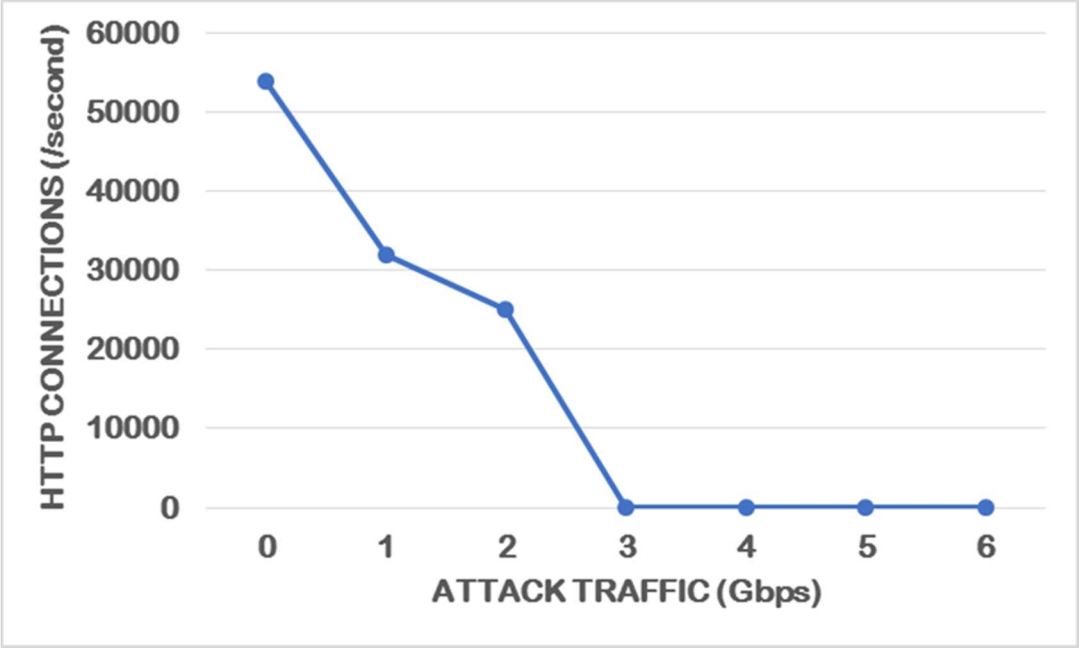


Figure 2.21 HTTP Connection Establishment of the victim server under TCP-SYN Flood Attack

With each increase in attack traffic load, the HTTP connections were recorded for different attack traffic magnitudes in the same fashion. Figure.2.21 shows the drop in the number of HTTP connections for various magnitudes of attack traffic. From the graph (Figure. 2.21) it can be observed that when 3 Gbps attack traffic was sent to the server, the server was unable to handle even a single HTTP connection request.

<b>MAGNITUDE OF ATTACK TRAFFIC (Gbps)</b>	<b>NUMBER OF HTTP CONNECTIONS (/second)</b>
Baseline or Nil Attack Traffic	60000
1	32000
2	25000
3	0
4	0
5	0
6	0

Figure 2.22 Number of HTTP connections handled by the server under TCP-SYN Flood Attack

The table (Figure 2.22) below displays the number of HTTP connections that the server could handle at different magnitudes of attack traffic. Five minutes after sending an attack traffic of magnitude 3 Gbps, when 4 Gbps attack traffic was sent, the server crashed in two minutes and displayed a Blue Screen of Death (BSoD) which displayed a Watchdog Violation Error (133) before restarting. The figure.2.23 displays the BSoD displayed by the server.

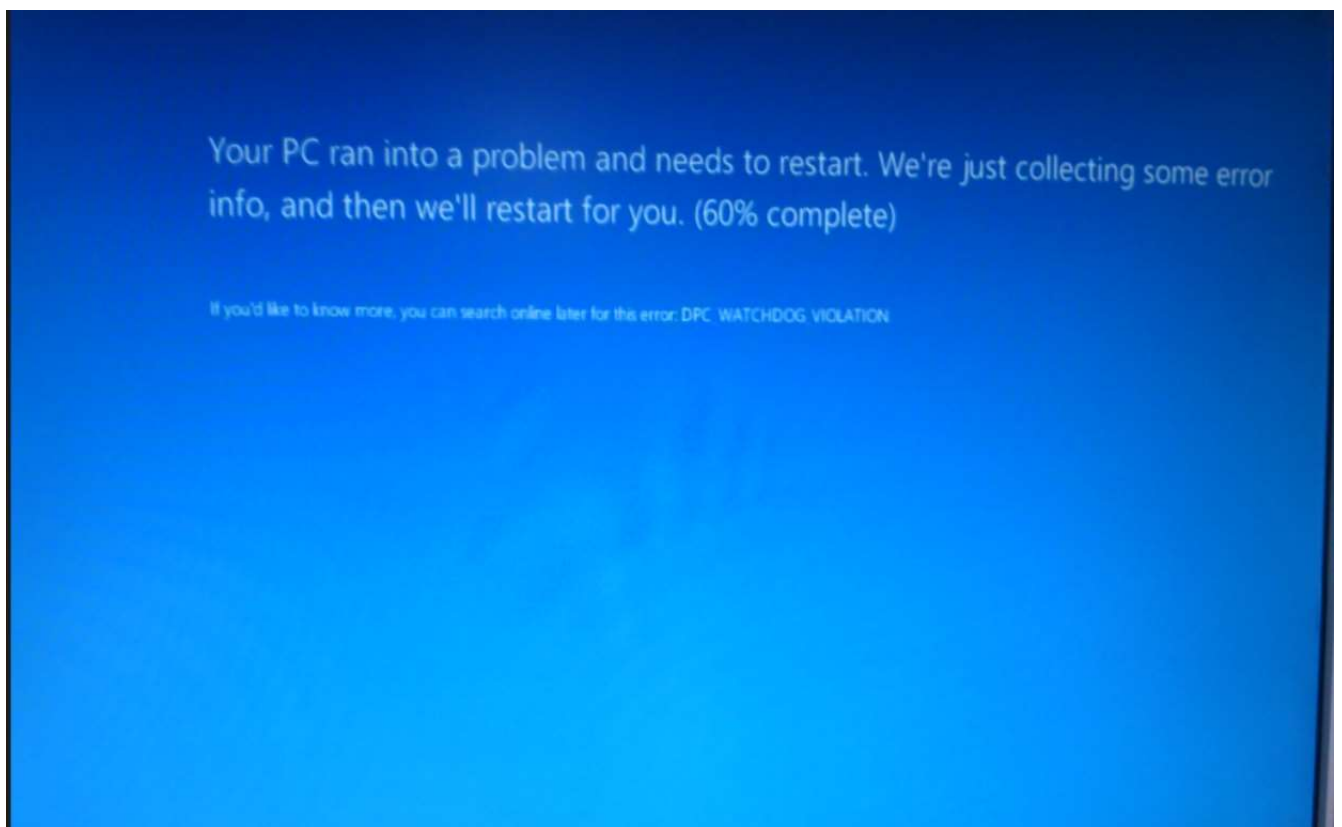


Figure 2.23 Blue Screen of Death (BSoD) displayed before the server crashed under 3.1 Gbps Attack Traffic

Once the server crashed at 4 Gbps attack traffic in two minutes, another set of experiments were carried out to determine the lowest magnitude of TCP- SYN attack traffic which forced the server to crash. Despite sending the attack traffic for thirty minutes, the server did not crash

while receiving 3 Gbps attack traffic. But when the attack traffic was increased by a magnitude of just one 1 Mbps, the server crashed in three minutes. Thus, the lowest magnitude of attack traffic that causes the Windows Server 2012 R2 to crash is 3.1 Gbps in a duration of three minutes. Following this, the attack traffic was sent at different magnitudes to determine the time the server is able to withstand the attack before crashing.

It was observed that with increasing magnitude of attack traffic, the server took lesser time to crash. The Figure.2.24 displays the time taken for the TCP-SYN flood attack traffic to crash the server starting from 3.1 Gbps and going up to 6 Gbps. It was determined that the server crashes in two minutes when it is attacked by 4 and 5 Gbps TCP-SYN attack traffic and crashes in just 1.5 minutes when 6 Gbps attack traffic is sent to the server.

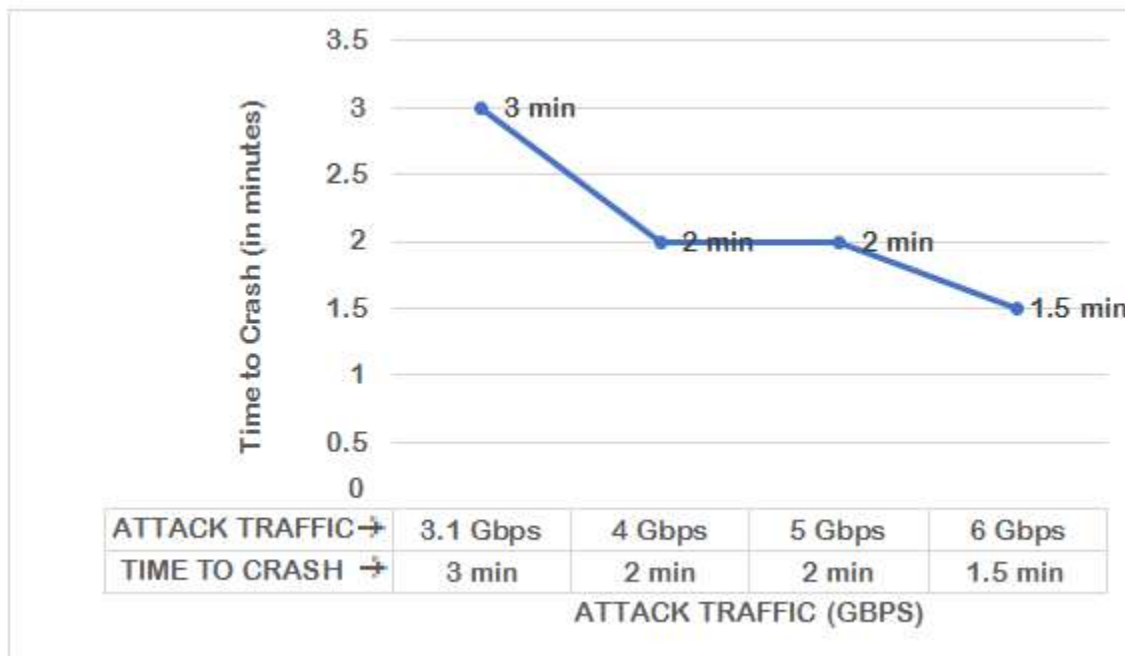


Figure 2.24 Duration of time the server is able to withstand the TCP-SYN Flood Attack Traffic before crashing

Over the years, Microsoft has improved its client and Server Operating Systems and made them less vulnerable to attacks compared to their predecessors. Being one of the highest used

server and client Operating Systems in the world, earlier versions of Microsoft Windows operating systems have been evaluated in the past [1]-[7]. After interpreting the results presented in the aforementioned publications it can be inferred that there has been a significant improvement in the protection mechanisms developed by Microsoft in the subsequent operating systems. Unfortunately, the ability to generate very high magnitudes of attack traffic has also improved and as a result the intensity of the attack on the target victim has increased multifold. The largest TCP-SYN flood attack observed in December 2015 had a magnitude of 155 million packets per second and peaked at 325 Gbps [9]. From the experiment it is evident that Windows Server 2012 R2, the latest server Operating System from Microsoft, would be not be to handle TCP-SYN attack traffic of magnitude higher than 3 Gbps.

#### **2.3.4 UDP Flood Attack**

For the same reason that the ports for receiving TCP traffic cannot be blocked, UDP traffic cannot be blocked either. TCP and UDP based DDoS attacks are the number one source of DDoS attacks [66]. The UDP flood attack was launched on the Windows Server 2012 R2 in three attack scenarios that are classified based on the number of ports of the server to which the attack traffic is sent.

In the first setup, processor utilization increases by ten percent going from seven percent to 17 percent once the attack is introduced, shown in Figure.2.25. Then the processor utilization kept increasing with increase in attack traffic magnitude. At the end of the first experiment, when the maximum attack traffic, 1 Gbps, was sent to two ports of the server, the processor utilization was 69 percent. From this experiment it can be inferred that the UDP flood attack has less impact on the server than Smurf and TCP-SYN attacks but more impact than the Ping attack.

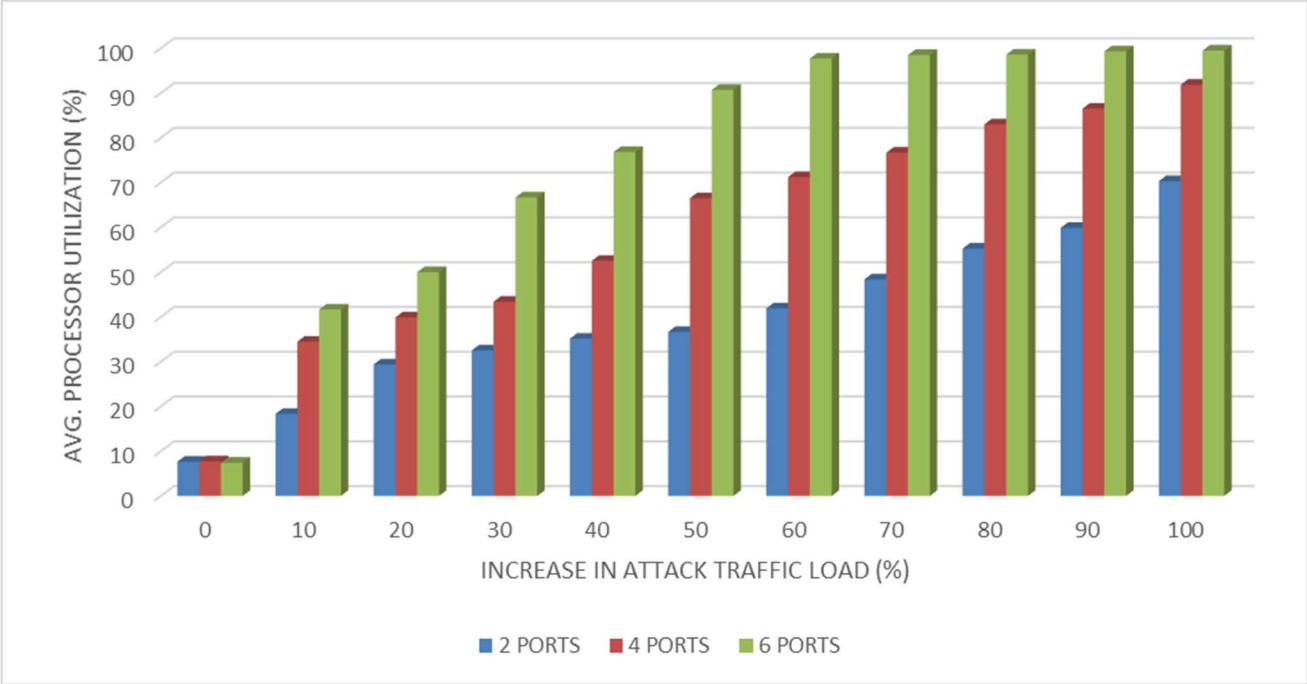


Figure 2.25 Average Processor Utilization of the server under UDP Flood Attack

In the second experiment, the processor is ninety percent utilized when an attack traffic of magnitude 4000 Mbps was sent to the server. Finally, in the third experimental setup, the processor utilization was 99 percent when 6 Gbps attack traffic was sent, but it does not force the server to stop the performance unlike in the case of TCP or Smurf. The core utilization corresponding to the average processor utilization is displayed in the Figure.2.26, this plot shows the results from the third setup where the attack traffic ranges from 600 Mbps to 6000 Mbps. All the four cores of the server are utilized equally throughout the duration of attack.



Figure 2.26 Core Utilization and Average Processor Utilization under UDP Flood Attack

It was deduced from the processor utilization of the server that the UDP flood attack has more impact than Ping attack but less impact than Smurf and TCP-SYN flood attacks. This can be confirmed from the memory utilization of the server under attack as shown in Figure 2.27.

Under Ping flood attack, the maximum number of nonpaged pool allocations was 960994 and for the UDP flood attack the nonpaged pool allocations are 1026362.



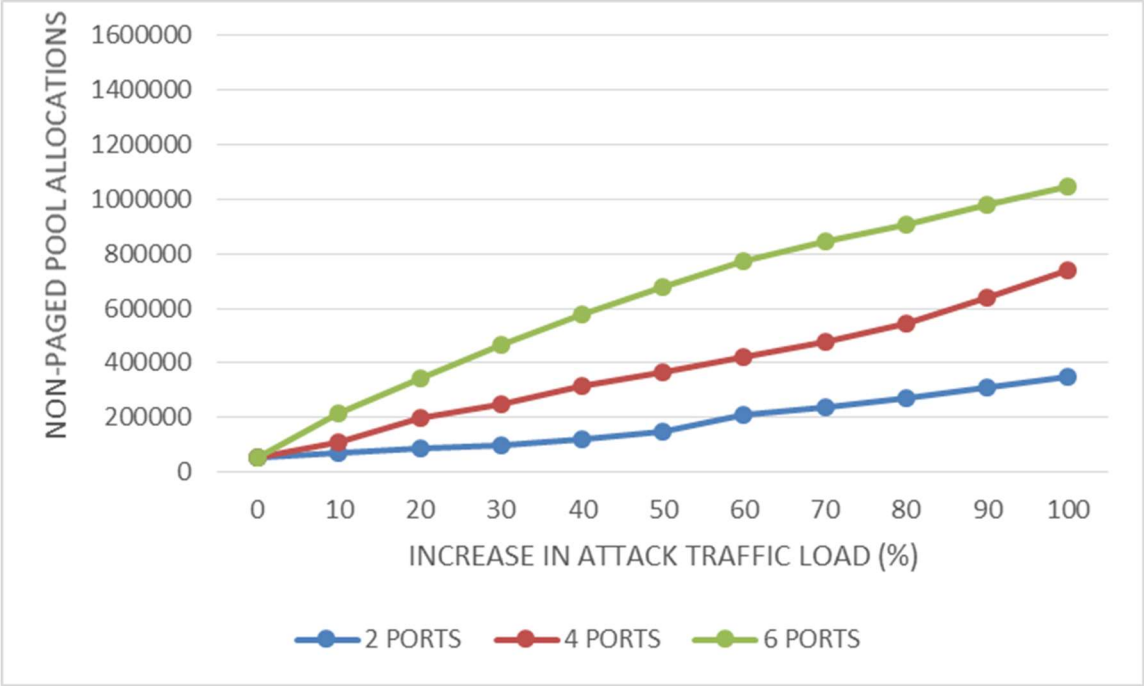


Figure 2.27 Nonpaged Pool Allocation in the server under UDP Flood Attack

Although the UDP attack is not as detrimental to the processor utilization as Smurf and TCP-SYN, it still has a very bad impact on the performance of the server operating system as a web server. This is reflected in the connection establishment and connection latency behavior displayed in Figures 2.28, 2.29 and 2.30. In the first experimental set up, the server is able to withstand an attack traffic of magnitude 200 Mbps and continue to establish 2500 connections per second after which the number of connections starts decreasing continuously. After receiving 1000 Mbps attack traffic each in two ports, the rate of HTTP connection establishment of the server started decreasing gradually such that the server was able to handle 1270 connections per second.

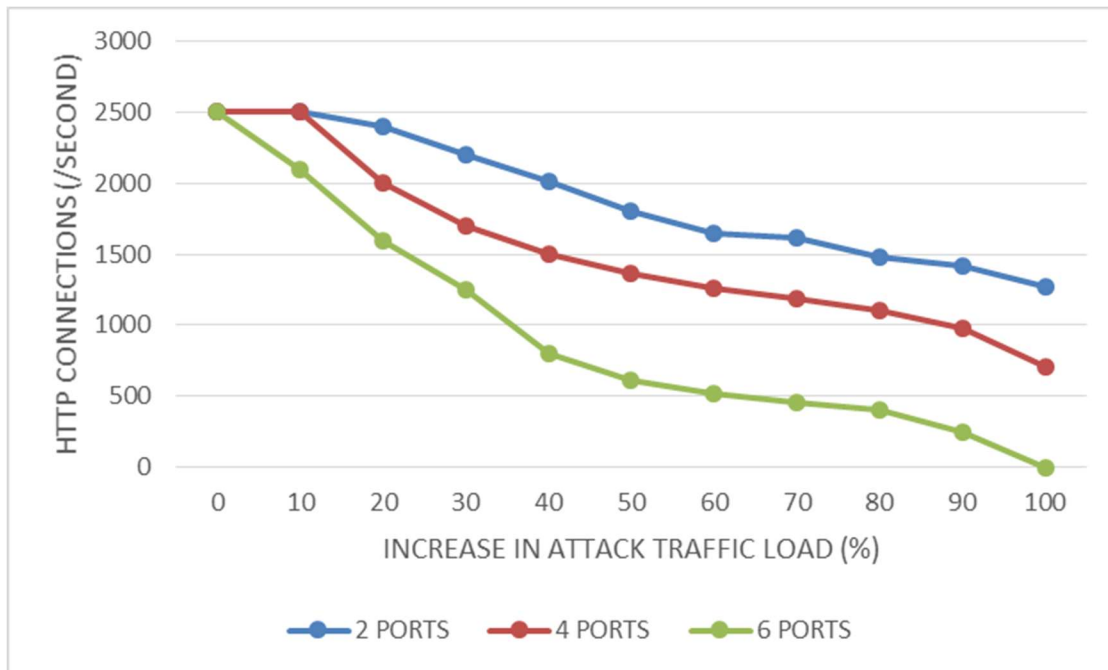


Figure 2.28 HTTP Connections established per second under UDP Flood Attack

In the second attack scenario, the server is able to sustain the connection establishment rate up to a ten percent increase in the attack traffic load after which the number of connections decreases considerably until it reaches the end of the experiment at which point the server is able to establish only 680 connections per second. Even though UDP flood attack does not cause the server’s processor utilization to reach 100 percent, the server becomes incapable of responding even to one legitimate client connection in the third scenario when 6000 Mbps traffic is sent to the server. Since the server does not establish any connections during the final increase in attack traffic load of the third scenario, the connection latency for the third scenario is shown separately in Figure.2.30 while the connection latencies for the first two setups are displayed in Figure.

2.29.

In case of the first scenario, there is no delay in the connection establishment until a thirty percent increase in the attack traffic load. When forty percent of the total attack load (200 Mbps)

is sent to the server, the server takes 200 milliseconds to establish a connection. After that the connection latency keeps increasing and reaches a value of 1480 milliseconds at the end of the first experiment when the server receives 2000 Mbps attack traffic. When the attack traffic is sent to four ports and then to all the six ports of the server in the second and third experiments, a connection latency can be observed from the time the attack traffic is introduced to the server. The server takes 2400 milliseconds to establish a connection at the end of the second experiment and takes 3300 milliseconds when ninety percent of the full load (6000 Mbps) is sent to the server in the third experimental setup. When hundred percent of the attack traffic load is sent to the server, the server does not establish any connection with the clients.

