12-2017

Security Evaluation of Virtualized Computing Platforms

Ganesh Reddy Gunnam
The University of Texas Rio Grande Valley

SECURITY EVALUATION OF VIRTUALIZED COMPUTING PLATFORMS

A Thesis

by

GANESH REDDY GUNNAM

Submitted to the Graduate College of
The University of Texas Rio Grande Valley
In partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE ENGINEERING

December 2017

Major Subject: Electrical Engineering

SECURITY EVALUATION OF VIRTUALIZED COMPUTING PLATFORMS

A Thesis
by
GANESH REDDY GUNNAM

COMMITTEE MEMBERS

Dr. Sanjeev Kumar
Chair of Committee

Dr. Wenjie Dong
Committee Member

Dr. Yoonsu Choi
Committee Member

December 2017

ABSTRACT

Gunnam, Ganesh Reddy, Security Evaluation of Virtualized Computing Platforms, Master of Science Engineering (MSE), December, 2017, 119 pp., 1 Table, 92 Figures, 81 References

In this thesis, security experiments were conducted to evaluate embedded security protocol performance of two leading server operating systems, Apple's MAC OS server LION Vs. Microsoft's Windows server 2012 R2 OS under different types of security attack. Furthermore, experiments were conducted to understand and evaluate the effect of virtualization using Hyper-V with Windows 2012 R2 OS on MAC hardware platform. For these experiments, connection rate, connection latency, non-paged pool allocations and processor core utilization for different OS, virtual machines, and under different traffic types were measured.

DEDICATION

The completion of my master's studies would not have been possible without the love and support of my family. I would like to dedicate my thesis to my parents, Annapurna, Krishna Reddy, and my brother, Madhusudhan Reddy and they wholeheartedly inspired, motivated and supported to accomplish this degree. I would like to thank P.V.M. Krishna Rao Sir (Late), Ramakrishna Sir, Rajashekar Reddy Sir, Srinivas Reddy Sir for your support and believe. Thank you for your love and patience.

ACKNOWLEDGEMENTS

I will always be grateful to Dr. Sanjeev Kumar, chair of my dissertation committee, for all his mentoring and advice. He encouraged me to complete this process through his infinite patience and guidance. My thanks go to my dissertation committee members: Dr. Wenjie Dong, and Dr. Yoonsu Choi. Their advice, input, and comments on my dissertation helped to ensure the quality of my intellectual work.

I would also like to thank my colleagues at the UTRGV library who helped me locate supporting documents for my research. I would like to thank all my friends at Network Research Laboratory in UTRGV. Also, I would like to acknowledge the many volunteers who participated in the focus group research.

Work in this thesis is supported in part by the grant awarded to Dr. Kumar by the National Science Foundation (NSF) under Grant No. 0521585 and Houston Endowment Chair in Science, Math and Technology Fellowship.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Recently there are many cyber-attacks happening in the world. Cyber threats are "growing more persistent, more diverse, more frequent and more dangerous every day," White House counter-terrorism adviser Lisa Monaco said at a cyber conference in New York on 26th of July 2016 [1]. The directive defines a significant cyber incident as one likely to harm national security or economic interests, foreign relations, public confidence, health safety or civil liberties, according to a White House fact sheet [1]. On 14 of June 2016 some hackers attacked on USA government by targeting Vermont Fish & Wildlife Department [2]. 470 gigabits per second (GBPS) distributed denial of service (DDoS) attack on an unnamed gambling website has been described as one of the largest and most complex assaults to date [3]. The perpetrators' multi-vector approach reached a packet-per-second peak of 110 million, although the assault was quickly mitigated by a security firm. The attack reportedly lasted just over four hours on 14 June 2016 and was notable not only for the strength of the assault, but also the multi-vector approach that mixed "nine different payload (packet) types". The security firm claims that only 0.2% of DDoS attacks from the first quarter of 2016 were multi-vector [3]. Blizzard's Battle.net experienced another outage on 20th Jun 2016 after facing similar issues a week before, leaving players unable to log in to popular games such as Over watch, Hearthstone and World of Warcraft due to an alleged DDoS attack [5]. Notorious hacker group Lizard Squad has claimed responsibility for the latest disruption [4]. On 25th May 2015 some unknown attackers have been

directing an ever-changing army of bots in a distributed denial of service (DDoS) attack against NS1, a major DNS and traffic management provider and attackers sent 50 million to 60 million lookup packets per second to NS1 [5]. On 4th April 2016, some hackers infiltrated the systems of W-2Express, a third-party vendor, and download the W-2 forms of 3,500 Stanford University employees [6]. Recently, Hollywood Presbyterian Medical Centre in Los Angeles paid US$17,000 to the ransomware attackers to regaining access to their patient's data [7].

In computing, a denial-of-service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. A distributed denial-of-service (DDoS) is where the attack source is more than one–and often thousands of-unique IP addresses.

In a denial of service attack, the user (Attacker) sends several authentication requests to the server. These all requests have spoofed return addresses, so that server can't find the user when it tries to send the authentication approval. The server waits for some time, before closing the connection. If server close the connection, the attacker sends a new request, and the process begins again tying up the service indefinitely [9]. A Denial of Service attack consumes a victim's system resource such as network bandwidth, CPU time and memory. In denial of service attack single attacker attacks single server.

Cloud computing has become a convenient way of accessing services, resources and applications over the internet. This cloud computing has shifted the focus of organizations and industries away from the deployment and day by day running of their IT facilities by providing an on-demand, self-service, pay as you go business model [8]. Cloud computing platform face cyber security attacks and DDoS attacks.

1.1 Statement of the Problem

It is important to understand the impact of cyber security attack on cloud computing. Server virtualization is the masking of server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. The server administrator uses a software application to divide one physical server into multiple isolated virtual environments [10]. Some servers have no built-in protection against some of DDoS attacks. In thesis two different servers Microsoft server OS and MAC server OS on Apple hardware were tested against different types of Distributed Denial of Service (DDoS) attacks. In this thesis we conduct experiments to understand the performance of virtualized server having Windows 2012 R2 operating system on MAC hardware, against security attacks and compared with non-virtualized server OS.

1.2 Distributed Denial of Service Attacks

In Distributed Denial of Service attack several computers attack one target server. First attacker develops the zombie or Daemon or Agent. It is a malicious software's which are instructed to attack the target at specific time. And then attacker tries to multiply the numbers of attackers by installing virtually the zombie at the internet user PCs which may be located at another external network to attack the target [11]. By doing this, the attacker network become giant. They are called "Botnet". And finally, Victim PCs wait for the command which is be sent by the attacker via the zombie to attack the Target. In DDoS attack, the target can be affected directly or indirectly. In indirect attack, Attacker can multiply the number of zombies to attack the single target [11], [72].

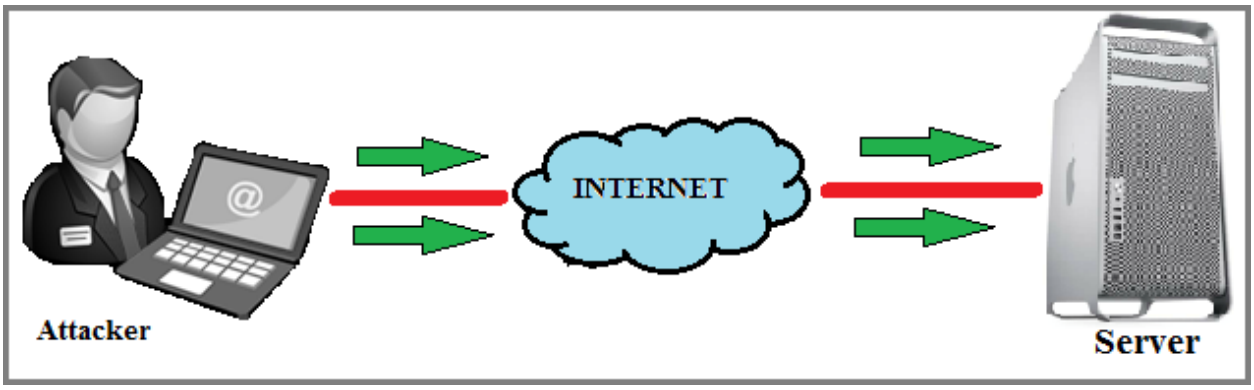

Figure 1.1 Denial of Service Attack

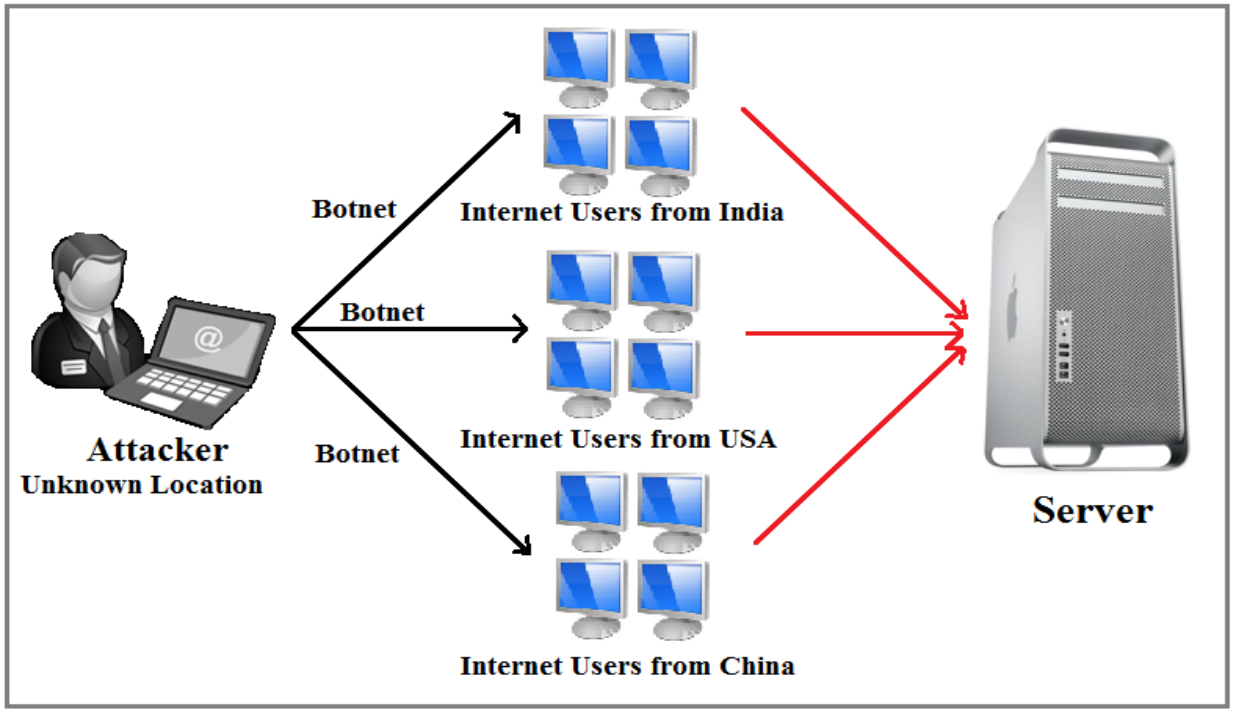

Figure 1.2 Distributed Denial of Service Attack [11]

1.2.1. ARP Flood Attack

The ARP protocol was designed for translation of addresses between the second and third layers of the OSI model. The Data link layer uses MAC addresses to communicate between different hardware devices directly on a small scale. The Network layer uses IP addresses to create large scalable networks that can communicate across the globe. ARP cache poisoning is

one of the oldest forms of modern MITM (Man in the middle) attack. It allows an attacker on the same subnet as its victims to eavesdrop on all network traffic between the victims [12].

The devices which were using ARP protocol will accept updates at any time whereas the devices with DNS protocol will accept only secure dynamic updates. This means that any device can send an ARP reply packet to another host and force that host to update its ARP cache with the new value. Sending an ARP reply when no request has been generated is called sending a gratuitous ARP. When malicious intent is present the result of a few well-placed gratuitous ARP packets used in this manner can result in hosts who think they are communicating with one host, but, communicating with a listening intruder.

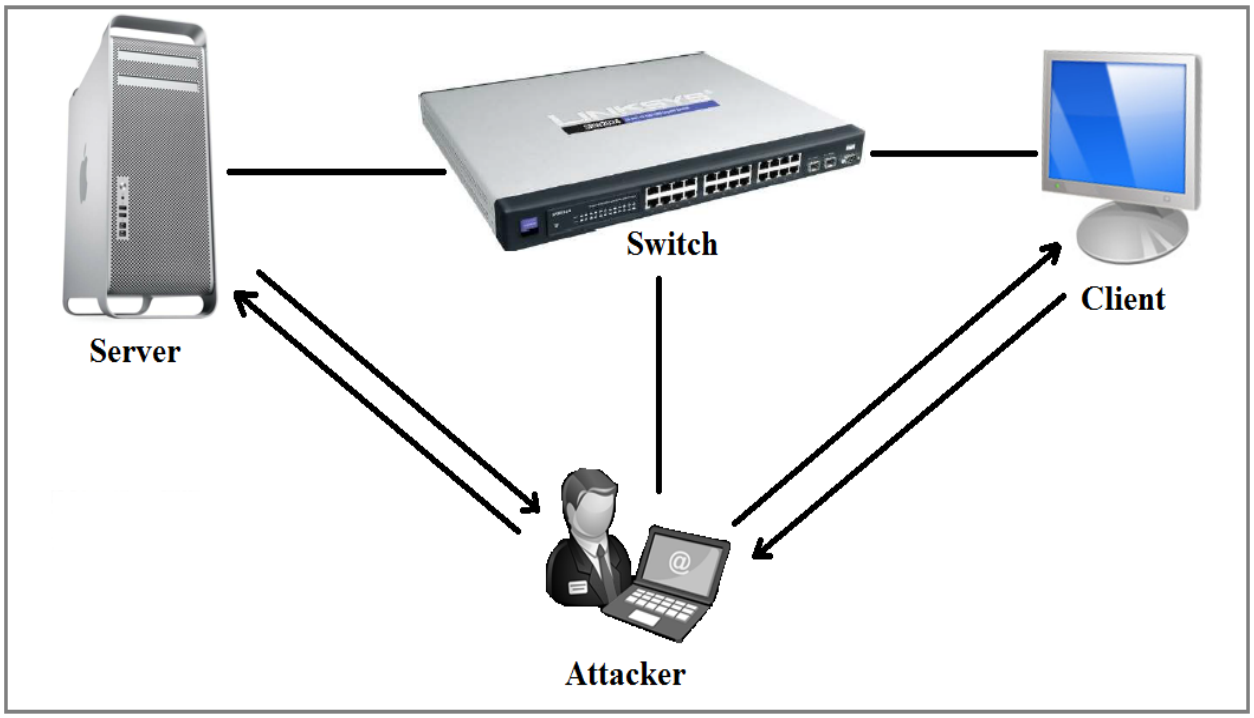

Figure 1.3 ARP Flood Attack

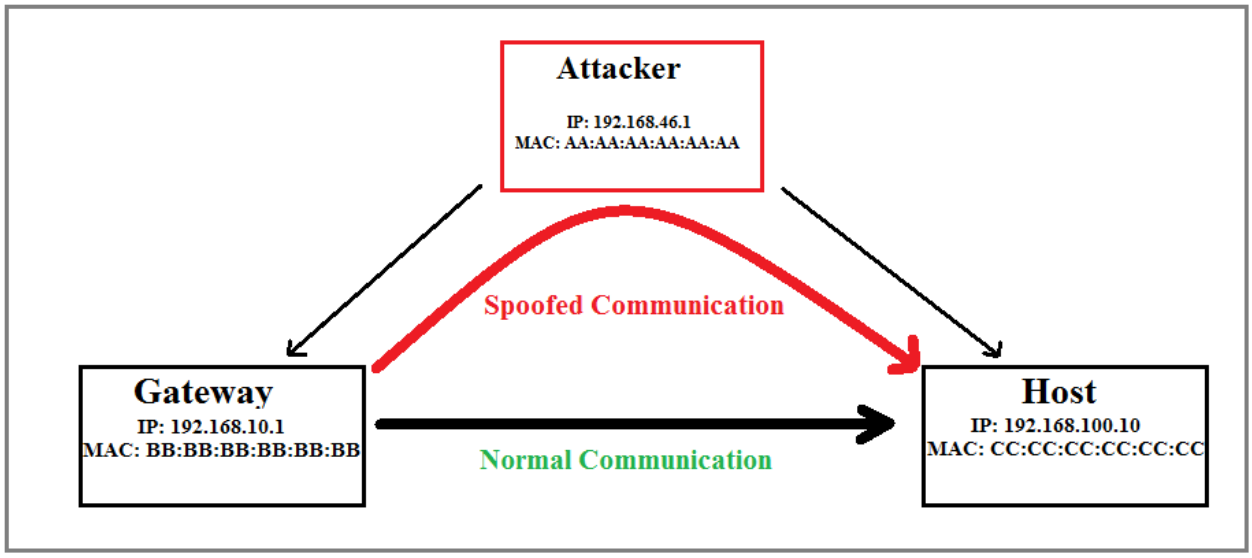

Figure 1.4 ARP Flood Attack Operation

The direct communication from Gateway to Host is the original standard traffic. The spoofing of the ARP Replies (the Gratuitous ARP Replies) convincing both sides they should send the data to the intruder. In a spoofed communication path, intruder listening in the middle of Gateway and Host.

1.2.2. Ping Flood based DDoS Attack

Ping Flood Attack is one of the oldest known network attacks, and its aim is to saturate the network with ICMP (Internet Control Message Protocol) traffic. ICMP Ping is used to verify the end-to-end internet path operation, where ICMP Echo request packet is sent to the target machine and an ICMP Echo Reply packet is expected to confirm communication between sender and receiver [13], [15].

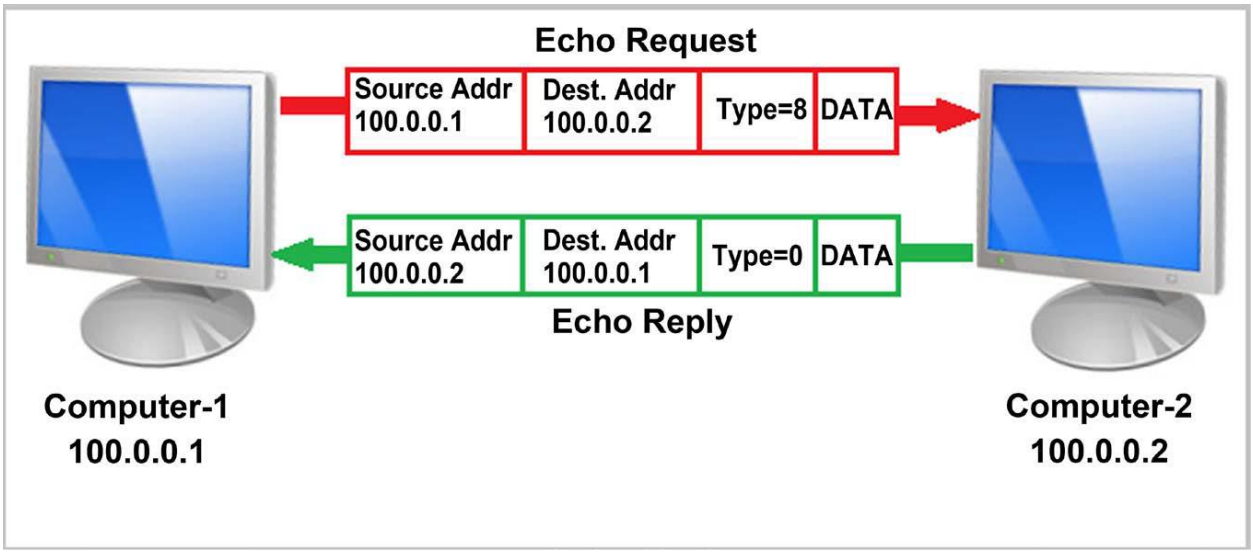

Figure 1.5 Ping Utility

A router, or a host, uses an ICMP echo request (ping) message to test a destination's reachability. A computer system that receives an ICMP echo request message will respond to it by sending an ICMP echo reply message back to the sender (Figure 1.5). Using this, an ICMP echo request and reply messages together can test the reachability of a computer on a network [14]. The ICMP echo request and reply messages are identified by the value of the type field in the ICMP message format [15]. If the value of type field is equal to 8, it becomes echo request, if the value of type field is equal to 0, it becomes an echo reply [14].

These Ping based DDoS attacks are flood of a large number of ping messages sent to target are known to be quite damaging to the availability of the web based services. The Ping attack can exhaust the target server's bandwidth and computing resources [15]. The victim computer continues receiving a Ping message that generates an ICMP echo reply message sent to the source address of the Echo Request.

1.2.3. Smurf Attack

A more sophisticated version of a DDoS attack is commonly known as a SMURF attack. A SMURF attack utilizes massive number of ICMP packets of spoofed source Internet Protocol (IP) addresses targeting the (Figure 1.6). This is achieved by altering the Echo Request sent to the botnet usingan IP broadcast address [14] [16]. The larger the Botnet is the faster and the bigger is the flood of Echo reply messages [17]. The increase of traffic reduces the target server's ability to respond, and can quickly cause a complete denial of service [18] [19].

In this attack both the ICMP echo request and ICMP echo reply messages are used. While the perpetrator sends ICMP echo request messages to an unprotected broadcast domain for amplifying the attack, the victim computer actually receives amplified attack traffic that comprises mainly of ICMP echo reply messages. If the broadcast domain has N number of computers, then for each ICMP echo request broadcasted in such a domain will generate N number of ICMP echo reply messages that are sent to the victim's server, due to the spoofed source address in the ICMP echo request messages [14].

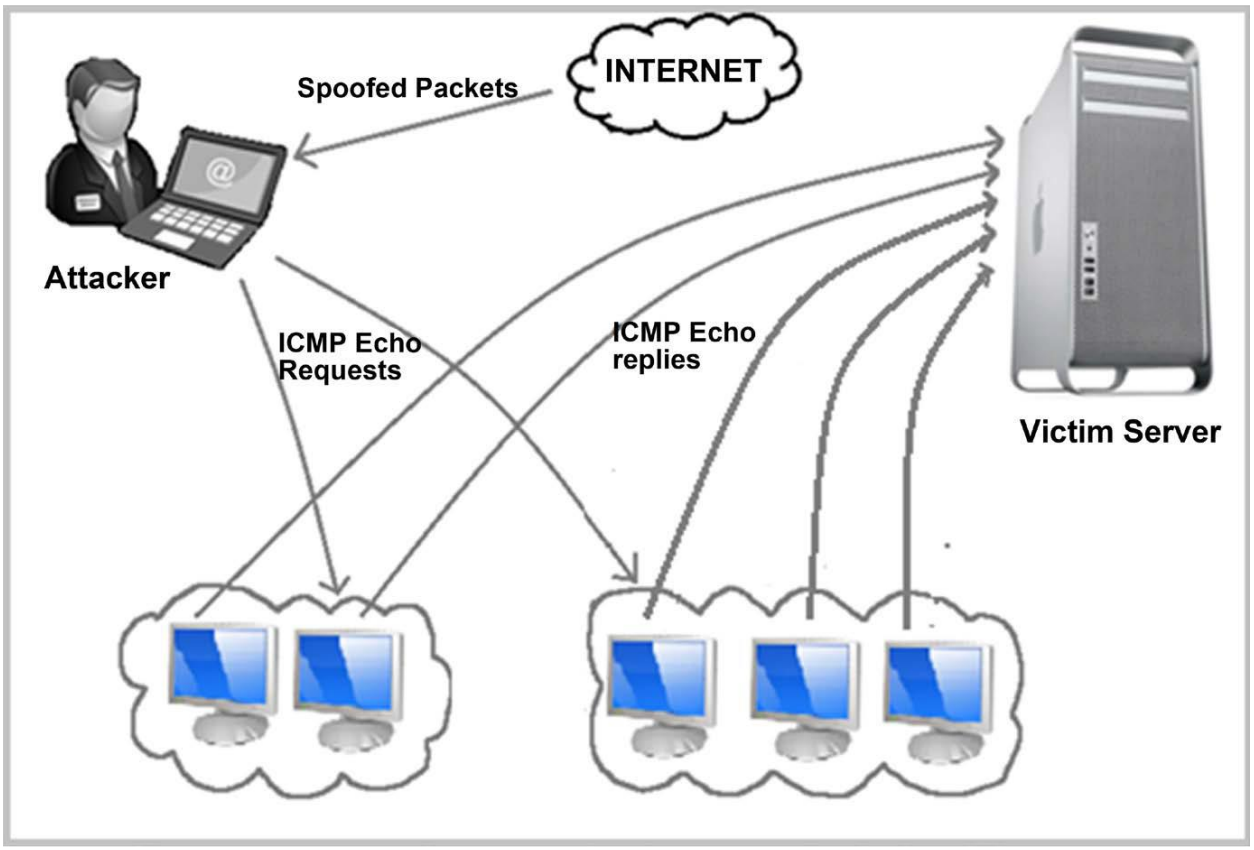

Figure 1.6 SMURF Attack [20]

1.2.4. TCP-SYN Flood Attack

The Transmission Control Protocol (TCP) connection-oriented transport-layer protocol that provides reliable byte-stream delivery between two hosts on a network [21]. TCP uses a three-way handshake to establish a network connection. Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. In this three-way handshake method first step is the active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A. The server sends back SYN-ACK (Synchronize-Acknowledgement) to the client [22]. The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B. And

finally, client sends an ACK back to server. The sequence number is set to the received acknowledgement value i.e. A+1, and the acknowledgement number is set to one more than the received sequence number i.e. B+1

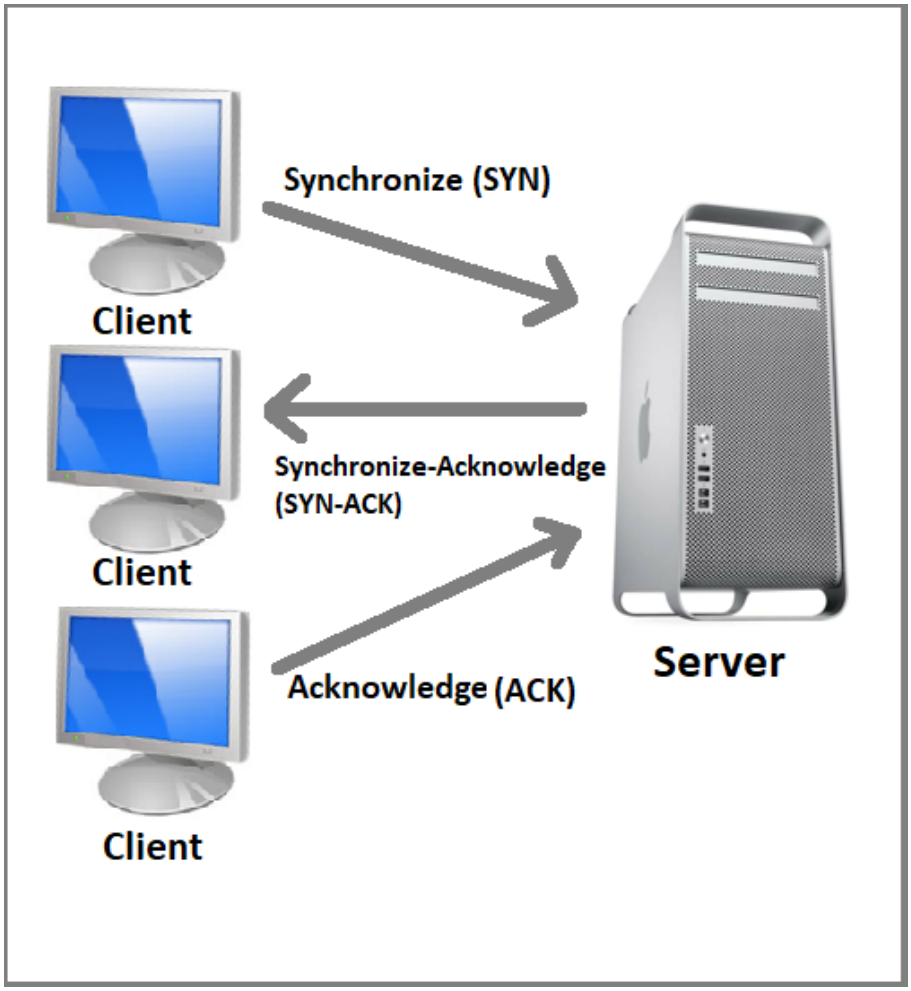

Figure 1.7 Normal 3-way handshake

In this TCP-SYN Attack, the attackers send SYN to server and then server sends SYN-ACK back to attacker, but the attacker won't send Acknowledge (ACK) back to server. Because of this the server is still waiting to get ACK to establish a connection. If the attacker keeps on doing this process the server is going to crash and it's not responds to legitimate users also.

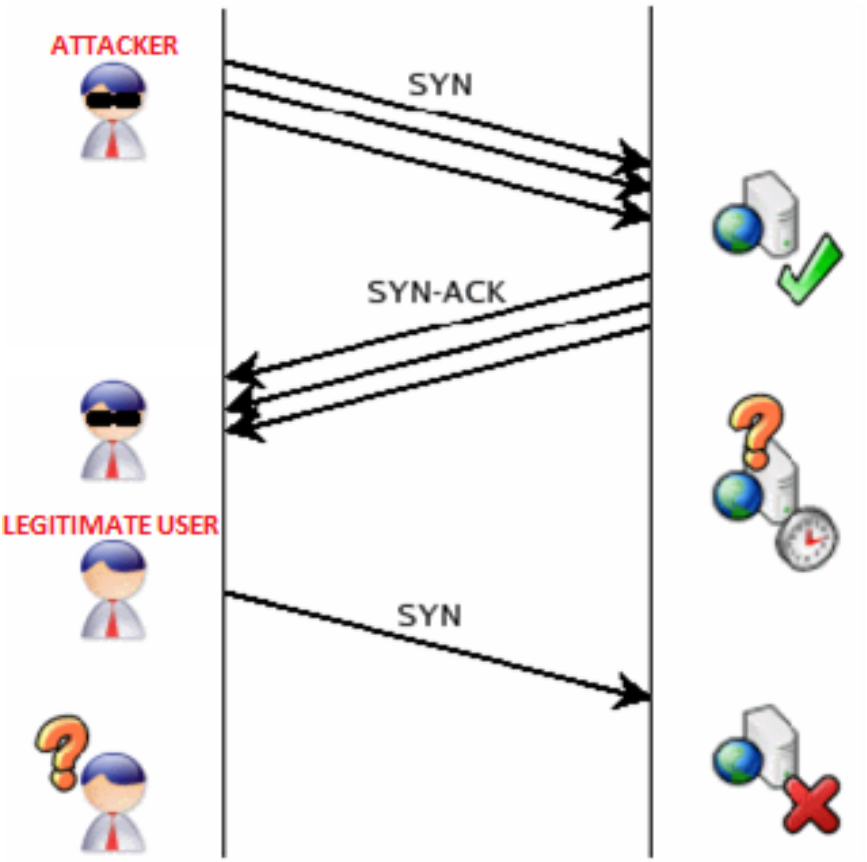

Figure 1.8 TCP/SYN Flood Attack

1.2.5. UDP Flood Attack

The User Datagram Protocol (UDP) is a connectionless computer networking protocol. The UDP is unlike TCP and there is no guarantee of delivering, ordering or duplicate protection [23]. The UDP Flood Attack is occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that the victim can no longer handle valid connections [24]. The main intention of UDP Flood Attack is to freeze the internet pipe.

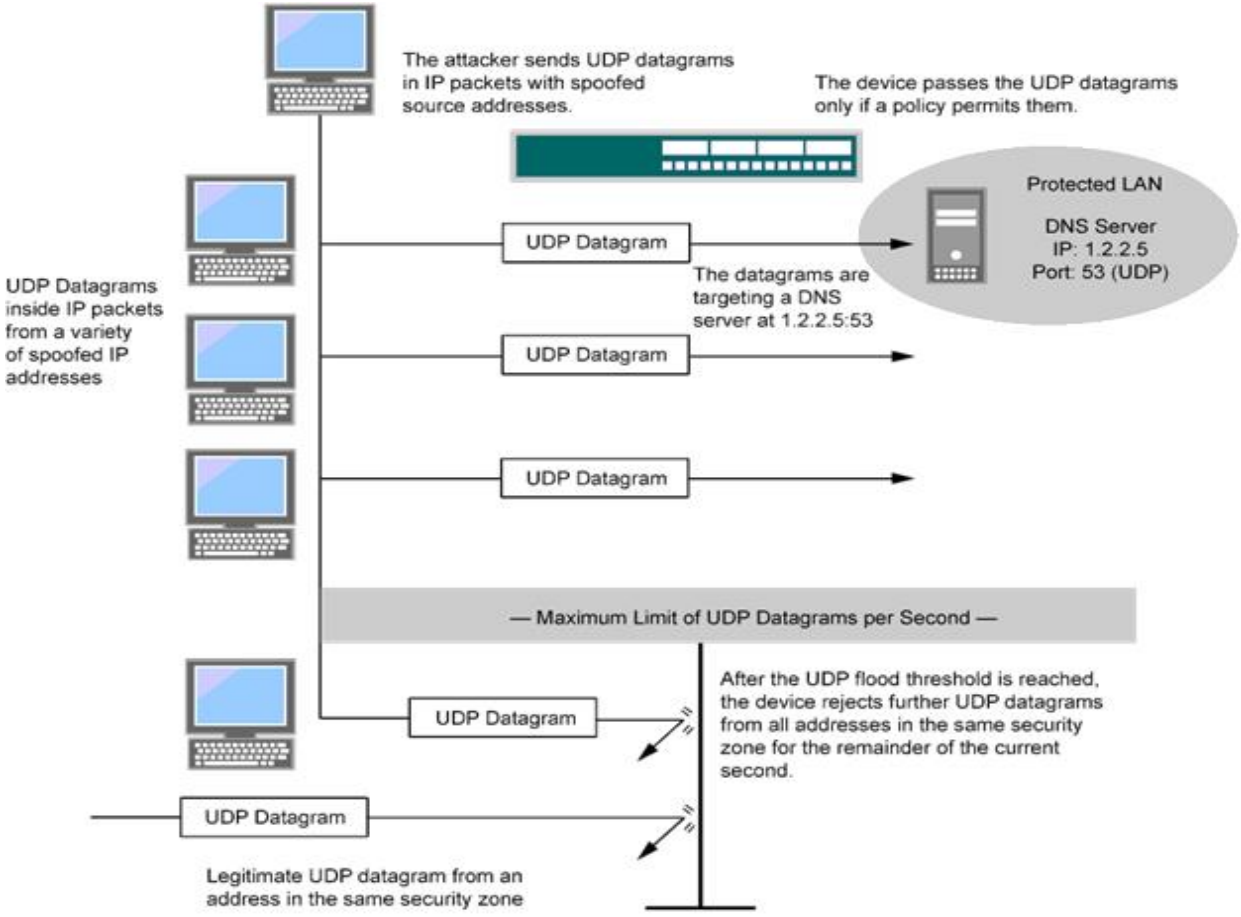

Figure 1.9 UDP Flood Attack [24]

In this UDP Flood Attack, an attacker sends UDP datagrams in IP packets with spoofed source addresses. These are all UDP datagrams targeting a DNS server. After reaching threshold limit of these datagrams, the DNS server will reject further UDP datagrams from all the addresses in the same security zone for the remainder of the current second. Because of this it will also reject legitimate UDP datagrams from an address in the same security zone.

1.3 Virtualization and Cloud Computing

From last fifty years, certain key trends created fundamental changes in how computing services are provided. In sixties and seventies mainframe processing played a big role. In eighties and nineties personal computers, client-server technology and digitalization of physical desktop were highlighted. After that internet came in to computer market. Now the new trend is cloud

computing and virtualization. Virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources [25]. Virtualization is how a computer application experiences its created environment. Virtual machine monitor (VMM) or virtual manager is the technology used for virtualization, this will separate compute environments from the actual physical infrastructure. John Livesay, vice president of Intranet, said virtualization makes servers, storage systems, and workstations independent of the physical hardware layers. Virtualization is the foundational element of cloud computing. Virtualization is the software that manipulates hardware, while cloud computing refers to a service that results from that manipulation. These are the most important trends in the today's business, because these are useful for reduce cost in IT industry while increasing efficiency, utilization and flexibility of their existing computer hardware [26] – [27].

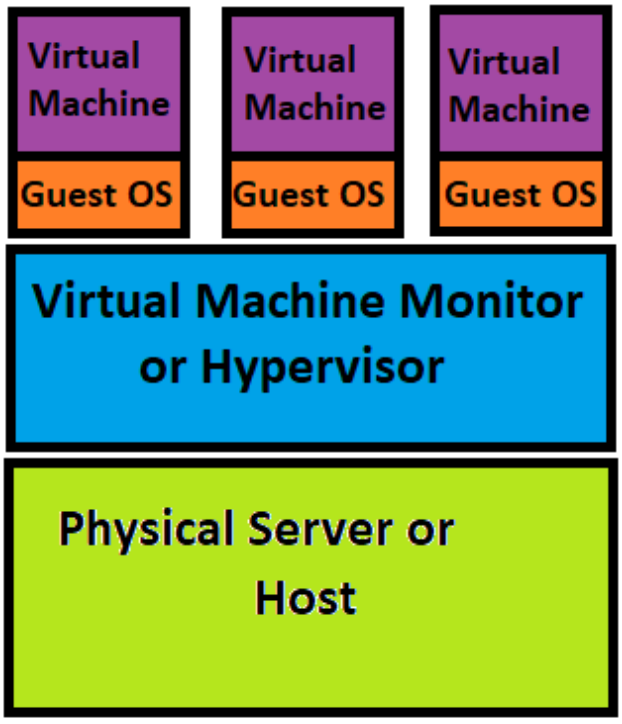

Figure 1.10 Virtualization of server

Virtualization software makes impossible for computers to run multiple operating systems and multiple applications on the same server at the same time. The first mainstream virtualization was done on IBM mainframes in the 1960s, but Gerald J. Popek and Robert P. Goldberg definition, virtualization allows many operating systems to run on the same server hardware at the same time, while keeping each virtual machine functionally isolated from all the others [27]. The first commercially available solution to provide virtualization for x86 computers came from VMware in 2001.

1.3.1 Cisco Strategy

Many enterprise customers are moving to cloud computing for improving their businesses. They need advantages of the cloud without the associated infrastucture, management, and technical issues.The Internet of Everything (IoE) — bringing together people, processes, data, and things to make networked connections more relevant and valuable than ever before it is also creating an entirely new set of requirements for globally distributed and highly secure clouds. Many cloud providing companies follows FAST (Flexible, Automated, Secure, Transformative) approch to their cloud infrastucture. Cisco launched the concept of the World of Many Clouds™ and along with thier partners they are helping customers shape their own journeys to the cloud. Cisco believes that each customer and situation requires a unique cloud solution such as public, private, hybrid, consuming services, or integrating multiple clouds together [28].

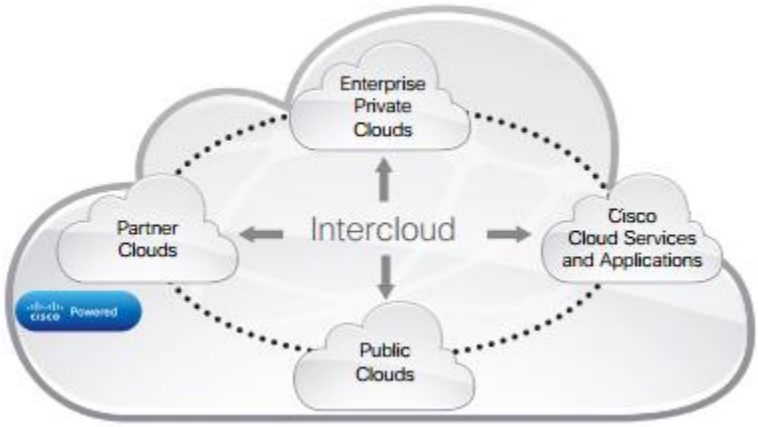

Figure 1.11 CISCO's cloud strategy is to build the platform for the Internet of Everything (IoE) [28]

The Intercloud is also driving the connection of multiple isolated clouds to a platform for the Internet of Everything (IoE) while increasing the choice of cloud consumption models for IT services.

Cisco's cloud strategy is to build the platform for the Internet of Everything with their partner ecosystem by connecting the World of Many Clouds into the Intercloud. This strategy enables businesses and cloud providers to increase ROI (Return Of Investment), reduce TCO (total cost of), lower risk, and enable business agility by using the increased efficiency, automation and management capabilities, enhanced security, transformative potential, and innovation edge that Cisco's cloud solutions, services, and partner ecosystem can provide.

Cisco's cloud strategy is based on Choice of consumption models, Intercloud infrastructure, Intercloud applications, Interoperability and open standards, and Security [28].

Cisco strategy is becoming popular because they are providing comprehensive range of options for delivering profitable services, reducing risk, increase operational reliability and availability and build flexible resource allocation.

1.3.2 Cisco Cloud Web Security

Malware can enter the Cisco network when an infected user PC connects over a direct link in the office or a VPN link from a remote location. For these connections, Cisco IT uses the Cisco® Web Security Appliance (WSA) to protect the network from malware intrusion [29]. But Web Security Appliance is not available if PC is connecting to direct Internet, without cisco network for example public networks such as restaurants. In this case user PC will be affected by malware, which may disrupt user activity. So, Cisco uses Cloud Web Security (CWS) to protect from malware.

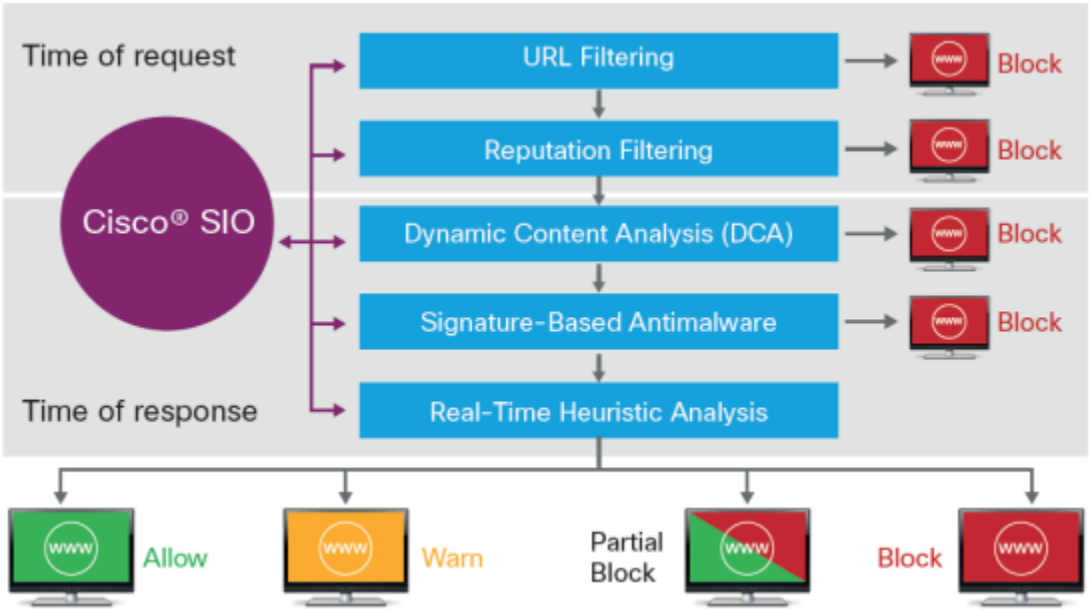

Figure 1.12 Cisco Cloud Web Security Key functionality [29]

The web-filtering service in Cisco CWS creates, enforces, and monitors web usage policies by applying real-time, rule-based filters and checking an up-to-date and accurate database for categorizing websites. By enforcing an organization's acceptable usage policy and

reducing the volume of inappropriate content, Cisco CWS helps avoid potential legal liabilities, reduces bandwidth congestion, and improves employee productivity.

When a user enters a URL in a web browser, that request is routed to the nearest Cisco CWS data center where the Cisco Security Intelligence Operations (SIO) service applies correlated detection technologies, automated machine-learning heuristics, and multiple scanning engines to detect and block known and unknown malware on websites. Cisco CWS operates independently from the user device; all control over URL access and malware detection is carried out in the cloud [29].

1.3.3 Types of common virtualization software

We have many virtualization software being used in cloud computing. For each virtualization software has its own console and its own building methods, operation and different tools [30].

1.3.3.1 GUEST OS/HOST OS

In this type of virtualization depends on an existing operating system, a third-party virtualization software technique, and creation of various guest operating systems. Each guest application runs on the host using shared resources donated by host. This virtualization deals with limited number of devices and drivers. The major disadvantage is that disk I/O suffers greatly.

VMware Server: VMware is a free offering from VMware and is considered an introductory package for use in small environments, testing, or for individuals. VMware server supports 64-bit machines as hosts and guests [30].

Sun x VM (VirtualBox): VirtualBox is also called as Sun x VM VirtualBox. This VirtualBox software is also free and cross-platform. This is unlike VMware server, it is an open source. Because of adjustable video memory, remote device connectivity, RDP (Windows Terminal Services) connectivity, and snappy performance, it may become best virtualization software [30].

1.3.3.2 HYPERVISOR

Hypervisor is a free bare-metal hypervisor that virtualizes the servers, so you can consolidate your applications on less hardware [31]. Bare-metal means server system hardware without any operating system or other software installed on it. If the hypervisor is compromised, any attached VMs will also be compromised, and the default configuration on the hypervisor isn't always the most secure [32].

Citrix Xen: Xen Server is an industry-leading, open source platform for cost-effective application, desktop, cloud, and server virtual infrastructures. This Xen software is open source hypervisor and it support para virtualization and hardware assisted virtualization. Xen allows the communication between the guest OS and the hypervisor by running modified guest kernel code with nonvirtualizable instructions replaced with calls to the hypervisor [33]. Xen Server enables organizations of any size or type to consolidate and transform compute resources into virtual workloads for today's data center requirements, while ensuring a seamless pathway for moving workloads to the cloud [34].

VMware ESX/VMware ESXi: The name ESX originated as an abbreviation of Elastic Sky X. ESXi is not a software application but it includes integrates vital OS components such as kernel. VMware ESXi is the industry-leading, purpose-built bare-metal hypervisor. ESXi installs directly onto your physical server enabling it to be partitioned into multiple logical servers

referred to as virtual machines [35]. The ESXi bare-metal hypervisor's management functionality is built into the VMkernel, reducing the footprint to 150 MB. This gives it a very small attack surface for malware and over-the-network threats, improving reliability and security [36].

Microsoft Hyper-V: Microsoft Hyper-V is windows based virtualization. We just need to install in-built Hyper-V application on windows. In this technology, we can able to run 64-bit or 32-bit virtual operating machines. We discussed more about this in later chapters.

KVM: Linux KVM (Kernel Virtual Machine) is a modified QEMU, but unlike QEMU, KVM uses virtualization processor extensions (Intel-VT and AMD-V). KVM supports many x86 and x86_64 architecture guest operating systems, including Windows, Linux, and FreeBSD. It uses the Linux kernel as a hypervisor and runs as a kernel loadable module. This KVM technology supports only hardware assisted virtualization but not para virtualization.

User-Mode Linux: User-mode Linux (UML) uses an executable kernel and a root file system to create a VM. To create a VM, you need a user-space executable kernel (guest kernel) and a UML-created root file system. These two components together make up a UML VM. The command-line terminal session you use to connect to the remote host system becomes your VM console. UML is included with all 2.6.x kernels.

Hypervisor/ Feature [9]	**Esxi**	**Hyper-V**	**XEN**	**KVM**
Version & base OS	5.50; vm kernel (Linux-based)	2012 R2; Windows Server	6.2; Linux (+QEMU)	2.6.32-279; Linux (+QEMU)
Architecture	Bare-Metal; Full, Para and H/W - Assisted Virtualization	Bare-Metal; Full, Para and H/W - Assisted Virtualization	Bare-Metal; Full, Para and H/W - Assisted Virtualization	Bare-Metal; Full, Para and H/W - Assisted Virtualization
CPU & Memory Features	Proportional Share-based Algorithm, Relaxed Co-Scheduling, Distributed Scheduler Cell	Scheduling Control with VM reservation, VM limit and relative weight	SEDF (Simple Earliest Deadline First), Credit	Linux schedulers (Fair Queuing Scheduler, round-robin, fair queuing, proportionally)
License	Commercial	Commercial	Commercial	License
Network Management	Priority-based Network I/O Control, TCP segmentation offload, net queue, distributed virtual switch	Fixed disks, pass through disks, dynamic disks	FIFO-based scheduling	FIFO-based scheduling
Storage Management	Latency-aware Priority-based scheduler, storage DRS	Fixed disks, pass through disks, dynamic disks	No-op, anticipatory, deadline, completely fair queue (CFQ)	No-op, anticipatory, deadline, completely fair queue (CFQ)
Live Migration	Yes	Yes	Yes	Yes, same CPU type
High availability	Yes	Yes	Yes	Yes
Network SRIOV	Enabled	Enabled	----	Yes

Logical processors	320	320	Unlimited	Unlimited
Physical memory	4TB	4TB	4/64TB/unlimited	4/64/unlimited
Virtual CPUs per Host	2048	2048	160 core/host 1096	160 core/host 4096
Virtual CPUs per VM	64	64	64	64
Memory per VM	ITB	1TB	2TB	2TB
Active VMs per Host	1024	512	Unlimited	Unlimited
Guest NUMA	Yes	Yes	Yes	Yes
Maximum Node	32	32	32	32
Maximum VM's	8000	4000	4000	4000

Table 1: Hypervisor comparison Table [37]

1.3.3.3 Hardware Emulation Software

Bochs: Bochs is a free, open source, Intel architecture x86 (32-bit) emulator that runs on UNIX and Linux, Windows, and Mac OS X, but only supports x86-based operating systems. Bochs is a very sophisticated piece of software and supports a wide range of hardware for emulating all x86 processors and x86_64 processor architecture. It also supports multiple processors but doesn't take full advantage of SMP now.

QEMS: QEMU is another free, open source emulation program that runs on a limited number of host architectures (x86, x86_64, and PowerPC) but offers emulation for x86, x86_64, ARM, Sparc, PowerPC, MIPS, and m68k guest operating systems.

1.3.3.4 Microsoft Virtual PC and Virtual Server:

Virtual PC is a free virtualization software package from Microsoft. Virtual PC uses emulation to provide its VM environment. These are good solutions for hosting a few VMs on a Windows XP

Workstation or Windows 2003 Server. It isn't a large environment solution by any stretch of the imagination, but it can get some VMs up and running cheaply and in very short order.

1.4 Thesis Outline

In this thesis, I investigate the performance of Apple MAC PRO server having Windows Server 2012 R2 operating system against deferent Distributed Denial of Service (DDoS) attacks. And I evaluate the security of virtualized server against DDoS attacks.

In Chapter-I, I discussed the introduction of Distributed Denial of Service attacks, Virtualization and Cloud computing. And I also mentioned different types of virtualization software's available and their comparison. In Chapter-II, I compared Windows Server 2012 R2 OS on Apple MAC PRO hardware platform with MAC OS X SERVER LION 10.7.5 against different DDoS attacks sent to the inbuilt ethernet ports on the victim server. In Chapter-III, I installed 4-port Broadcom gigabit ethernet NIC adapter to Apple MAC PRO server and then evaluated the security against different DDoS attacks were sent to one port, two ports and four ports respectively. In Chapter-IV, I virtualized the Windows Server 2012 R2 operating system on Apple MAC PRO hardware by installing Hyper-V. And then, I installed four Virtual Machines having Windows Server 2012 R2 operating system on it. And later, I evaluated the performance of virtual machines by sending different DDoS attacks. In Chapter-V, I compare the performance of Virtualized server with the performance of Non-Virtualized server against DDoS attacks. And I compare multiple Virtual Machines affected by the DDoS security attacks.