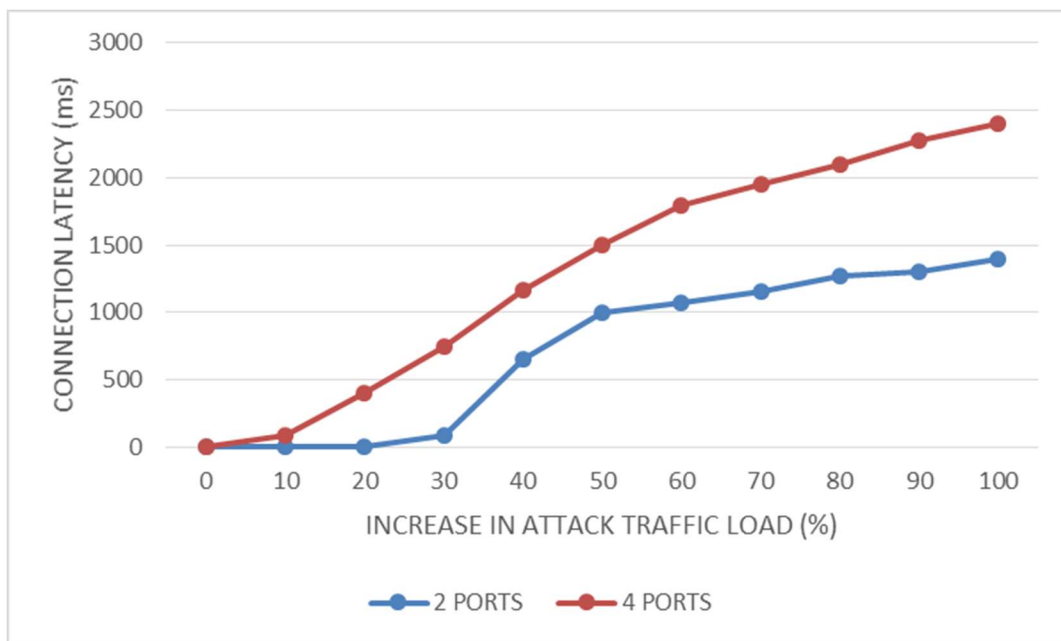


Figure 2.29 Connection Latency of the server under UDP Flood Attack

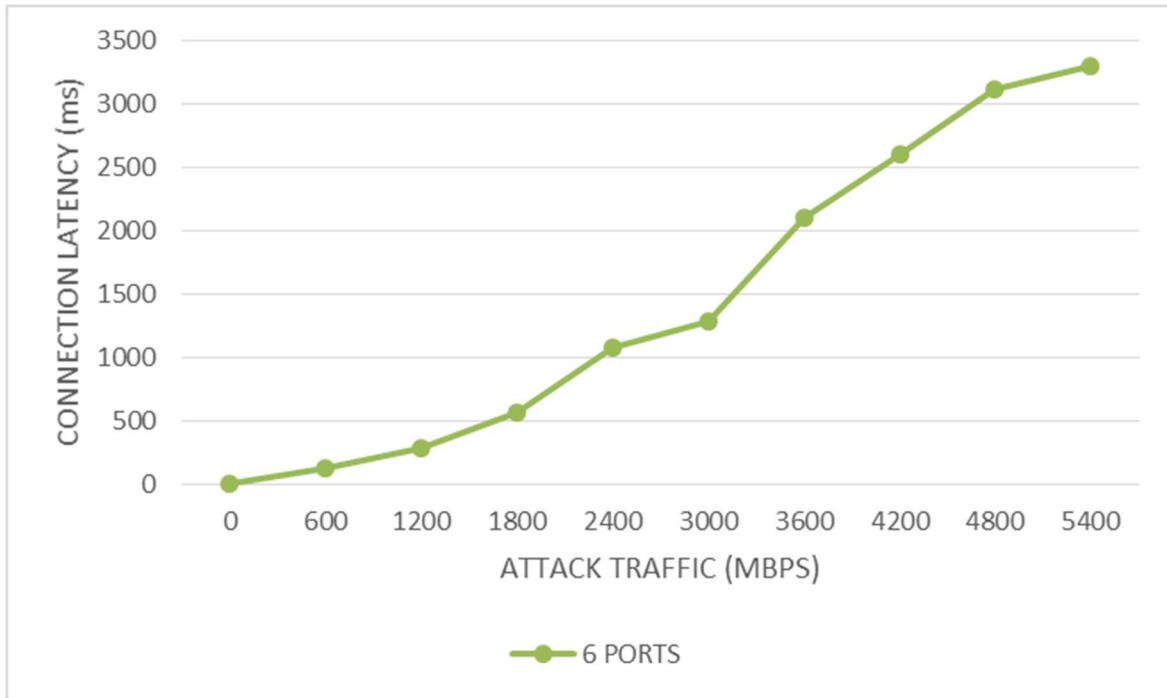


Figure 2.30 Connection Latency of the server when UDP Flood Attack traffic is sent to six ports

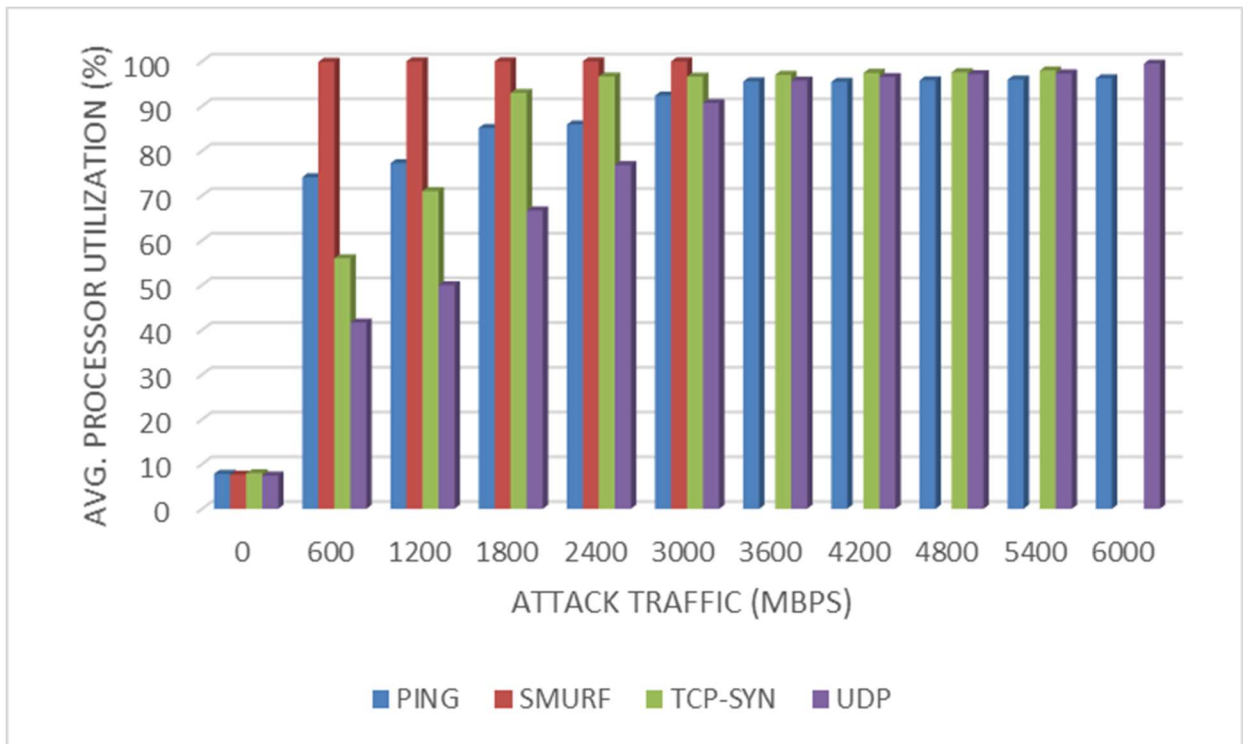


Figure 2.31 Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on Average Processor Utilization

Figures 2.31, 2.32 and 2.33 display the Average Processor Utilization, Nonpaged Pool Allocations and HTTP Connection establishment with the clients. If the four attacks have to be listed in the increasing order of impact on the victim server, the order would be Ping flood, UDP flood, TCP-SYN flood and Smurf. Smurf attack has the highest impact on the target victim but it does not crash the server. If Smurf attack is the most dangerous of all the four attacks, why does it not crash the server? How does TCP-SYN flood attack which is comparatively less effective than Smurf crash the server? And why does TCP-SYN flood attack alone crash the server and not any of the other three attacks?

The answer to all the above questions lies in the fact that only TCP-SYN packets have the connection establishment mechanism which involves half open connections wherein the server must allocate memory for the TCP-SYN packets that it receives. The other three attacks utilize the memory only through nonpaged pool allocation, but the TCP-SYN flood attack, in addition to utilizing memory through nonpaged pool allocation, also forces the server to utilize memory for maintaining the half open connections for the barrage of TCP-SYN packets the server receives. Since the ways for memory utilization for TCP-SYN flood attack was twofold, it proved to be fatal to the server causing it to crash. TCP-SYN attack did not affect the server when it was sent in increments of ten percent attack traffic load since the increase in nonpaged pool allocation was linear. The Blue Screen of Death was observed when 3.1 Gbps attack traffic was introduced to the server when it was only receiving legitimate client traffic. This caused an exponential increase in the memory which resulted in crashing the server.

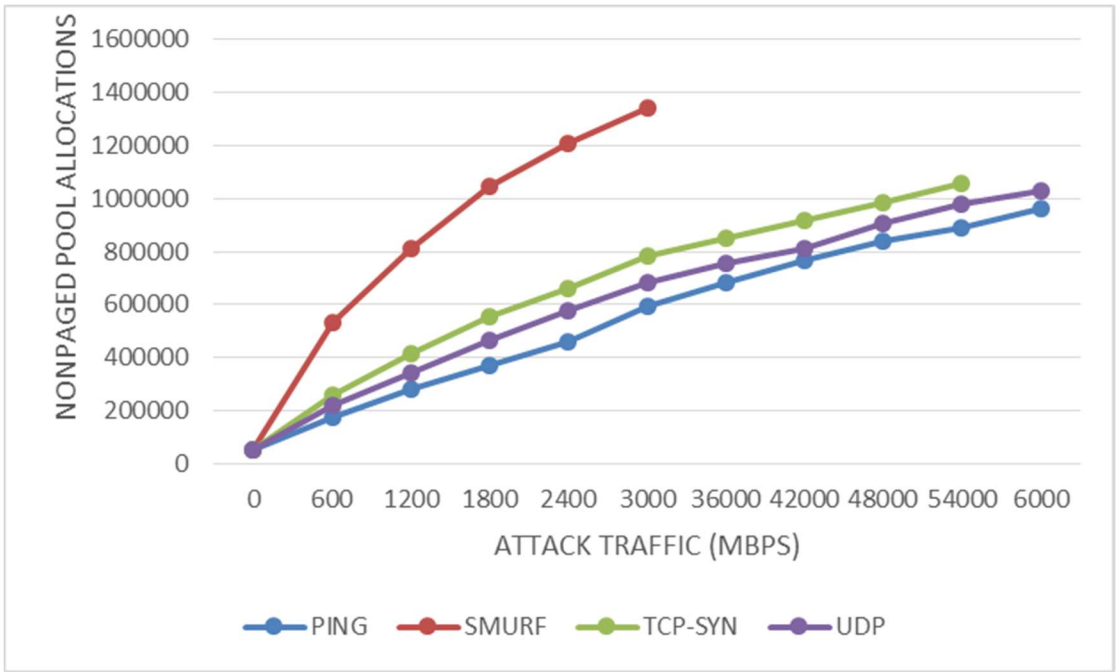


Figure 2.32 Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on Nonpaged Pool Allocations

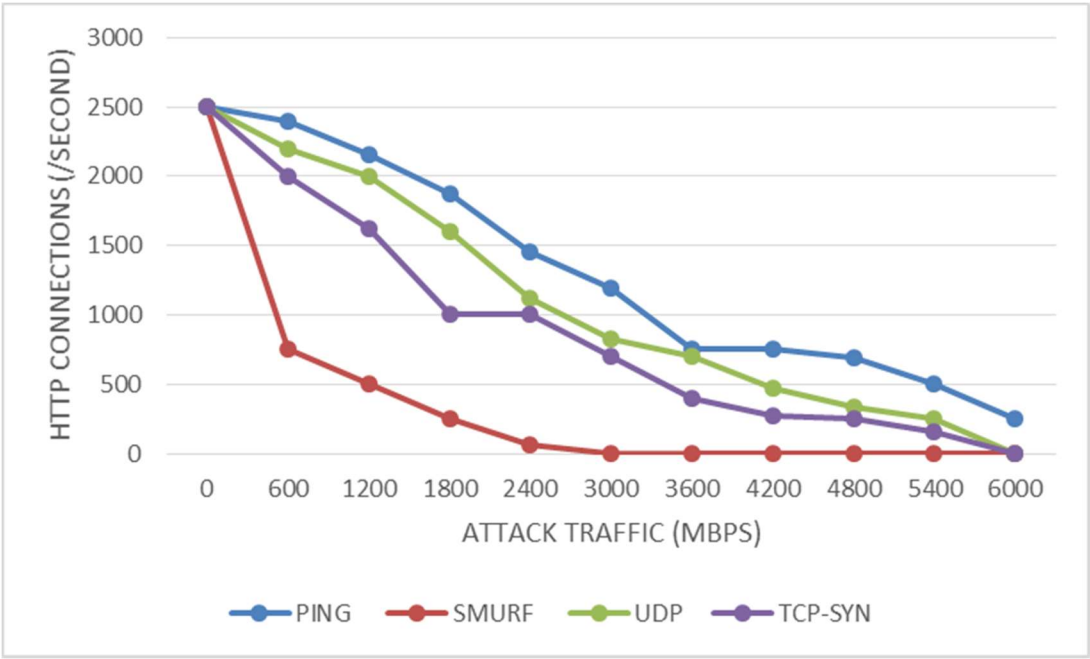


Figure 2.33 Comparison of the Effect of DDoS Attacks on Windows Server 2012 R2 based on HTTP connection establishment

## 2.4 Chapter Summary

This chapter analyzed the effect of four most popular DDoS attacks on the latest Microsoft server Operating System, Windows Server 2012 R2. Although it looked like the increase in processor utilization was the reason why the performance of the server was affected, it was also the memory utilization of the server that targeted by the DDoS attacks. The DDoS attacks aim to deplete the physical memory of the victim server to alarmingly low levels, the effect is popularly known as memory leak [25], [26]. This drop in the memory triggers an increase in the processor utilization and affects the overall efficiency of the victim server which is reflected through the connection establishment and connection latency of the server.

Thus, this chapter analyzed the impact of DDoS attacks on the Windows Server 2012 R2 operating system before virtualization. In the next chapter, the changes in the defense of the same operating system (Windows Server 2012 R2) is analyzed when it is a Virtual Machine installed in Hyper-V. The comparison of the results obtained in this chapter with the results obtained after virtualization of the server would help evaluate the changes introduced by virtualization to a system in the network security front.

## CHAPTER III

### EVALUATION OF THE EFFECT OF VIRTUALIZATION ON THE AVAILABILITY OF SERVERS UNDER DDoS ATTACKS

The *raison d'être* of virtualization is to maximize the utilization of hardware resources such as processing power, memory, input/output devices. Each Virtual Machine or guest operating system installed on a hardware is allocated only a portion of the hardware resources. Since more than one guest operating system shares the resources of a single hardware, it might be possible that the performance of the same operating system could be different when it is a virtual machine and when it is not.

In general, a virtual machine is allocated virtual processors based on the role of the virtual machine, depending on the capacity of the hardware and the requirements of the virtual machine, multiple virtual machines are installed on a single hardware. However, in this chapter, only one virtual machine with Windows Server 2012 R2 Datacenter operating system was installed in the server in addition to the host operating system. This virtual machine was allocated all the cores in the server hardware to ensure that there is no difference in the processing power that was available to the non-virtualized server and the virtual machine.

To achieve more similarity between the virtualized and non-virtualized systems that are under comparison, the Operating Systems of the non-virtualized system and that of the virtual



machine were both chosen to be the same, Windows Server 2012 R2. By allocating equal processing power to the virtual machine and the non-virtualized system and running the same Operating Systems, the overhead due to virtualization can be observed. Thus the goal of this chapter is to study the extent to which virtualization affects a system by comparing the impact of well-known DDoS attacks on a virtualized and a non-virtualized system.

### **3.1 Experimental Setup**

The Windows Server 2012 R2 Standard operating system is installed in a Dell PowerEdge T320 [37] hardware with Intel Xeon E5-2407 v2 quad core 2.4 GHz processor [38] and 8 GB RAM. The built-in firewall of the server was enabled with the default settings throughout all the experiments. The experimental set up is shown in Figure 3.1. The attack traffic was simulated in a controlled environment at the Network Research Lab at the University of Texas Rio Grande Valley (UTRGV).

The Windows Server 2012 R2 Operating System was initially tested against four most popular DDoS attacks, Ping Flood, Smurf, TCP/SYN and UDP Flood attacks, in Chapter II. Now, the Windows server OS is virtualized in order to test the effect of the same four attacks on a virtual machine. In order to install virtual machines, the Hyper-V manager is installed in the server OS. The Hyper-V manager is installed in the Windows OS through the Server manager console [31].

To install Hyper-V, the Add roles and features option under the Configure the local server is clicked and the role-based or feature-based installation option is chosen. Click on Next and choose the select a server from the server pool option and under Roles, choose Hyper-V. Continue to click on Next until you reach the finish installation screen and then restart the server.

From the server manager console, under the Tools tab choose Hyper-V Manager to open the Hyper-V manager. In the Hyper-V manager, under the actions tab, click on New → Virtual Machine, this will open a New Virtual Machine Wizard. Click on Next in the Before you Begin Window. The name and location for the Virtual Machine can be entered and then click on Next to choose the Generation of the Virtual Machine depending on the type of OS. If the Virtual Machine or the Guest Operating System is a 64-bit version of Windows 8 or Windows Server 2012 or later, then choose Generation 2, else choose Generation 1. It is important to note that the generation of a virtual machine cannot be changed after the virtual machine has been created. The generation of the virtual machine is to be chosen correctly to ensure that support is provided for features such as SCSI boot, Secure Boot and PXE boot using a standard network adapter. The Operating System of the Virtual Machine is Windows Server 2012 R2, hence generation 2 was selected.

Next, the startup memory for the virtual machine is assigned [32]. The requirement of the startup memory is decided based on the role of the virtual machine and the Operating System that the Virtual Machine will run. For the Windows Server 2012 R2 guest OS that was being installed, 512 MB was assigned as the startup memory. Click on Next and choose a Virtual switch for the virtual machine. Virtual Switches are broadly classified into three types: External, Internal and Private [33]. For the purpose of the thesis, an external switch is designated to the VM. Following this step, the location from which the image of the virtual machine is to be installed is specified and then the installation options are selected. Finally, the virtual machine with the Windows Server 2012 R2 operating system is created.

As the goal of the chapter is to determine the overhead of virtualization, the VM is allocated all the cores available in the server hardware, four physical cores. This is to make sure

that the virtual machine has access to the same amount of processing power that was available to the now host operating system which was previously tested under different DDoS attacks in Chapter II. The server hardware (Dell PowerEdge T320) used in the thesis consists of a quad core Intel Xeon E5-2407 v2 processor. As the processor in the server hardware does not support hyper-threading [33], the number of logical processors is equal to the number of physical cores. By allocating the same amount of processing power to the virtual machine as the non-virtualized server, the overhead due to virtualization can be observed. Although the same processing power was allocated to the virtual machine, the memory allocated to the VM was different from the memory available to the non-virtualized system. Although both the processor and the memory allocation are decided based on the need of the virtual machine, a startup memory of 512 MB was allocated to the virtual machine in order to ensure optimal performance [32].

The virtual machine is also set as a web server like the non-virtualized system which is now acting as the host Operating System on the same hardware. The victim server OS, Windows Server 2012 R2, in the virtual machine was set up as a Web server, as a result, the latest version of Internet Information Services (IIS 8.0) service was installed in the server OS following the instructions in [14].

A sample webpage called index.html is created in the victim web server which is running the IIS service. Once the sample web page is created, the IIS manager could be accessed from the Server Manager console. The IIS manager provides users with the option of adding a new webpage to the web server. Under the server name, in the tab called web pages, the newly created sample webpage “index” is added. Then it is verified from the application pools tab if an application pool was created for index. Now another sub category called index would appear below webpages. Under the webpage settings, an option for a default document icon is chosen.

In the default document icon, the entry index.html is moved up so that it can be accessed through an HTTP request.

Whenever the server receives a Hyper Text Transfer Protocol (HTTP) request from a client requesting the webpage index.html, the server responds to the request by sending the webpage through an HTTP reply. In order to recreate a typical web server environment, the HTTP requests were sent by means of simulating the users or clients in the lab. Throughout the thesis, the terms legitimate traffic or client traffic are also used to refer to HTTP requests.

The most well-known DDoS attacks, Ping, Smurf, TCP/SYN flood and UDP flood attacks, were launched on the target server. The attack traffic was sent at the range of 100 to 1000 Mbps or 1 Gbps to the virtual machine. The legitimate or client traffic was sent at the rate of 2500 HTTP requests per second to the VM server. The experimental setup is shown in figure 3.1

Initially, only the legitimate HTTP connections were sent to the server in order to establish the baseline behavior of the victim server in the absence of any attack traffic. The legitimate HTTP connections requests were sent to the server for five minutes. Once the baseline of the server was established, the attack traffic was injected into the network along with the legitimate traffic. The attack traffic magnitude sent to the guest OS ranged from 100 Mbps to 1000 Mbps. The attack traffic load sent was increased in increments of 100 Mbps for every five minutes and was sent to the server simultaneously along with the HTTP client requests.

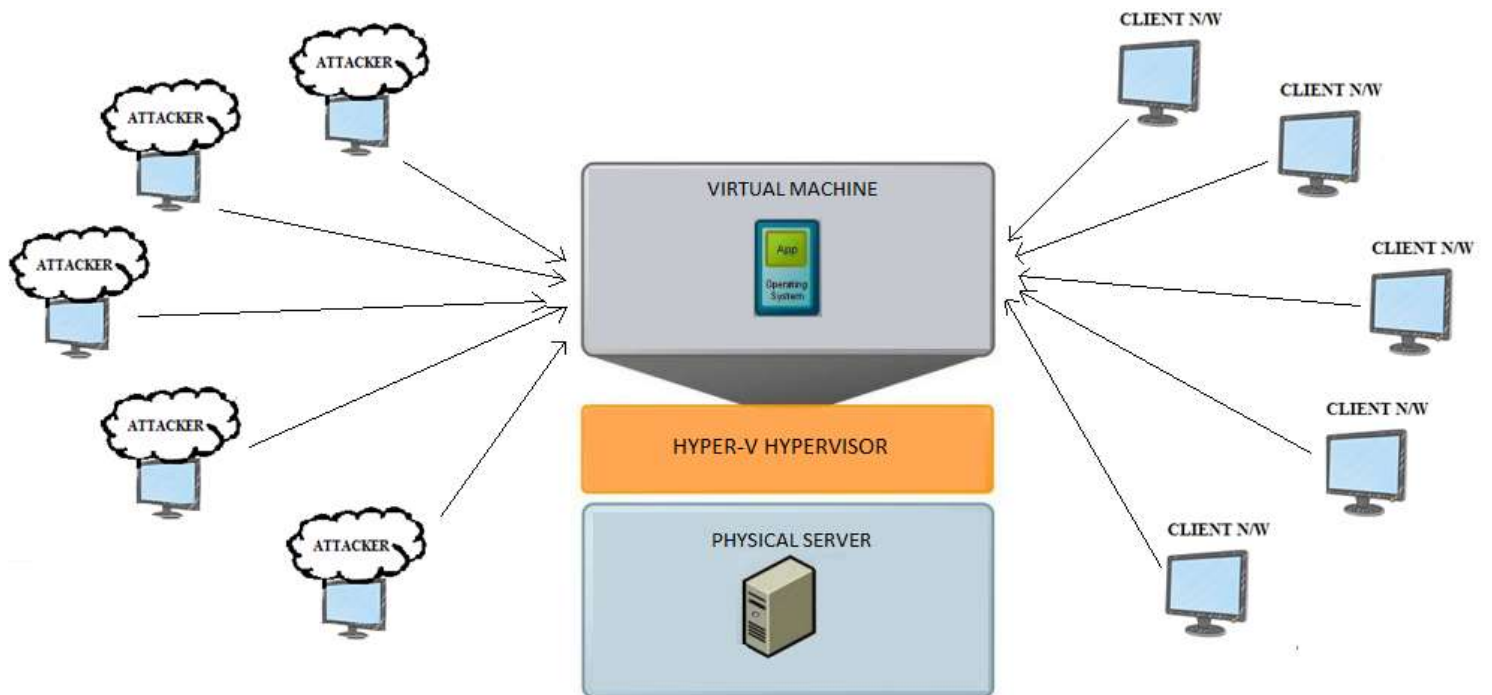


Figure 3.1 Experimental Setup

Thus, once the baseline behavior was captured, an attack traffic load of 100 Mbps was introduced to the server with the legitimate traffic, then after five minutes, the attack traffic load was changed to 200 Mbps and sent along with the client traffic. Each attack traffic load was sent to the server for a uniform duration of five minutes. This process was repeated until an attack load of 1000 Mbps was sent to the victim virtual machine. Hence, the total duration of the experiment was fifty five minutes, five minutes for obtaining the baseline and five minutes for each increased attack traffic load. Various parameters of the web server were monitored and recorded to enable the comparative evaluation of the virtual machine and the server before it was virtualized.