CHAPTER II

EFFECT OF DISTRIBUTED DENIAL OF SERVICE ATTACKS ON MAC PRO SERVER WITH INBUILT ETHERNET PORTS

Windows Server 2012, codenamed "Windows Server 8", is the sixth release of Windows Server. It is the server version of Windows 8 and succeeds Windows Server 2008 R2. Two pre-release versions, a developer preview and a beta version, were released during development. The software was generally available to customers starting on September 4, 2012 [38].

Microsoft introduced Windows Server 2012 and its developer preview in the BUILD 2011 conference on September 9, 2011. However, unlike Windows 8, the developer preview of Windows Server 2012 was only made available to MSDN subscribers. It included a graphical user interface (GUI) based on Metro design language and a new Server Manager, a graphical application used for server management. On February 16, 2012, Microsoft released an update for developer preview build that extended its expiry date from April 8, 2012 to January 15, 2013 [39].

The successor to Windows Server 2012 is called Windows Server 2012 R2, was released along with windows 8.1 in October 2013. Windows Server 2012 has a new version of Task Manager along with older version. And it’s having a new version of Hyper-V, consists of many new features includes network virtualization, multi-tenancy, storage resource pools, cross-premises connectivity, and cloud backup.

2.1 Experimental Setup

In the experiment, simulated PING/SMURF attack traffic is sent to the victim server simultaneously from multiple networks. In the process of evaluating the impact of the attack in an organization-like environment, legitimate or client traffic is also sent to the server simultaneously along with attack traffic [40]. We measured the processor utilization, memory utilization and HTTP transactions for different loads of attack traffic ranging from 0 Mbps to 1Gbps over a gigabit Ethernet link connected to the victim computer [41]- [42]. So much work has been done on different operating systems with DDoS attacks [43]- [59], [69]-[82] but the companies are still not able to correct all problems that have been observed.

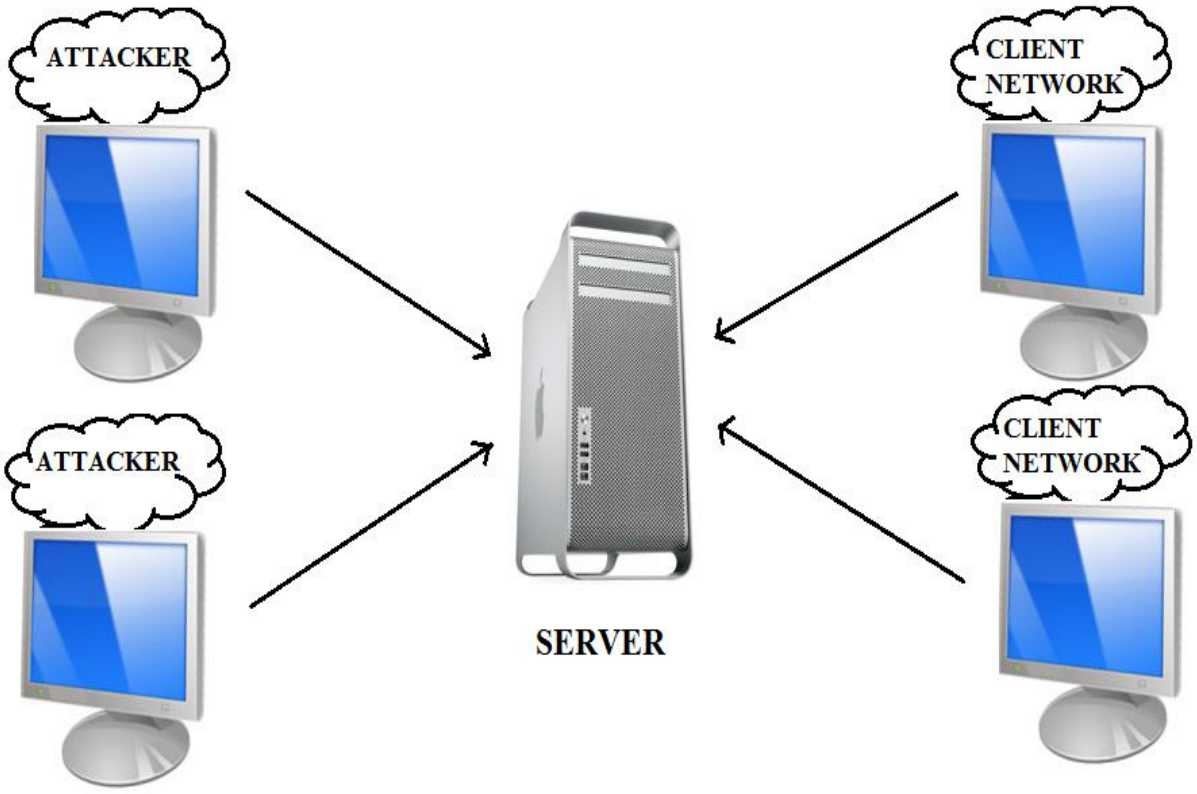

Figure 2.1 Experimental Setup

The PING and SMURF attacks were performed using the experimental set up is shown in Figure.2. The victim server is an Apple MAC PRO, Two 2.4GHz Quad-Core Intel Xeon E5620 "Westmere" processors server, 8 logical processor and 12 GB RAM [60]- [61]. As mentioned earlier, Windows 2012 R2 Standard Operating System and Apple MAC PRO server to MAC OS X SERVER LION 10.7.5 (11G63) has been installed in the victim server. Because we were going to compare these two operating systems in terms of HTTP transaction rate, CPU utilization and memory utilization response to Different DDoS attacks. The only protection mechanism that was active on the server platform was firewall in the both operating systems.

In order to communicate with the client, first a sample webpage namely Index.html was created in the victim server with support of IIS service [63]. And then, this sample webpage was accessed through the Hyper Text Transfer Protocol (HTTP) request from a Client. The victim server was responds to the clients whenever server received a HTTP request from client. In this thesis, we created a controlled environment requesting web server request to the victim server. We calculated server capacity by means of HTTP connections per second, CPU utilization, memory utilization and non-paged pool allocations. Distributed Denial of service(DDoS) attacks were sent to the victim server in two different scenarios in this chapter. In these two different experimental setups, the legitimate or client traffic is sent at the rate of 3000 HTTP requests per second to the server.

2.2 Performance parameters for Evaluation

In this experiment, the parameters that are used to evaluate the performance were Memory utilization, CPU utilization, Non-paged pool allocation and HTTP transactions per second. Some of these parameters we collect from Performance monitor present in that particular Operating system. In performance monitor, click on "Data Collector Sets" option, then select

"User Defined" to create new data collector set. And then we can create manually by selecting those performance parameters.

CPU Utilization (CPU Usage in %): % Processor Time is 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). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It should be noted that the accounting calculation of whether the processor is idle is performed at an internal sampling interval of the system clock (10ms). On today's fast processors, % Processor Time can therefore underestimate the processor utilization as the processor may be spending a lot of time servicing threads between the system clock sampling intervals. Workload based timer applications are one example of applications which are more likely to be measured inaccurately as timers are signaled just after the sample is taken. The Processor utilization is amount of usage to the total central processing unit (CPU). This will evaluate whether that attack traffic is effect on the CPU. If CPU utilization is more, that attack traffic is CPU intensive attack. The name of counter that is used to evaluate processor utilization is known as \Processor (_Total) \% Processor Time.

The Processor performance object consists of counters that measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The processor object represents each processor as an instance of the object. This particular MAC PRO server has 8 logical processors, 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, \Processor (3)\%Processor Time, \Processor (4)\%Processor Time, \Processor (5)\%Processor Time, \Processor (6)\%Processor Time, \Processor (7)\%Processor Time.

Memory Utilization (RAM Usage in MBytes): Available MBytes is the amount of physical memory, in Megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free and zero-page lists. The memory utilization is the amount of RAM usage with respect to total random-access memory available assigned to that particular operating system. If the memory utilization is more then we can say that this particular attack is called Memory intensive attack. The name of the counter that is used to evaluate memory utilization is known as \Memory\Available MBytes.

HTTP transaction per second: This HTTP transactions are referred to the number of legitimate connections established by the server. This parameter will give the number of connections per second established by the server for different amount of attack traffic ranging from 1 Mbps to 1 Gbps. This parameter helps in determine whether the server has reached to its saturation point.

2.3 Results and discussion

2.3.1 Windows Server 2012 R2 OS on MAC platform

In this section, different Distributed Denial of Service Attacks sent to Windows Server 2012 R2 operating system is installed on Apple Server platform. In this chapter, we used Apple inbuilt Ethernet ports in the Apple MAC PRO 2010. We discussed the evaluation of different DDoS attacks on this setup.

2.3.1.1 Ping Flood Attack

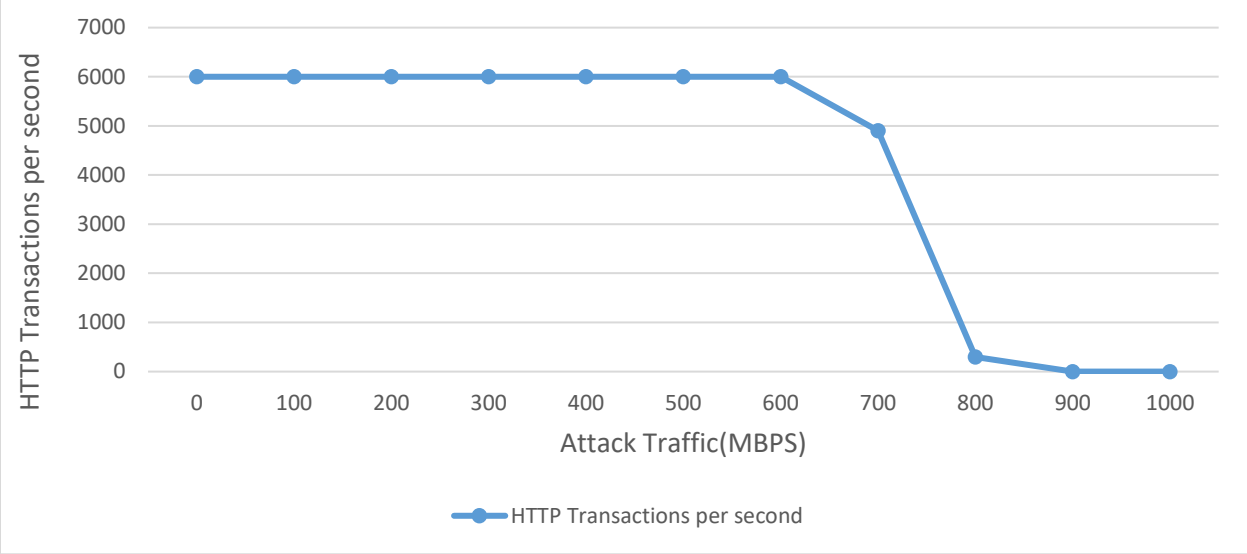

Figure 2.2 Number of HTTP connections established by the server under Ping Flood Attack on Windows 2012 R2 OS on Apple Server Platform

In this scenario, we used the Windows Server OS on the Apple's server hardware platform. To analyze the effectiveness of an attack on the server, we found the maximum number of HTTP connections that can be establish on the server without the presence of attack traffic (baseline performance), and then this result was compared with the results obtained in presence of the attack traffic. In the beginning, the legitimate HTTP connections were established with the server in the absence of attack traffic, and then the simulated attack traffic was introduced in the network and intensity was measured. To evaluate the impact of the ICMP based attack traffic,

the number of HTTP connections that the server could handle was recorded for various amount of attack traffic ranging from 100 Mbps to 1 Gbps. The baseline performance of the server with no attack traffic was measured to be 6000 HTTP connections per second. After baseline HTTP connections were established, simulated attack traffic was introduced in the range of 100 Mbps to 1 Gbps to the network. Traffic intensity was measured in the steps of 100 Mbps. When the PING attack traffic was introduced as shown in Figure 2.2, the base line performance of 6000 HTTP connections of the Windows server was maintained up to 600 Mbps of PING attack traffic. However, as the PING flood was increased beyond 600 Mbps, the server's baseline performance was found to decline. When the attack traffic reached 700 Mbps, the number of HTTP connections declined to 4950 HTTP connections. At 800 Mbps of attack traffic the legitimate connections declined to 350 only. Finally, at higher PING flood intensity greater than 800 Mbps, no legitimate connections could be established with the server (Figure 2.2).

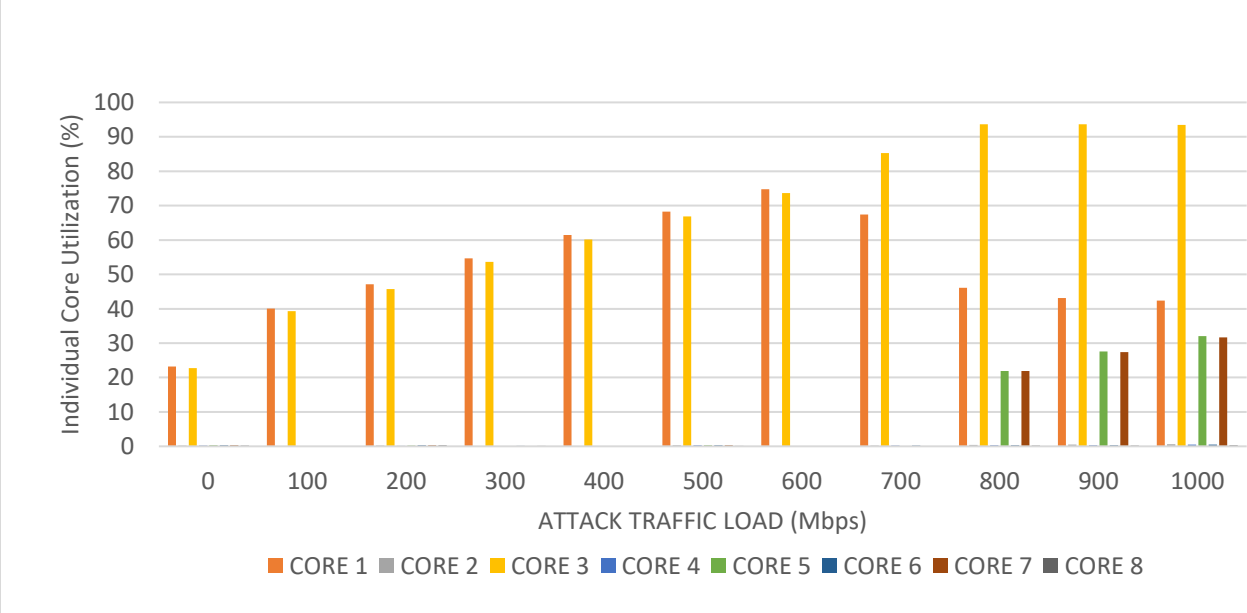

Figure 2.3 Individual Core Utilization under Ping Flood Attack when sent to the Victim Server on Windows 2012 R2 OS on Apple Server Platform

In the initial stage of individual core utilization only two logical processors are participated. At 800 Mbps of Ping attack traffic, two more processors are start sharing utilization because the number of HTTP transactions are almost zero.

2.3.1.2 Smurf Attack

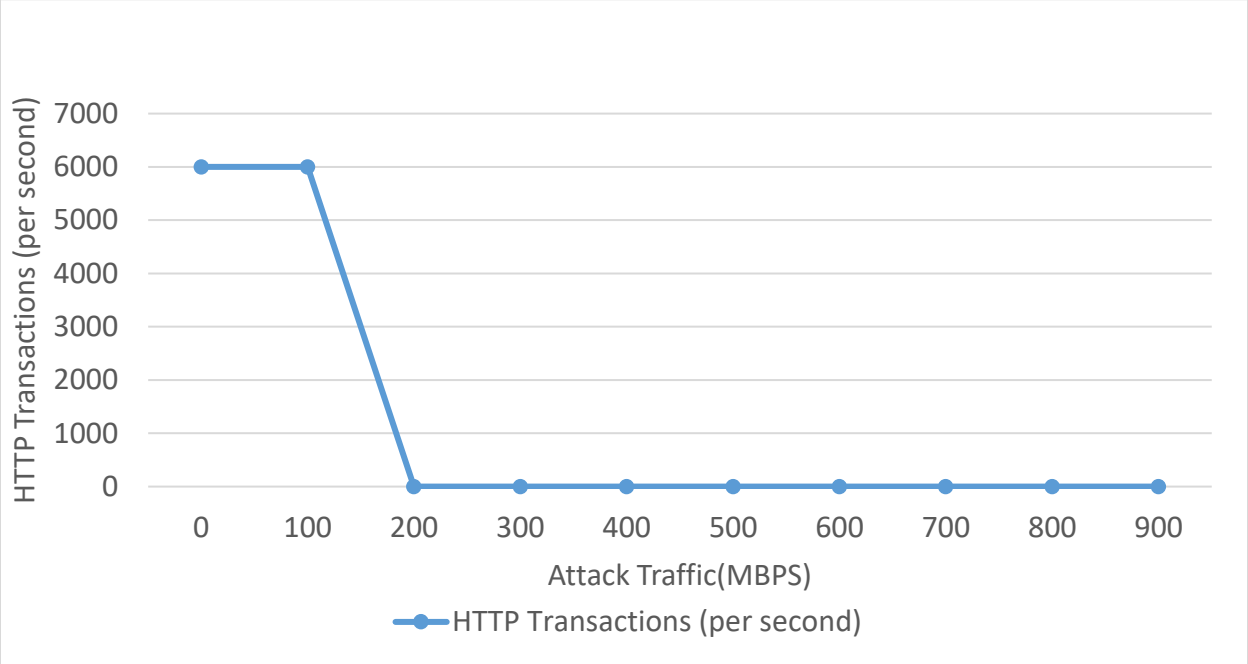

Figure 2.4 Number of HTTP connections established by the server under Smurf Attack on Windows 2012 R2 OS on Apple Server Platform

In scenario, the Smurf flood attack was used to evaluate Windows Server OS 2012 R2 on the same server hardware platform from Apple Inc. A drastic change was observed in Microsoft's Windows server performance under the Smurf flood attack compared to its previous performance under PING flood attack. In this scenario, the baseline server performance of the number of legitimate connections fell sharply as the Smurf attack traffic increased beyond 100 Mbps. All legitimate client connections were lost at 150 Mbps of Smurf attack traffic, which is a relatively

low attack bandwidth compared to 1000 Mbps or 1 Gbps being common these days. No legitimate client connections could be established with the Microsoft's server OS running on the same hardware platform from Apple Inc. (Figure 2.4) for Smurf traffic higher than 150 Mbps.

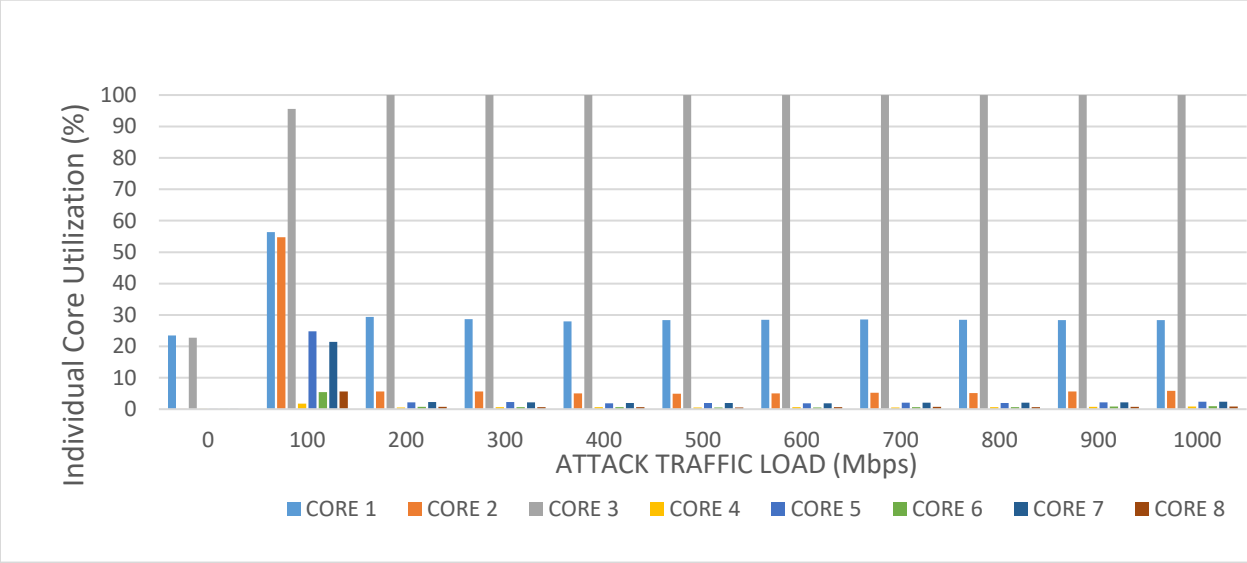

Figure 2.5 Individual Core Utilization under Smurf Attack when sent to the Victim Server on Windows 2012 R2 OS on Apple Server Platform

This seemed quite unusual in the beginning knowing the fact that the server hardware deployed 8 core processors but the whole server system became unresponsive under relatively small volume of Smurf attack traffic of 150 Mbps. Further analysis of the core utilization showed that one of the core maxed out and other cores didn't share the excess load of the Smurf flood. It was not clear if it was due to the inability of the Window's server OS in handling the Smurf flood or was it due to the inability of the Apple's hardware platform in sharing the excess load. In one of the literatures issued by Apple Inc [62], Apple gave a statement saying "It's not possible to split a single thread across multiple cores, although a single core may run multiple threads at the same time. This is one reason that you may sometimes see uneven load distributions across the available cores on your computer".

2.3.1.3 TCP-SYN Flood Attack

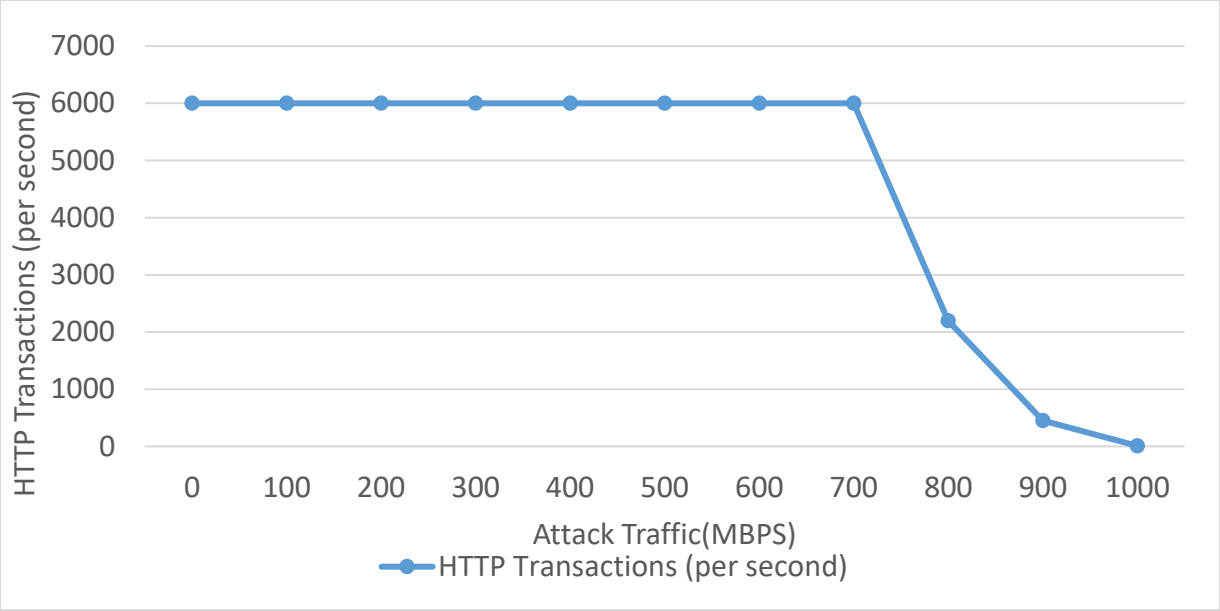

Figure 2.6 Number of HTTP connections established by the server under TCP-SYN Flood Attack on Windows 2012 R2 OS on Apple Server Platform

In scenario, the TCP-SYN flood attack was used to evaluate Windows Server OS 2012 R2 on the same server hardware platform from Apple Inc. It remains full 6000 HTTP connections until 700 Mbps of TCP-SYN flood attack. At 800 Mbps of TCP-SYN flood attack, the number of HTTP transactions per second were start declining and HTTP connections continued to decrease with increased attack traffic. Coming to Individual processor utilization, there were only two logical cores that were sharing processor utilization from the baseline to 700 Mbps of attack traffic. And later four cores start sharing the processor utilization because of HTTP transactions were started declined at 800 Mbps of TCP-SYN flood attack traffic.