### 3.2 Parameters of Performance Evaluation

The evaluation of the difference in the performance of the non-virtualized server and the Virtual Machine was done through the comparison of five key performance parameters. The parameters that were monitored were the Processor Utilization, Core Utilization, Memory Utilization in terms of Nonpaged Pool Allocations, Number of HTTP connections that the server can establish with the client per second and the Connection Latency. Similar to the previous chapter, the HTTP connection establishment and the connection latency are both measured using the client simulation software.

The parameters that were monitored during the experiment were the Average Processor Utilization, Core Utilization, Nonpaged Pool Allocations, the number of HTTP (Hyper Text Transfer Protocol) connections handled by the server and the Connection Latency experienced by the clients. The aforementioned counters help in the determination of the performance of the non-virtualized server but some additional counters were employed to determine the performance of the virtual machine. These additional counters are run in the host operating system. All these parameters were measured throughout the duration of the experiment starting from the baseline behavior. These parameters were recorded by using the Data Collector Sets available in the performance monitor of the Windows Server 2012 R2 and the counters available in the client systems.

The Processor Utilization of a computer is analogous to the heartbeat and has a strong influence on the performance of the server. The name of the counter that is used to monitor processor utilization is known as `\Processor(_Total)% Processor Time` and is defined as “The percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then

subtracting that value from 100%. (Each processor has an idle thread that consumes cycles when no other threads are ready to run)” [10]. The Total Processor Utilization is the average of core utilization of all the cores in a server, in this case the server has four cores. The Central Processing Unit (CPU) has to be functional at all times for the server to be able to deliver its most efficient performance. Monitoring the processor utilization enables a person to accurately observe the effect that an attack has on the server. Whenever the CPU utilization exceeds its optimal value, it will start impacting the efficiency of the server.

The second parameter that was monitored during the course of the experiments was the Core Utilization. The core utilization counter, available under the % Processor Time, can be used to determine the processor consumption of each individual core present in the system. The counter \Processor(\_Total)\% Processor Time is the average of the core utilization of all the cores in a processor. The processor used for the thesis consists of four cores hence, the counters that were used to monitor the core utilization of the server are \Processor(0)\%Processor Time, \Processor(1)\%Processor Time, \Processor(2)\%Processor Time and \Processor(3)\%Processor Time. Although the term Total processor utilization might be misunderstood as the sum total of core utilization of all the cores in a server, Total Processor Utilization actually refers to the average of the average of the core utilizations in a server.

To monitor the processor utilization of the virtual machine, the performance counter called \Hyper-V Hypervisor Logical Processor(\_Total)\% Total Run Time found in the host operating system is used [34]. This counter can be used to accurately determine the processor utilization of the guest operating system installed in Hyper-V. There are three thresholds that can be used to determine the state of the guest operating system. If the percentage of total runtime is less than 60 percent, then the guest operating system is considered to be healthy. When 60 to 89 percent of

the total run time is used, then the guest operating system has to be monitored, if 90 to 100 percent of the time is used, then the state of the virtual machine is considered to be critical.

If processor utilization can be used to observe the effect of an attack on the server, the change in the memory usage of the server will throw light on the root cause of the issue, impact of DDoS attacks on servers. Analyzing the memory utilization of the server will help explain the reason behind why an attack has such a huge impact on the performance of the server. The Non-Paged pool and the Paged pool are the two memory resources used by an Operating System and its device drivers for storing data structures. The Non-Paged pool in the memory can only be allocated in the physical memory and not in the virtual memory unlike in the case of Paged pool [11]. As a result, the number of non-paged allocations is considered to be a representation of the memory utilization and is monitored throughout the experiment. The performance counter used is called `\Memory\Pool Nonpaged Allocs`. The same counter is run in the virtual machine to determine the memory utilization of that guest operating system.

Depending on the type of services installed in a server, the performance efficiency of the server can be analyzed using different parameters. The role installed in the Windows Server 2012 R2 Server OS both in the non-virtualized server and the Virtual Machine is the Internet Information Services (IIS) Manager. The IIS Manager is used for the configuration and management of the services offered by a web server. Therefore, the number of HTTP connections that the server is able to establish with the clients is chosen as one of the parameters to measure the performance of the server. The web server hosts a URL (Uniform Resource Locator) for the web page called `index.html`. The simulated client network records the statistics of the communication with the server. The clients monitor the number of positive HTTP



responses received from the server, which in turn can be used to judge the efficiency of the web server as the attack progresses.

The delay caused in responding to an HTTP request, also known as Connection Latency, is considered as one of the deciding factors to determine the efficiency and quality of a web server. Therefore, the connection latency is also monitored to analyze the strain that the attack causes to the server and how it affects the speed of response. The Connection Latency is defined as “The average time elapsed between the time the client sends a SYN packet and the time it receives the SYN/ACK.” Connection latency is measured in microseconds in the counter available in the client. In this thesis, the connection latency is represented in milliseconds.

### **3.3 Results and Discussion**

#### **3.3.1 Ping Flood Attack**

The Ping flood attack was launched by sending a flood of ICMP echo request messages to a victim system affecting the bandwidth and the processing power of the targeted victim. As mentioned earlier, the aim of the chapter is to study the effect of virtualization. The same number of HTTP connections were sent to the non-virtualized system and the virtual machine for the same duration. Once the baseline was established, the Ping flood attack traffic was introduced to the victim web server as mentioned in the experimental setup.

The figure 3.2 shows the comparison of the processor utilization of the virtual machine and a non-virtualized system. When the HTTP connections were sent in the absence of attack traffic, the average processor utilization was approximately seven percent in a non-virtualized system but in case of the virtual machine, the processor utilization was nearly 15 percent. The same number of HTTP connections are sent to both the systems, hence the difference in the magnitude of processor utilization is the overhead due to virtualization. Once the attack traffic of magnitude

100 Mbps was introduced, the processor utilization of the virtual machine immediately shot to 42 percent, which is approximately three times the processor utilization in the absence of attack traffic. When the non-virtualized system was attacked, the processor utilization increased to nearly twice the baseline processor utilization and reached 17.5 percent.

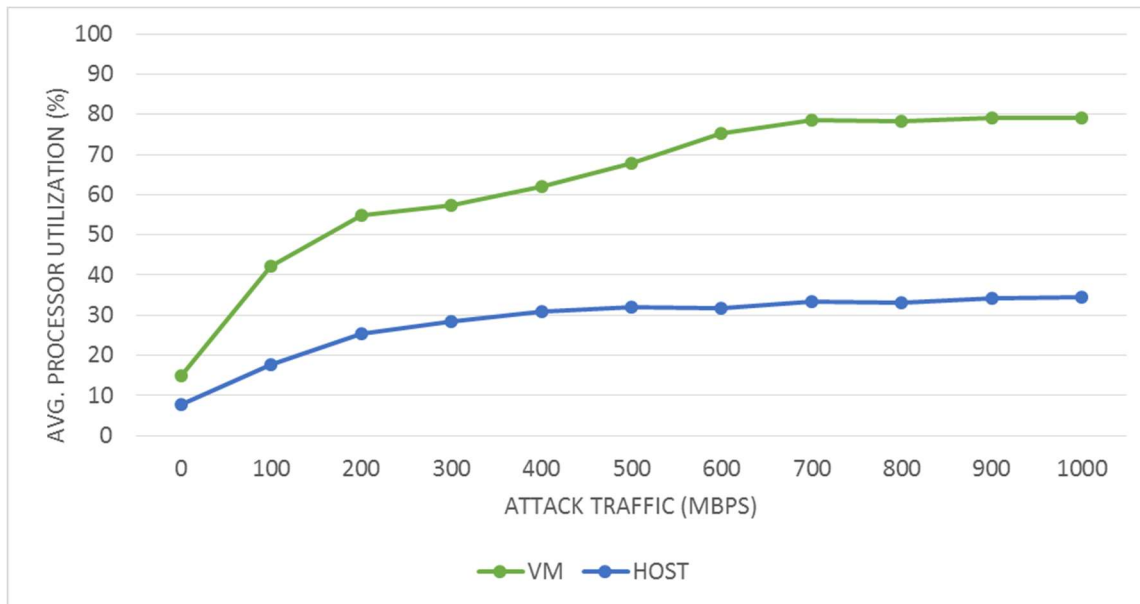


Figure 3.2 Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack

The attack traffic magnitude was increased by 100 Mbps every five minutes and the effect of the attack on both the systems was analyzed. With increasing attack traffic, the average processor utilization of both the systems increased, but the increase was much higher in the virtual machine. After the rapid increase in utilization when the attack traffic was introduced, with each 100-Mbps increment in the attack traffic magnitude, the processor utilization of the virtual machine increased by five to ten percent from its previous value.

Initially, the processor usage of the non-virtualized system also increased in a similar fashion but after the magnitude of attack traffic was increased from 300 Mbps, the processor utilization was nearly constant through the remaining duration of the experiment. Hence, even

though the processor utilization increased with increasing attack traffic in the non-virtualized system as well, since the increase in utilization was marginal it can be inferred that the effect of the attack on the non-virtualized system was less pronounced.

When the maximum attack traffic magnitude, 1000 Mbps, was sent to the servers, the average processor utilization was 80 percent and 35 percent for the virtual machine and the non-virtualized system respectively. This high difference in the processor usage in the two systems indicates that virtualization decreases the ability of the operating system to defend against Ping flood attack.

The figure 3.3 displays the processor utilization of the virtual machine in the form of core utilization and average processor utilization. As indicated by the legend in the graph, the light blue dotted line and the scale on the right indicate the Average Processor Utilization. The scale on the left and the bars indicate the utilization of each core during the baseline and throughout the duration of the attack. The average processor utilization is the average of the core utilization. From the graph it can be observed that at any given time, the four cores in the processor are not equally utilized.

When only the legitimate traffic was sent to the virtual machine to obtain the baseline, the core utilization was the highest in core 1 followed by core 3, core 4. The core 2 was the least utilized of the four cores. Once the attack traffic was introduced, the core 2 became the most used and the core 1 was the least used of the four cores. This can be because the throughout the duration of the experiment, the core 2 was the most utilized followed by core 4 and then after 600 Mbps attack traffic was sent, the core 3 and the core 4 are almost equally utilized. Initially, the core 1 was used more than core 3, the core 1 was the least used of the four cores after an attack traffic of magnitude 500 Mbps was sent to the virtual machine.

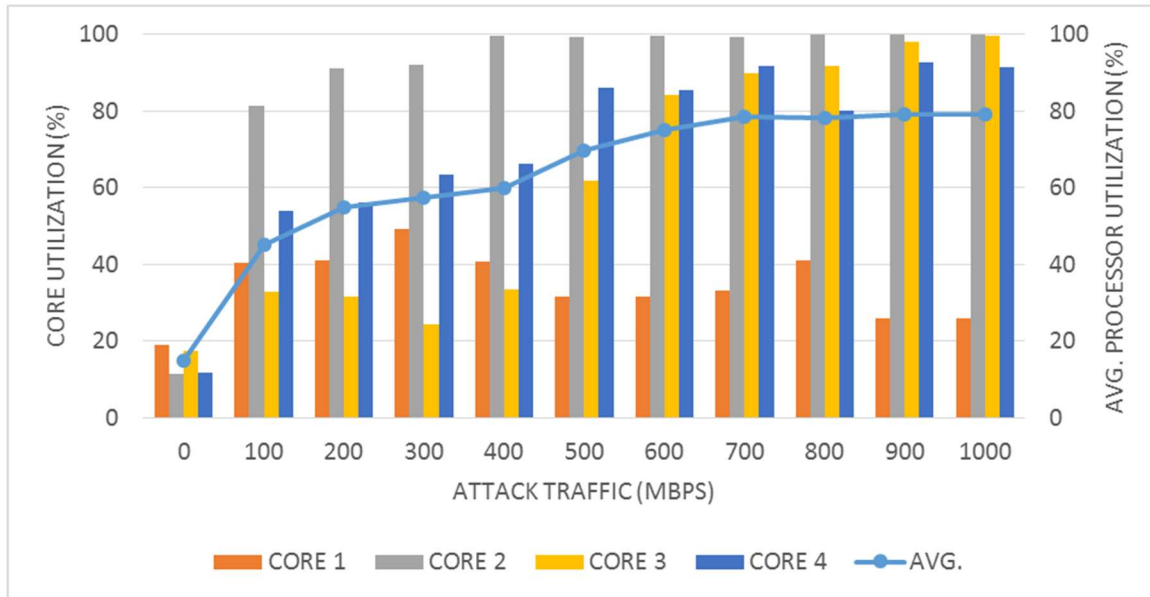


Figure 3.3 Processor Utilization of the Virtual Machine under Ping Flood Attack

The figure 3.4 shows the processor utilization of the non-virtualized machine acting as a web server before it was virtualized. One striking difference that can be noticed between the core utilization before and after virtualization is that before virtualization the four cores were nearly equally utilized, but after virtualization, there was an uneven utilization of the four cores in the server. Upon further examination, before virtualization, the cores are equally used in the absence of attack traffic, after the attack traffic was introduced, cores 2 and 4 are utilized more, although marginally, compared to cores 1 and 3.

From this observation it might seem that the uneven core utilization is triggered due to the attack traffic, although this is a plausible inference from the results obtained from the non-virtualized server, this does not explain the unequal core usage in the virtual machine. If the unequal utilization is due to the attack traffic, then in the absence of attack traffic, the cores must be equally utilized but this is not the case with the virtual machine. Hence the reason for the uneven core utilization in the virtual machine is the virtualization.

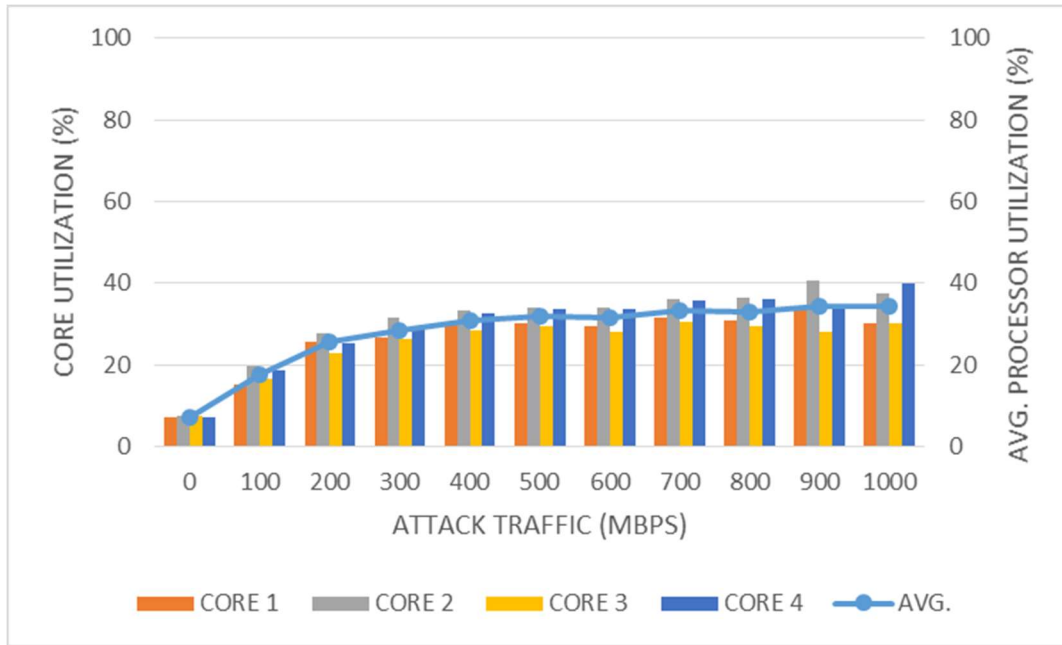


Figure 3.4 Processor Utilization of the Non-Virtualized Server under Ping Flood Attack

Thus, in the virtual machine, the unequal core utilization is contributed both by virtualization and the attack traffic. Even though attack traffic causes unequal core utilization, it is only marginal as shown in figure.3.4, but virtualization intensifies the effect. This uneven utilization of cores, caused due to virtualization, is the one of the reasons why the same DDoS attack has a much worse effect on a virtual machine than on a non-virtual machine both of which are running the same Operating System.

It is interesting that the same attack has different levels of impact on the performance of a web server when it is installed in a non-virtual server and a virtual machine. It was already mentioned that one of the reasons for the variation could be due to the difference in the core utilization of the processor in the two cases. The most significant parameter that affects the performance of a system is the memory allocation.

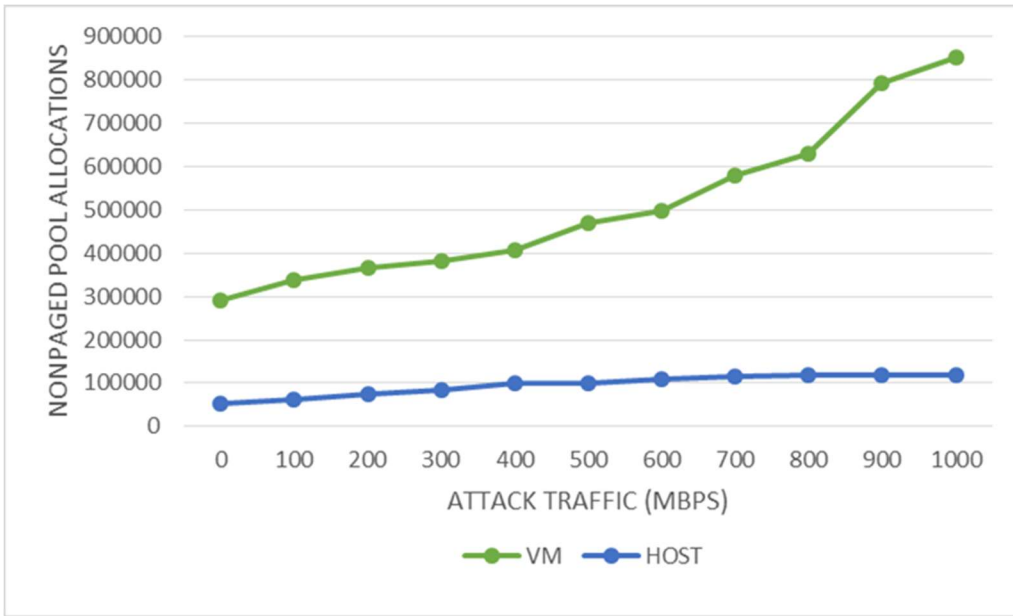


Figure 3.5 Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack

In this thesis, the memory allocation throughout the duration of each experiment has been monitored through the number of Pool Nonpaged allocations. There are two types of memory allocations in a system, paged and nonpaged, of which the nonpaged allocations take place in the Random Access Memory of the Central Processing Unit. Hence, the performance of the server is affected with increasing number of nonpaged memory allocations since the number of nonpaged allocations is directly proportional to the memory utilization.

The nonpaged pool allocations during the experiment are shown in figure 3.5. When 2500 connections are sent, in addition to 100000 allocations, the virtual machine needs 200000 more nonpaged pool allocations to establish the same number of connections with the clients. From the difference in the number of allocations in a virtual machine, it might be inferred that the difference is because of the overhead due to virtualization.

As soon as the ping attack traffic was introduced, there was a visible increase in the number of allocations in case of the virtual machine, this corresponds with the increase in the average

processor utilization from fifteen percent to fifty percent. At the same time, there was no significant increase in the memory utilization of the non-virtualized server which was reflected by the comparatively lower increase in processor utilization in the non-virtualized server shown in figure 3.2.

As the attack traffic was increased, there was very minimal increase in the number of allocations which correlates with the low increase in the processor usage of the non-virtualized server under attack. On the other hand, the number of allocations increases linearly with each increase in the attack traffic magnitude when the attack traffic is sent to the VM. This in turn causes an increase in the average processor utilization of the virtual machine. These two events would affect the performance of the virtual machine as a web server which can be evaluated by analyzing the connection establishment behavior of the web servers collected from the clients.

The figure 3.6 shows the trend of connection establishment throughout the duration of the attack before and after virtualization. To establish the baseline, initially 2500 HTTP connection requests were sent to the non-virtualized system and the virtual machine from the simulated clients. Both the systems were able to handle the attack traffic effectively without any decrease in the number of connections until 400 Mbps magnitude attack traffic was sent.

Once the attack traffic was increased to 400 Mbps, the number of connections established per second decreased to 2350. Following this, the number of legitimate client connections kept on decreasing with increasing attack traffic. The non-virtualized system was able to withstand the attack up to 500 Mbps after which the number of connections started decreasing. When attack traffic of magnitude 1 Gbps was sent, the virtual machine was able to establish 1100 connections and the non-virtualized system established 1600 connections per second with the clients.

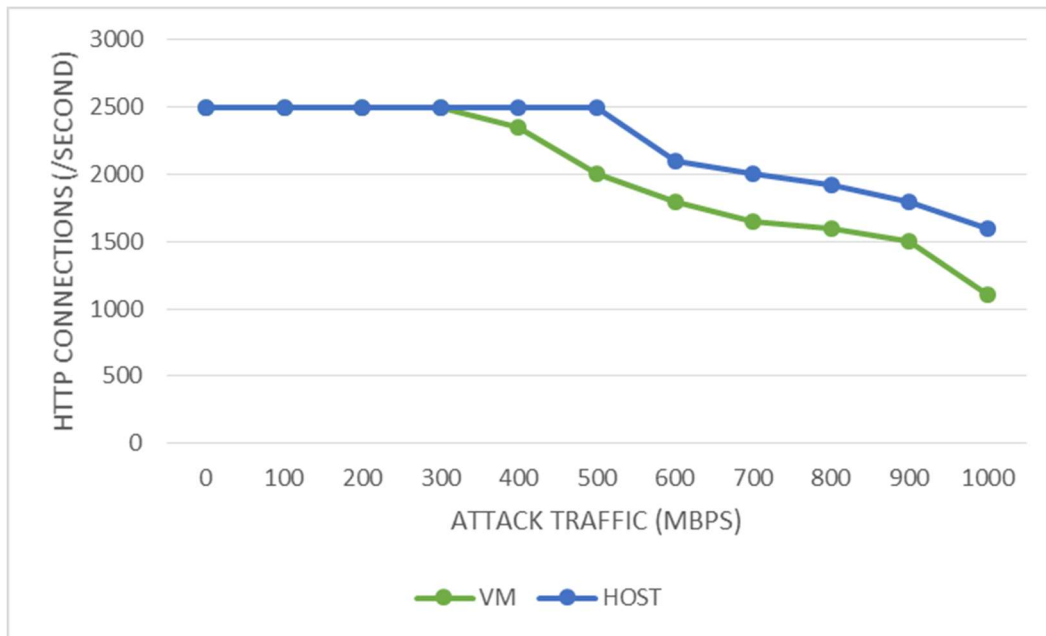


Figure 3.6 Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack

With the heavy competition in today’s E-commerce industry, connection latency plays a crucial role in the success of an organization. Nowadays, organizations face the risk of losing a customer to another company that has a faster website with lower or zero delay in responding to the client’s request. Although a delay of a few seconds may not seem to be significant, Amazon and Google have lost millions of dollars annually due to a delay of one second [16]. Hence, connection latency is an important factor in evaluating the performance of a web server.