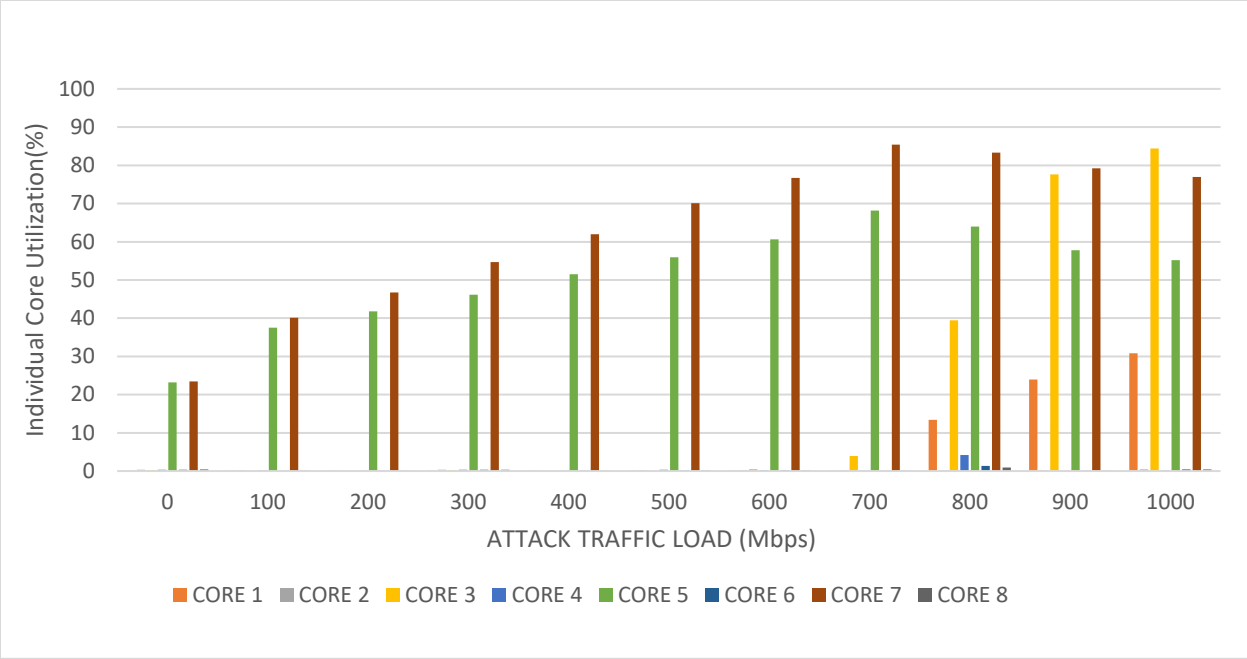

Figure 2.7 Individual Core Utilization under TCP-SYN Flood Attack when sent to the Victim Server on Windows 2012 R2 OS on Apple Server Platform

2.3.2 MAC OS on MAC server

2.3.2.1 Ping Flood Attack

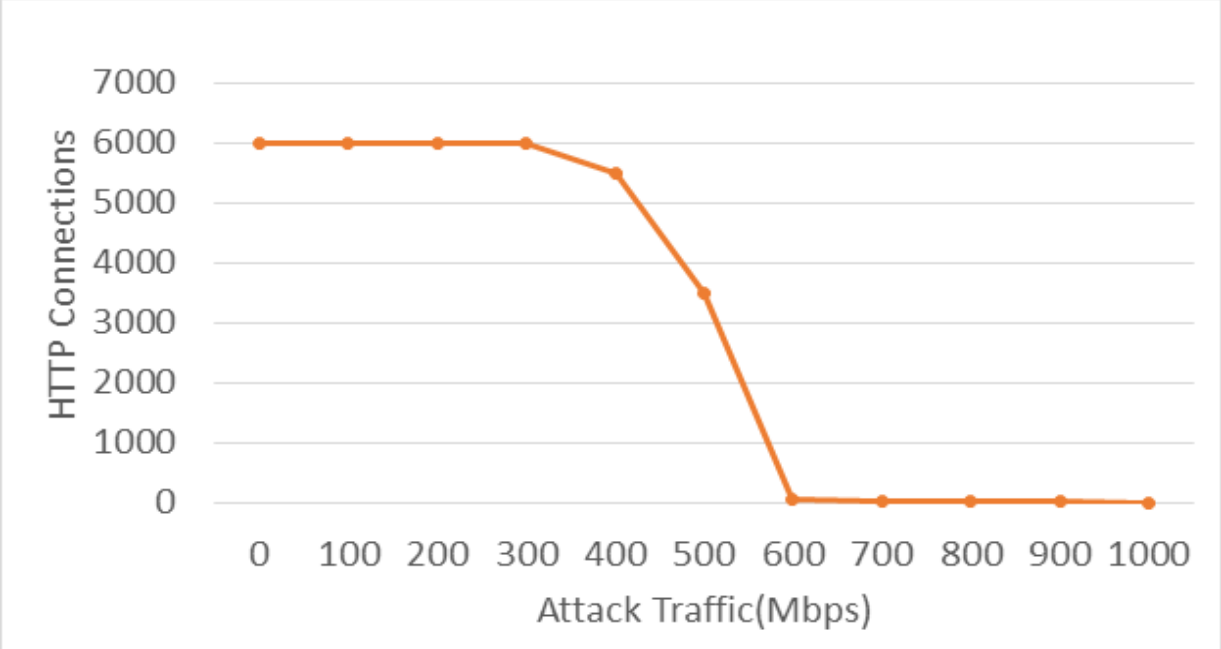

Figure 2.8 Number of HTTP connections established by the server under Ping Flood Attack on Mac OS on Apple Server Platform

For this scenario, we used the Apple's native MAC OS for the same Apple's server hardware platform. Comparatively, the Mac OS results were found to be different from that of Windows Server 2012 R2 for the same hardware platform. Baseline performance could be maintained till 500 Mbps of the PING flood. A significant decline in the number of legitimate connections was found at 600 Mbps supporting only 50 legitimate connections under Ping attack (Figure 2.8). This kind of significant decline in the legitimate connections was found to be at 800 Mbps for Windows Server 2012 R2 OS on Apple's hardware server platform. Inferring from the performance data, it showed that the Microsoft's Windows Server 2012 R2 was performing better than Apple's Mac OS on its native Apple.

2.3.2.2 Smurf Attack

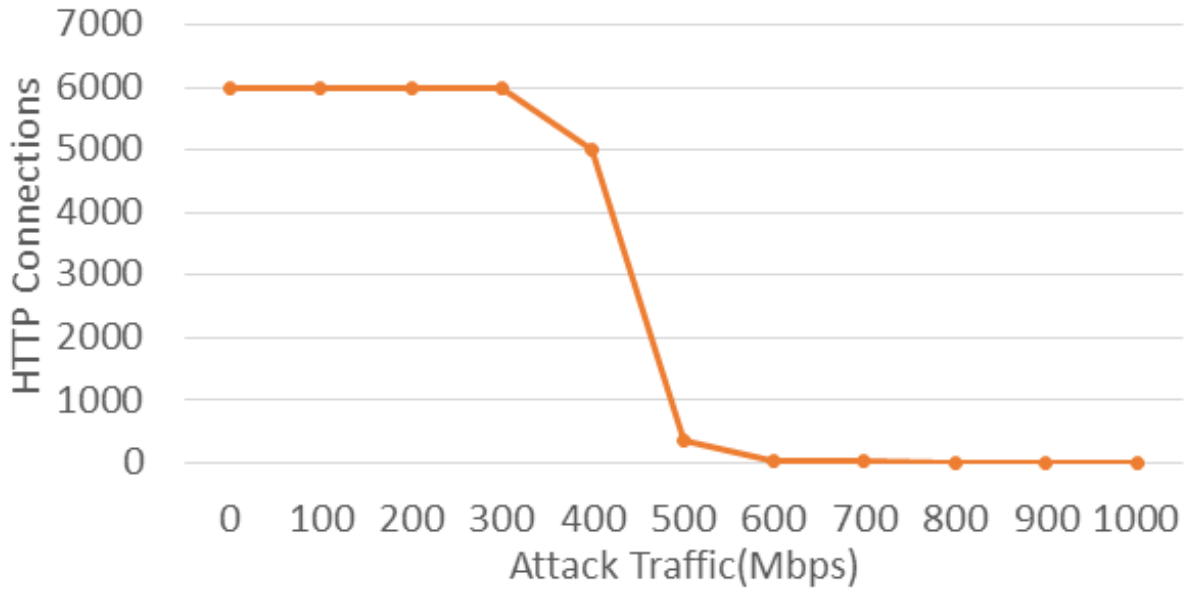

Figure 2.9 Number of HTTP connections established by the server under Smurf Attack on Mac OS on Apple server platform

In this scenario, we used native Mac OS on the same Apple's server hardware platform. A Smurf attack on Mac OS produced relatively improved resilience of the server compared to the crashing of Windows Server 2012 R2 at 150 Mbps of the smurf attack load. Compared with Windows OS, Mac OS could sustain the Smurf attack till 300 Mbps by supporting the baseline performance. When the attack traffic increased, the number of legitimate connections started declining, and all legitimated connections were completely lost after the attack traffic increased beyond 500 Mbps (Figure 2.9).

2.3.2.3 TCP-SYN Flood Attack

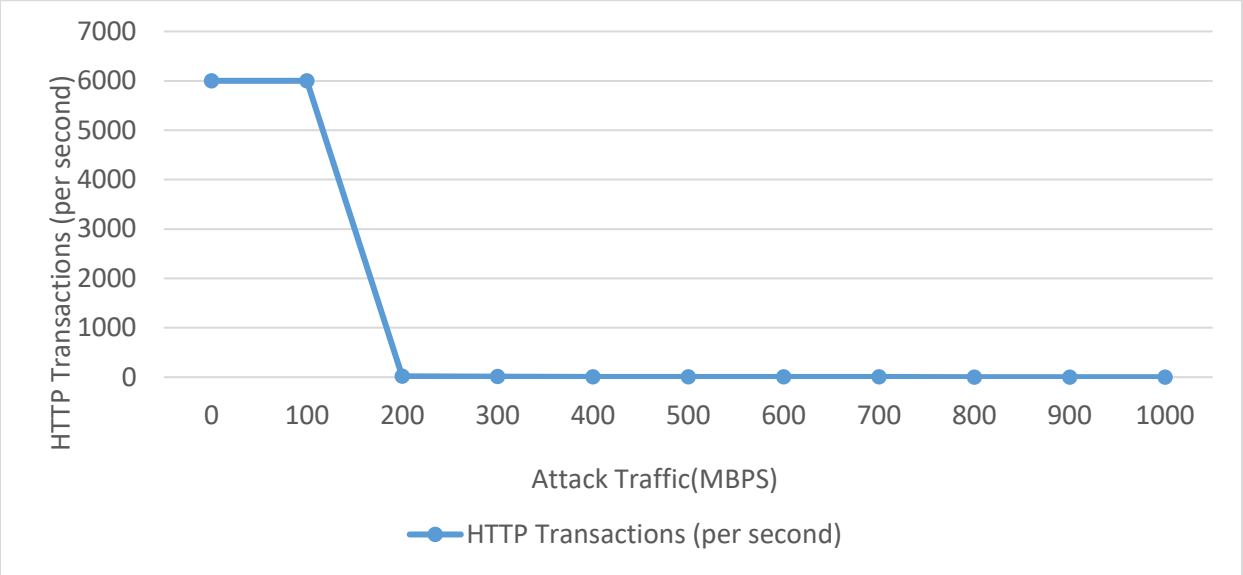

Figure 2.10 Number of HTTP connections established by the server under TCP-SYN Flood Attack on Mac OS on Apple Server Platform

In this scenario, we used native Mac OS on the same Apple's server hardware platform. A TCP-SYN attack on Mac OS behaves like Smurf attack on Windows 2012 Server R2 OS on Apple server platform. Above graph shows at 200 Mbps of TCP-SYN attack traffic, the number of HTTP transactions are almost becoming zero. Whereas at 200 Mbps of TCP-SYN attack traffic on Windows Server 2012 R2 OS on Apple server platform had full 6000 HTTP connections.

This shows that, TCP-SYN attack on Windows Server 2012 R2 OS on Apple server platform is having better security than TCP-SYN attack on Mac OS on Apple Server platform.

2.3.3 Comparison of results

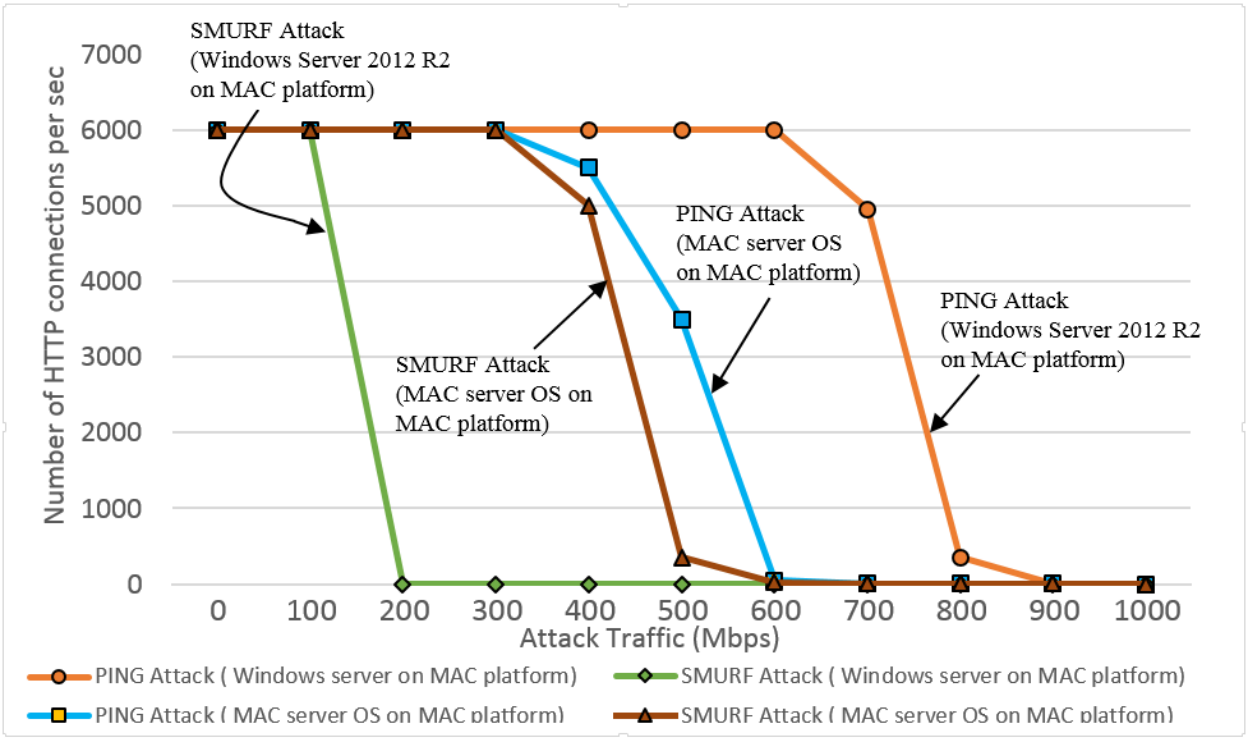

Figure 2.11 Comparison of legitimate HTTP Connections supported by different configurations.

It is important to compare the performance of different servers under different types of ICMP attacks to obtain a better picture of protection provided by these leading server platforms. Comparative performance is shown in Figure 2.11 for two server OS under two different types of ICMP based attacks. Under Ping attack, the Microsoft's Windows Server OS 2012 R2 on Apple's server hardware performs better than Mac LION OS on its own native Apple server hardware. It is found that for the Microsoft's Windows OS, the number of legitimate connections start declining from its baseline of 6000 connections for attack traffic higher than 600 Mbps. However, for Mac OS on the same Mac hardware platform, the number of legitimate connections

starts declining from its baseline of 6000 connections when the Ping flood intensity exceeds 300 Mbps. Under Smurf attacks, the Microsoft's Windows server OS on MAC hardware platform is found to crash at relatively low Smurf attack intensity of 150 Mbps. However, under the Smurf attack, the Apple's MAC LION OS performs much better on the same Apple's Mac Pro hardware platform. The MAC OS lost all legitimate connections but at much higher attack traffic i.e. 600 Mbps. comparatively, under Smurf attack traffic, Mac OS on Apple's server hardware platform shows higher survivability compared to that for Windows Server OS 2012 R2 on Apple's server hardware platform.

2.4 Chapter Summary

It is observed that different server operating systems perform differently under different types of ICMP based flood attacks. Windows Server 2012 R2 is one of the most popular server used today, hence even though Apple server platform has its own operating system, it is common to use Windows Server 2012 R2 operating system on Apple Server hardware platform. It is shown in this paper, the Microsoft's Windows Server OS performed better in term of survivability (number of legitimate connections supported under attack) when compared with that of Apple's Server OS under Ping based ICMP attack traffic. However, under Smurf based ICMP attack, the Window's Server OS crashed at a relatively low Smurf traffic of 150 Mbps. For the same smurf attack the Apple's Server OS survived under the same scenario of 150 Mbps. However, it also dropped all legitimate connections rather at higher Smurf traffic intensity. The results presented in this paper show that the built-in protection mechanism of Windows Server 2012 R2 is not effective on its own against a SMURF flood attack. We conclude that both server OS need to deploy more efficient protection mechanisms especially against ICMP based Cyber-attacks without depending on external security devices.

CHAPTER III

EFFECT OF DDoS ATTACKS ON WINDOWS 2012 R2 OS ON MAC HARDWARE PLATFORM WITH 4 PORT BROADCOM NIC ADAPTER

In this chapter, I had installed 4-port Broadcom gigabit ethernet adapter on Mac server Hardware platform. And I evaluate the performance of Mac server with windows server 2012 R2 operating system under different DDoS attacks.

In the experiment, simulated DDoS attack traffic is sent to the victim server simultaneously from multiple networks. In the process of evaluating the impact of the attack in an organization-like environment, legitimate or client traffic is also sent to the server simultaneously along with attack traffic [40]. We measured the processor utilization, memory utilization and HTTP transactions for different loads of attack traffic ranging from 0 Mbps to 1Gbps over a gigabit Ethernet link connected to the victim computer [41]- [42].

The Distributed Denial of Service(DDoS) attacks were performed using the experimental set up is shown in Figure.2. The victim server is an Apple MAC PRO, Two 2.4GHz Quad-Core Intel Xeon E5620 “Westmere” processors server, 8 logical processor and 12 GB RAM [60]-[61]. As mentioned earlier, Windows 2012 R2 Standard Operating System and Apple MAC OS X SERVER LION 10.7.5 (11G63) has been installed in the victim server. Because we were going to compare these two operating systems in terms of HTTP transaction rate, CPU utilization and memory utilization response to Different DDoS attacks. The only protection mechanism that was active on the server platform was firewall in the both operating systems.

3.1 Experimental Setup

The experimental set up was 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).

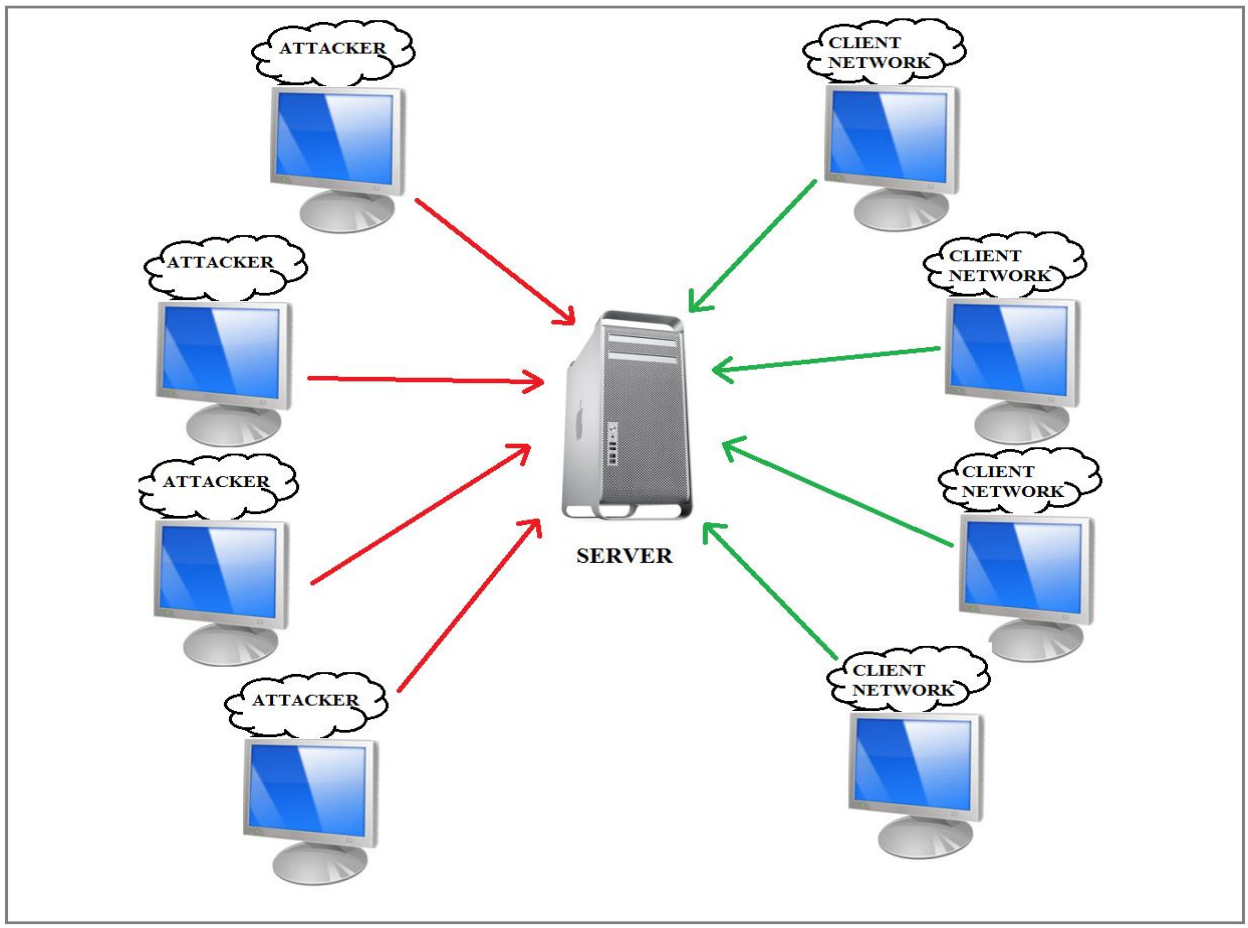

Figure 3.1 Experimental Setup

In order to communicate with the client, first a sample webpage namely Index.html was created in the victim server with support of IIS service [63]. And then, this sample webpage was accessed through the Hyper Text Transfer Protocol (HTTP) request from a Client. The victim server was responds to the clients whenever server received a HTTP request from client. In this thesis, I created a controlled environment requesting web server request to the

victim server. I calculated server capacity by means of HTTP connections per second, CPU utilization, memory utilization and non-paged pool allocations.

The apple server consists of two port inbuilt Gigabit Ethernet adapters. And I installed 4-port Broadcom Gigabit Ethernet adapter to the same apple server. As I mentioned earlier, I was tested the server with two inbuilt Gigabit Ethernet ports in the previous chapter.

In this chapter, I used two different networks such as Class B and Class C networks. Class B network has consisted of 16 network bits and 16 host bits. This Class B address start with "10" and 128.0.0.0 was starting address and 191.255.255.255 was its ending address. And its default subnet mask is 255.255.0.0. Class B network having 16,384 networks and 65,536 hosts per network. So, in total 1,073,741,824 possible addresses were available in Class B network.

Class C network has consisted of 24 network bits and 8 host bits. This Class B address start with "110" and 192.0.0.0 was starting address and 223.255.255.255 was its ending address. And its default subnet mask is 255.255.255.0. Class C network having 2,097,152 networks and 256 hosts per network. So, in total 536,870,912 possible addresses were available in Class C network.

Distributed Denial of service(DDoS) attacks were sent to the victim server in three different scenarios with two different networks such as Class B and Class C in this chapter. 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 3000 HTTP requests per second to the server.

In the first scenario, the attack traffic was sent to one out of the four external ports in the server. First, to measure a baseline of the experiment by sending five minutes of legitimate

traffic in the form of HTTP requests without any attack traffic. Once the baseline of the experiment established, the attack traffic was introduced to the victim server. Initially, attack traffic intensity of 100 Mbps was applied from simulated attack network for five minutes. Later, it was increased to 200 Mbps of attack traffic for five minutes and this process will continue until to reach 1000 Mbps maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 100 Mbps of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes.

In the second scenario, the attack traffic was sent to two out the four external ports in the server. First, to measure a baseline of the experiment by sending five minutes of legitimate traffic in the form of HTTP requests without any attack traffic. Once the baseline of the experiment established, the attack traffic was introduced to the victim server. Initially, attack traffic intensity of 200 Mbps was applied from simulated attack network (100 Mbps from each external Gigabit Ethernet adapter) for five minutes. Later, it was increased to 400 Mbps of attack traffic for five minutes and this process will continue until to reach 2000 Mbps maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 200 Mbps of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes.

In the third scenario, the attack traffic was sent to all four external ports in the server. First, to measure a baseline of the experiment by sending five minutes of legitimate traffic in the form of HTTP requests without any attack traffic. Once the baseline of the experiment established, the attack traffic was introduced to the victim server. Initially, attack traffic intensity of 400 Mbps was applied from simulated attack network (100 Mbps from each external Gigabit Ethernet adapter) for five minutes. Later, it was increased to 800 Mbps of attack traffic for five minutes and this process will continue until to reach 4000 Mbps maximum bandwidth of Gigabit

Ethernet adapter by increasing regular interval of 400 Mbps of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes.

3.2 Performance parameters for Evaluation

In this experiment, the parameters that are used to evaluate the performance were Memory utilization, CPU utilization, Non-paged pool allocation and HTTP transactions per second. Some of these parameters that were collect from Performance monitor present in that particular Operating system. In performance monitor, click on “Data Collector Sets” option, then select “User Defined” to create new data collector set. And then we can create manually by selecting those performance parameters.

CPU Utilization (CPU Usage in %): % Processor Time is 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). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It should be noted that the accounting calculation of whether the processor is idle is performed at an internal sampling interval of the system clock (10ms). On today’s fast processors, % Processor Time can therefore underestimate the processor utilization as the processor may be spending a lot of time servicing threads between the system clock sampling intervals. Workload based timer applications are one example of applications which are more likely to be measured inaccurately as timers are signaled just after the sample is taken. The Processor utilization is amount of usage to the total central processing unit (CPU). This will evaluate whether that attack traffic is effect on the CPU. If CPU

utilization is more, that attack traffic is CPU intensive attack. The name of counter that is used to evaluate processor utilization is known as \Processor (_Total) \% Processor Time.

The Processor performance object consists of counters that measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The processor object represents each processor as an instance of the object. This MAC PRO server has 8 logical processors, hence the counters that were used to monitor the multi core utilization of the server are \Processor (0) \%Processor Time, \Processor (1)\%Processor Time, \Processor (2)\%Processor Time, \Processor (3)\%Processor Time, \Processor (4)\%Processor Time, \Processor (5)\%Processor Time, \Processor (6)\%Processor Time, \Processor (7)\%Processor Time.

Memory Utilization (RAM Usage in MBytes): Available MBytes is the amount of physical memory, in Megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free and zero-page lists. The memory utilization is the amount of RAM usage with respect to total random-access memory available assigned to that particular operating system. If the memory utilization is more then we can say that this attack is called Memory intensive attack. The name of the counter that is used to evaluate memory utilization is known as \Memory\Available MBytes.

Non-paged pool allocation: Pool Nonpaged Allocs is the number of calls to allocate space in the nonpaged pool. The nonpaged pool is an area of system memory area for objects that cannot be written to disk, and must remain in physical memory if they are allocated. It is measured in numbers of calls to allocate space, regardless of the amount of space allocated in each call. This

counter displays the last observed value only; it is not an average. The name of the counter that is used to evaluate Non-paged pool Allocation is known as \Memory\Pool Nonpaged Allocs.

HTTP transaction per second: This HTTP transactions are referred to the number of legitimate connections established by the server. This parameter will give the number of connections per second established by the server for different amount of attack traffic ranging from 1 Mbps to 1 Gbps. This parameter helps in determine whether the server has reached to its saturation point.

Connection Latency: 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.

3.3 Results and Discussion

3.3.1 Ping Flood Attack

Ping based DDoS attacks are flood of a large number of ping messages sent to target are known to be quite damaging to the availability of the web based services. The ping attack can exhaust the target server's bandwidth and computing resources. Class C network's Ping attack was sent to three different scenarios as I mentioned earlier.

3.3.1.1 Class C Network Ping Flood Attack

Class C network has 256 hosts per network. In this experiment, I used one class C network that generates 256 networks.

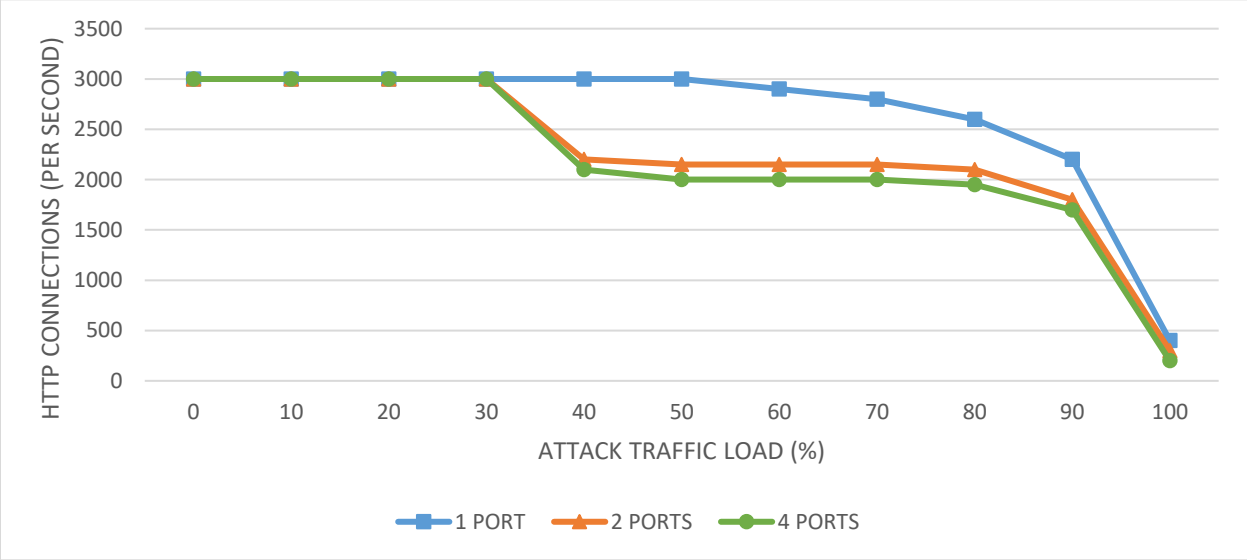

Figure 3.2 Number of HTTP connections established by the server under Ping Flood Attack when sent from Class C Network

The above Figure 3.2 shows, the number of HTTP connections per second under class C of Ping flood attack traffic. There was no much effect on HTTP connection rate in the first scenario until 900 Mbps of attack traffic and there were approximately 400 HTTP connections per second at

100% of attack traffic load. In 2 ports and 4 ports scenario, the connection rate was declined at 40% of Ping flood attack traffic and the connection rate was 200 at 100 % of attack traffic load.

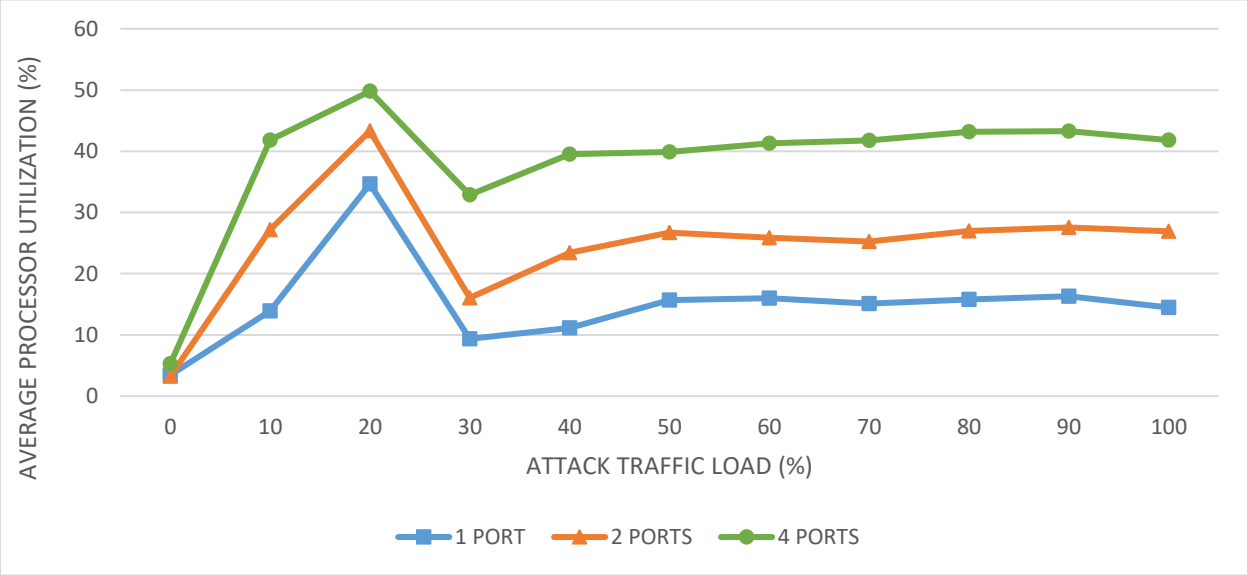

Figure 3.3 Average Processor Utilization under Ping Flood Attack when sent from Class C Network

The Average processor utilization of three different scenarios under Class C network Ping attack traffic was approximately 5 % of total processor usage at baseline of no attack traffic as shown in Figure 3.3. Interestingly, at 20 % of Ping attack traffic the average processor utilization was reached to its maximum usage throughout the experiment in all three scenarios. It was reached to 50 % of processor utilization at 800 Mbps (20% of 4 Gbps) of attack traffic in 4 ports scenario. The Figure 3. 4 shows the number of non-paged pool allocations under class C network Ping flood attack in all three scenarios. The number of Non-paged pool allocations were approximately same in all three scenarios until 40% of Ping attack traffic load. When server reached to 4 Gbps of Ping attack traffic, the number of non-paged pool allocations were approximately 217000. The number of non-paged pool allocations were increasing while increasing the number of ports as shown in Figure 3.4.

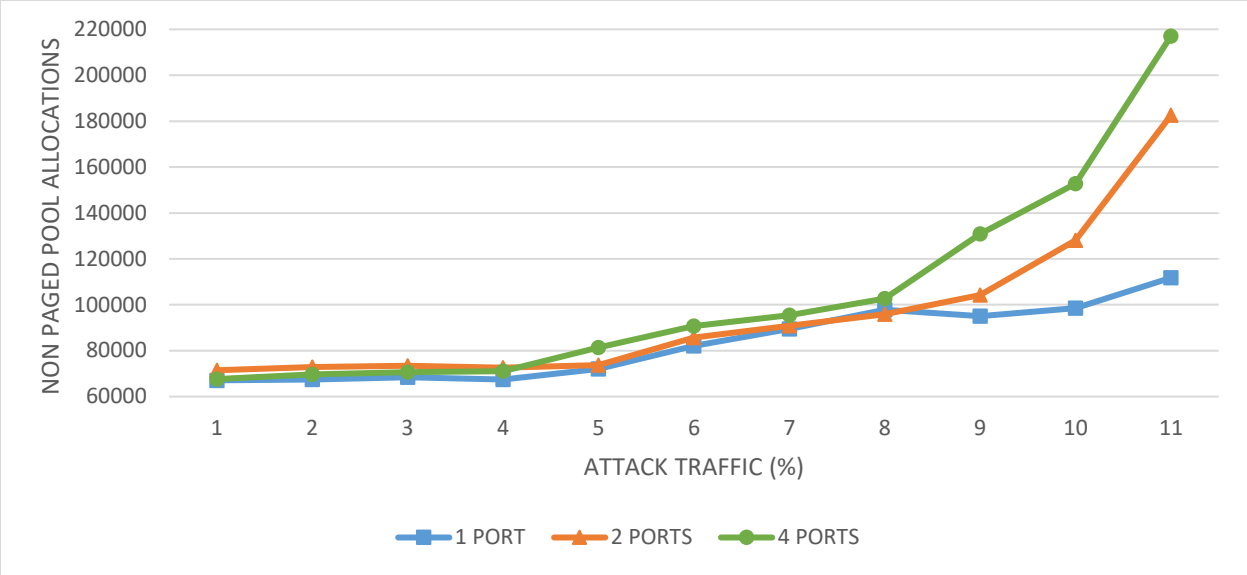

Figure 3.4 Number of Non-Paged Pool Allocations under Ping Flood Attack when sent from Class C Network

3.3.1.2 Class B network Ping Flood Attack

Class B network has 65,536 hosts per network. In this experiment, I used one class B network that generates 65,636 different networks.

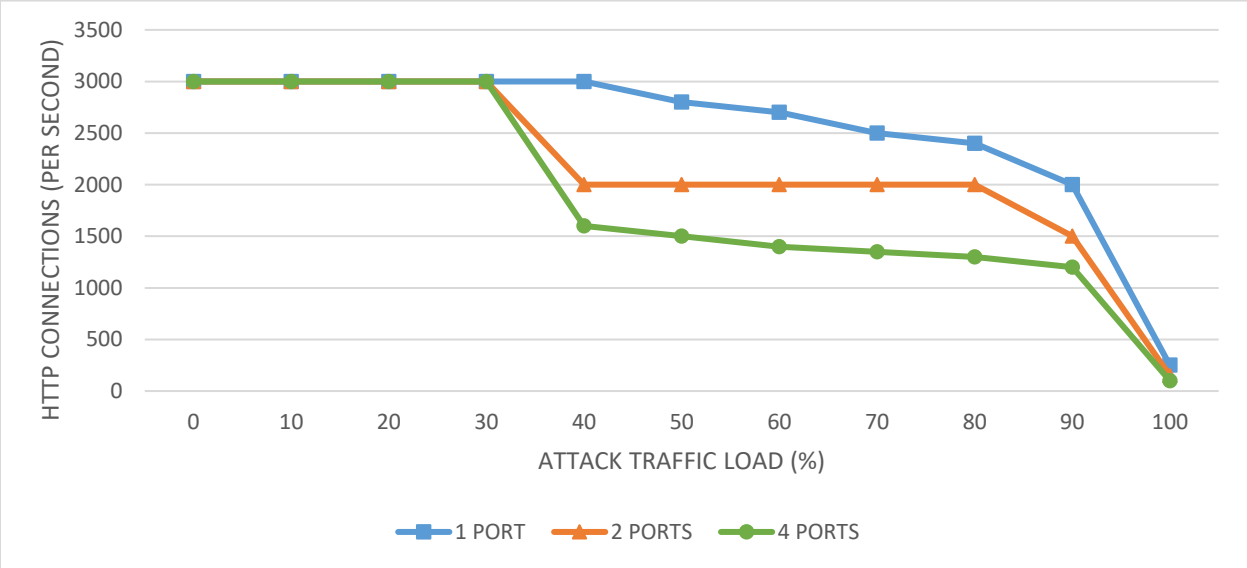

Figure 3.5 Number of HTTP connections established by the server under Ping Flood Attack when sent from Class B Network

The above Figure 3.5 shows the HTTP connection rate under class B network of Ping flood attack. The number of HTTP connections per second were maximum of 3000 until 30% of Ping attack traffic in case of all three scenarios. In the first scenario, the HTTP connections per second were started declining at 50% and it was reached approximately 250 HTTP connections at 1000 Mbps (100% of 1Gbps) of Ping flood attack. In 4 ports scenario, the HTTP connection rate was sharp declined at 40% of attack traffic load and there were about 100 HTTP connections per second at maximum attack traffic load.

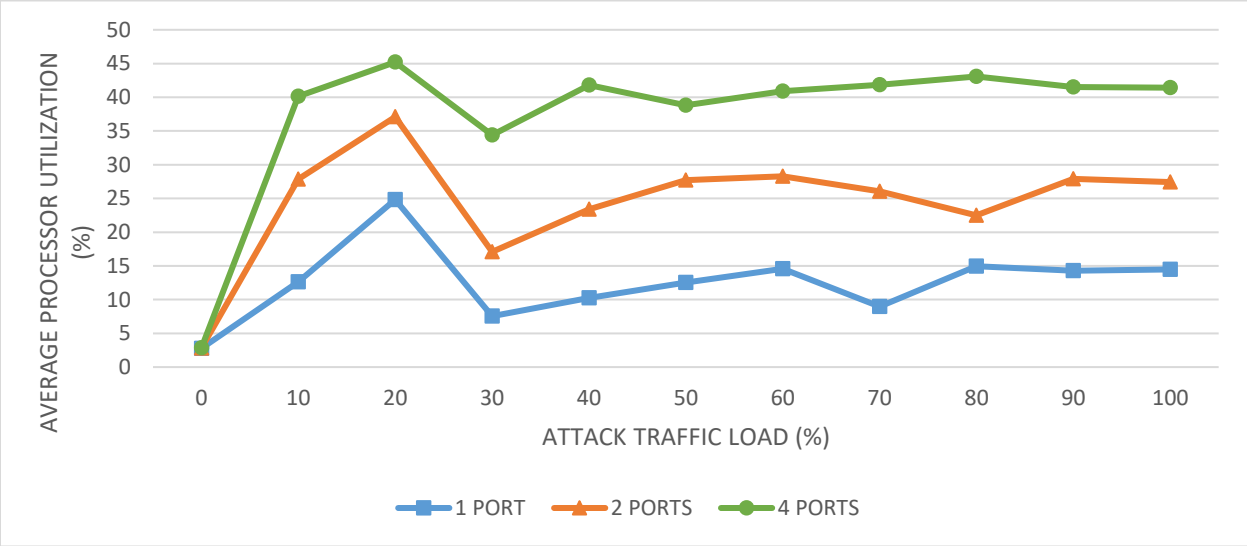

Figure 3.6 Average Processor Utilization under Ping Flood Attack when sent from Class B Network

The above Figure 3.6 shows the average processor utilization under class B network Ping flood attack traffic in all three scenarios. When ping attack traffic sent to single port, approximately 15% of total processor utilized. And then I sent ping attack traffic to two ports, its increased to 25%. And alter I sent ping attack traffic to all four ports, its used approximately 40% of total processor. This clearly shows amount of total processor utilization increases by increasing number of attacking ports.

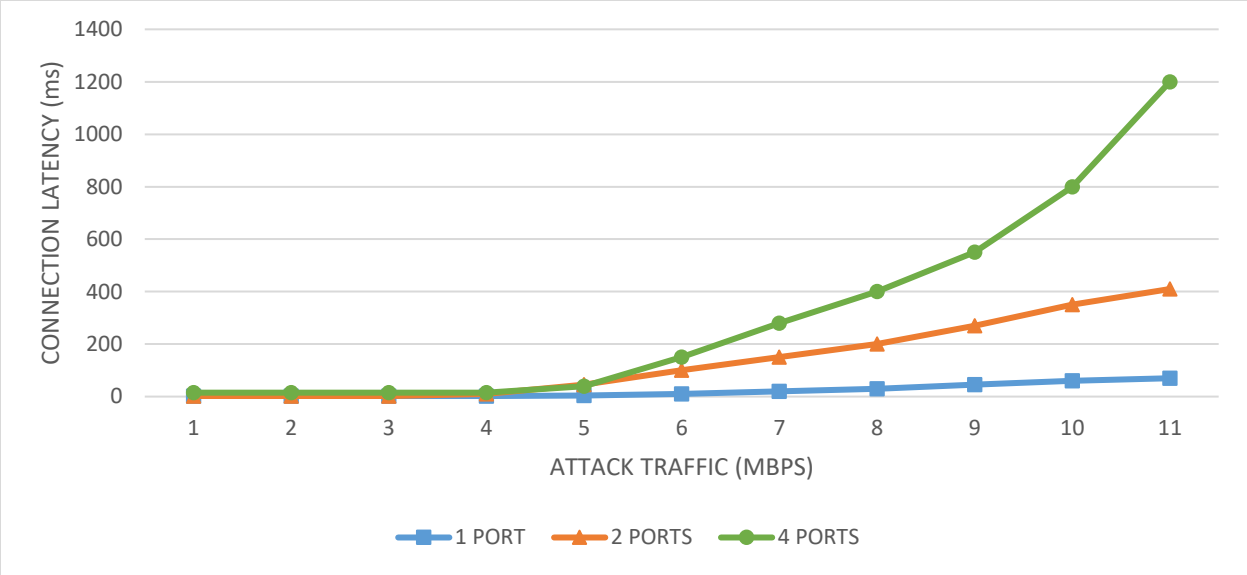

Figure 3.7 HTTP Connection Latency under Ping Flood Attack when sent from Class B Network

The connection Latency of the victim server shown above Figure 3.7 is used to evaluate the ability of the server to handle the class B network Ping flood attack traffic. At 50 % of Ping attack traffic, all three scenarios the connection latency was almost zero. In 4 ports scenario, connection latency was reached to 1200 milliseconds at 4 Gbps of Ping attack traffic. Whereas 1 port scenario, the connection latency was reached to 70 milliseconds. This clearly shows, HTTP connection latency was increasing with increasing number of ports.

3.3.1.3 Comparison of Class B with Class C Networks under Ping flood attack

The HTTP connection rate was affected more in Class B Ping flood attack traffic, because of number of different networks generated by class C. If it is large network, the server will have to respond more times. As a result, the HTTP connection rate was more affected with class B network. I observed, there were 100 HTTP connections at 4 Gbps (100% of 4Gbps) of Ping attack traffic with class B netwok whereas in class C, there were 200 connections at 4 Gbps of attack traffic.

3.3.2 Smurf Attack

A more sophisticated version of a DDoS attack is commonly known as a SMURF attack. A SMURF attack utilizes massive number of ICMP packets of spoofed source Internet Protocol (IP) addresses targeting the victim server's IP address This is achieved by altering the Echo Request sent to the botnet using an IP broadcast address. The larger the Botnet is the faster and the bigger is the flood of Echo reply messages. The increase of traffic reduces the target server's ability to respond, and can quickly cause a complete denial of service.

3.3.2.1 Class C network Smurf Attack

The below Figure 3.8 shows the HTTP connection rate under class C Smurf attack traffic in all three scenarios. In the first scenario, the HTTP connections were started declining at 300 Mbps of attack traffic. The connection rate was reached t0 1750 at 900 Mbps of attack traffic and then it declined to approximately 400 connections per second.

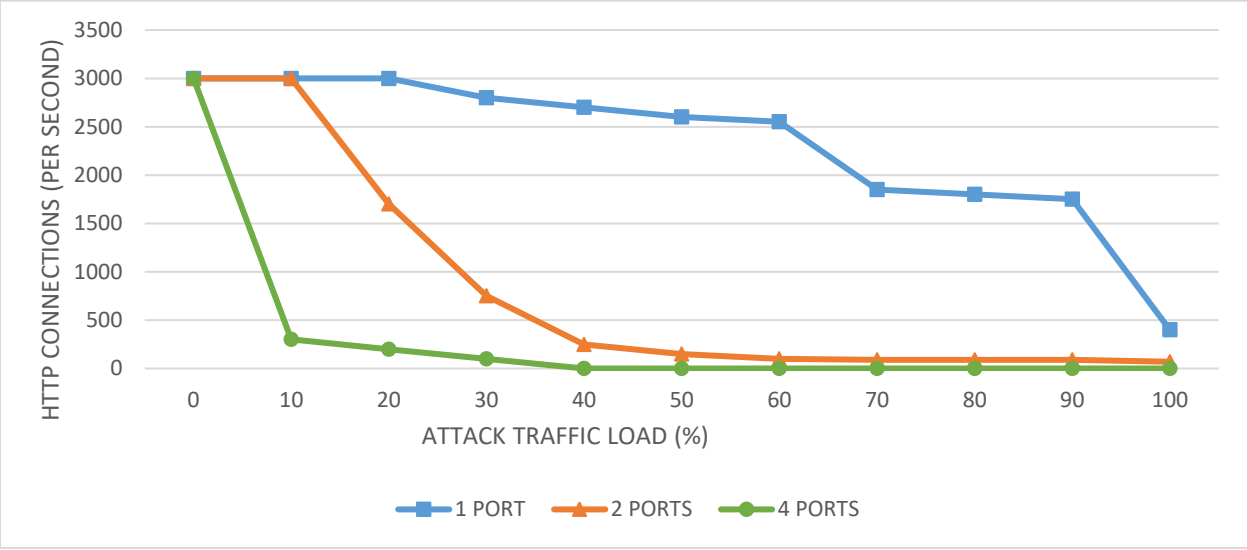

Figure 3.8 Number of HTTP connections established by the server under Smurf Attack when sent from Class C Network

In 2 ports scenario, the number of HTTP connections were started decling at 200 Mbps of attack traffic and at 1200 Mbps (60% of 2 Gbps) of Smurf attack traffic there was almost no connections. In 4 ports scenario, there were only 200 HTTP connections at 400 Mpbs of attack and there were no HTTP connections at only 40% of attack traffic load.

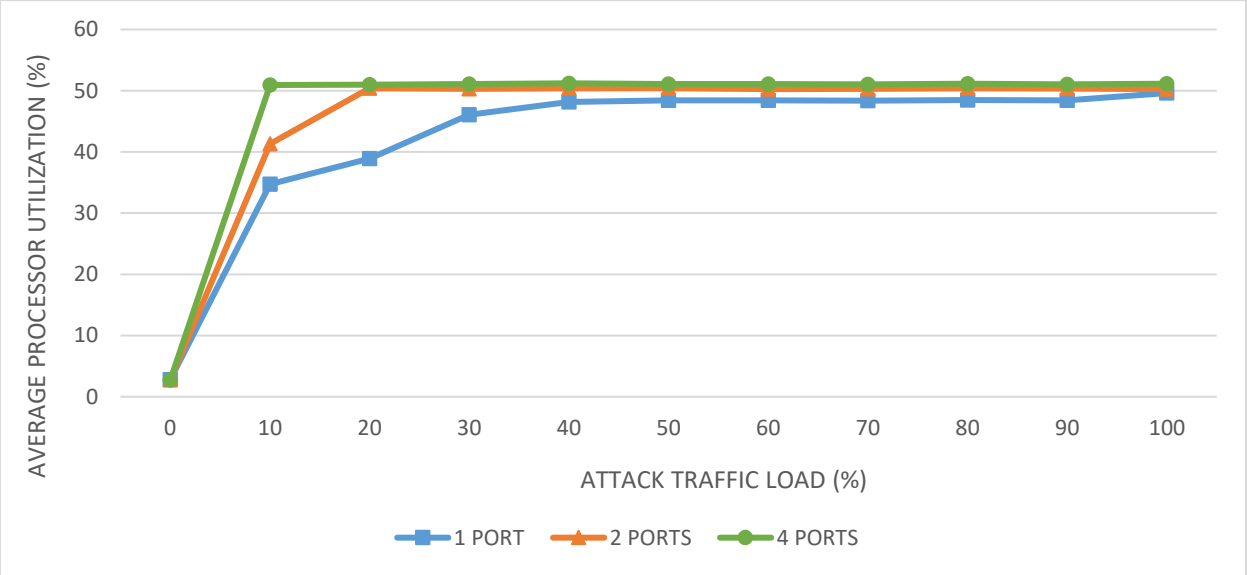

Figure 3.9 Average Processor Utilization under Smurf Attack when sent from Class C Network

The above Figure 3.9 shows the average processor utilization under class C network Smurf attack in all three scenarios. The average processor utilization in single port scenario was approximately 35% at 100 Mbps of attack traffic. And it was reached to 48% at 400 Mbps of attack traffic load and maintained same for rest of the experiment. In other two scenarios, the average processor utilization was almost 50% once the Smurf attack traffic was introduced.

3.3.2.2 Class B network Smurf Attack

The Figure 3.10 shows the HTTP connection rate under Class B network Smurf attack traffic under one port and four port scenarios. The number of HTTP connections was started declining at 30% of attack traffic load in the single port scenario and the connection rate was keep declining while increasing the attack traffic load. Approximately there were 200 HTTP connections per second established in single port scenario at 1000 Mbps of attack traffic load. Whereas in 4 ports scenario, the HTTP connection was started declining at 400 Mbps of Smurf attack traffic and reached to 250 HTTP connections per second. And at 800 Mbps (20% of 2 Gbps) of Smurf attack traffic load, there were no HTTP connections established by the server.