The connection latency experienced by the simulated clients when the web servers were under attack is shown in figure 3.7. Initially, there was no time delay between the servers sending HTTP response to the client requests. The same trend continued until an attack traffic of magnitude 400 Mbps was sent to the virtual machine. When the virtual machine received 400 Mbps of Ping attack traffic, the time taken to establish a connection increased from 0 to 100 milliseconds. But, the non-virtualized web server continued to respond to client requests at the same magnitude of attack traffic. The connection latency of the non-virtualized web server



continued to be zero until 700 Mbps attack traffic was sent to the server while the connection latency of the VM kept increasing linearly with increasing magnitude of Ping attack traffic.

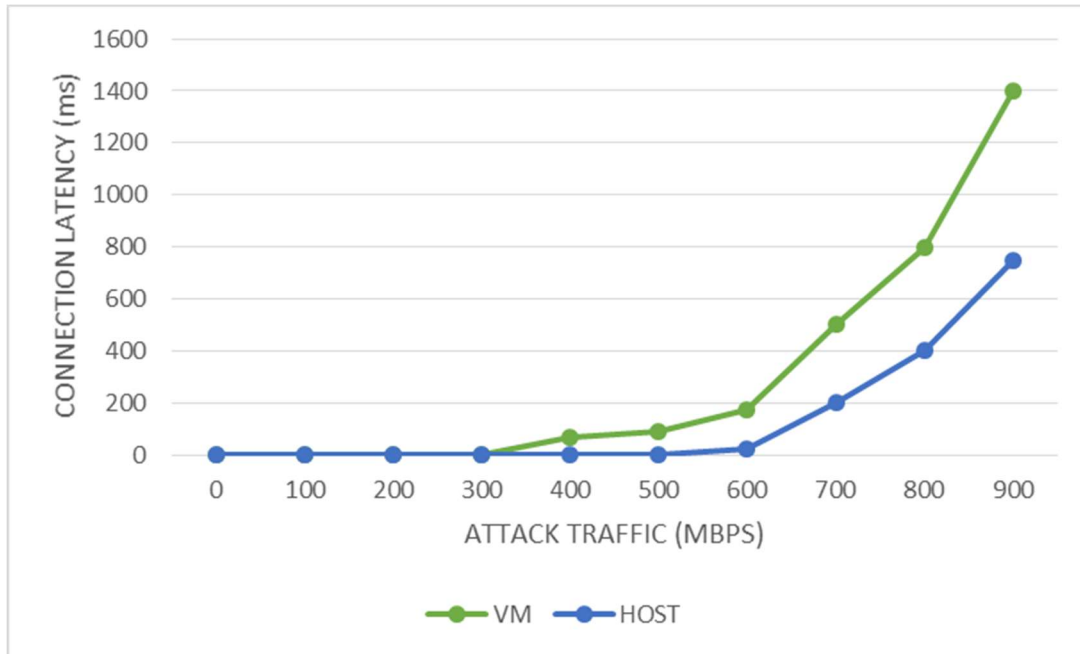


Figure 3.7 HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under Ping Flood Attack

When 900 Mbps attack traffic was sent, the virtualized server took 1400 milliseconds to respond to the TCP-SYN requests of clients, while the non-virtualized server took 800 milliseconds to respond to requests.

### 3.3.2 Smurf Attack

The Smurf attack is launched by sending a barrage of ICMP echo reply messages to a targeted victim server. Although the receiver of a reply message is not required to act on the received message unlike in the case of an echo request message or Ping, surprisingly the Smurf attack generally has a much higher impact on a victim than the Ping Flood Attack. Since the Smurf attack had the highest impact on the server of the four attacks, Ping Flood, Smurf,

TCP-SYN and UDP Flood attacks when it was non-virtualized, it was expected that the performance of the virtual machine might be affected heavily under the influence of the Smurf Attack.

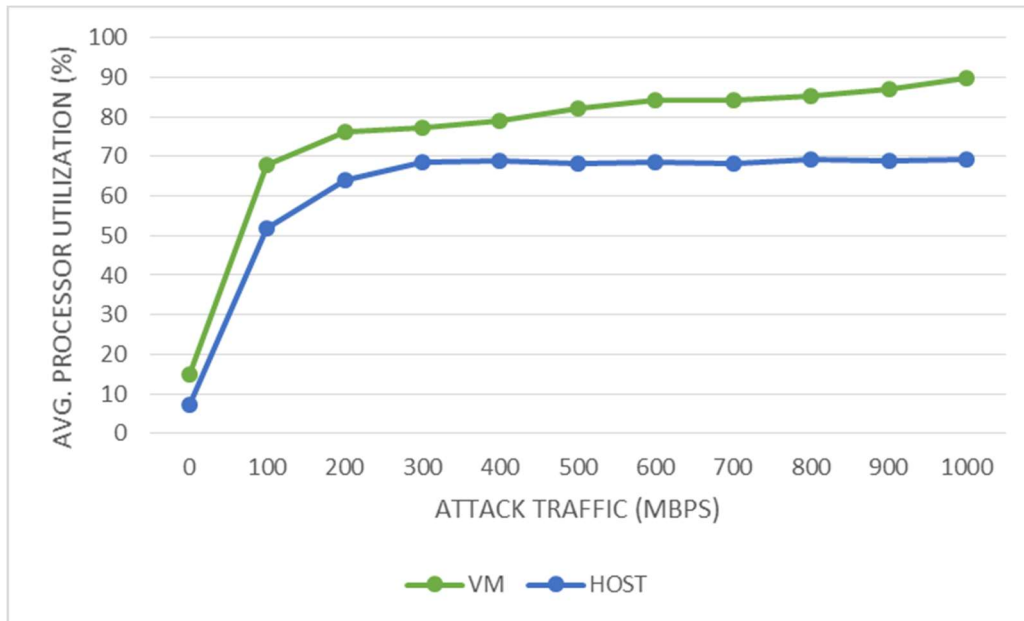


Figure 3.8 Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under Smurf Attack

The average processor utilization of the web server in a non-virtualized server operating system and in a virtual machine is shown in figure 3.8. Unlike in the case of the Ping flood attack in which only the processor usage of the VM increased as soon as the attack traffic was introduced, the processor utilization of both types of servers increased by approximately forty percent from their respective baseline utilization. With further increase in the attack traffic magnitude, the average processor utilization of the non-virtualized server and the virtual machine increase marginally with increase in the attack traffic magnitude.

When the attack traffic was increased further after 300 Mbps, the average processor utilization of the virtual machine kept increasing but that of the non-virtualized server remained the same through the remaining duration of the experiment. At the end of the experiment when

1000 Mbps magnitude smurf attack traffic was sent, the processor usage of the VM had reached 80 percent while the non-virtualized server had used 70 percent of the processing power.

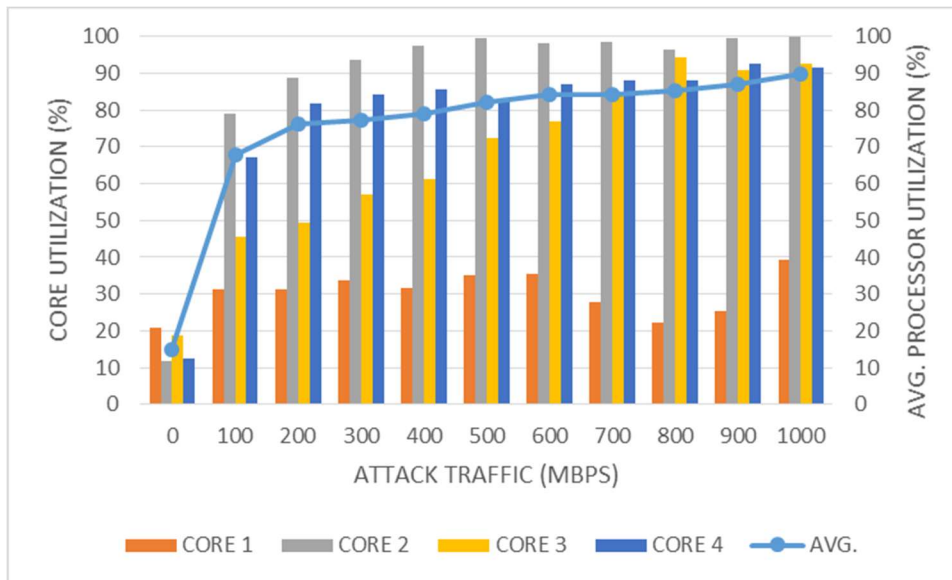


Figure 3.9 Processor Utilization of the Virtual Machine under Smurf Attack

As observed in Ping flood attack, the core utilization of the virtual machine was uneven under Smurf attack as well. The figure 3.9 shows the processor utilization of the virtual machine. During the baseline, the core 1 was the most utilized and the core 2 was the least used of the four cores. As a result, once the attack traffic was introduced, the core 2 became the highest used core and the core 1 was least used throughout the remaining period of the experiment. In the same way, the core 3 had the second highest utilization after core 1 while the core 4 was used less compared to the core 3. Hence, after receiving the attack traffic, the core 4 became the second most used while the core 3 was less utilized than core 4. This trend of core utilization continued until the server received 1000 Mbps magnitude of smurf attack traffic.



Figure 3.10 Processor Utilization of the Non-Virtualized Server under Smurf Attack

The core utilization of the non-virtualized server is similar to that observed in Ping attack since under both the attacks all the four cores have been used equally throughout the duration of the experiment. The core utilization of the server before virtualization under the influence of the smurf attack is shown in figure 3.10.

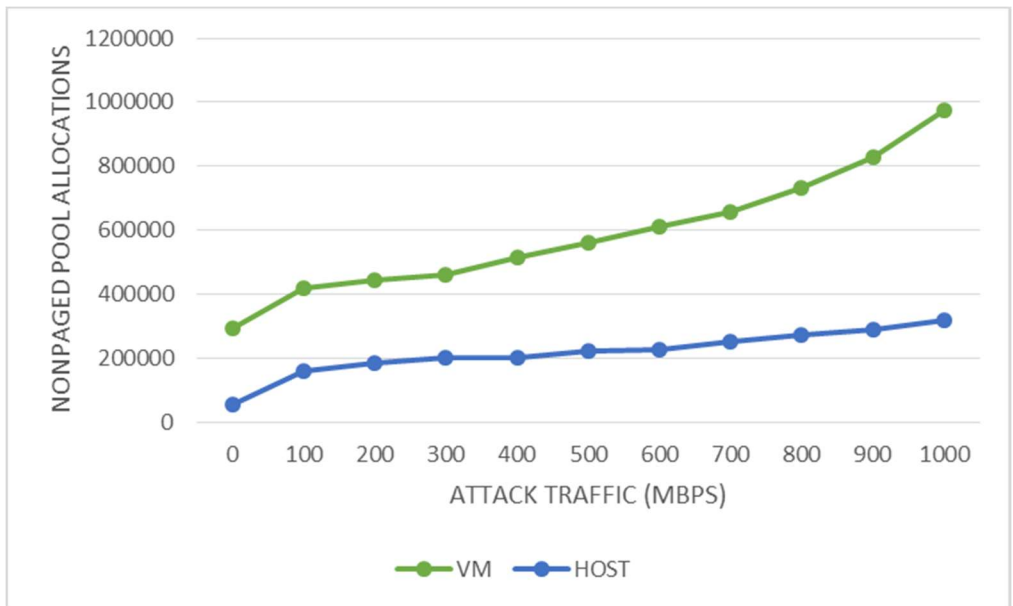


Figure 3.11 Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under Smurf attack

As mentioned earlier, the memory utilization of the server plays a crucial role in the performance of the server. Figure 3.11 shows the memory usage of the virtual machine and the non-virtualized server under smurf attack. There was a spike in the number of nonpaged pool allocations in both the non-virtualized server and the virtual machine as soon as the attack traffic was introduced unlike in the case of ping attack where there was no sudden increase in the number of allocations observed in the non-virtualized server.

The memory utilization increased with increase in magnitude of the smurf attack. When 1000 Mbps attack traffic was sent to the virtual machine and the non-virtualized server, the number of nonpaged pool allocations were 1000000 and 325000 respectively.

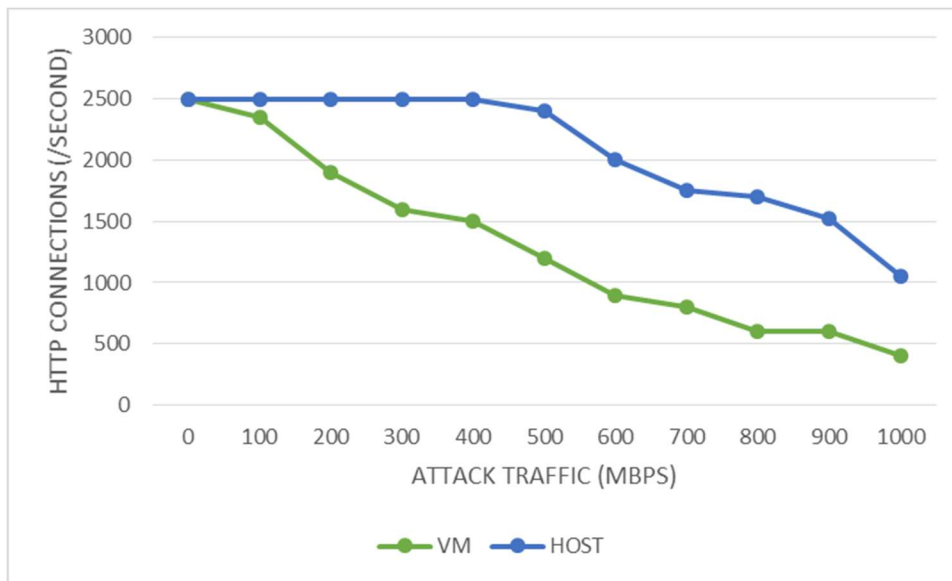


Figure 3.12 Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under Smurf Attack

The performance of a web server can be detected by the number of connections that a server can establish and the presence or lack of delay in establishing those connections. The figures 3.12 and 3.13 show how the non-virtualized server the virtual machine perform on these grounds. As soon as the attack traffic was introduced, the virtual machine could no longer maintain its baseline behavior and dropped its connection rate from 2500 to 2400 connections

per second. Following this, the number of connections established by the virtual machine kept decreasing with increase in the attack traffic magnitude. Although the processor utilization of the non-virtualized server also increased by nearly 40 percent from its baseline, it was able to continue to establish 2500 connections per second with the clients while receiving 100 Mbps of smurf attack traffic.

There was a decrease in the number of HTTP connection rate of the non-virtualized server when it was targeted with smurf attack traffic of magnitude 500 Mbps. From that point in the experiment, the number of connections kept on decreasing with increase in the magnitude of attack traffic. When 1 Gbps smurf attack traffic was sent to the virtual machine, it could only establish 450 connections per second with the clients while the non-virtualized server was able to maintain more than twice the number of connections, 1000 connections per second, when it was receiving the same magnitude of attack traffic.

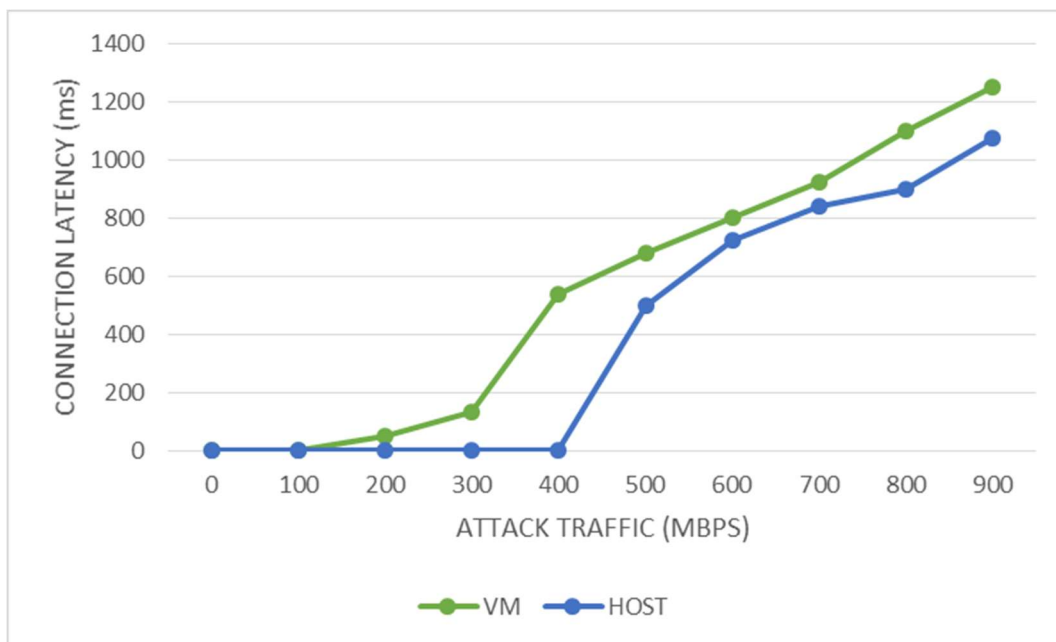


Figure 3.13 HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under Smurf Attack

There was nil delay in the time taken by the server and the virtual machine to respond to the TCP-SYN requests of simulated clients in the absence of attack traffic, shown in figure 3.13. When 200 Mbps of smurf attack traffic was sent to the virtual machine, the VM took 50 milliseconds longer to send SYN-ACK to the SYN requests that it received. The non-virtualized server was able to manage the attack without letting it affect the performance of the server until 500 Mbps of attack traffic was sent to it. Upon receiving an attack traffic of magnitude 500 Mbps, the connection latency of the non-virtualized server increased to 500 milliseconds from zero or nil delay. With increasing magnitude of attack traffic, the servers took longer to respond to the legitimate clients with acknowledgments. While receiving 900 Mbps of attack traffic, the VM and the non-virtualized server had connection latencies of 1200 and 1050 milliseconds respectively in sending acknowledgements to clients.

### **3.3.3 TCP/SYN Flood Attack**

The average processor utilization of the virtual machine and the server before virtualization is shown in figure 3.14. The processor usage increased to 50 percent and 20 percent respectively after the TCP-SYN attack traffic of magnitude 100 Mbps was sent to the VM and the non-virtualized server. Then with each 100 Mbps increment in the magnitude of attack traffic sent, the processor utilization increased in both the server and the VM but only by 10 percent in the virtual machine and 5 percent in the non-virtualized server.

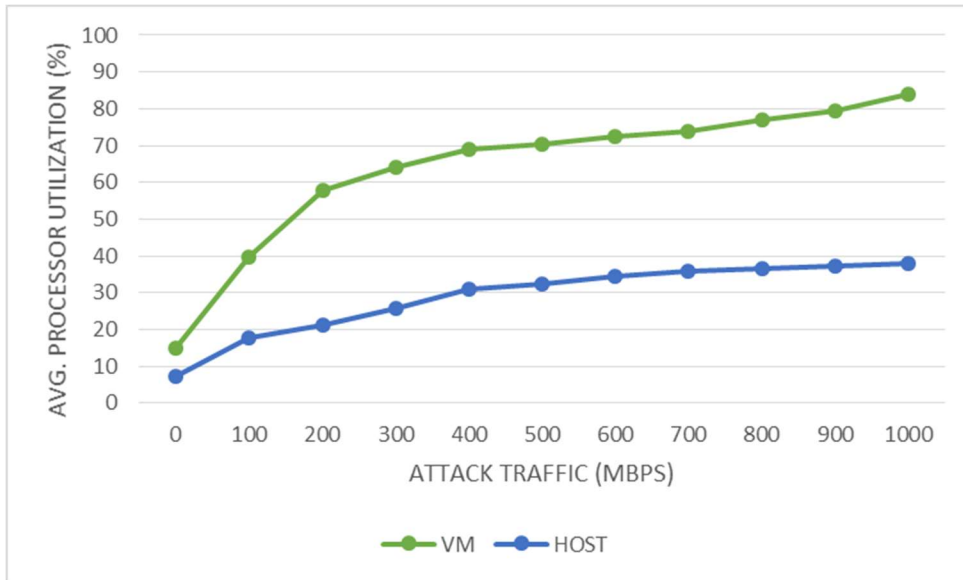


Figure 3.14 Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack

After 200 Mbps, the increase in processor usage of the virtual machine also dropped to 5 percent throughout the remaining period of the experiment. When 1000 Mbps of TCP-SYN flood was sent to the non-virtualized server and the virtual machine, the average processor utilization was 40 percent and 85 percent respectively.

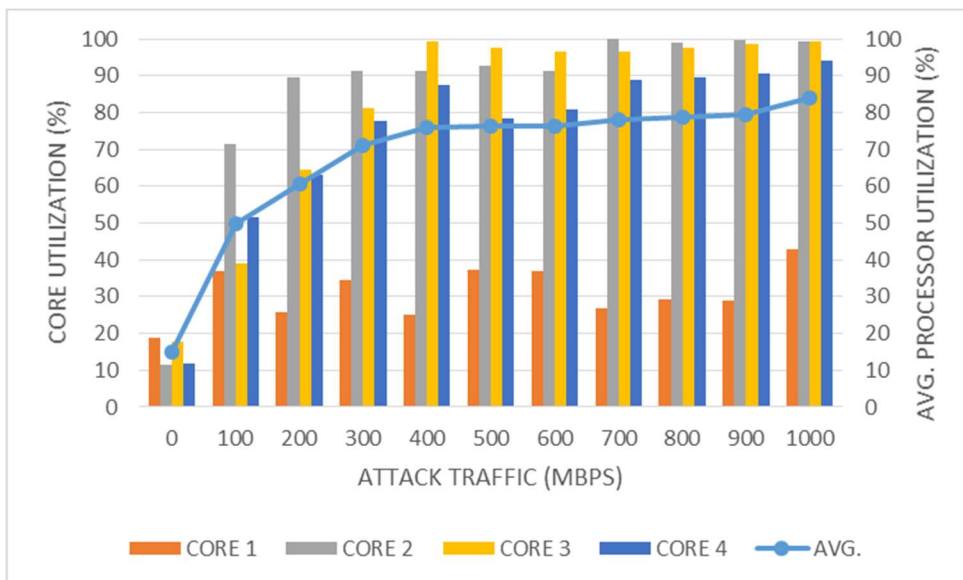


Figure 3.15 Processor Utilization of the Virtual Machine under TCP-SYN Flood Attack



The figure 3.15 shows the core utilization of the VM. The four cores are not equally utilized both before and after the attack traffic was introduced. Once again, the core 1 was the most used before the attack began and the least used after the attack traffic was received by the virtual machine.



Figure 3.16 Processor Utilization of the Non-Virtualized Server under TCP-SYN Flood Attack

Before the server was virtualized, all the four cores of the server were equally utilized when it was receiving only legitimate connections from clients as shown in figure 3.16. Even after the attack traffic was introduced, the core utilizations for any given attack traffic magnitude differed only by two or three percent which is much lower compared to the huge differences in the core utilization observed in virtual machine when it was under the influence of TCP-SYN flood attack traffic.