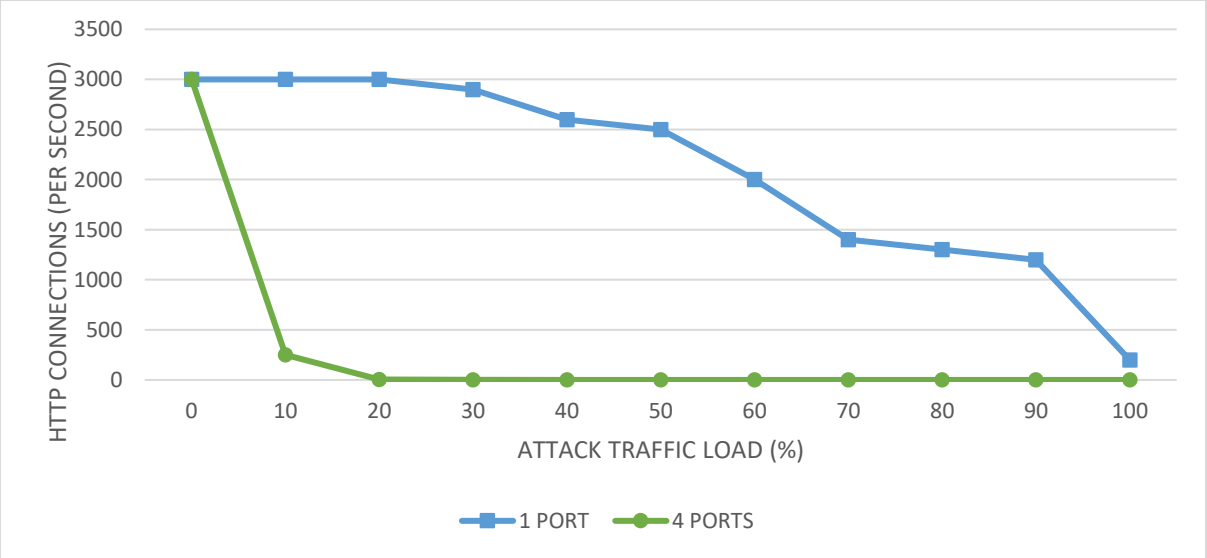

Figure 3.10 Number of HTTP connections established by the server under Smurf Attack when sent from Class B Network

The average processor utilization under class B network Smurf attack was similar to class C and there was no much difference. The Figure 3.11 shows the non-paged pool allocations under class C network Smurf attack in one port and four port scenarios. It shows that the non-paged pool allocations were decreasing while increase the number of ports.

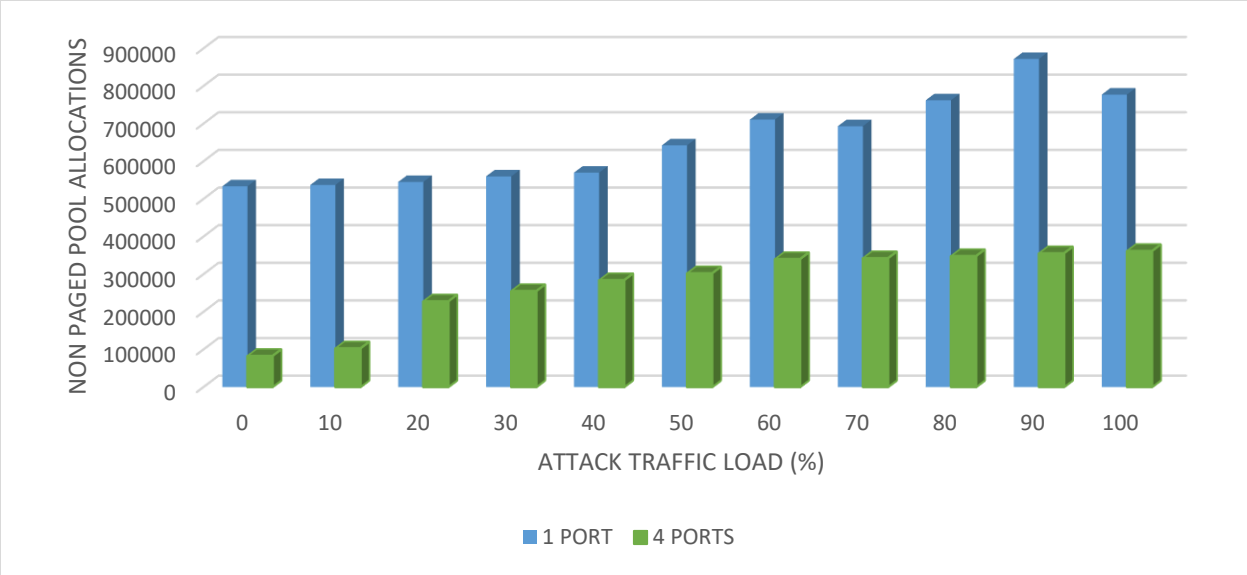

Figure 3.11 Number of Non-Paged Pool Allocations under Smurf Attack when sent from Class B Network

The HTTP connection latency under class B network Smurf attack in case of all three scenarios was shown in Figure 3.12. Connection latency was increasing with the increase of attack traffic in all three scenarios. 9500 milliseconds of connection latency were registered in case of 4 ports scenario at 100% of Smurf attack traffic load.

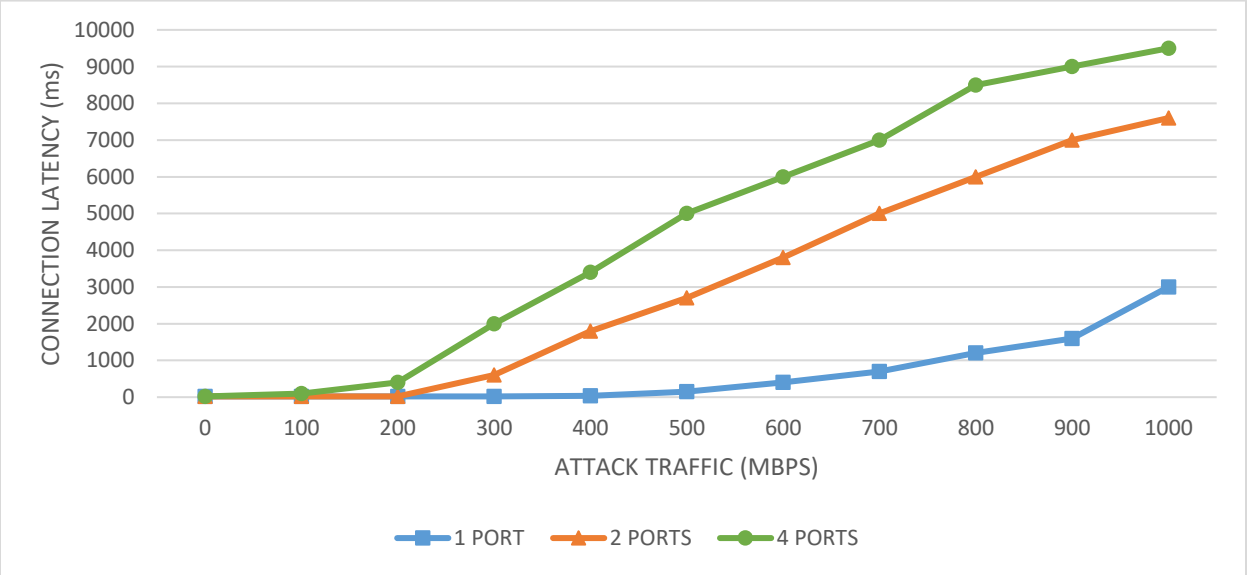

Figure 3.12 HTTP Connection Latency under Smurf Attack when sent from Class B Network

3.3.3 TCP-SYN Flood Attack

The Transfer Control Protocol (TCP) is a connection oriented protocol which is belongs to layer 4 of OSI reference model. TCP uses a three-way handshake to establish a network connection. Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. In TCP_SYN attack, attacker won't respond for acknowledgement. Which results, the number of half open connections will exist, and server will wait for some time to receive acknowledgement from attacker. This will keep server busy all the time. Now will see how this TCP-SYN flood attack will affect Windows Server 2012 R2 operating system on MAC hardware in case Class B and Class C network.

3.3.3.1 Class C network TCP-SYN Flood Attack

The Figure 3.13 shows the HTTP connection rate established against class C TCP-SYN flood attack in all three different scenarios. This clearly shows the connection rate was decreasing in all three scenarios.

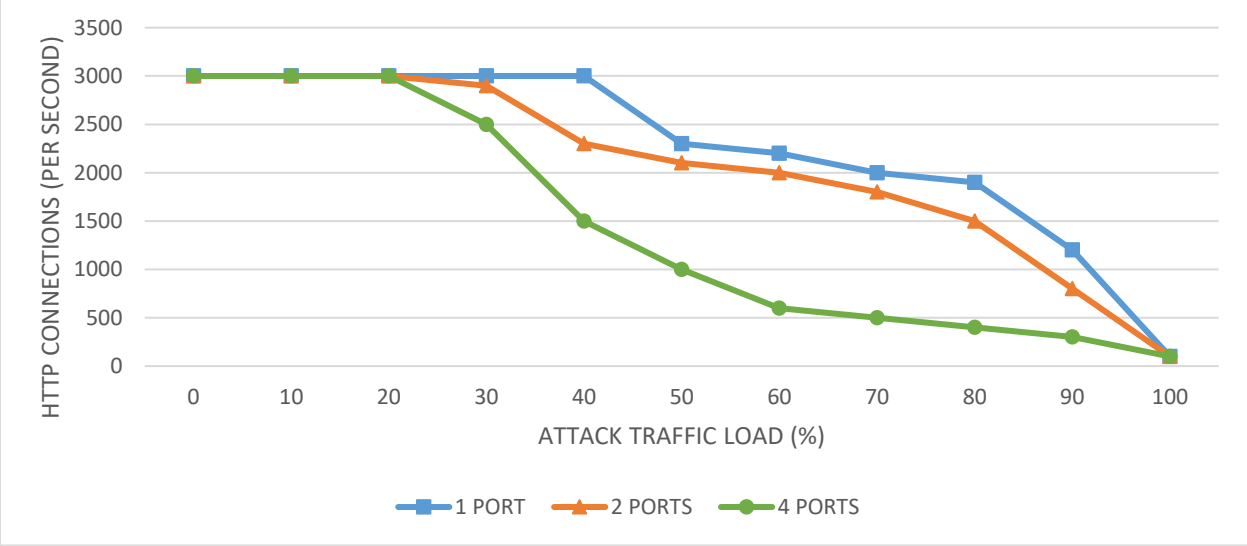

Figure 3.13 Number of HTTP connections established by the server under TCP-SYN Flood Attack when sent from Class C Network

In the first scenario, the HTTP connection rate was started declining at 50% of TCP-SYN attack traffic load and it was keep declining at higher attack traffic. In the second scenario, the HTTP connection rate was started declining at 40% of attack traffic load. In third scenario, the connection rate was started declining at only 30% of attack traffic load. At 100% of TCP-SYN attack traffic, the HTTP connection rate was approximately zero in all three scenarios.

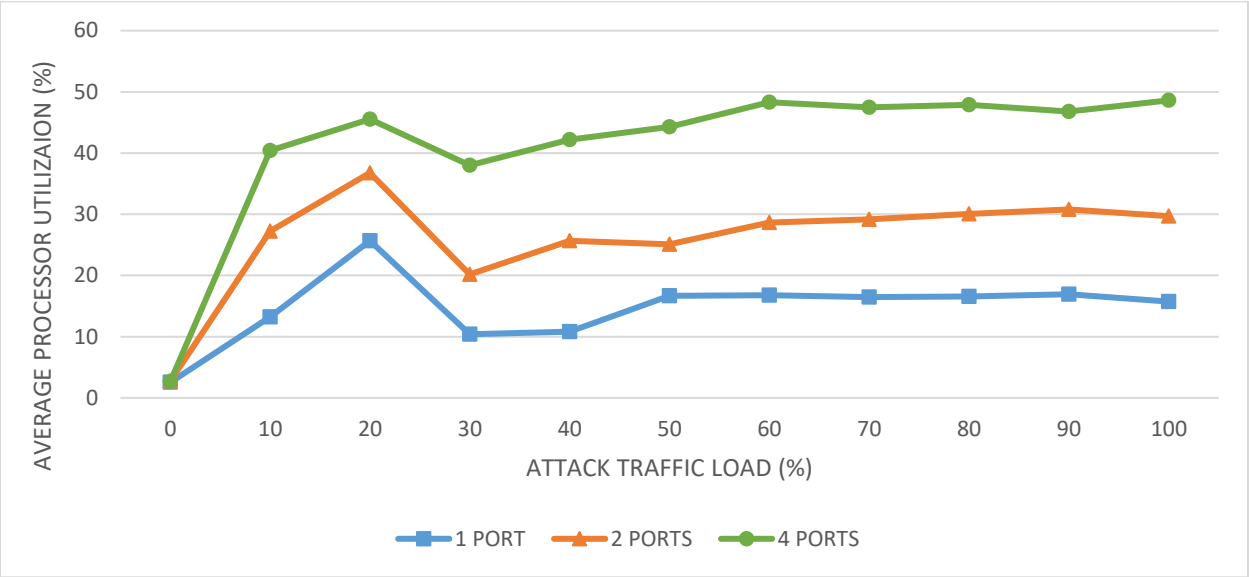

Figure 3.14 Average Processor Utilization under TCP-SYN Flood Attack when sent from Class C Network

The Figure 3.14 shows the average processor utilization under class C TCP-SYN attack traffic in three scenarios. The average processor utilization in the first scenario was approximately 17% and 20% in the 2 ports scenario, 47% in the 4 ports scenario. The average processor utilization increasing by increasing the number of ports. The Figure 3.15 shows the HTTP connection latency under class C TCP-SYN flood attack traffic in all three scenarios. The connection latency was increasing with the increasing number of ports. In the 4 ports scenario, the connection latency was almost 1000 milliseconds this means the server took 1 second to respond to the client network.

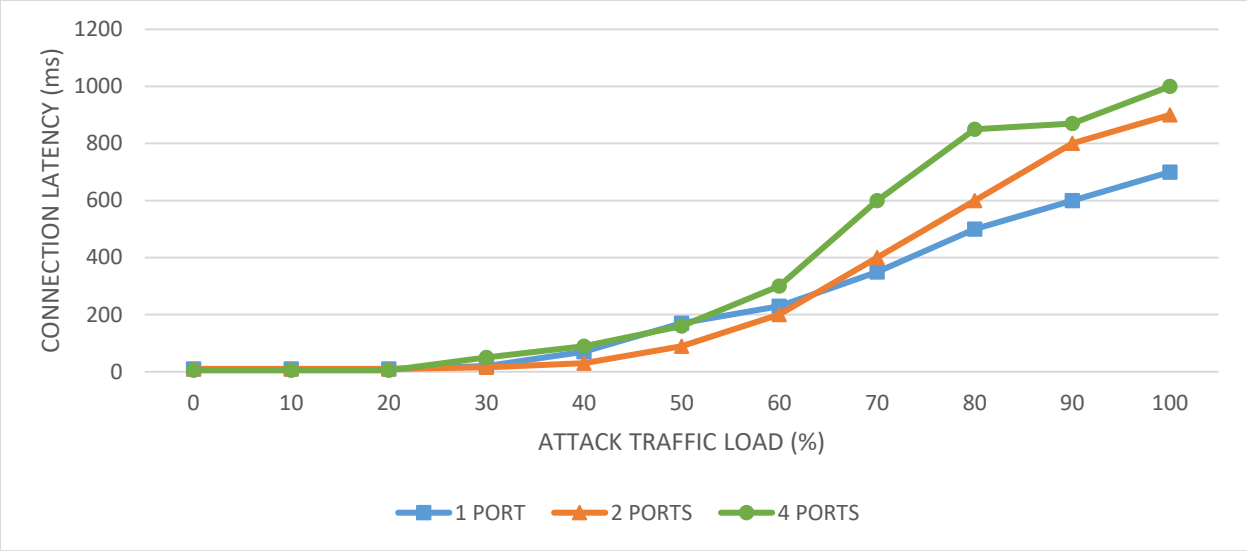

Figure 3.15 HTTP Connection Latency under TCP-SYN Flood Attack when sent from Class C Network

3.3.3.2 Class B network TCP-SYN Flood Attack

In TCP-SYN flood attack, the number of half open connections will be more with more number of different hosts. Class B network provides more number of hosts per network than class C network. So, will see class C impact on the victim server under TCP-SYN flood attack traffic.

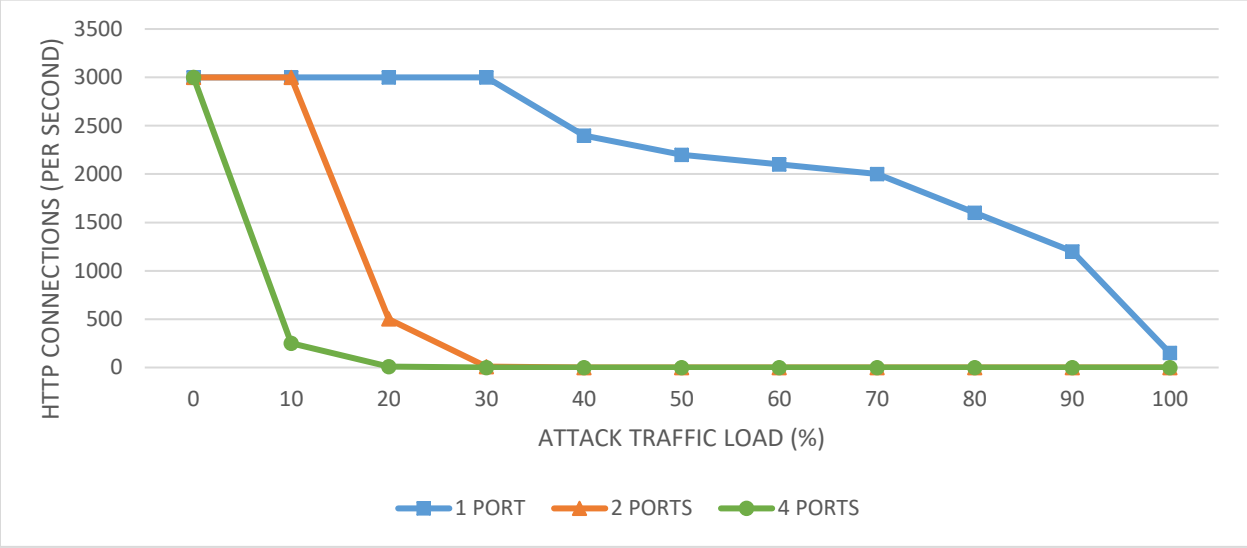

Figure 3.16 Number of HTTP connections established by the server under TCP-SYN Flood Attack when sent from Class B Network

The Figure 3.16 shows the number ofHTTP connections established per second under class B network TCP-SYN attack. The HTTP connection rate was started declining at 40% of attack traffic load in the single port scenario. By the increasing number of half-open connections, the HTTP connection rate was drastically declined at 20% of TCP-SYN attack traffic in the two ports scenario and server became connectionless at only 30% of attack traffic load. In the 4 ports scenario, there were approximately 250 HTTP connections at only 10% of attack traffic load and became connectionless at 20% of TCP-SYN attack traffic load.

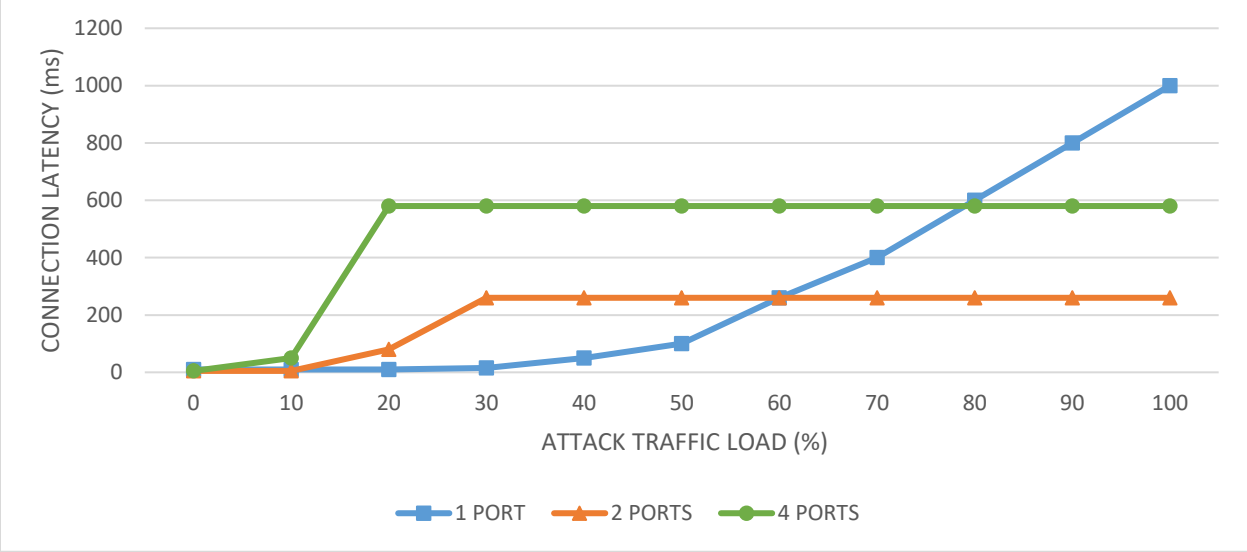

Figure 3.17 HTTP Connection Latency under TCP-SYN Flood Attack when sent from Class B Network

The above Figure 3.17 shows the connection latency under class B network TCP-SYN attack traffic in three different scenarios. In the 4 ports scenario, the connection latency reached to 600 milliseconds at 1200 Mbps and later became constant because of no HTTP connections established. In the 2 ports scenario, the connection latency reached to 260 milliseconds at 800 Mbps and later became constant because of no HTTP connections. In the 1 port scenario,the HTTP connection latency was increasing because there were still some HTTP connections established by the victim server even at higher attack traffic load.

3.3.3.3 Comparison of Class B with Class C networks under TCP-SYN attack traffic

Because of having more number of different host connections, the class B has more impact on server than class C network. The number of half-open connections will be more in class B even though the attack intensity was same in case of class C. Because the TCP-SYN attack will create a half-open connection for each different host. I observed that, there were still some connections established even at 100% of attack traffic in four ports scenario in case of class C network but the server became connectionless at only 20% of TCP-SYN flood attack traffic in case of class B network. The below Figure 3.18 shows the number of non-paged pool allocations under TCP-SYN flood attack traffic in the class B and class C networks. The number of non-paged pool allocations were more in the class B than class C network. It shows there was no impact on non-paged pool allocations in class C network even after introduced the attack traffic.

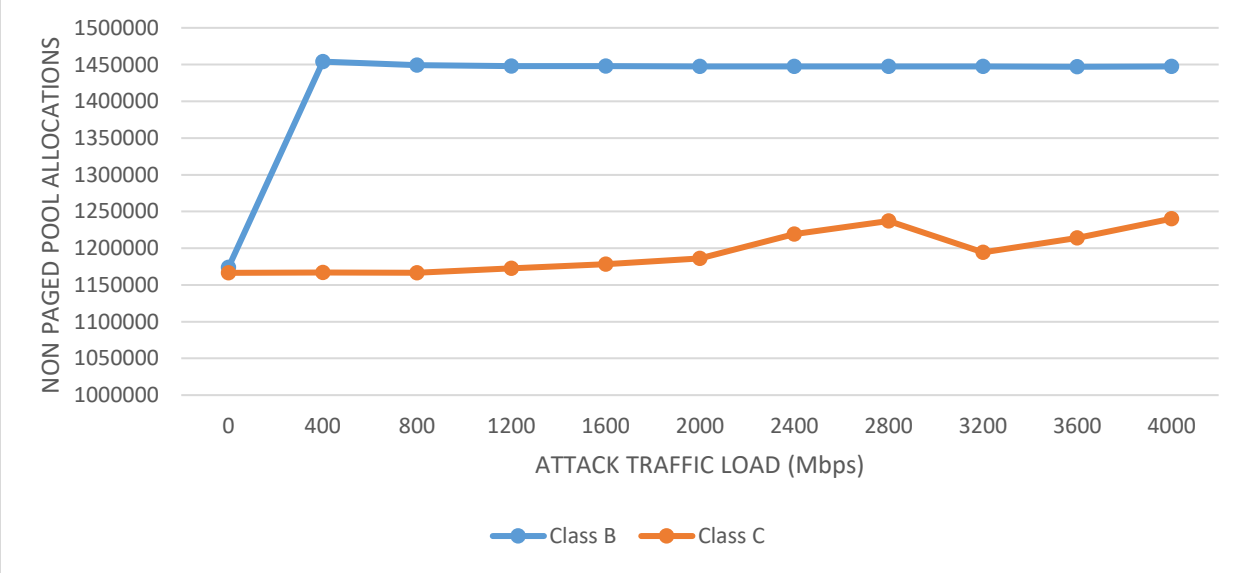

Figure 3.18 Number of Non-Paged Pool Allocations under TCP-SYN Flood Attack when sent from Class B and Class C Networks

3.3.4 UDP Flood Attack

The User Datagram Protocol (UDP) is a connectionless computer networking protocol. The UDP is unlike TCP and there is no guarantee of delivering, ordering or duplicate protection. The UDP Flood Attack occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that the victim can no longer handle valid connections. The main intention of UDP Flood Attack is to freeze the internet pipe. In this UDP Flood Attack, an attacker sends UDP datagrams in IP packets with spoofed source addresses. These are all UDP datagrams targeting a DNS server. After reaching threshold limit of these datagrams, the DNS server will reject further UDP datagrams from all the addresses in the same security zone for the remainder of the current second. Because of this it will also reject legitimate UDP datagrams from an address in the same security zone.

3.3.4.1 Class C network UDP Flood Attack

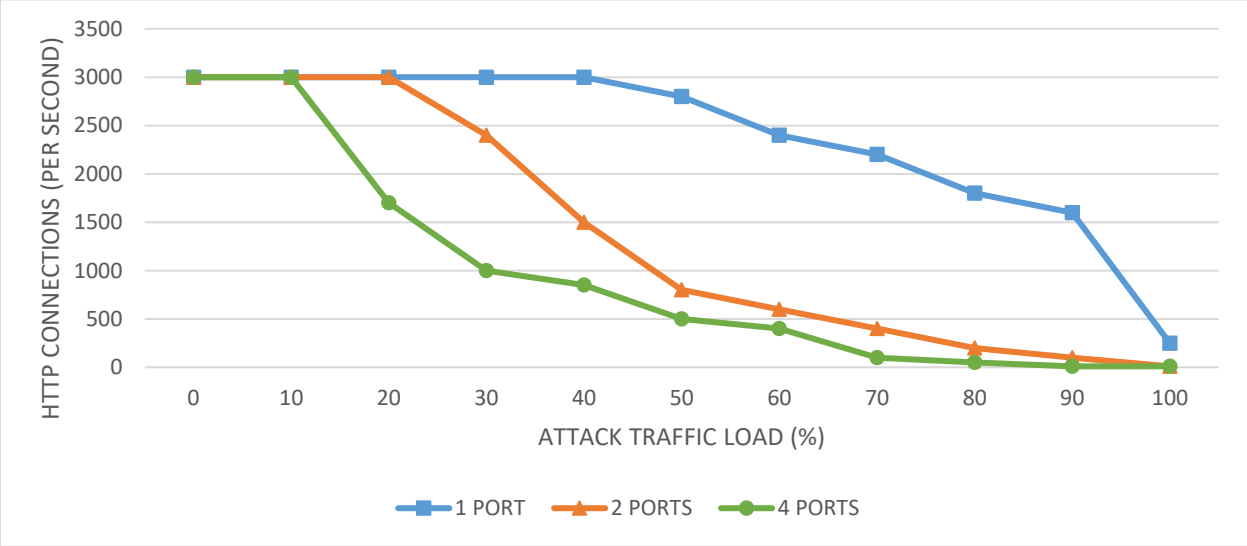

Figure 3.19 Number of HTTP connections established by the server under UDP Flood Attack when sent from Class C Network

The above Figure 3.19 shows the HTTP connection rate of victim server under class C UDP flood attack traffic in three scenarios. The HTTP connection rate was started declined at 50% of UDP flood attack traffic. There were approximately 1600 connections were established by the server at 90% of attack traffic load in one port scenario and there was sudden decline at 100% of attack traffic load. In two ports scenario, the victim server has started declining HTTP connections at 30% of attack traffic load and victim server became connectionless at 90% of attack traffic load. In the 4 ports scenario, the HTTP connection rate was started declining at only 20% attack traffic load and became connectionless at 70% of UDP flood attack traffic.

3.3.4.2 Class B network UDP Flood Attack

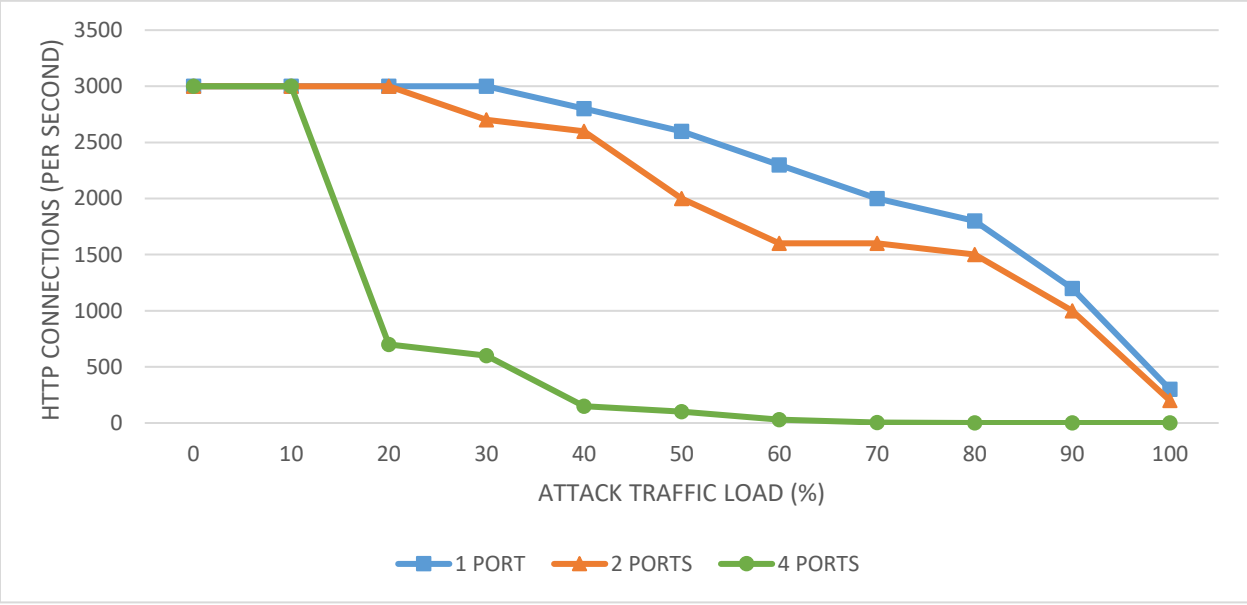

Figure 3.20 Number of HTTP connections established by the server under UDP Flood Attack when sent from Class B Network

The above Figure 3.20 shows the HTTP connection rate under class B UDP flood attack traffic in all three scenarios. In the first scenario, the connection rate was started declining at 40% of attack traffic load. In the two ports scenario, the connection rate was started declining at

30% of attack traffic load. In the 4 ports scenario, the HTTP connection rate was sharp declined at 20 % of UDP attack traffic load and victim server became connectionless at 50% of attack traffic load. The connection latency was decreasing by increasing the number of ports as shown in the Figure 3.21.

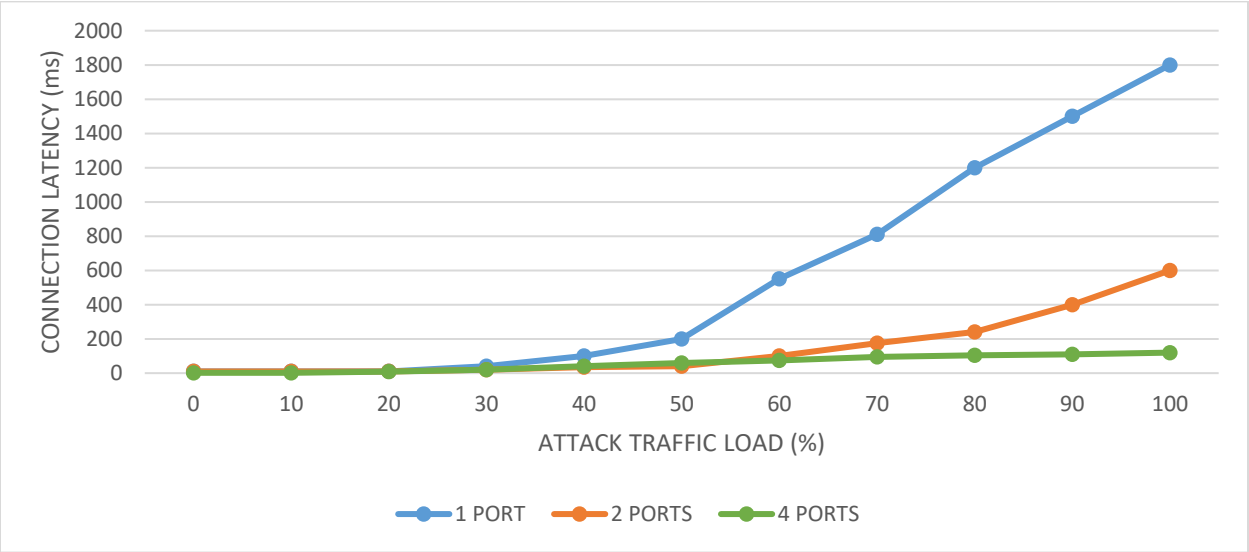

Figure 3.21 HTTP Connection Latency under UDP Flood Attack when sent from Class B Network

3.3.4.3 Comparison of Class B with Class C networks under UDP flood attack traffic

The HTTP connection rate was affected more in class B network than class C network in the two ports and four ports scenarios. In the first scenario, there was no much difference of HTTP connections established by the victim server with both the class C and class B networks. In class C network server became connectionless at 70% of attack traffic load whereas victim server became connectionless at 50% of attack traffic load in case of class B network.

3.4 Comparison of Results

Now will see the comparison between the different DDoS attacks under class B and class C networks when it sent to four gigabit ethernet ports. The Figure 3.22 shows the class C network and Figure 3.23 shows class B network. The Smurf attack was affecting the victim server more than all other DDoS attacks in both class C and class B networks. There was no much security against Smurf attack on this victim server.

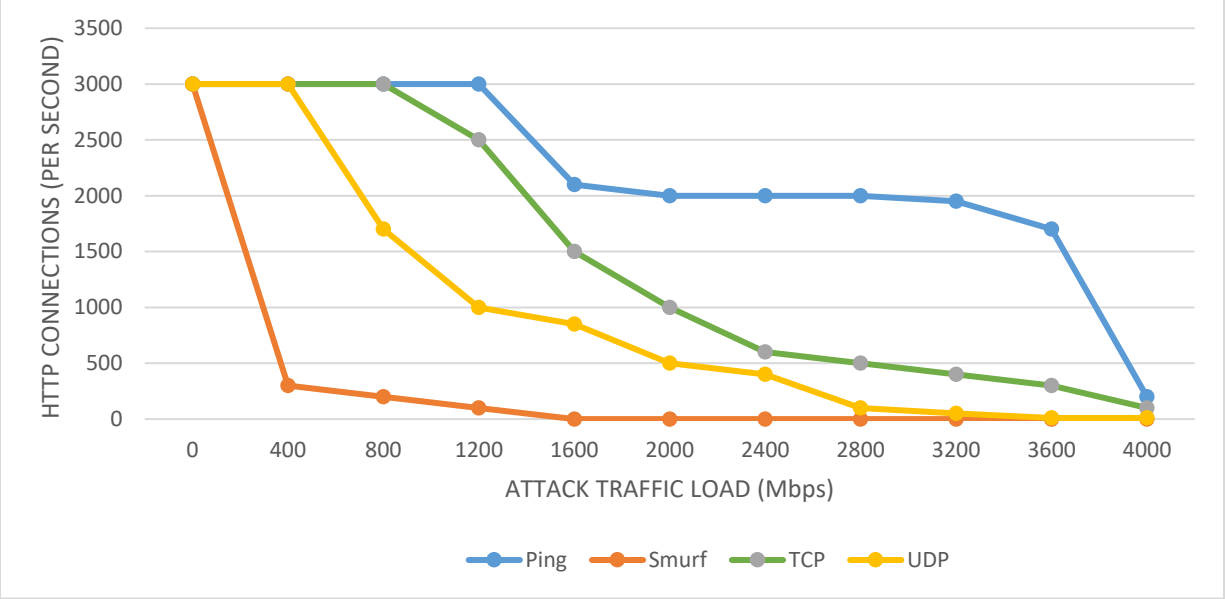

Figure 3.22 Comparison of HTTP connections under different DDoS attacks when sent to four ports from class C network

Because of having more number of half-open connections, the HTTP connection rate was more affected by class B network TCP-SYN flood attack traffic. This shows that, there was not enough security on MAC hardware having Windows Server 2012 R2 OS, against TCP-SYN based DDoS attacks. It shows there was no much impact on HTTP connection rate in Ping flood attack traffic compared to other DDoS attacks in both the networks. This clears that, the Windows Server 2012 R2 operating system on MAC hardware platform having better security against Ping flood attack.

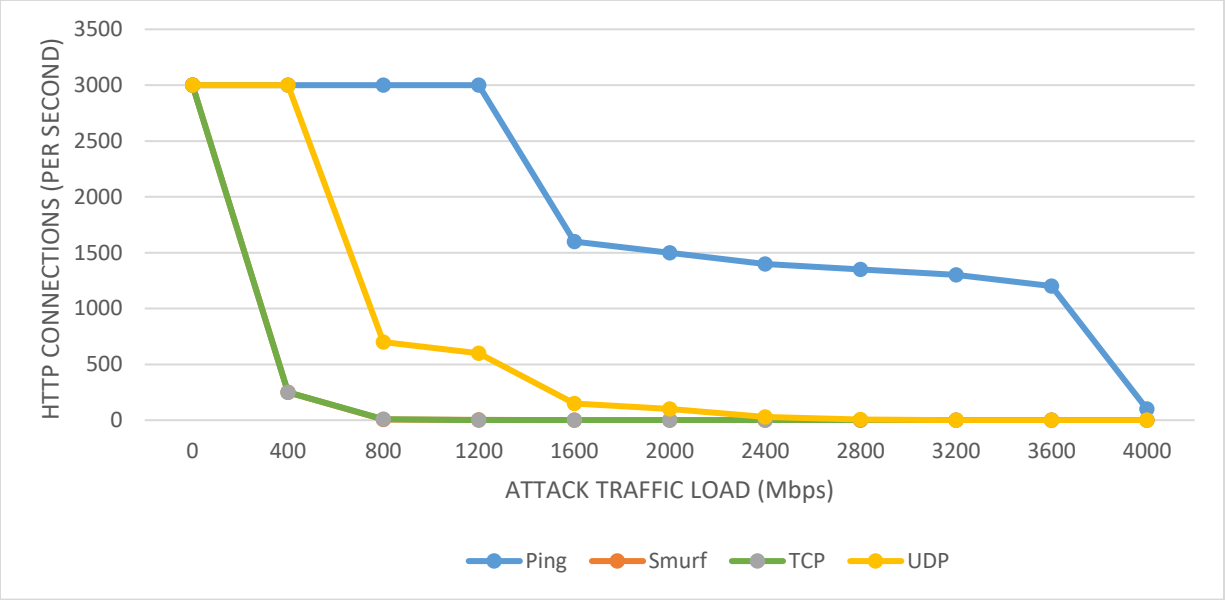

Figure 3.23 Comparison of HTTP connections under different DDoS attacks when sent to four ports from class B network

UDP flood attack was also affecting victim server in both cases irrespective of different number of hosts networks in class B. A little bit less impact on class C as compared to class B network against UDP flood attack. In the class B network, approximately there was no HTTP connections in all DDoS attacks except Ping flood attack at only 1600 Mbps (40% of 4 Gbps) of attack traffic intensity.

3.5 Chapter Summary

The behavior of Windows Server 2012 R2 operating system on MAC hardware platform with external Broadcom ethernet adapters was different with inbuilt ethernet adapters. There was a more probability of getting affected to all ethernet ports if one ethernet port was compromised. It was observed that, if victim server was affected by class B network, it will affect more on HTTP connection rate than with class C network. In this chapter, there was different results with different scenarios was observed in both the class C and class B Network Distributed Denial of Service Attacks. For some DDoS attacks, there was not enough security in the victim server.

CHAPTER IV

EFFECT OF SECURITY ATTACKS ON VIRTUALIZED WINDOWS SERVER 2012 R2 OS ON MAC HARDWARE PLATFORM WITH 4 PORT BROADCOM NIC ADAPTER

Server virtualization enables multiple server instances to run concurrently on a single physical host; yet server instances are isolated from each other. Each virtual machine essentially operates as if it is the only server running on the physical computer. Network virtualization provides a similar capability in which multiple virtual network infrastructures run on the same physical network (potentially with overlapping IP addresses), and each virtual network infrastructure operates as if it is the only virtual network running on the shared network infrastructure [64]. Hyper-V is the virtualization software from Microsoft Corporation and works very well within a Microsoft environment. The hypervisor itself can run almost all OSs as VMs but is lacking when it comes to virtual networking and management options for a standalone system [65].

4.1 Experimental setup

The victim server is an Apple MAC PRO, Two 2.4GHz Quad-Core Intel Xeon E5620 "Westmere" processors server, 8 logical processor and 12 GB RAM [60]- [61]. As mentioned earlier, Windows 2012 R2 Operating System was installed on the MAC hardware platform which was used as the target server.

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 Non-Virtualized Windows Server 2012 R2 Operating System on MAC hardware platform was initially tested against four most popular DDoS attacks, Ping Flood, Smurf, TCP/SYN and UDP Flood attacks. In Chapter, Virtualization was introduced, and Virtualized Windows Server was used. Now I will test virtualized Windows Server 2012 R2 operating system on Apple MAC hardware platform. As I mentioned in Chapter I, there are many virtualization techniques available out there. In this chapter, Hyper-V technique was used. The Hyper-V manager was installed in the Windows OS through the Server manager console [66].

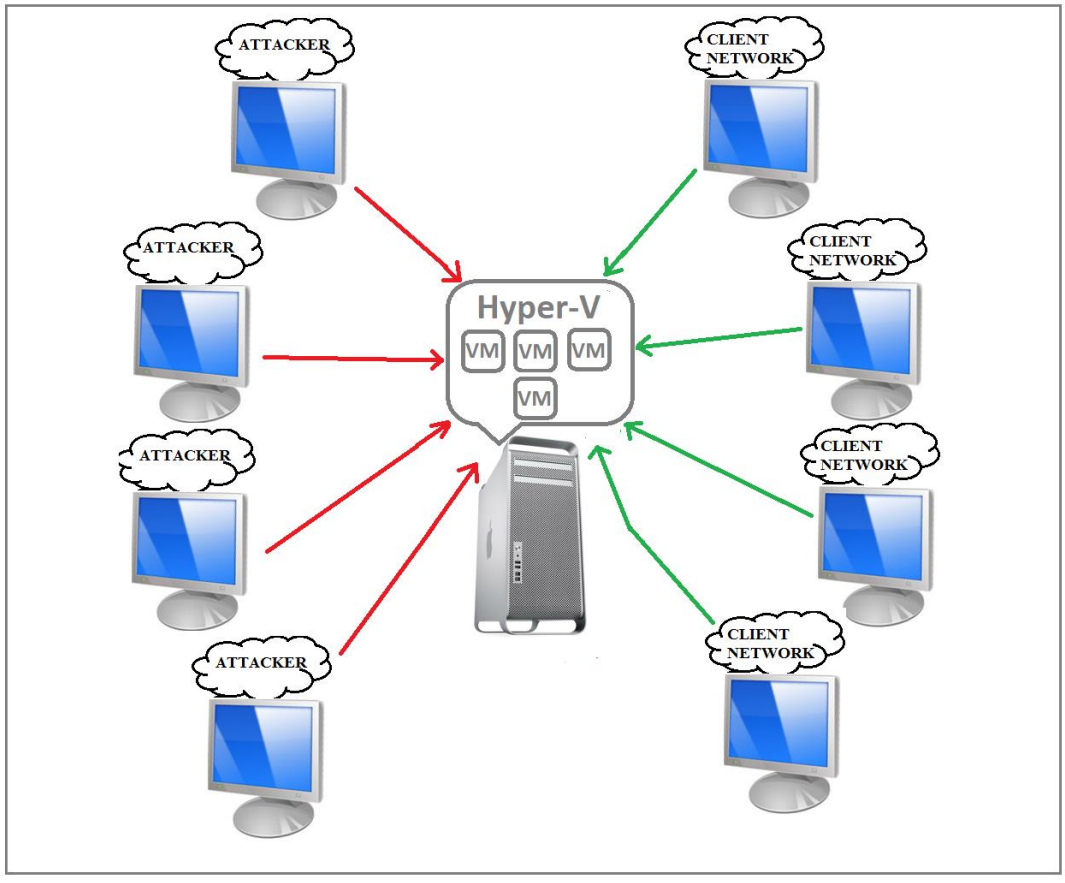

Figure.4.1 Experimental setup

Hyper-V Installation: Configuration click through the following options <Configure this local server> » <Add roles and features> » <role-based or feature-based installation option> » <Next> » < a server from the server pool option > » < Roles> » <Hyper-V> and continue to click on <Next> until you reach the finish installation screen and then restart the server. And then click <Server manager console> » <Tools tab> » <Hyper-V manager> » <Actions tab> » <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 to 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. Note: It is important to note that the generation of a virtual machine cannot be changed after the virtual machine has been created. It has 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 [67]. 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 <Virtual switch> for the virtual machine. Virtual Switches are broadly classified into three types: External, Internal and Private [68]. For this 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.

In this Chapter, I evaluate the effect of virtualization on Windows Server 2012 operating system on Apple MAC hardware platform. As I mentioned earlier, the victim server hardware platform is an Apple MAC PRO, two 2.4GHz Quad-Core Intel Xeon E5620 “Westmere” processors server having 8 logical processor and 16 GB of RAM. I allocated all available processors (8 logical processors) to Virtual Machines. Different amount of memory was allocated for different scenarios discussed below. The virtual machine was set up 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) was installed in the server OS following the instructions in [46].

In order to communicate with the client, first a sample webpage namely Index.html was created in the victim server with support of IIS service. And then, this sample webpage was accessed through the Hyper Text Transfer Protocol (HTTP) request from a Client. The victim server responds to the clients whenever server received a HTTP request from client. In this thesis, we used a controlled environment which was requesting web server request from the victim server. We calculated server capacity by means of HTTP connections per second, CPU utilization, memory utilization and non-paged pool allocations. 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.

For experiments in this chapter, I installed four Virtual Machines (Windows Server 2012 Operating System) on the victim Server. A sample webpage was created in all the VM's. I created four virtual switches by using Hyper-V virtual switch manager. The four Broadcom Gigabit Ethernet ports were assigned to four virtual switches created by Hyper-V virtual switch manager and I assigned four different virtual switches to four different virtual machines. In this chapter, I used three different scenarios to test these Virtual Machines. In all three different scenarios, the legitimate or client traffic was sent at the baseline rate of 3000 HTTP requests per second to the server.

Scenario I: In the first scenario, the attack traffic and legitimate traffic was sent to only one VM. In this case, 8 GB RAM was allocated to VM and remaining 8 GB was allocated to physical server. First, to measure a baseline of the experiment by sending five minutes of legitimate traffic in the form of HTTP requests without any attack traffic to VM. In this scenario, all 3000 HTTP connections were sent to one VM. Once the baseline of the experiment was established, the attack traffic was introduced to the Virtual Machine. Initially, attack traffic intensity of 100 Mbps was applied from simulated attack network for five minutes. Later, it was increased to 200 Mbps of attack traffic for five minutes and this process continued until reached to 1000 Mbps maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 100 Mbps of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes.

Scenario II: In the second scenario, the attack traffic and legitimate traffic was sent to two VMs. In this case, 5 GB RAM was allocated to each of VM and remaining 6 GB was allocated to physical server. First, to measure a baseline of the experiment by sending five minutes of legitimate traffic in the form of HTTP requests without any attack traffic to VM. In this scenario,

the number of HTTP connections per second were shared by two VM's which allowed baseline value of 1500 HTTP connections per second to each Virtual Machine. Once the baseline of the experiment was established, the attack traffic was introduced to the Virtual Machine. Initially, attack traffic intensity of 100 Mbps per VM was applied from simulated attack network for five minutes. Later, it was increased to 200 Mbps per VM of attack traffic for five minutes and this process continued until reached to 1000 Mbps per VM (Total of 2 Gbps) maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 100 Mbps per VM of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes.

Scenario III: In the third scenario, the attack traffic and legitimate traffic was sent to all four VM's. In this case, 3 GB RAM was allocated to each of VM and remaining 4 GB was allocated to physical server. First, to measure a baseline of the experiment by sending five minutes of legitimate traffic in the form of HTTP requests without any attack traffic to VM. In this scenario, the number of HTTP connections per second were shared by four VM's which allowed baseline value of 750 HTTP connections per second to each Virtual Machine. Once the baseline of the experiment was established, the attack traffic was introduced to the Virtual Machine. Initially, attack traffic intensity of 100 Mbps per VM was applied from simulated attack network for five minutes. Later, it was increased to 200 Mbps per VM of attack traffic for five minutes and this process continued until reached to 1000 Mbps per VM (Total of 4 Gbps) maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 100 Mbps per VM of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes. Different 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.

4.2 Performance parameters for Evaluation

In this experiment, the parameters that are used to evaluate the performance were Memory utilization, CPU utilization, Non-paged pool allocation and HTTP transactions per second. Some of these parameters we collect from Performance monitor are present in that particular Operating system. For performance monitor click through the following options <Data Collector Sets> » <User Defined> » <create new data collector set>. And then we can create manually by selecting those performance parameters.

CPU Utilization (CPU Usage in %): % Processor Time is 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). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It should be noted that the accounting calculation of whether the processor is idle is performed at an internal sampling interval of the system clock (10ms). On today's fast processors, % Processor Time can therefore underestimate the processor utilization as the processor may be spending a lot of time servicing threads between the system clock sampling intervals. Workload based timer applications are one example of applications which are more likely to be measured inaccurately as timers are signaled just after the sample is taken. The Processor utilization is amount of usage to the total central processing unit (CPU). This will evaluate affect of attack traffic on the CPU. If CPU utilization is more, that attack traffic is CPU intensive attack. The name of counter that is used to evaluate processor utilization is known as \Processor (_Total) \% Processor Time.

The Processor performance object consists of counters that measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The processor object represents each processor as an instance of the object. This particular MAC PRO server has 8 logical processors, hence the counters that were used to monitor the multi core utilization of the server are \Processor (0)\%Processor Time,\Processor (1)\%Processor Time,\Processor (2)\%Processor Time,\Processor (3)\%Processor Time,\Processor (4)\%Processor Time,\Processor (5)\%Processor Time,\Processor (6)\%Processor Time,\Processor (7)\%Processor Time.

Memory Utilization (RAM Usage in MBytes): Available MBytes is the amount of physical memory, in Megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free and zero-page lists. The memory utilization is the amount of RAM usage with respect to total random-access memory available assigned to that particular operating system. If the memory utilization is more then we can say that this attack is called Memory intensive attack. The name of the counter that is used to evaluate memory utilization is known as \Memory\Available MBytes.