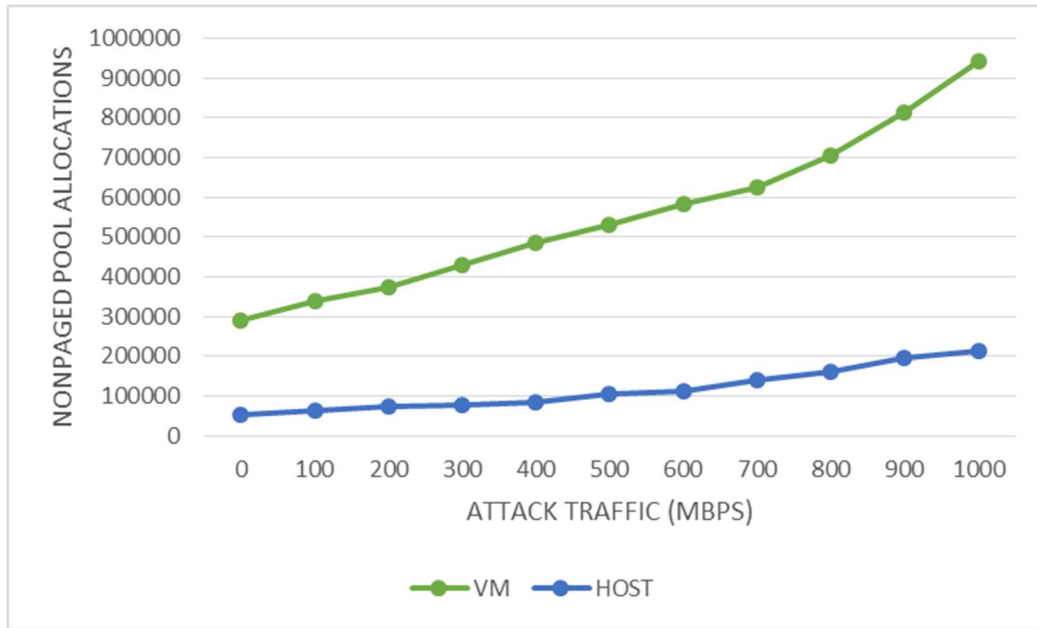


Figure 3.17 Number of Nonpaged allocations in the Virtual Machine and Non-Virtualized Server under TCP-SYN Flood Attack

The number of nonpaged pool allocations in the virtual machine and the server kept on increasing with increase in the attack traffic magnitude. The memory usage of the non-virtualized server was approximately the same through the duration in which the server received 300 Mbps to 600 Mbps of attack traffic. In the end of the experiment, the number of nonpaged pool allocations in the virtual machine was 9500000 and in the non-virtualized server was 200000 as shown in figure 3.17, which shows that virtualization causes a higher increase in the memory utilization of the virtual machine compared to the non-virtualized server for the same magnitude of attack traffic.

After the servers established a baseline of 2500 legitimate connections per second, only the non-virtualized server was able to continue to maintain the same number of connections after the attack traffic was introduced to the serve shown in figure 3.18. The number of legitimate connections immediately dropped to 2200 connections per second after the attack traffic was sent to the virtual machine

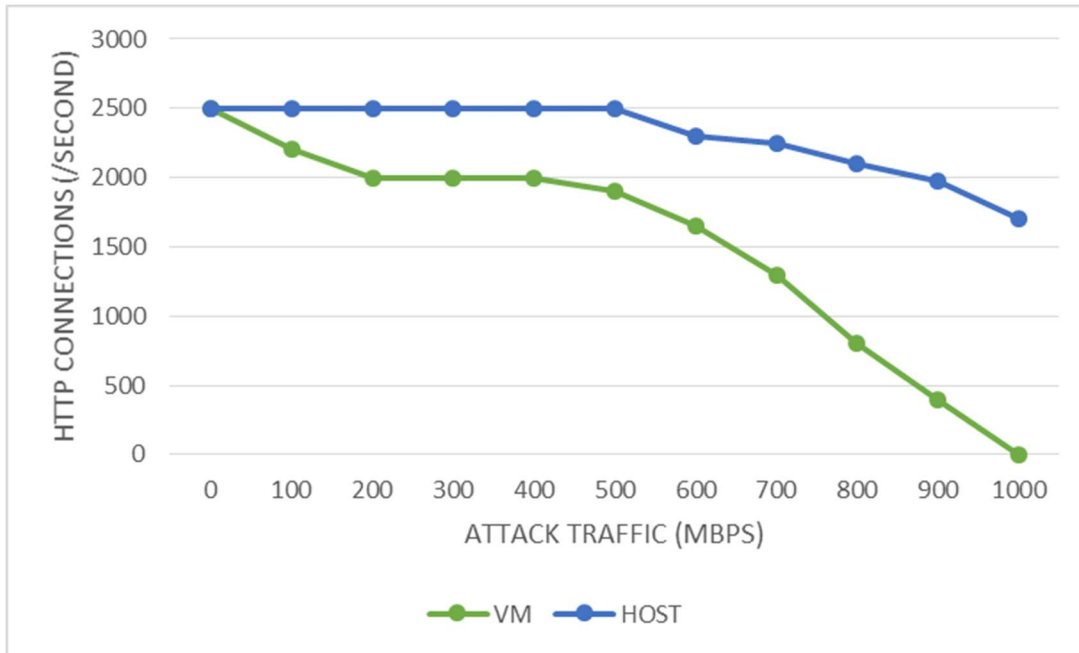


Figure 3.18 Number of HTTP Connections established by the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack

As opposed to this, the non-virtualized server was able to maintain the same number of legitimate connections as that established during the baseline until an attack traffic of magnitude 500 Mbps was sent

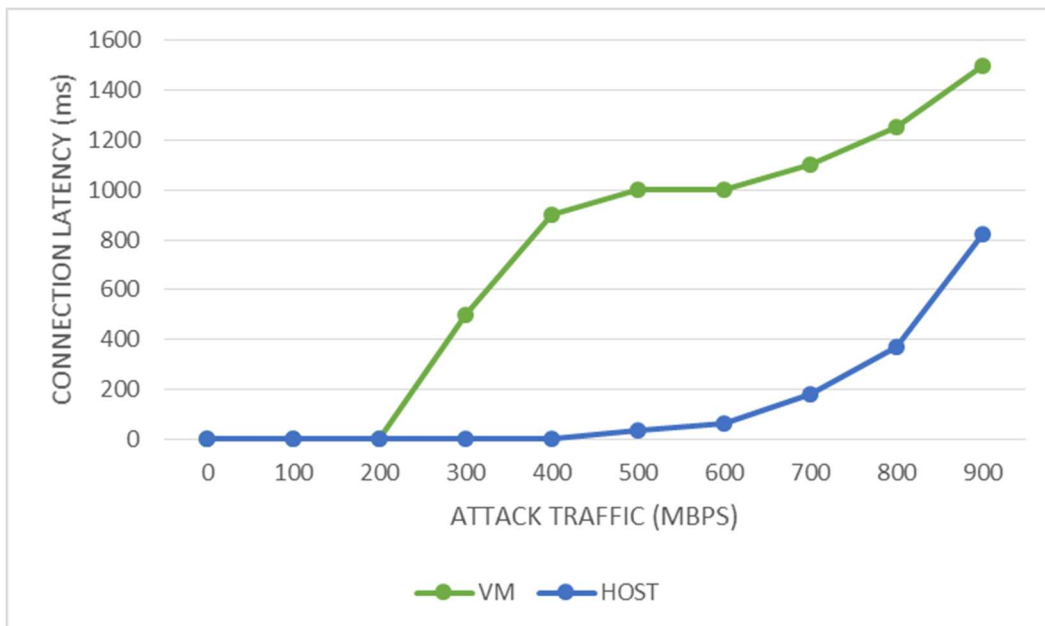


Figure 3.19 HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under TCP-SYN Flood Attack

The delay in replying to the legitimate TCP-SYN requests of the simulated clients, also known as connection latency, is shown in figure 3.19. There was no delay in the connection establishment in the beginning when the TCP-SYN attack traffic was sent to the non-virtualized server or the virtual machine. Once the attack traffic was increased to 300 Mbps, there was a 500-millisecond-delay caused by the virtual machine in sending the acknowledgement (SYN-ACK) to the SYN requests sent by the clients. After that, with each 100 Mbps increase in the attack traffic magnitude, the connection latency kept increasing. When 900 Mbps attack traffic was sent to the servers, the connection latencies observed in the virtual machine and the non-virtualized server were 1500 and 800 milliseconds respectively.

#### **3.3.4 UDP Flood Attack**

The User Datagram Protocol became popular as an alternative to the TCP since UDP gave the sender greater control over the time at which the data could be sent to the receiver and there was no restriction on the amount of data that it could send to the receiver. This is because, UDP does not employ congestion control (which controls the amount of data sent) and does not have to establish a connection before sending data to the receiver (which restricts the sender from sending the data immediately since the sender has to wait for the completion of the three-way handshake). Although the lack of reliable data delivery was a drawback in UDP, it was compensated by incorporating reliability in the application layer. In much the same way that UDP gives more control to the sender, which is usually the clients, an attacker can also exploit UDP to attack a targeted victim.

When the UDP flood attack was sent to both types of servers, there was an immediate increase in the processor utilization of the virtual machine and a relatively lower increase in the processor usage of the non-virtualized server. Another noticeable increase in the processor

utilization of the virtual machine was observed when the attack traffic was increased from 300 Mbps to 400 Mbps when it changed from 50 percent to 60 percent as shown in figure 3.20.

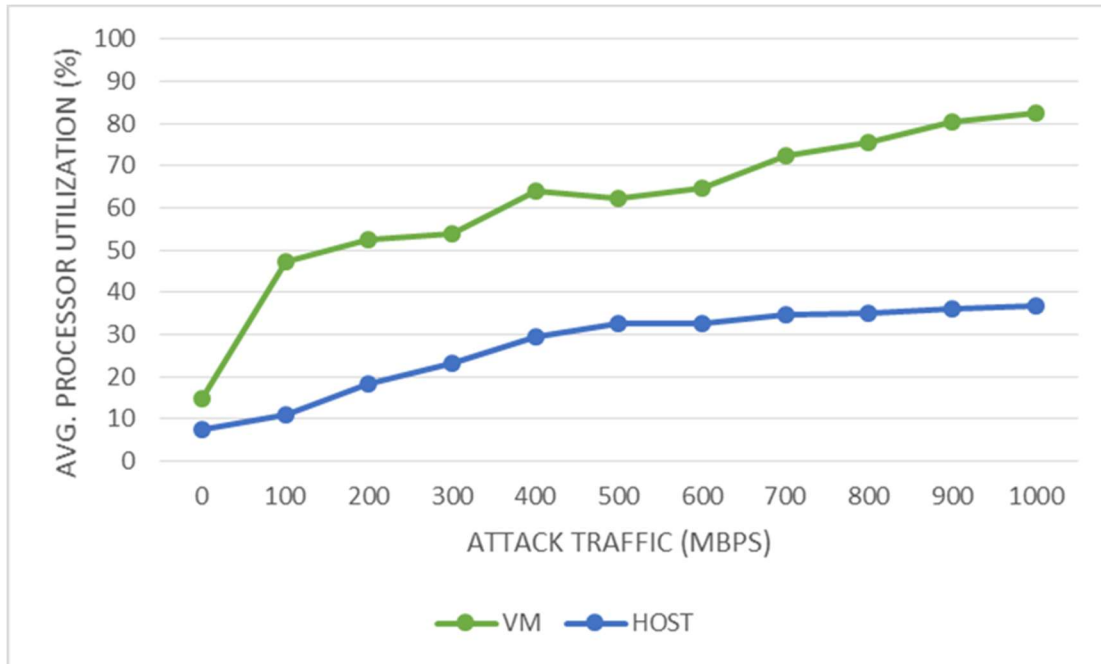


Figure 3.20 Average Processor Utilization of the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack

Through the remainder of the experiment, the increase in the average processor utilization was linearly proportional to the increase in the magnitude of attack traffic. When 1000 Mbps magnitude UDP flood attack was sent, the processor utilization of the VM and the non-virtualized server were 82 percent and 37 percent respectively.

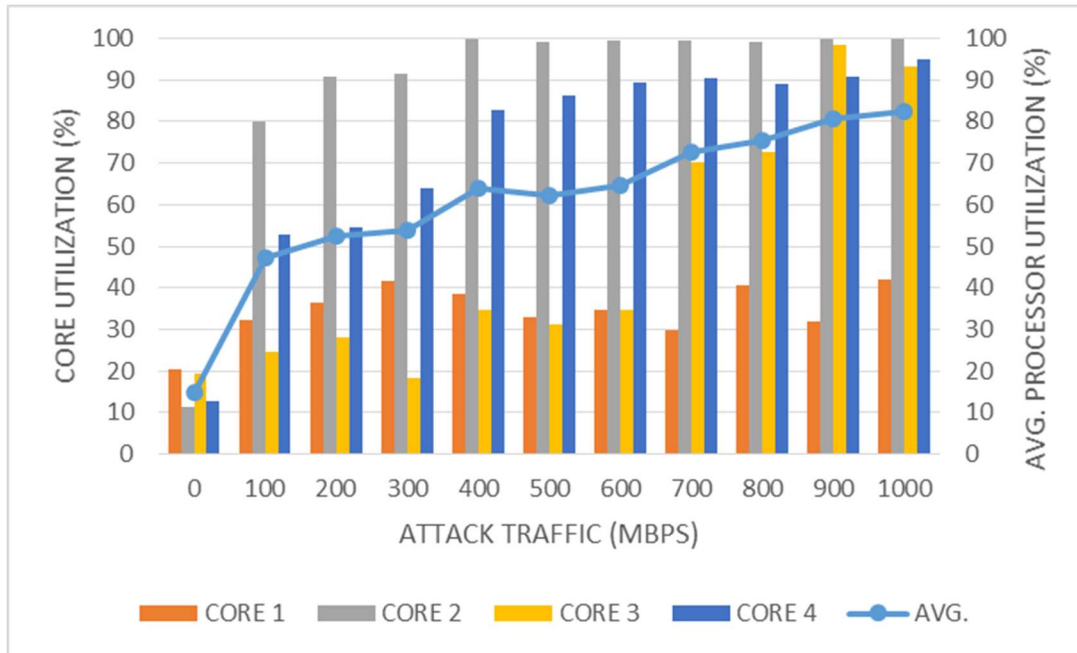


Figure 3.21 Processor Utilization of the Virtual Machine under UDP Flood Attack

The four cores of the virtual machine are not utilized equally as shown in figure 3.21. Before the UDP flood attack traffic was introduced, the cores 1 and 3 were the most used. After the virtual machine received the attack traffic, the least used of the four cores, cores 2 and 4 were utilized more than the other two cores. This trend continued throughout the experiment except during the measurement of the baseline. It can also be inferred that between the transition from the baseline traffic to the attack traffic, a mechanism had been used to choose the core that was least used during the baseline and use it the most and do the vice versa for the core that was most used during the baseline measurement.

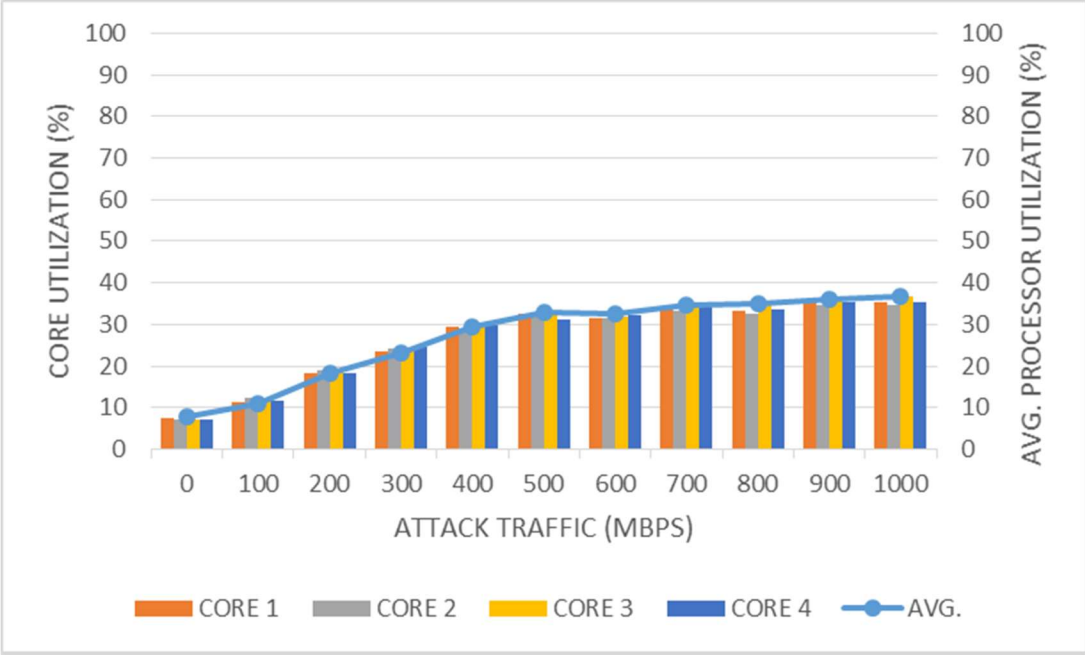


Figure 3.22 Processor Utilization of the Non-Virtualized Server under UDP Flood Attack

The figure 3.22 shows the representation of the processor utilization in the form of core utilization. All the four cores were equally utilized both during the baseline measurement and after the attack traffic was sent to the non-virtualized server.

It has been observed from these experiments that the memory allocation plays a crucial role in the efficiency of the server since it directly impacts the processor utilization. As with the earlier results, there was difference of nearly 200000 allocations between the virtual machine and the non-virtualized server. The figure 3.23 shows the memory utilization of the VM and the non-virtualized server under the influence of the UDP flood attack. Once the attack traffic was introduced, the memory utilization of the VM increased by 100000 allocations while there was a marginal increase in the case of the non-virtualized server. Then the memory usage increased linearly with increase in attack traffic in both types of servers.

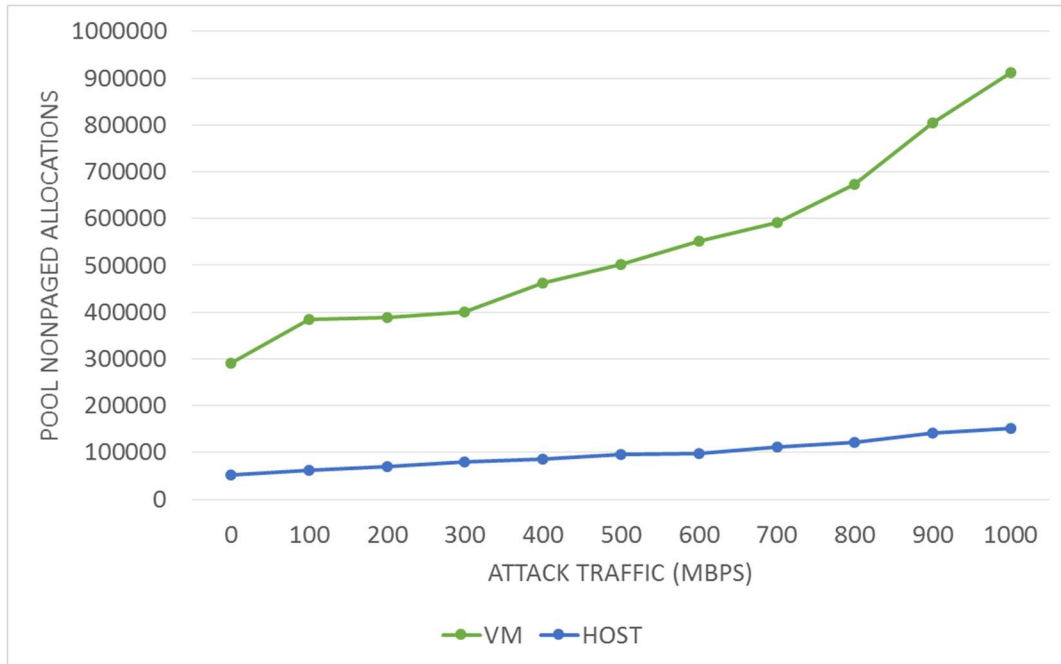


Figure 3.23 Number of Nonpaged Pool Allocations in the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack

When the attack traffic magnitude was increased from 300 Mbps to 400 Mbps, there was quite a high increase in the number of nonpaged allocations which lead to a nearly twenty percent increase in the processor utilization which was discussed earlier in figure.3.a. The number of nonpaged pool allocations kept increasing at a high rate finally reaching 90000 nonpaged pool allocations when it received the maximum attack traffic of magnitude 1000 Mbps. At the end of the experiment, the number of allocations had gone from 300000 to 900000 in the virtual machine, there was a comparatively lower increase in memory usage when the server was not virtualized where the nonpaged allocations ranged from 50000 to 150000.



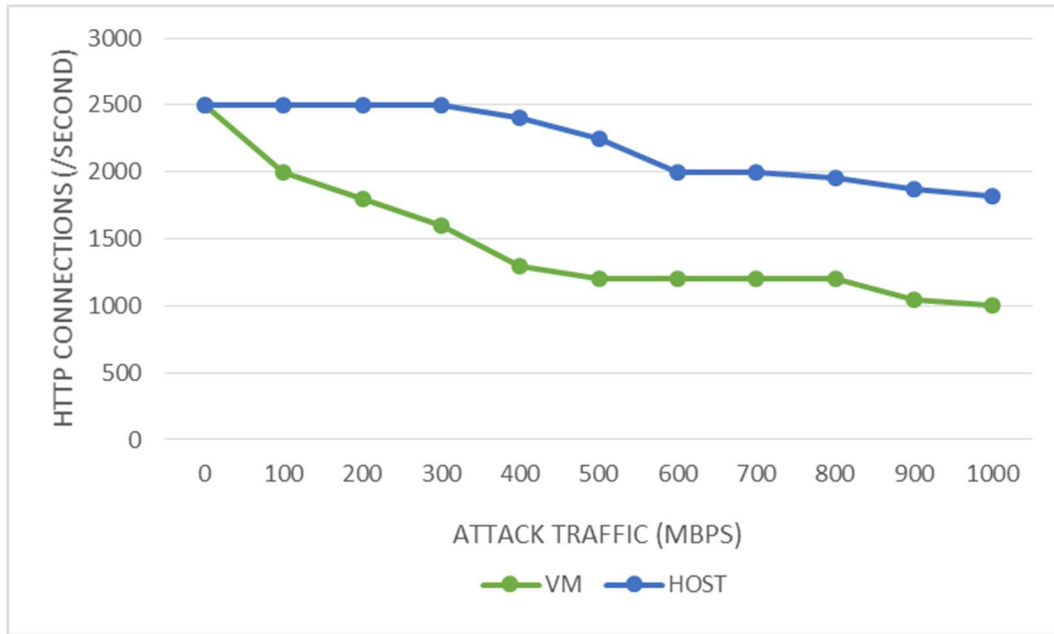


Figure 3.24 Number of HTTP Connections Established by the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack

The web servers were initially able to establish 2500 connections per second in the absence of attack traffic as shown in figure 3.24. After an attack traffic of 100 Mbps was sent, the non-virtualized server continued to maintain the same number of connections with the clients but the number of connections established by the virtual machine with the clients dropped to 2000 connections per second. As it was previously observed with the other attacks, the performance of both the virtualized and the non-virtualized server worsened with increasing attack traffic, but the attack had a greater impact on the virtual machine.

The non-virtualized server could not continue to establish 2500 connections after 400 Mbps UDP attack traffic was sent to it. The rate of connection establishment dropped to 2400 connections per second after which the number of connections kept dropping further with increasing attack traffic. When 1000 Mbps of UDP attack traffic was sent, the non-virtualized server was able to establish 1800 connections per second while the virtual machine was able to establish 1000 connections per second with the clients.

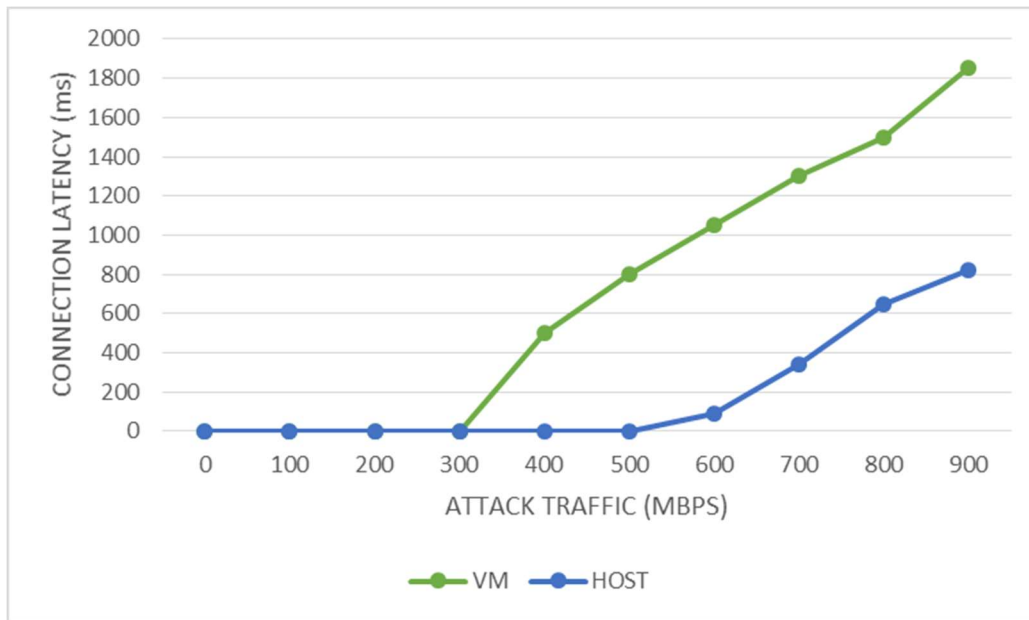


Figure 3.25 HTTP Connection Latency of the Virtual Machine and the Non-Virtualized Server under UDP Flood Attack

The figure 3.25 shows the connection latency of the comparison of the connection latency of the web servers installed in a virtual machine and a non-virtualized server. There was no delay in the connection establishment in the first three increments in the attack traffic magnitude, when 400 Mbps UDP attack traffic was sent, the virtual machine took 500 milliseconds to respond to the SYN requests sent by the clients. With increasing attack traffic magnitude, more and more delay was experienced by the clients. In the case of the web server before virtualization, there was no connection latency until the server started receiving 600 Mbps attack traffic but the delay was 50 milliseconds, but it was still lower than a delay of 1000 milliseconds caused by the VM when it received the same magnitude of attack traffic. When 900 Mbps attack traffic was sent, the connection latency in the VM was 1800 milliseconds and that in the non-virtualized server was 800 milliseconds, less than half the connection latency due to the virtual machine.

### 3.4 Chapter Summary

The Chapter III analyzed the effect of popular DDoS attacks, first on a non-virtualized server operating system and then on a Virtual machine that had the same server operating system. It was observed that there was a difference between a VM and a non-virtualized server in terms of the processor utilization and the memory utilization irrespective of the presence or absence of attack traffic. When the processor and memory usage is higher for the same amount of legitimate traffic, it can be inferred that there is an overhead due to virtualization.

When a VM receives only legitimate traffic, the processor utilization is higher than in a non-virtualized server hence, when a VM is under attack, the increase in processor and memory utilization is accelerated. From the analysis it is evident that virtualization makes a server more vulnerable to Distributed Denial of Service attacks. Hence it is extremely significant that networking professionals exercise caution with provisioning hardware resources to each virtual machine and with the number of virtual machines installed in the hardware since virtual machines are much more susceptible to succumb to DDoS attack than a non-virtualized server installed on the same hardware. The chapter IV evaluates the comparison of various Windows Server Operating Systems over time Windows Server 2008 R2, 2012 R2 and 2016.

## CHAPTER IV

### EVALUATION OF THE IMPACT OF DDoS ATTACKS IN A MULTI-VM ENVIRONMENT

Virtualization has enabled organizations to maximize their profit and dramatically cut down on their CapEx and maintenance costs because of virtualization. It is very difficult to exactly predict the server and client systems required for an organization, as a result, companies end up spending money to buy more hardware than needed and expansion of an organization was also difficult. After the advent of virtualization, all these issues were tackled. Depending on the requirement and the role of each server or client, an organization decides to install several number of virtual machines in a hardware and allocates hardware resources accordingly.

In this chapter, three different Windows Server Operating Systems released from 2008 till 2016 have been compared for their performance under the effect of DDoS attacks. Unlike in the previous chapter where the Windows Server 2012 R2- virtual machine was allocated all the four cores in the hardware, in this chapter, all the three virtual machines have been allocated one core each.

This difference in the number of cores also allows for a comparison between the changes in performance of the same virtual machine, in this case, Windows Server 2012 R2, when it is allocated different number of cores. The comparison of the impact of DDoS attacks on the

processor utilization of the same virtual machine when it was allocated four cores and when it was allocated one core has also been analyzed in this chapter.

#### **4.1 Experimental Setup**

The Windows Server 2012 R2 Standard operating system is installed in a Dell PowerEdge T320 [37] hardware with Intel Xeon E5-2407 v2 quad core 2.4 GHz processor [38] and 8 GB RAM. The built-in firewall of the server was enabled with the default settings throughout all the experiments. The experimental set up is shown in Figure 4.1. The attack traffic was simulated in a controlled environment at the Network Research Lab at the University of Texas Rio Grande Valley (UTRGV).

The Windows Server 2012 R2 Operating System was initially tested against four most popular DDoS attacks, Ping Flood, Smurf, TCP/SYN and UDP Flood attacks, in Chapter II. Now, the Windows server OS is virtualized in order to test the effect of the same four attacks on a virtual machine. In order to install virtual machines, the Hyper-V manager is installed in the server OS. The Hyper-V manager is installed in the Windows OS through the Server manager console [31].

Once Hyper-V is installed, from the server manager console, under the Tools tab choose Hyper-V Manager to open the Hyper-V manager. In the Hyper-V manager, under the actions tab, click on New → Virtual Machine, this will open a New Virtual Machine Wizard. Click on Next in the Before you Begin Window. The name and location for the Virtual Machine can be entered and then click on Next to choose the Generation of the Virtual Machine depending on the type of OS. If the Virtual Machine or the Guest Operating System is a 64-bit version of Windows 8 or Windows Server 2012 or later, then choose Generation 2, else choose Generation 1. It is

important to note that the generation of a virtual machine cannot be changed after the virtual machine has been created. The generation of the virtual machine is to be chosen correctly to ensure that support is provided for features such as SCSI boot, Secure Boot and PXE boot using a standard network adapter. The Operating System of the Virtual Machine is Windows Server 2012 R2, hence generation 2 was selected.

Next, the startup memory for the virtual machine is assigned [32]. The requirement of the startup memory is decided based on the role of the virtual machine and the Operating System that the Virtual Machine will run. For the Windows Server 2012 R2 guest OS that was being installed, 512 MB was assigned as the startup memory. Click on Next and choose a Virtual switch for the virtual machine. Virtual Switches are broadly classified into three types: External, Internal and Private [33]. For the purpose of the thesis, an external switch is designated to the VM. Following this step, the location from which the image of the virtual machine is to be installed is specified and then the installation options are selected. Finally, the virtual machine with the Windows Server 2012 R2 operating system is created. A similar procedure was followed to create virtual machines with Windows Server 2008 R2 and Windows Server 2016 Technical Preview 4.0.

The Windows Server 2016 operating system has not yet been released and the latest Technical preview of the operating system was released on November 19, 2015 [35]. At this point of time, since only the preview of the server is available, the server operating system could not be used as a web server as was done with the Windows Server 2012 R2 in the previous chapters.

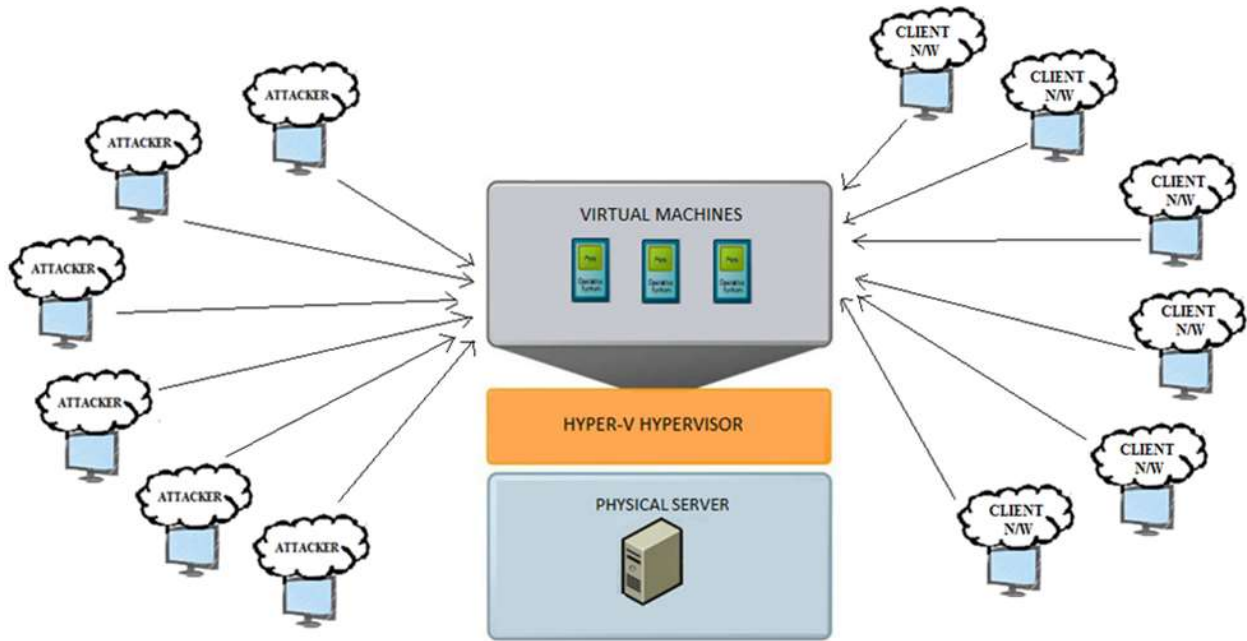


Figure 4.1 Experimental Setup

In order to ensure that all the three virtual machines run the same applications to perform the comparison between the three operating systems (2008 R2, 2012 R2 and 2016), none of the virtual machines were used as a web server in this chapter. The four types of DDoS attacks that were launched on the virtual machines were, Ping Flood attack, Smurf attack, TCP-SYN Flood attack and UDP Flood attack. The figure 4.1 shows the experimental setup which shows three virtual machines that were installed on the same hardware.

In chapters 2 and 3, the virtualized or the non-virtualized server was set up as a web server and as a result, HTTP request traffic was sent along with the attack traffic. In this chapter however, since the 2016 server could not be set up as a web server, the three virtual machines receive only the attack traffic throughout the experiment. Initially, the performance counters were run for five minutes in the absence of any traffic in order to obtain the baseline. After the

baseline behavior of the virtual machines were recorded, attack traffic was increased by 25 Mbps for every five minutes until the processor utilization of the virtual machine reached hundred percent. The parameters that were recorded to keep track of the performance of the virtual machines under the DDoS attacks are described in the following section.

#### **4.2 Parameters of Performance Evaluation**

The parameters that were monitored during the experiment is the Average Processor Utilization of the virtual machines. Both the parameters were measured throughout the duration of the experiment starting from the baseline behavior. These parameters were recorded by using the Data Collector Sets available in the performance monitor of the virtual machine operating systems.

In the Chapter III, all the parameters were measured from the performance counters present in the host operating system. The counter `\Hyper-V Hypervisor Logical Processor(_Total)\% Total Run Time` would be more useful to determine the total processing power utilized by all the virtual machines in the host. In this chapter, since the goal is to compare the performance of individual virtual machines installed on the hardware, performance monitors were run in the guest operating systems or virtual machines and not in the host operating system.

The Processor Utilization of a computer is analogous to the heartbeat and has a strong influence on the performance of the server. The name of the counter that is used to monitor processor utilization is known as `\Processor(_Total)\% Processor Time` and is defined as “The percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then



subtracting that value from 100%. (Each processor has an idle thread that consumes cycles when no other threads are ready to run)” [10].

The Total Processor Utilization is the average of core utilization of all the cores in a server, in this case the server has four cores. The Central Processing Unit (CPU) has to be functional at all times for the server to be able to deliver its most efficient performance. Monitoring the processor utilization enables a person to accurately observe the effect that an attack has on the server. Whenever the CPU utilization exceeds its optimal value, it will start impacting the efficiency of the server.

## **4.3 Results and Discussion**

### **4.3.1 Ping Flood Attack**

Ping Flood attack is launched by sending ICMP echo requests to a targeted victim server from several compromised botnets by an attacker. The geographically distributed botnets with a wide range of source IP addresses makes it very difficult to block the incoming flood of ICMP requests.

The Ping Flood attack was launched on each virtual machine individually and the processor utilization has been compared through the graph shown in figure 4.2. Initially, when 25 Mbps attack traffic was introduced, the processor utilization of the three virtual machines increased to nearly 25 percent. Following that there was an increase of 10 percent in the processor utilization in all the three virtual machines for each increment in the attack traffic until 175 Mbps attack traffic was sent. When 175 Mbps ping attack traffic was sent, the virtual machines running 2008 R2 and 2012 R2 reached 100 percent utilization but the processor usage of 2016 was 96 percent.

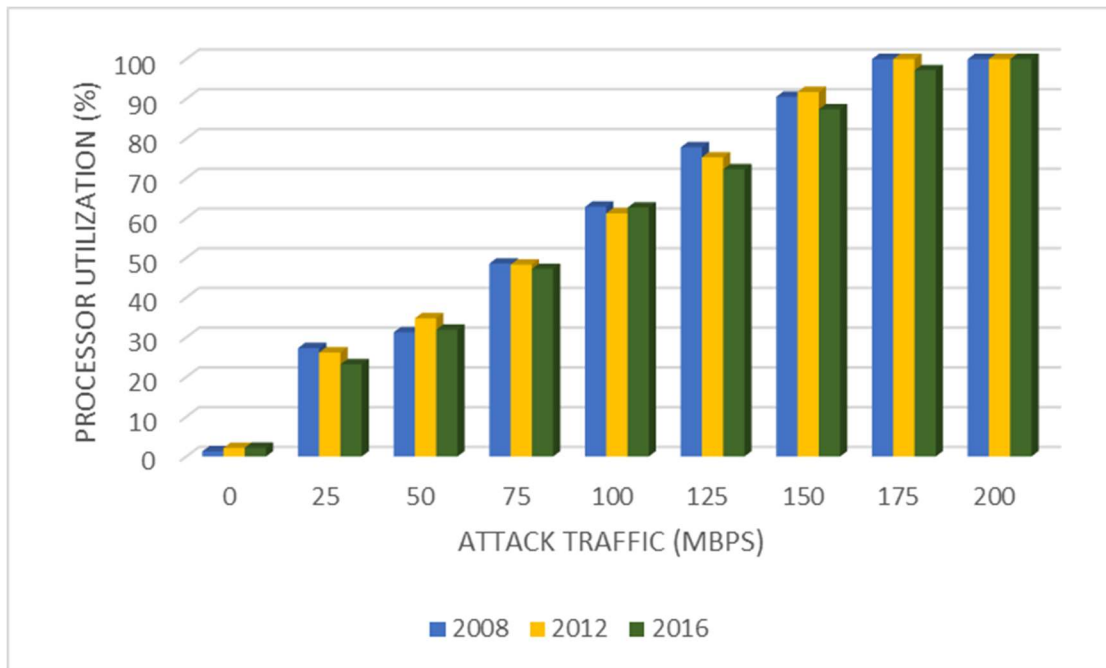


Figure 4.2 Comparison of Processor Utilization of the Virtual Machines under Ping Flood Attack

The processing power was completely consumed in 2016 also when the ping flood attack traffic was sent at a rate of 200 Mbps. Hence, the latest Operating system from Microsoft, Windows Server 2016, is able to withstand the ping flood attack comparatively longer, although marginally, than its predecessors, Windows Server 2008 R2 and Windows Server 2012 R2.

### 4.3.2 Smurf Attack

When a barrage of ICMP echo replies are sent to a system, it is called as a Smurf attack. When this attack was launched on the server at a magnitude of 25 Mbps like in the case of ping attack, it caused a high increase in the processor utilization. Hence, in order to monitor the increase in processor utilization more precisely, the attack traffic was increased in steps of 10 Mbps instead of 25 Mbps. Once the attack traffic of 10 Mbps was sent, in all the three virtual machines, the processor utilization increased from two percent to approximately 38 percent.

With further increase of smurf attack traffic magnitude to 20 Mbps, there was a slightly higher increase observed in the case of 2016 server OS compared to the other two virtual

machines. When 30 Mbps attack traffic was sent, the processor utilization of the virtual machine with the 2008 server operating system reached nearly 90 percent, but the other two VMs exhibited a lower increase in the processor utilization.

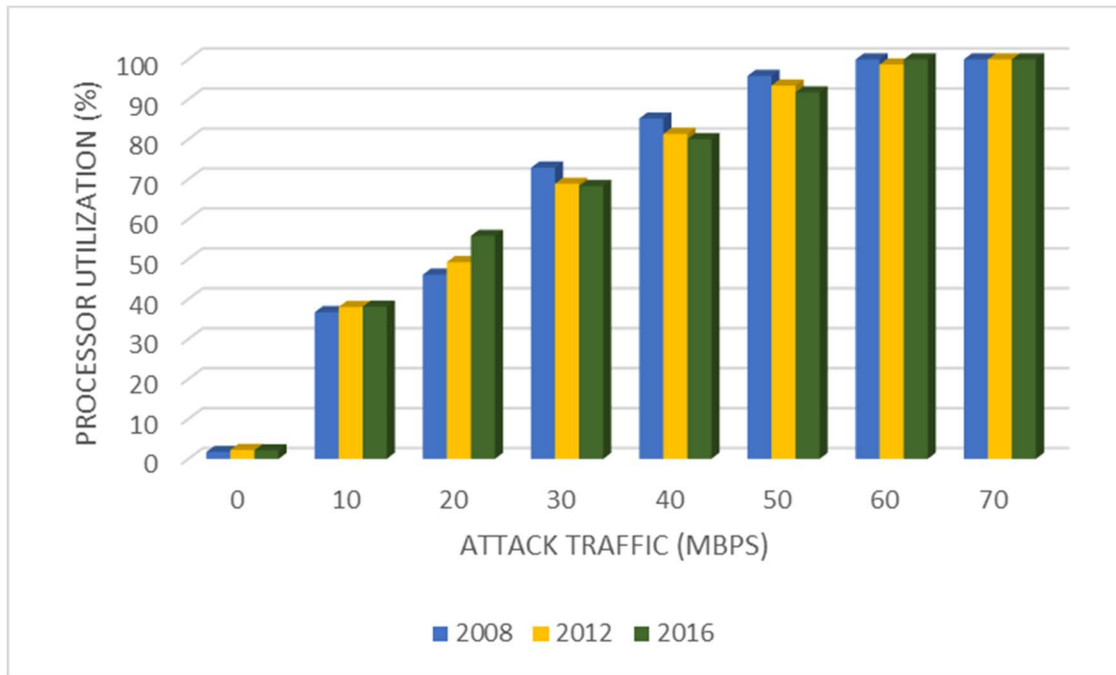


Figure 4.3 Comparison of Processor Utilization of Virtual Machines under Smurf Attack

The processor usage kept on increasing with increasing attack traffic and the 2008 Server OS virtual machine reached 100 percent at 50 Mbps attack traffic. The processor usage of the 2012 and 2016 server OSs were 98 and 100 percent respectively when they received 60 Mbps of smurf attack traffic. The processor utilization of the 2012 Windows Server OS VM reached completely consumed when 70 Mbps of smurf attack was sent to it.

### 4.3.3 TCP-SYN Flood Attack

The TCP-SYN attack is launched by flooding a victim server with TCP-SYN requests forcing the victim to respond with SYN-ACK packets and as a result making it busy to respond to legitimate client requests. The attack traffic was sent to all the ports in increments of 25 Mbps every five minutes. Initially, after establishing the baseline, the attack traffic is introduced and

the processor utilization of all the three virtual machines increased by almost 35 percent. The processor usage increased proportionately with each increase in the attack traffic magnitude in the three virtual machines.

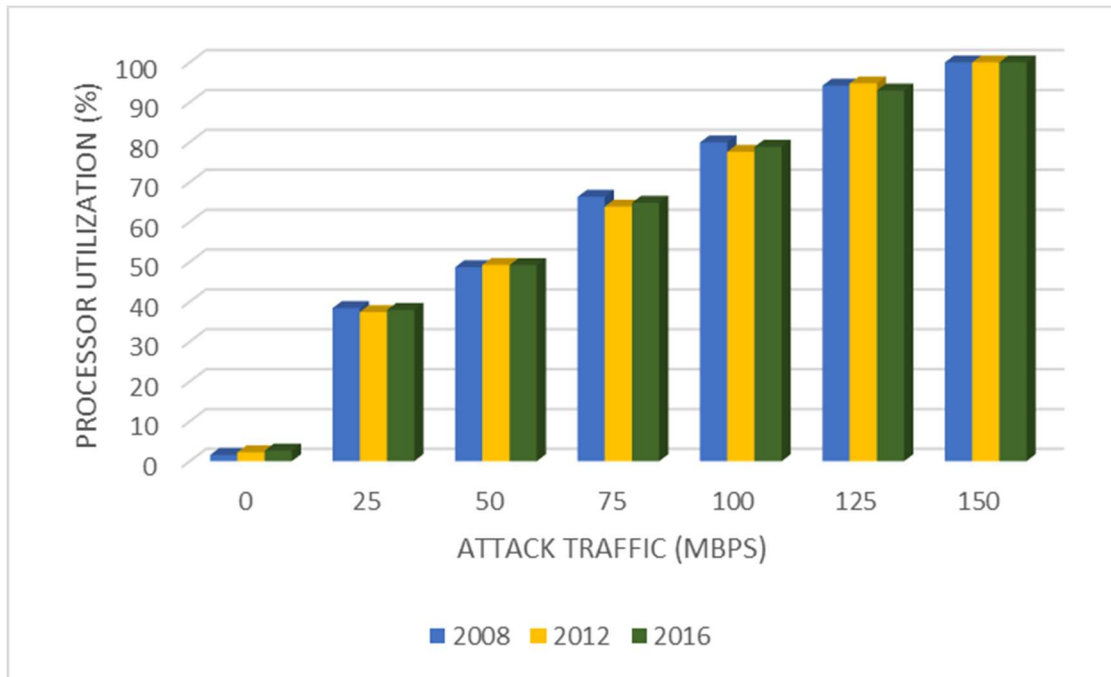


Figure 4.4 Comparison of Processor Utilization of Virtual Machines under TCP-SYN Flood Attack

The processor utilization of the VM with the 2008 server OS was higher by 3 to 4 percent compared to the other two virtual machines when it was receiving 75 Mbps and 100 Mbps attack traffic. When 125 Mbps magnitude attack traffic was sent, the processor utilization of the VM with 2016 server OS was comparatively lower than the other two virtual machines. Finally, the processor utilization all the three virtual machines reached 100 percent when an attack traffic of magnitude 150 Mbps was sent.

#### 4.3.4 UDP Flood Attack

UDP is a connectionless protocol that serves the transport layer. The UDP flood attack is launched by sending a torrent of UDP packets to the victim. In this experiment, when 25 Mbps

magnitude of attack traffic was sent to the three virtual machines, the processor utilization increased to approximately 30 percent. With further increase in the attack traffic magnitude, the processor utilization of all the three virtual machines increased uniformly in correspondence with the increase in attack traffic magnitude. When 175 Mbps attack traffic was sent, the virtual machine with the 2008 R2 operating system had reached a processor utilization of 100 percent.

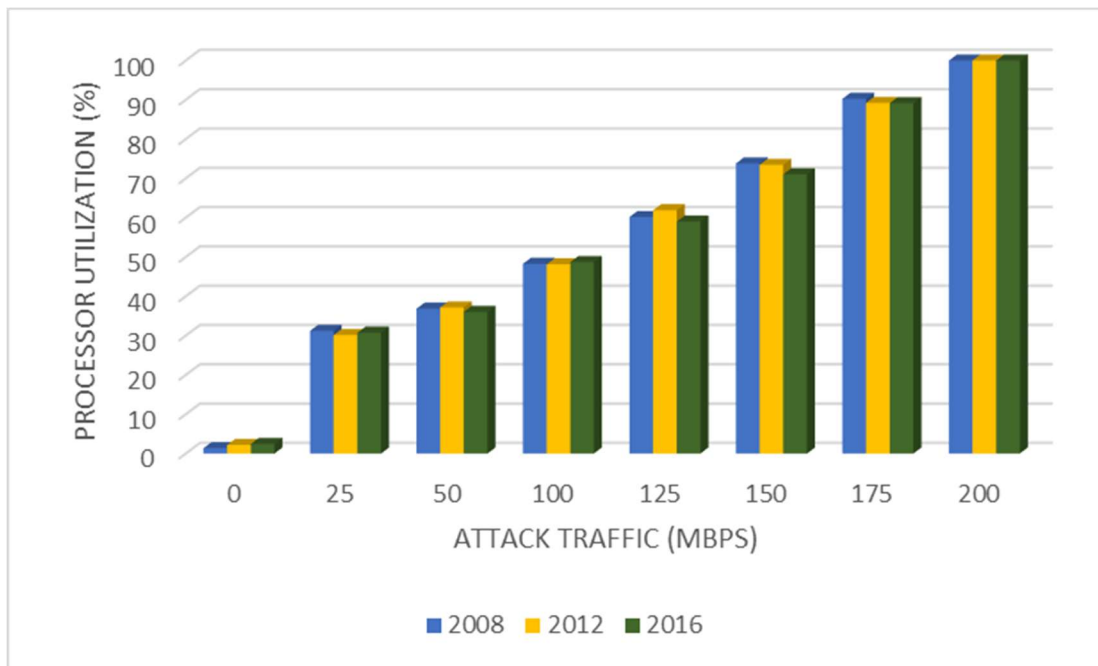


Figure 4.5 Comparison of Processor Utilization of Virtual Machines under UDP Flood Attack

The 2012 R2 and 2016 server Operating Systems were able to withstand UDP attack traffic with a magnitude of 175 Mbps without reaching 100 percent processor utilization. The processor usage of both the virtual machines reached hundred percent when 200 Mbps attack traffic was sent to them as shown in figure 4.5.

#### 4.3.5 Effect of the number of cores allocated on the Performance of a Virtual Machine under DDoS Attacks

In chapter III, only one virtual machine (Windows Server 2012 R2) was installed in the server hardware, but in this chapter, in addition to Windows Server 2012 R2, two other server operating systems were installed as virtual machines, Windows Server 2008 R2 and Windows

Server 2016 Technical Preview. Although one virtual machine has the same operating system (Windows Server 2012 R2) in chapter 3 and chapter 4, the number of cores allocated to the virtual machine are different. In chapter 3, all the cores (four) in the hardware but in this chapter, the same virtual machine is allocated one core, the same amount of memory has been allocated to the virtual machine in both the cases.

In order to analyze the effect of the number of cores allocated to a virtual machine, the results of chapter 3 and chapter 4 pertaining to Windows Server 2012 R2 have been compared. From the figure 4.6 it can be seen that the processor utilization of the virtual machine reached 100 percent utilization under Ping flood attack traffic of magnitude 200 Mbps but the same virtual machine when it had four cores is not affected as much by the ping attack. At the same attack traffic magnitude, 200 Mbps, the processor utilization of the VM with 4 cores is approximately 52 percent half the processor usage of the VM with one core. Another important result that needs to be taken note of is that the processor utilization of the VM with 4 cores does not reach 100 percent but, 75 percent at 1000 Mbps Ping flood attack traffic.

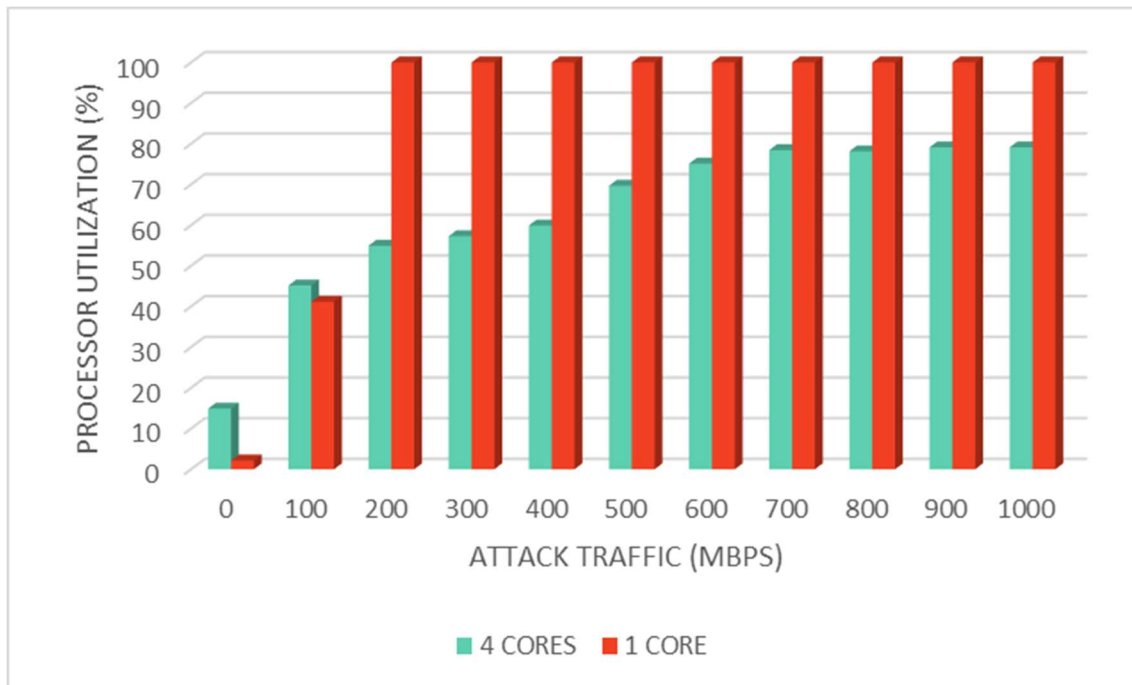


Figure 4.6 Comparison of VM with 4 cores and 1 core under the effect of Ping Flood Attack

When the smurf attack was launched on the virtual machine, the results showed that Smurf had a comparatively higher effect on the processor utilization compared to Ping attack. The results of the smurf attack are shown in figure 4.7. For the VM with one core, the processing power allocated to it was completely utilized when 100 Mbps of smurf attack traffic was sent.

The processor utilization did not reach hundred percent even when 1000 Mbps of smurf attack traffic was sent to it, the processor utilization was 89 percent. Although smurf attack is considered to have the highest impact on a system compared to the other three attacks, when a virtual machine is allocated more processor cores, it is able to withstand the attack nearly ten times better than a virtual machine which has been allocated only a single core.

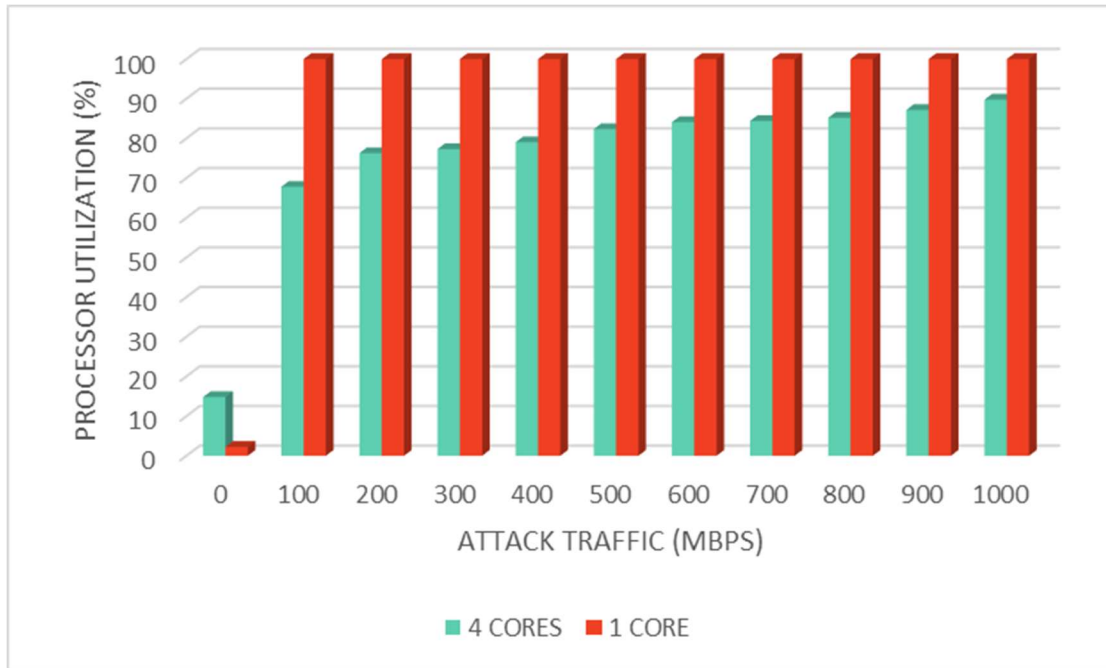


Figure 4.7 Comparison of VM with 4 cores and 1 core under the effect of Smurf Attack

In the case of the TCP-SYN attack, the virtual machine was first allocated one core and then two cores and then three and finally all the four cores were allocated to the VM. The comparison of the impact of TCP-SYN flood attack on the virtual machine is shown in figure 4.8. The figure shows at what magnitude of attack traffic the processor utilization of the VM reached one hundred percent under each different core allocation. The processor utilization of the VM with one core reached 100 percent when it received an attack traffic of magnitude 120 Mbps.



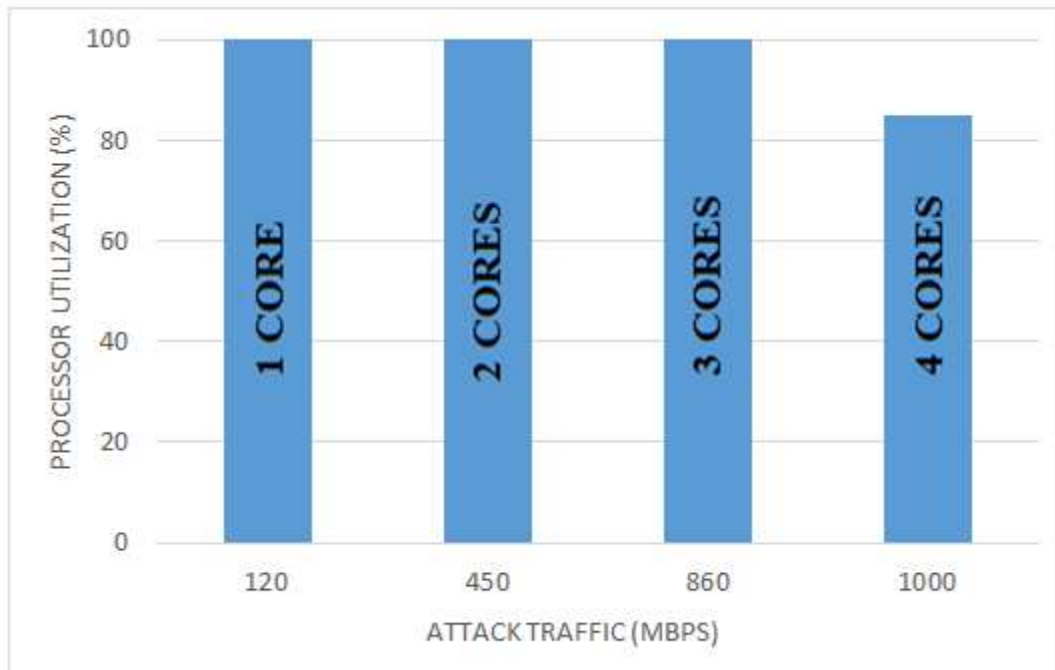
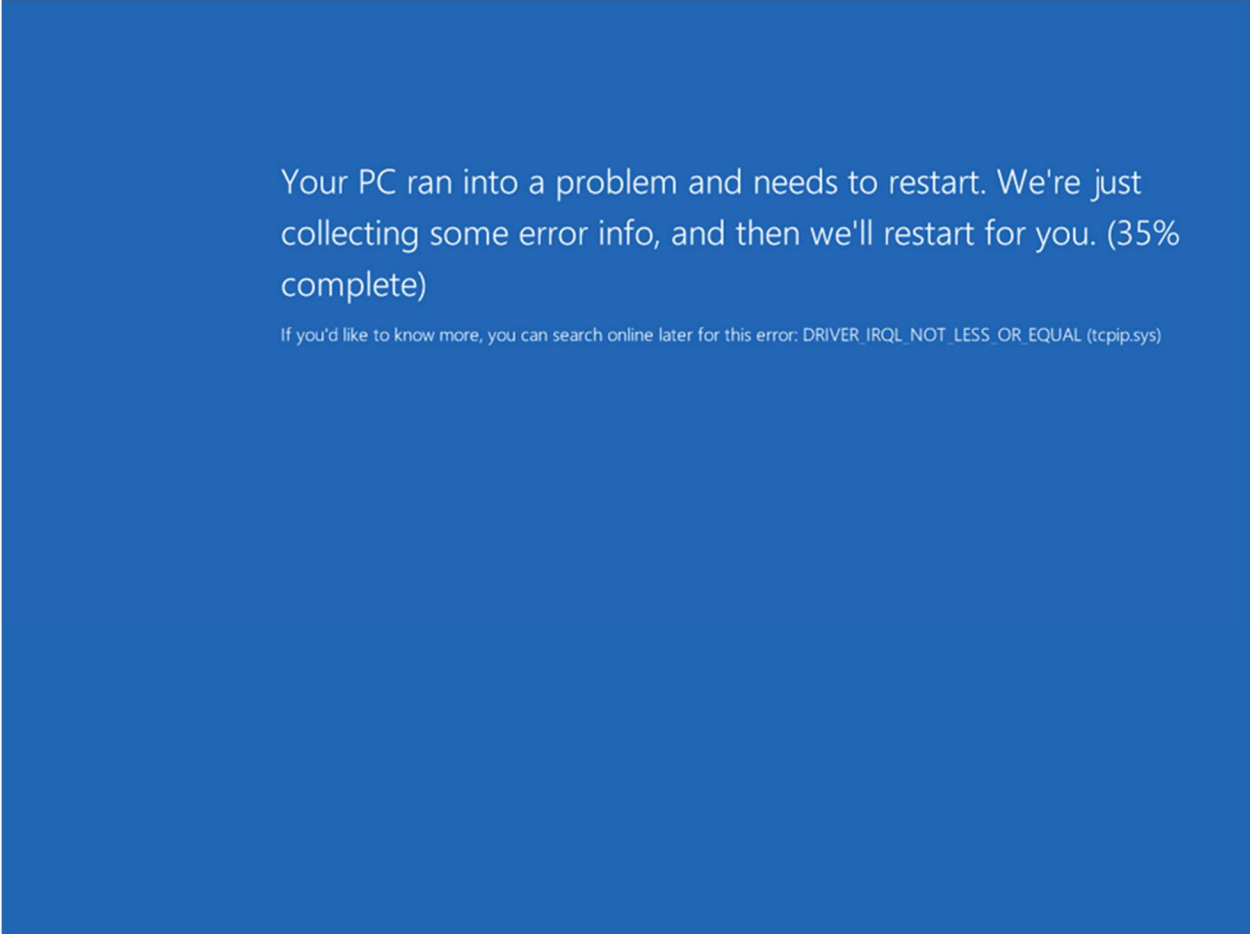


Figure 4.8 Comparison of the Processor Utilization of a VM with different number of cores under the effect of TCP-SYN Flood Attack

In chapter 2, before the server was virtualized, the TCP-SYN attack caused Blue Screen of Death (BSoD) in the 2012 R2 server operating system when an attack traffic of magnitude 3100 Mbps was sent to the server. Since it was not a virtual machine, all the four cores were available to the operating system. Now that the server has been virtualized, the same server Operating System, Windows 2012 R2, has been allocated a single core instead of four cores.



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (35% complete)

If you'd like to know more, you can search online later for this error: DRIVER\_IRQL\_NOT\_LESS\_OR\_EQUAL (tcpip.sys)

Figure 4.9 Blue Screen of Death in the Virtual Machine running 2012 Windows Server Operating System under 220 Mbps TCP-SYN Flood Attack

When the TCP-SYN flood attack traffic was sent to the virtual machine running Windows Server 2012 R2, the Blue Screen of Death occurred at an attack magnitude of 230 Mbps, which is less than one tenth of 3100 Mbps, attack traffic at which 2012 R2 OS crashed before virtualization. The figure 4.9 shows the error message displayed before the virtual machine restarted due to the TCP-SYN attack traffic. Although the same OS was attacked before and after virtualization, the virtualization and the decrease in processor resources caused the server to crash at ten times lower attack traffic.

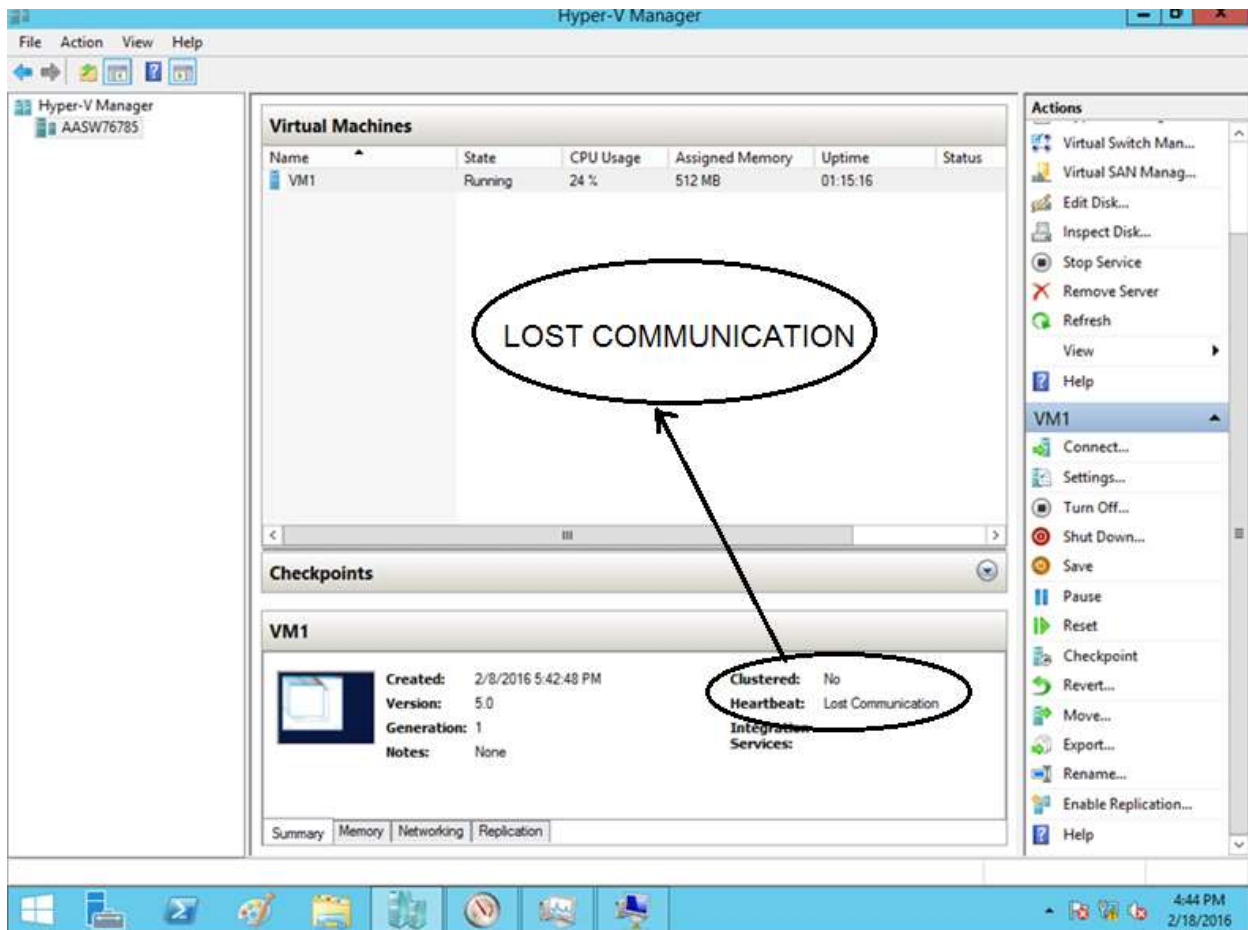


Figure 4.10 Hyper-V manager loses control over the VM which is under the impact of a DDoS Attack

When the VM had more cores, it was able to withstand a higher magnitude of TCP-SYN attack traffic before its processor utilization reached 100 percent. When 1 Gbps attack traffic was sent to the VM with four cores, the processor utilization was 83 percent. The Hyper-V host is the only interface through which the VMs installed in a host can be controlled. Hence, it is very important that the Hyper-V manager is able to communicate with the VMs. However, when the processor of a virtual machine is exhausted, then the Hyper-V Manager will not be able to communicate with such a VM as shown in figure 4.10. As a result, when a VM is under the influence of an attack, it will not be possible to take any action on the VM through the host.

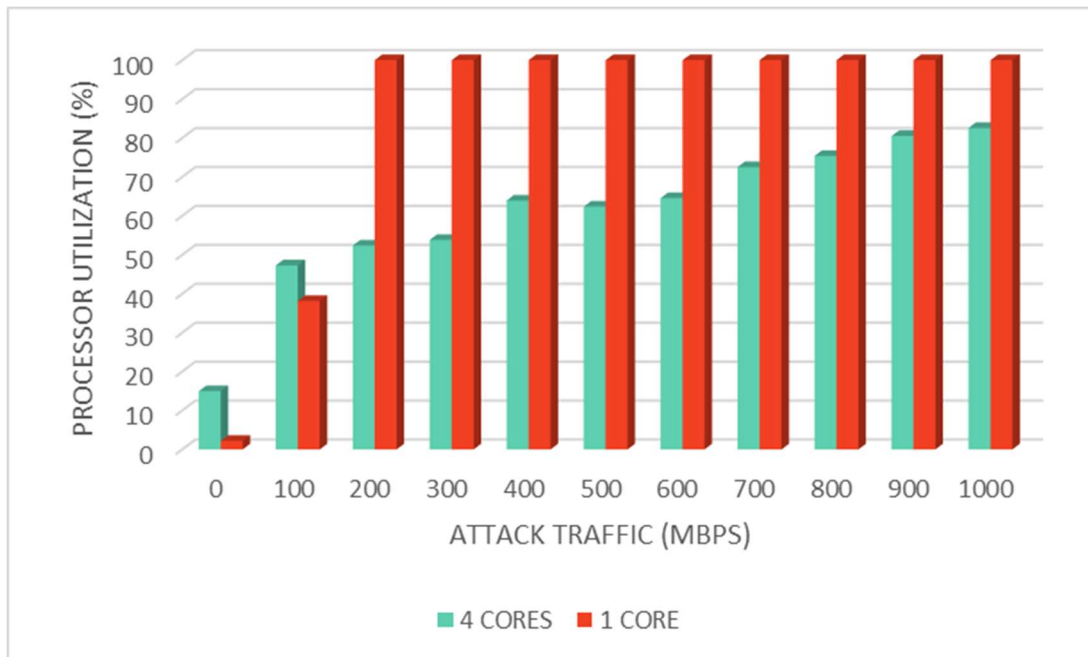


Figure 4.11 Comparison of VM with 4 cores and 1 core under the effect of UDP Flood Attack

Similar to the Ping flood attack, the processor utilization of the virtual machine with one core reached 100 percent when it received 200 Mbps of UDP attack traffic, as shown in figure 4.11. On the other hand, the processor utilization of the VM with four cores was only 51 percent while it received the same magnitude of attack traffic, 200 Mbps. The processor utilization of the 4 core-virtual machine kept increasing proportionately with increase in the attack traffic magnitude and reached 80 percent while it received 1000 Mbps of UDP attack traffic.

#### 4.3.6 Effect of the number of Virtual Machines on the Hyper-V Host under DDoS Attacks

As shown in the figure 4.12, six virtual machines were installed on the same Hyper-V host. Previously, in this chapter, the effect that DDoS attacks have on the processor utilization of the virtual machines have been analyzed. The previous section highlighted the significance of the number of cores that are allocated to a VM. Since multiple virtual machines are installed on a single hardware, it is extremely important to monitor the parameters of the host. This section consists of the analysis of how the performance of the Hyper-V host is affected based on the number of virtual machines installed in the host.

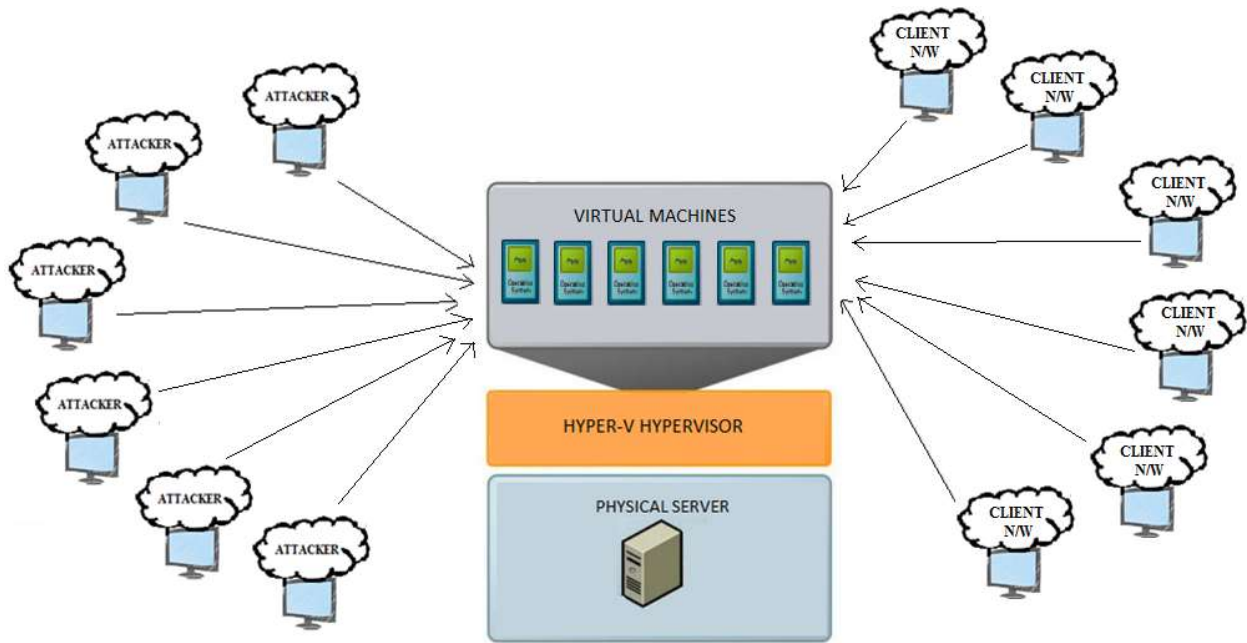


Figure 4.12 Hyper-V Host with six Virtual Machines

Initially, legitimate traffic was sent to all the virtual machines to obtain the baseline. The CPU usage of the host was 7 percent as shown in figure 4.13. Then the attack traffic was sent to one virtual machine until the processor utilization of the VM reached 100 percent, now the processor utilization of the host increased to 18 percent.

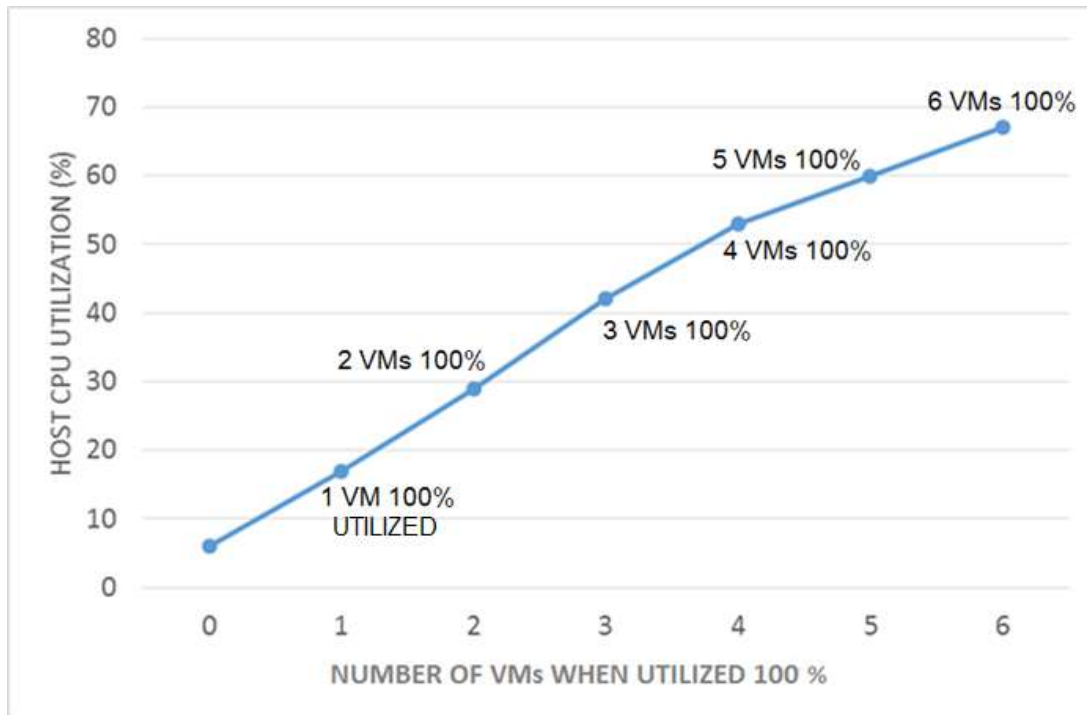


Figure 4.13 Processor Utilization of the Hyper-V Host when the attack traffic is sent to Virtual Machines installed in the Host

Next the attack traffic was simultaneously sent to two virtual machines, this caused the host CPU usage to increase to 29 percent. When a virtual machine is under attack, the attack is contained in that VM and it does not affect the other virtual machines installed in the same host. When the attack traffic was sent to more and more virtual machines, the processor utilization of the host kept increasing linearly. Finally, when the attack traffic was sent to all the six VMs in the host, the processor utilization of the host was 67 percent. Hence, the CPU utilization of the host has to be monitored to ensure the optimal functioning of the Hyper-V host.

#### 4.4 Chapter Summary

The chapter IV consists of the analysis of the comparison of the impact of DDoS attacks on virtual machines with three different Windows Server Operating Systems, 2008 R2, 2012 R2 and 2016 Technical Preview. From the results, it was observed that the virtual machine with the 2008

R2 server OS was more affected by the attacks compared to the other two virtual machines. The latest server Operating System from Microsoft, Windows Server 2016, does not seem to have considerable improvement in defending against DDoS attacks compared to its predecessor, Windows Server 2012 R2. Another significant observation is that the host Operating loses control over the virtual machine that is under attack when the processor utilization of the virtual machine reaches 100 percent. In addition to that, the attack on the virtual machine also has an effect on the Host Operating system.

In this chapter, the effect of the number of cores allocated to a virtual machine and the role it plays on the ability of a virtual machine to defend itself against Denial of Service attacks has also been analyzed. The performance of the same virtual machine is considerably affected based on the processing power allocated to it. The same virtual machine with Windows Server 2012 R2 operating system was allocated four cores in chapter III and in this chapter it was allocated one core. This caused a huge change in the effect of the attack traffic on the virtual machine. Since the number of cores allocated is only one as opposed to four, the lowered processing power available to the virtual machine caused the virtual machine to be completely exhausted when it received only one fifth of the magnitude of attack traffic.

## CHAPTER V

### CONCLUSIONS AND FUTURE WORK

The Distributed Denial of Service attacks are becoming more and more common despite the increased awareness and the myriad of network security tools such as firewalls, intelligent Intrusion Detection Systems, Intrusion Prevention Systems. It can be inferred that in addition to the many anti-DDoS tools that are employed, host-based protection against DDoS attacks needs to be improved. With virtualization becoming ubiquitous in the Information Technology field, it is essential to analyze the changes that virtualization has brought to the host-based or built-in security capabilities of operating systems as virtual machines.

In chapter 2, the performance of the Windows Server 2012 R2 operating system as a web server was analyzed under the impact of four most popular DDoS attacks, Ping flood attack, smurf attack, TCP-SYN flood attack and UDP flood attack, before it was virtualized to compare the effect of virtualization on the server. It was found that the Smurf attack was the most effective of the four DDoS attacks. It was also determined that the TCP-SYN flood attack caused the Blue Screen of Death (BSOD) at a minimum attack traffic of 3.1 Gbps within five minutes of receiving the attack traffic. From the results gathered it was observed that the root cause for the increase in processor utilization was due to the increase in number of nonpaged pool allocations which causes a decrease in the random access memory of the server.



In order to determine the overhead due to virtualization, the same operating system was installed as a virtual machine in the server hardware in chapter 3. In addition to using the same operating system, the processing power allocated to the virtual machine was also equal to that available to the operating system before virtualization. It was observed that although the same four attacks that were launched on the non-virtualized server were launched on the virtualized server, the processor utilization and the memory utilization increased at a rapid rate compared to the non-virtualized server. This kind of analysis through the allocation of all the cores in the server hardware to the virtual machine helped in effectively determining the changes introduced by virtualization to the host-based protection, or in this case VM-based protection, against DDoS attacks.

This analysis has been done to throw light on the changes introduced by virtualization with the hope that the awareness would help network security engineers be more prepared to handle and mitigate DDoS attacks in a virtualized environment. Although Hyper-V is successful in containing the effect of the attack to the attacked VM, consideration of the security aspects during the allocation of hardware resources to a virtual machine would further improve the chances of withstanding the attack.

Future work can be done in the direction of comparing the performance of the Windows Server 2016 Operating System. Other platforms for virtualization such as VMWare and other open source virtualization platforms could also be evaluated for performance and compared with the performance of Hyper-V.

## REFERENCES

- [1]. Zubair A. Baig, Farid Binbeshr, Controlled Virtual Resource Access to Mitigate Economic Denial of Sustainability (EDoS) Attacks against Cloud Infrastructures, December, International Conference on Cloud Computing and Big Data (CloudCom-Asia), 2013
- [2]. DDoS-for-hire costs just \$38 per hour, (<http://www.infosecurity-magazine.com/news/ddosforhire-costs-just-38-per-hour/>), last access on: Mar-4, 2016
- [3]. Igal Zeifman, Q4 2015 Global DDoS Threat Landscape, (<https://www.incapsula.com/blog/ddos-report-q4-2015.html>) last access on: Mar-4, 2016
- [4]. Swati Khandelwal, 602 Gbps! This May Have Been the Largest DDoS Attack in History, The hacker News, Jan 8, 2016. Available online at (<http://thehackernews.com/2016/01/biggest-ddos-attack.html>)
- [5]. John Woodrow Cox, Possible ‘ransomware’ attack still crippling some MedStar hospitals’ computers, The Washington Post, Mar 30, 2016. Available online at ([https://www.washingtonpost.com/local/likely-ransomware-cyberattack-still-crippling-medstar-health-computers-at-some-hospitals/2016/03/30/a82c9fa8-f687-11e5-8b23-538270a1ca31\\_story.html](https://www.washingtonpost.com/local/likely-ransomware-cyberattack-still-crippling-medstar-health-computers-at-some-hospitals/2016/03/30/a82c9fa8-f687-11e5-8b23-538270a1ca31_story.html))
- [6]. Ransomware attacks to grow in 2016, Security Magazine, Nov 23, 2015. Available online at (<http://www.securitymagazine.com/articles/86787-ransomware-attacks-to-grow-in-2016>)
- [7]. Rakesh Krishnan, Ransomware attacks on Hospitals put Patients at Risk, Apr 3, 2016. Available online at (<http://thehackernews.com/2016/04/hospital-ransomware.html>)
- [8]. Sanjeev Kumar, Raja Sekhar Reddy Gade, Evaluation of Microsoft Windows Servers 2008 & 2003 against Cyber Attacks, Journal of Information Security, Vol.6 No.2, Page(s):155-160, 2015
- [9]. Sanjeev Kumar, Sirisha Surisetty, Microsoft vs. Apple: Resilience against Distributed Denial-of-Service Attacks, IEEE Security & Privacy, Vol. 10, Issue 2, Page(s):60-64, 2012
- [10]. Sanjeev Kumar, Sirisha Surishetty, Apple's Leopard Versus Microsoft's Windows XP: Experimental Evaluation of Apple's Leopard Operating System with Windows XP-SP2 under Distributed Denial of Service Security Attacks, Information Security Journal: A Global Perspective, Vol.20 No.3, Page(s):163-172, 2011

- [11]. Hari Krishna Vellalacheruvu, Sanjeev Kumar, Effectiveness of Built-in Security Protection of Microsoft's Windows Server 2003 against TCP SYN Based DDoS Attacks, Journal of Information Security, Vol.2 No.3, Page(s):131-138, 2011
- [12]. Raja Sekhar Reddy Gade, Hari Krishna Vellalacheruvu, Sanjeev Kumar, Performance of Windows XP, Windows Vista and Apple's Leopard Computers under a Denial of Service Attack, 4<sup>th</sup> International Conference on the Digital Society (ICDS 2010), Page(s):188-191, 2010
- [13]. Sanjeev Kumar, Einar Petana, Mitigation of TCP-SYN Attacks with Microsoft's Windows XP Service Pack2 (SP2) Software, Seventh International Conference on Networking (ICN 2008), IEEE Computer Society, Page(s): 238-242, 2008
- [14]. Rodolfo Baez Junior, Sanjeev Kumar, Apple's Lion vs Microsoft's Windows 7: Comparing Built-In Protection against ICMP Flood Attacks, Journal of Information Security, Vol.5, No. 3, July 2014
- [15]. Sanjeev Kumar: PING attack - How bad is it? Computers & Security Journal, Vol.25, July 2006.
- [16]. Microsoft Windows Server 2012 R2 Performance Monitor Data Collector Set, Counter Description
- [17]. Mark Russinovich, David A. Solomon, Alex Ionescu, Microsoft Windows Internals Part 1, Sixth Edition, 2012
- [18]. Indiana University, Knowledge Base (<https://kb.iu.edu/d/ajmi>), last access on: Mar-4, 2016
- [19]. IBM HTTP Server Performance Tuning ([http://publib.boulder.ibm.com/htpserv/ihsdiag/ihs\\_performance.html#tcp\\_conn](http://publib.boulder.ibm.com/htpserv/ihsdiag/ihs_performance.html#tcp_conn)), last access on: Mar-4, 2016
- [20]. Transmission Control Protocol, RFC 793, September 1981
- [21]. Common Vulnerabilities and Exposures-2015-0206
- [22]. Common Vulnerabilities and Exposures-2014-3513
- [23]. Common Vulnerabilities and Exposures-2015-3508
- [24]. Maddy Keith, Global E-Commerce Sales, Trends and Statistics 2015, (<http://www.remarkety.com/global-ecommerce-sales-trends-and-statistics-2015>) last access on: Feb-21, 2016
- [25]. Denial of Service, Open Web Application Security Project, (OWASP), last revision: Feb-2, 2015
- [26]. Memory Leak, Open Web Application Security Project, (OWASP), last modified on: Jan-22, 2016
- [27]. Determining the source of Bug Check 0x133 (DPC\_WATCHDOG\_VIOLATION) errors on Windows Server 2012, MSDN blogs  
<http://blogs.msdn.com/b/ntdebugging/archive/2012/12/07/determining-the-source-of-bug-check-0x133-dpc-watchdog-violation-errors-on-windows-server-2012.aspx>

- [28]. Windows stop error 133 occurs on Windows Server 2012, Knowledge Base Dell Support <http://www.dell.com/support/article/us/en/04/SLN291258/EN>
- [29]. Knowledge Base (KB) 2789962: You receive a "DPC\_WATCHDOG\_VIOLATION (133)" Stop error message on a Windows Server 2012-based computer, Article ID: 2789962 - Last Review: 12/12/2012 15:22:00 - Revision: 4.0
- [30]. Knowledge Base (KB) 301379: Stop error when there's faulty hardware in Windows 8.1 or Windows Server 2012 R2, Article ID: 3013791 - Last Review: 07/14/2015 18:13:00 - Revision: 3.0
- [31]. Install Hyper-V and create a Virtual Machine, TechNet Library, (<https://technet.microsoft.com/en-us/library/hh846766.aspx>) last access on: Mar-16, 2016
- [32]. Brien. M. Posey, Virtualization: Optimizing Hyper-V Memory Usage, TechNet magazine, Issue December 2011 (<https://technet.microsoft.com/en-us/magazine/hh709739.aspx>), last access on: Mar-16, 2016
- [33]. Windows Platform Design Notes, Design Information for the Microsoft® Windows® Family of Operating Systems, White Paper
- [34]. Measuring Performance on Hyper-V, Microsoft Developer Network, ([https://msdn.microsoft.com/en-us/library/cc768535\(v=bts.10\).aspx](https://msdn.microsoft.com/en-us/library/cc768535(v=bts.10).aspx)), last access on: Jan-08, 2016
- [35]. Mike Neil, Make innovation easier with Windows Server 2016 and System Center 2016 Technical Preview 4, (<https://blogs.technet.microsoft.com/server-cloud/2015/11/19/make-innovation-easier-with-windows-server-2016-and-system-center-2016-technical-preview-4/>), last access on: Apr-16, 2016
- [36]. The NIST Definition of Cloud Computing (<http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>), last access on: Mar-7, 2016
- [37]. Dell PowerEdge T320 Tower server (<http://www.dell.com/us/business/p/poweredge-t320/pd>)
- [38]. Intel Xeon Processor ([http://ark.intel.com/products/75782/Intel-Xeon-Processor-E5-2407-v2-10M-Cache-2\\_40-GHz](http://ark.intel.com/products/75782/Intel-Xeon-Processor-E5-2407-v2-10M-Cache-2_40-GHz))
- [39]. Installing IIS 8.0 on Windows Server 2012 (<http://www.iis.net/learn/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012>)
- [40]. RFC 4987, TCP SYN Flooding Attacks and Common Mitigations, Aug 2007
- [41]. Eddy, TCP SYN Flooding Attacks and Common Mitigations, Request for Comments (RFC)-4987, August 2007. Available online at (<https://tools.ietf.org/html/rfc4987>)
- [42]. Igal Zeifman, Q2 2015 Global DDoS Threat Landscape Report: Assaults Resemble Advanced Persistent Threats, Incapsula Blog, Bots & DDoS, Jun 9, 2015. Available online at (<https://www.incapsula.com/blog/ddos-global-threat-landscape-report-q2-2015.html>)

- [43]. Daemon9, Route and Infinity, Project Neptune, Phrack Magazine, Volume Seven, Issue 48, File 13 of 18, July 1996. Available online at (<http://phrack.org/issues/48/13.html>)
- [44]. Bernstein.D.J, SYN cookies, December 2005. Available online at (<https://cr.yp.to/syncookies.html>)
- [45]. Jonathan Lemon, Resisting SYN flood DoS attacks with a SYN cache, BSD Conference, February 2002
- [46]. Sirisha Surisetty, Sanjeev Kumar, “Is Apple’s iMac with its Leopard Operating System Secure under Network based Security Attacks?”, The Fifth International Conference on Internet Monitoring and Protection , (ICIMP 2010),scheduled on May 9 - 15, 2010 - Barcelona, Spain.
- [47]. Sirisha Surisetty, Sanjeev Kumar, “Is McAfee Security Center/Firewall Software Providing Complete Security for your Computer?” 2010 Fourth International Conference on Digital Society, (ICDS 2010), St.Maarten, Netherlands, February 10-16, 2010.
- [48]. S. Kumar: Impact of distributed denial of service (DDoS) attack due to ARP storm, 4th International Conference on Networking (ICN), 2005.
- [49]. Dr. S Kumar, —Can Microsoft’s Service Pack 2 (SP2) security software prevent Smurf attacks? IEEE computer society, Sep 2006.
- [50]. Possible LAND attack vulnerability affects Windows XP and 2003: available online at ([HTTP://ARTICLES.TECHREPUBLIC.COM.COM/5100-10878\\_11-5611467.HTML](HTTP://ARTICLES.TECHREPUBLIC.COM.COM/5100-10878_11-5611467.HTML))
- [51]. Sanjeev Kumar: Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. Second International Conference on Internet Monitoring and Protection (ICIMP 2007).
- [52]. Mark Russinovich, NonPaged Pool Allocation in Windows, Mar 10, 2009. Available online at (<http://blogs.technet.com/markrussinovich/archive/2009/03/26/3211216.aspx>)
- [53]. CC Hameed, Memory Management – Understanding Pool Resources, Mar 7, 2007. Available online at (<https://blogs.technet.microsoft.com/askperf/2007/03/07/memory-management-understanding-pool-resources/>)
- [54]. Hemanth Tarra, Understanding Processor (% Processor Time) and Process (%Processor Time), Aug 13, 2012. Last Revision: Jun 13, 2014. Available online at (<http://social.technet.microsoft.com/wiki/contents/articles/12984.understanding-processor-processor-time-and-process-processor-time.aspx>)
- [55]. J. Postel: Internet Control Message Protocol, DARPA Internet program protocol specifications, RFC 792, September 1981.
- [56]. Keisuke Kato, Vitaly Klyuev, Large-scale network packet analysis for intelligent DDoS attack detection development, Page(s): 360-365, 9th International Conference for Internet Technology and Secured Transactions (ICITST), London,, Dec 8-10, 2014.
- [57]. Omkar P. Badve, B. B. Gupta ; Shingo Yamaguchi ;Zhaolong Gou, DDoS detection and filtering technique in cloud environment using GARCH model, Page(s): 584-586, IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Oct 27-30, 2015.

- [58]. Sidharth Sharma, Santosh Kumar Sahu ; Sanjay Kumar Jena, On selection of attributes for entropy based detection of DDoS, Page(s): 1096-1100, International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, Aug 10-13, 2015
- [59]. Xinlei Ma, Yonghong Chen, DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy, Page(s): 114-117, IEEE Communications Letters (Volume: 18, Issue: 1), Dec 6, 2013. Current Version: Jan 20, 2014
- [60]. Muhammad Agung Tri Laksono, Yudha Purwanto ; Astri Novianty, DDoS detection using CURE clustering algorithm with outlier removal clustering for handling outliers, Page(s): 12-18, International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Aug 27-29, 2015
- [61]. Chenxi Li , Jiahai Yang ; Ziyu Wang ; Fuliang Li ; Yang Yang, A Lightweight DDoS Flooding Attack Detection Algorithm Based on Synchronous Long Flows, Page(s): 1-6, 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, Dec 6-10, 2015
- [62]. Vinayaka Jyothi, Xueyang Wang; Sateesh K. Addepalli ; Ramesh Karri, BRAIN: Behavior Based Adaptive Intrusion Detection in Networks: Using Hardware Performance Counters to Detect DDoS Attacks, Page(s): 587-588, 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, Jan 4-8, 2016
- [63]. Luo Ya-Dong, Study on Detection Algorithm of DDoS Attack for Cloud Computing, Page(s): 950-953, 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), Hunan, Jun 15-16, 2014
- [64]. E-Commerce Industry Outlook 2016 (<http://www.criteo.com/media/3552/criteo-commerce-industry-outlook-2016.pdf>) last access on: Feb-21, 2016
- [65]. How One Second Could Cost Amazon \$1.6 Billion In Sales, Fast Company (<http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>) last access on: Feb-23, 2016
- [66]. Global DDoS Threat Landscape Q4 2015, Incapsula, (<https://www.incapsula.com/ddos-report/ddos-report-q4-2015.html>) last access on: Mar-4, 2016

## BIOGRAPHICAL SKETCH

Koushicaa Sundar was born on October 29, 1991. She completed her Bachelor of Engineering in Electrical and Electronics Engineering from Anna University, India in May 2013. She finished her Master of Science in Electrical Engineering from The University of Texas Rio Grande Valley, Edinburg, Texas, US on May 12, 2016. She has also served as a Teaching Assistant for the Computer Engineering Department at UTRGV from August 2015 to May 2016.

Her current mailing address is,

941, N.Sugar Rd, Veranda Place,  
Edinburg, TX- 78541.

Her publication during her Masters is,

Koushicaa Sundar & Dr. Sanjeev Kumar, "Blue Screen of Death observed for Microsoft Windows 2012 R2 under DoS Security Attack," Accepted, *Journal of Information Security*