Non-paged pool allocation: Pool Nonpaged Allocs is the number of calls to allocate space in the nonpaged pool. The nonpaged pool is an area of system memory area for objects that cannot be written to disk, and must remain in physical memory if they are allocated. It is measured in numbers of calls to allocate space, regardless of the amount of space allocated in each call. This counter displays the last observed value only; it is not an average. The name of the counter that is used to evaluate Non-paged pool Allocation is known as \Memory\Pool Nonpaged Allocs.

HTTP transaction per second: This HTTP transactions are referred to the number of legitimate connections established by the server. This parameter will give the number of connections per second established by the server for different amount of attack traffic ranging from 1 Mbps to 1 Gbps. This parameter helps to determine whether the server has reached its saturation point or not.

Connection Latency: 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 also 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.

4.3 Results and Discussion

4.3.1 Ping Flood Attack

Ping based DDoS attacks are flood of a large number of ping messages sent to target are known to be quite damaging to the availability of the web based services. The ping attack can exhaust the target server's bandwidth and computing resources. After the baseline established, the ping flood attack traffic was introduced to the Virtual Machine.

Figure 4.2 shows number of HTTP connections per second established by the Virtual Machine in all three scenarios. The number of HTTP connections were 3000 at 800 Mbps (80% of 1 Gbps) of Ping flood attack traffic and later it declined to 1000 HTTP connections per second at 1000 Mbps. Figure 4.3 shows Individual core utilization under Ping flood attack in the first scenario. We can see there was only one core that experienced most of the load, even though some other cores were trying to share the load. The maximum utilization of that core was 70 at 900 Mbps of Ping flood attack traffic.

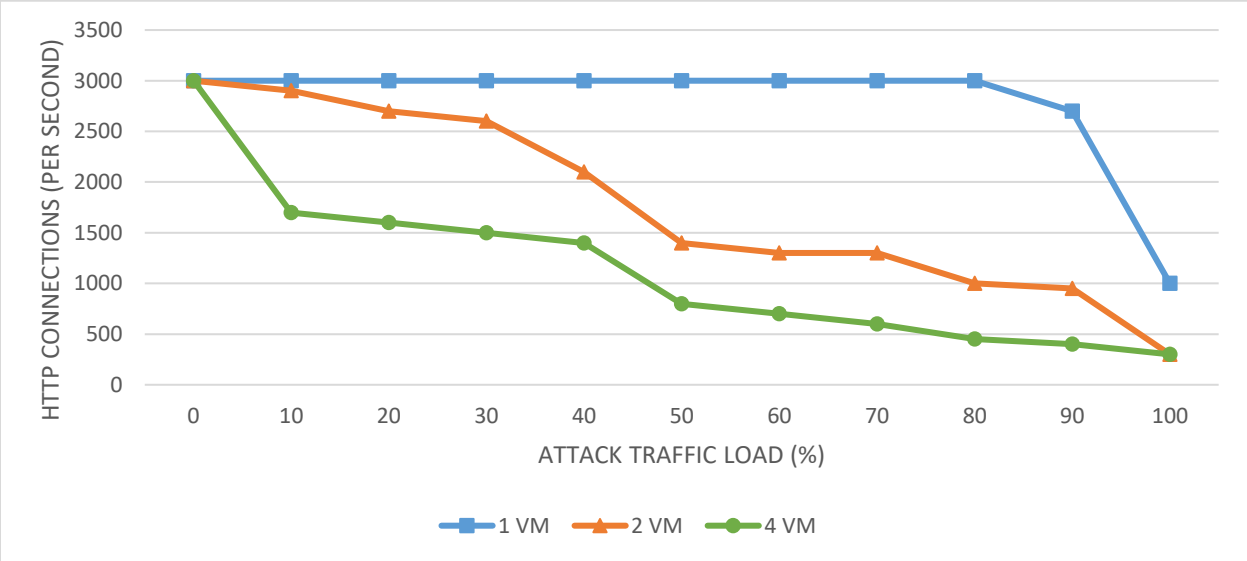

Figure 4.2 Number of HTTP connections established by the server under Ping Flood Attack when sent to Virtual Machines

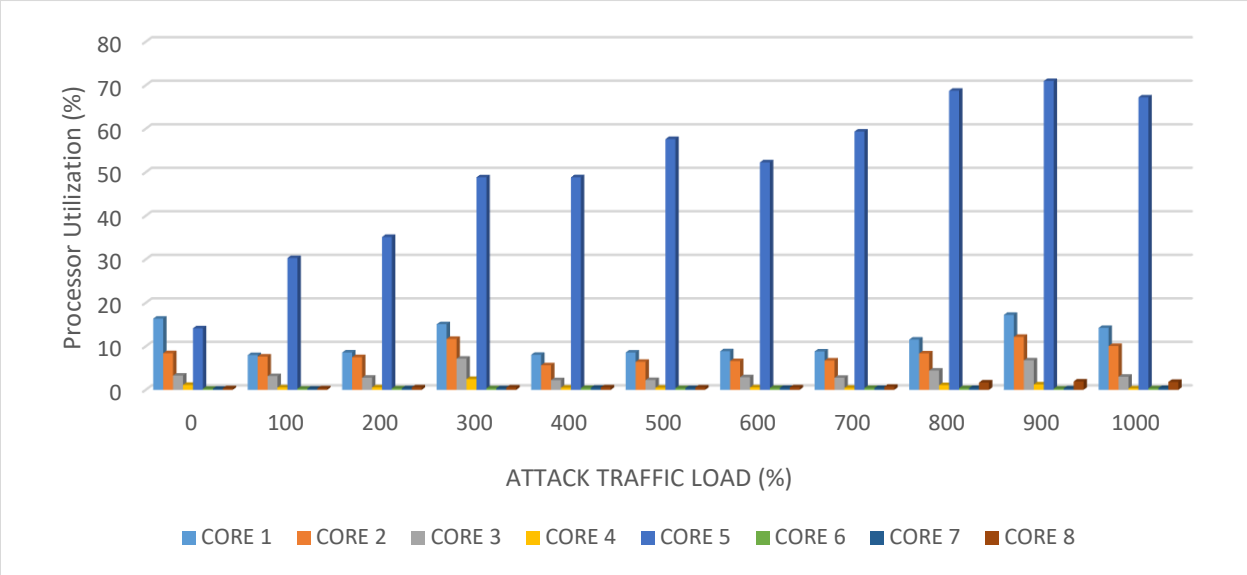

Figure 4.3 Individual Core Utilization under Ping Flood Attack when sent to one Virtual Machine

In the second scenario, the number of HTTP connections per second were started declining at 200 Mbps (10 % of 2 Gbps) of Ping flood attack traffic and it continued declining till there were approximately 300 HTTP connections per second at 2 Gbps of Ping flood attack traffic as shown in Figure 4.2. And Figure 4.4 shows individual core utilization under Ping flood attack of 2nd VM while using two Virtual Machines. Like first scenario, there was only one core which experienced most of the load. That one core (core 5) reached its maximum utilization at 400 Mbps of Ping flood attack. And later, it maintained constant utilization of 25%.

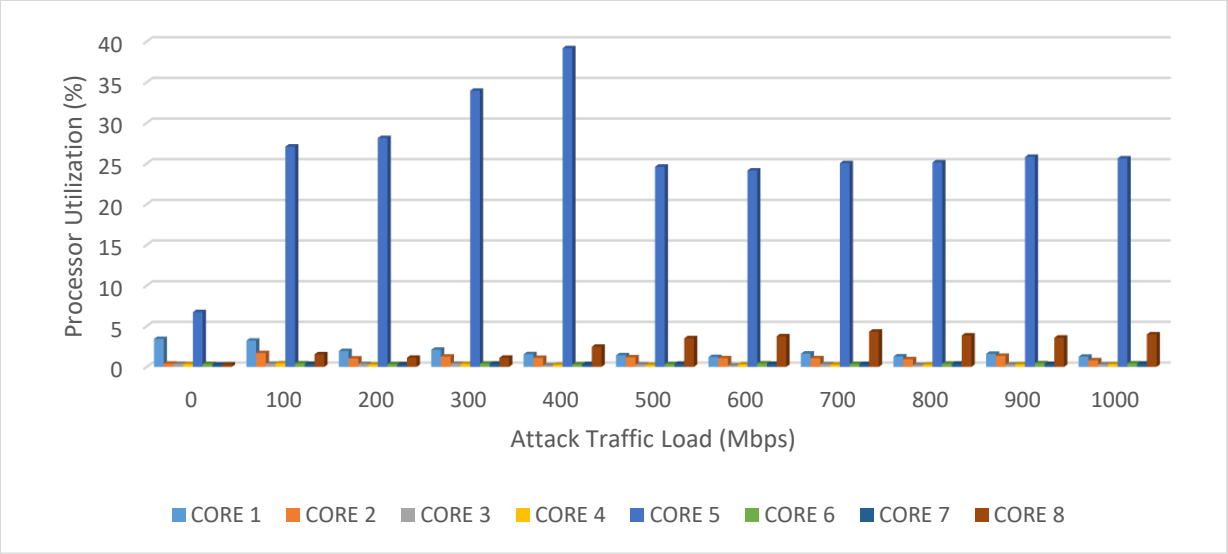

Figure 4.4 Individual Core Utilization under Ping Flood Attack of 2nd VM while using 2 Virtual Machines

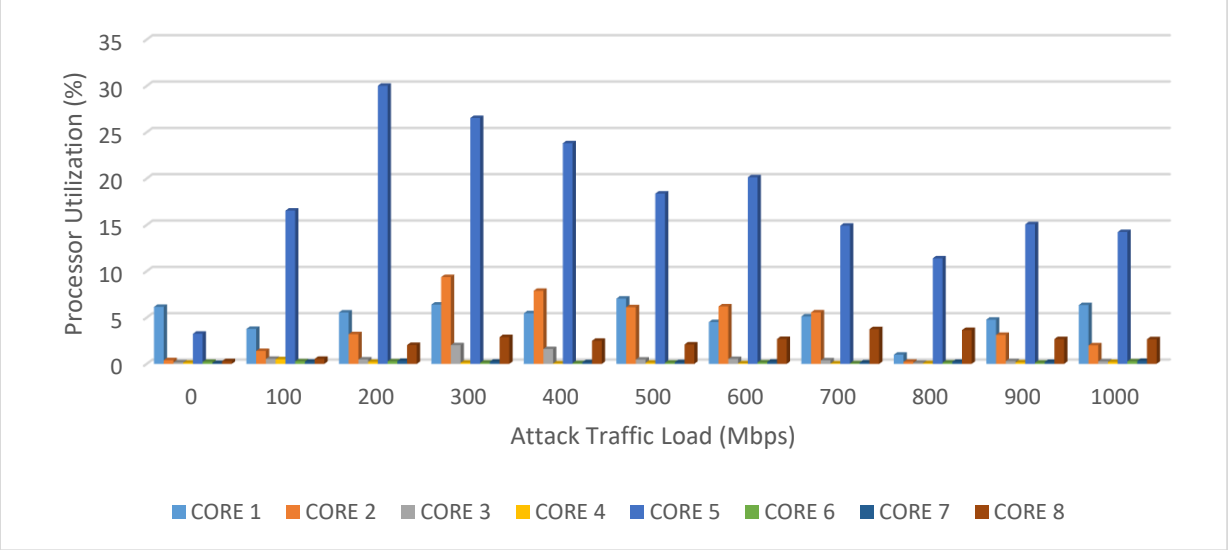

Figure 4.5 Individual Core Utilization under Ping Flood Attack of 4th VM while using Virtual Machines

In case of four Virtual Machines, the number of HTTP connections per second decline from 3000 HTTP baseline connections per second to 1700 HTTP connections per second at 400 Mbps (10% of 4Gbps) of Ping flood traffic as shown in figure 4.2. Later it declined at regular intervals and there were about 300 HTTP connections per second. As compared to behavior of two VM's,

four VM's HTTP connection rate was worst. In this third scenario, the individual core utilization was similar as compared to previous two scenarios. I observed that, the maximum core utilization of core 5 was 30%. And later it was keep declined as shown in Figure4.5.

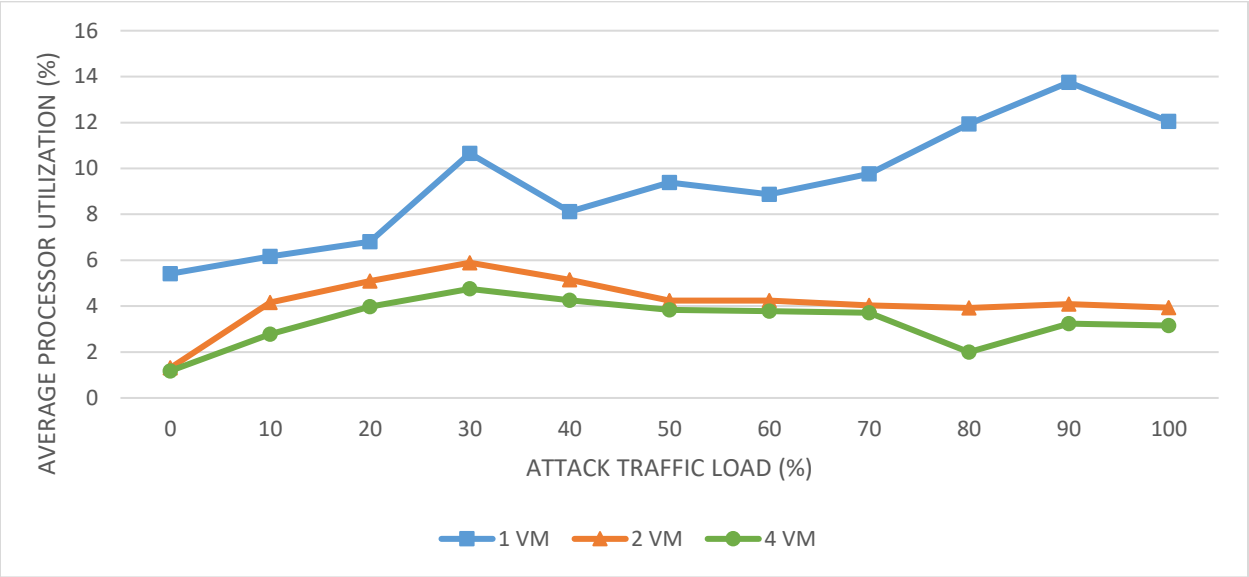

Figure 4.6Average Processor Utilization under Ping Flood Attack when send it Virtual Machines

Figure 4.6 shows the average processor utilization in all three scenarios under Ping flood attack. In the first scenario, the average processor utilization was 6 % when there was no attack traffic. And then, it started increasing once attack traffic was introduced and it reached to 12% at 1000 Mbps of Ping flood attack traffic. In second scenario, the average processor utilization increased from 2% to 4% at 100% of attack traffic load. And then, in third scenario the average processor utilization was little bit less as compared to second scenario with two VMs. I observed that, the average processor utilization was decreasing with the increase in number of Virtual Machines.

4.3.2 Smurf Attack

A more sophisticated version of a DDoS attack is commonly known as a SMURF attack. A SMURF attack utilizes massive number of ICMP packets of spoofed source Internet Protocol (IP) addresses targeting the victim server's IP address This is achieved by altering the Echo Request sent to the botnet using an IP broadcast address. The larger the Botnet is the faster and the bigger is the flood of Echo reply messages. The increase of traffic reduces the target server's ability to respond, and can quickly cause a complete denial of service.

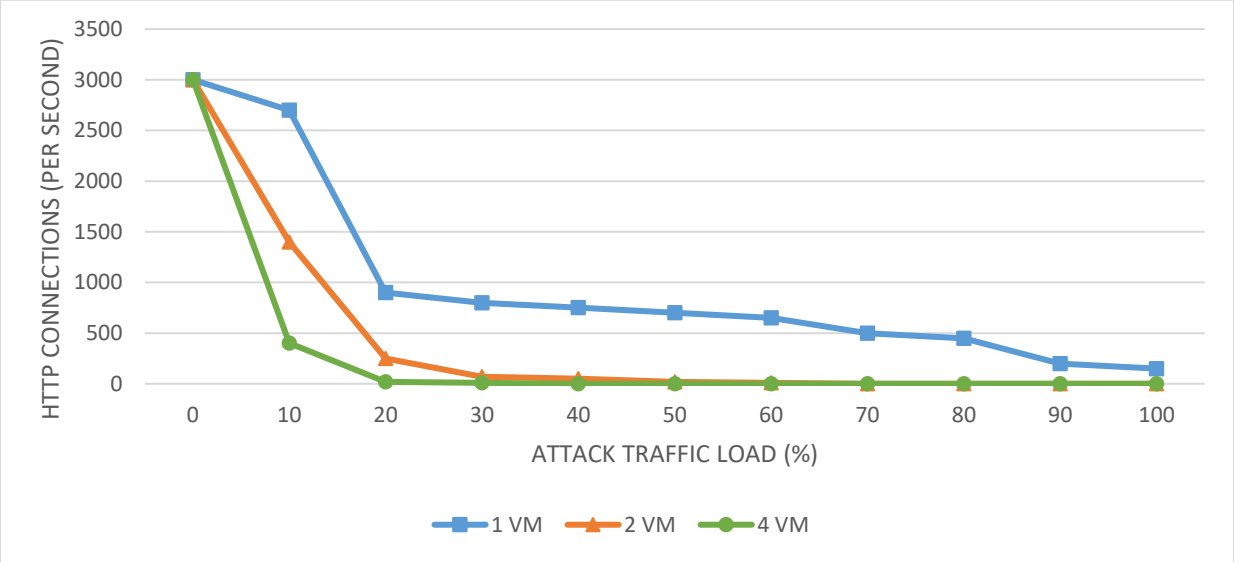

Figure 4.7 Number of HTTP connections established by the server under Smurf Attack when sent to Virtual Machines

Figure 4.7 shows the number of HTTP connections per second under Smurf attack in all three scenarios. In the first scenario, the number of HTTP connections per second were 3000 at the baseline without any attack traffic. Once the Smurf attack traffic was introduced, the number of HTTP connections were declined a little bit, approximately 2700 connections per second at 100 Mbps of Smurf attack traffic. There was a sudden decline of 900 HTTP connections per second

at 200 Mbps (20% of 1 Gbps) of attack traffic. And later, it was gradually declined, and it reached 150 HTTP connections per second at 1000 Mbps (100% of 1Gbps). Figure 4.8 shows individual core utilization of first scenario. It can be observed that; three cores were experienced load at baseline without attack traffic. Once, attack traffic was introduced, core 3 utilization reached to its maximum. This core 5 was experienced its maximum load throughout the experiment even though other cores were trying to share the load.

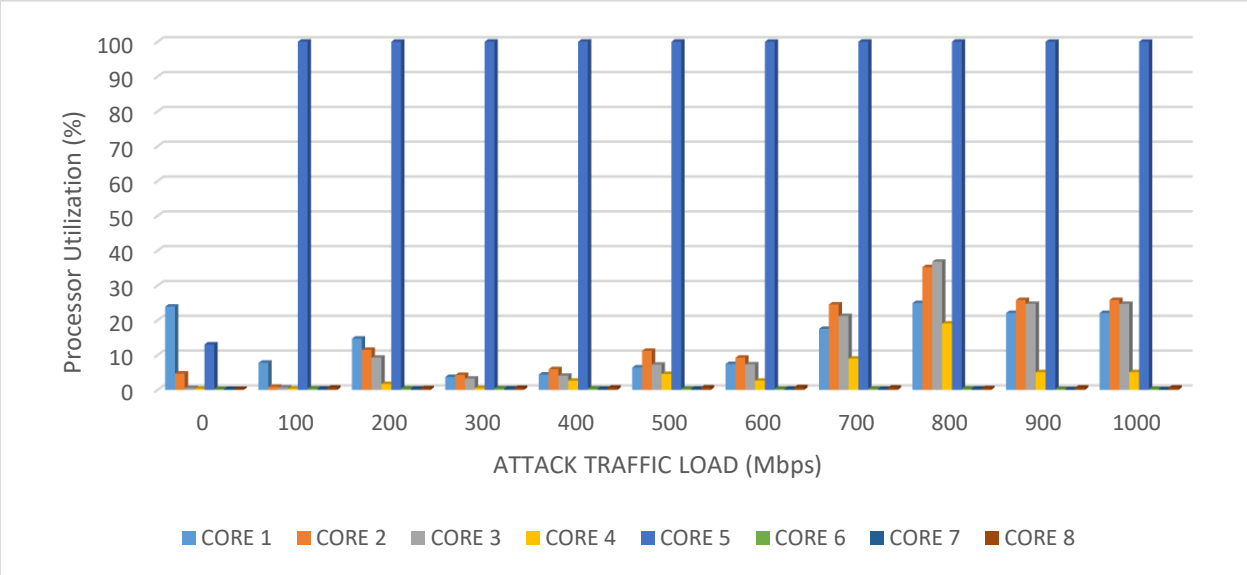

Figure 4.8Individual Core Utilization under Smurf Attack when sent to one Virtual Machine

In the second scenario, the number of HTTP connections were declined from 3000 to approximately 1500 at 200 Mbps (10% of 2 Gbps) of Smurf attack traffic. Later, it dropped to 250 HTTP connections at 400 Mbps of attack traffic. And then there were no HTTP connections at 1000 Mbps of Smurf attack traffic. Figure 4.9 shows the individual core utilization of second scenario. It is similar to first scenario, the core 5 was only core experienced most of the load. Because of having some connections, the other cores were also shared little bit of load until 800 Mbps of Smurf attack traffic.

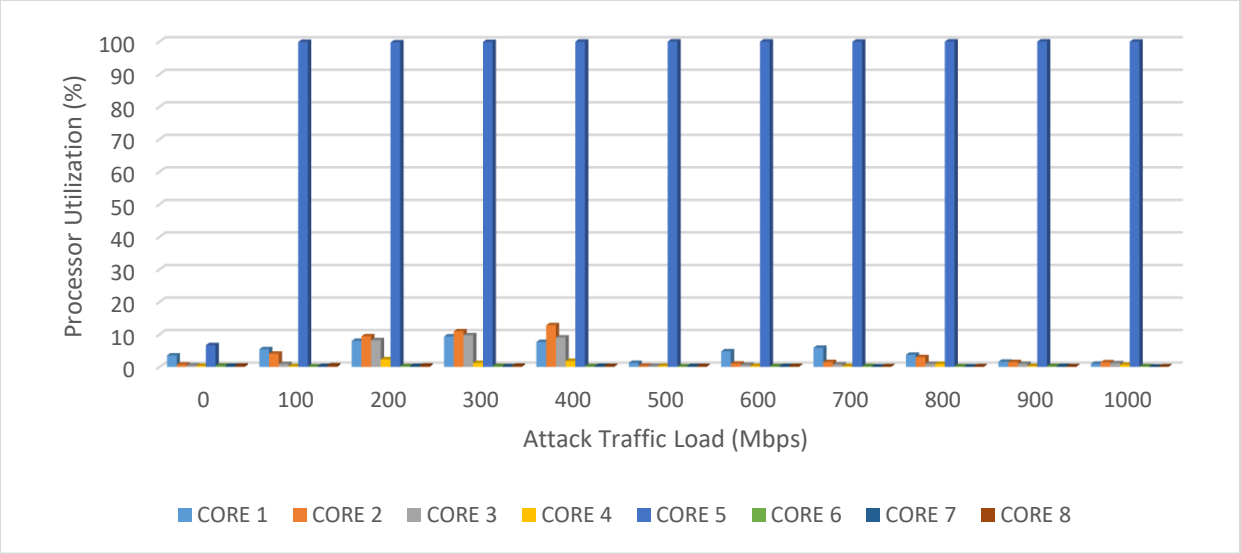

Figure 4.9 Individual Core Utilization under Smurf Attack of 2nd VM while using 2 Virtual Machines

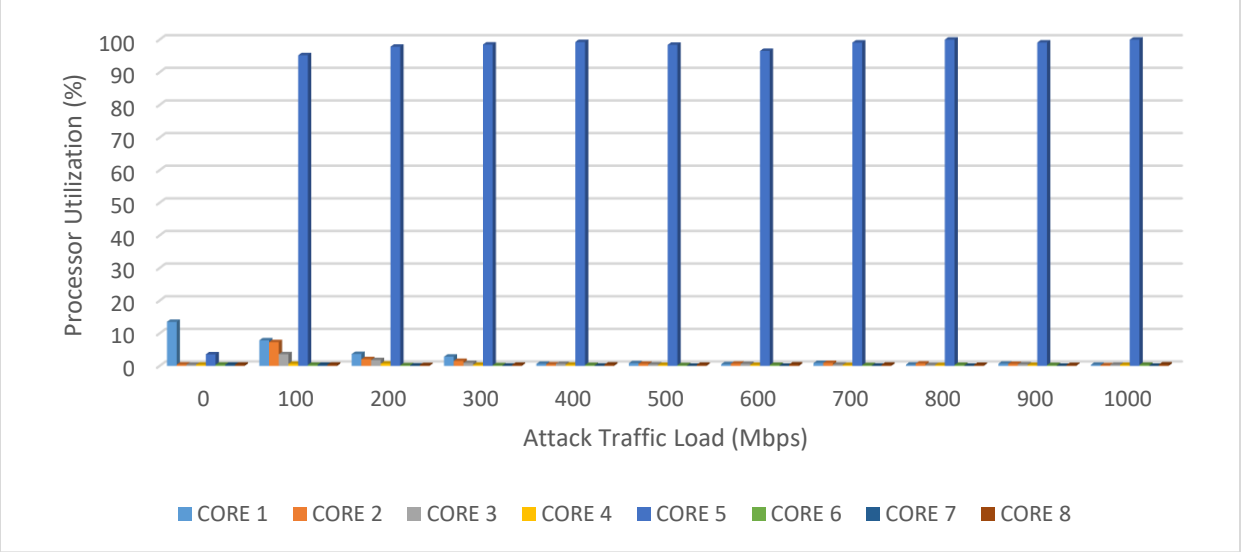

Figure 4.10 Individual Core Utilization under Smurf Attack of 4th VM while using Virtual Machines

When Smurf attack sent to four Virtual machines, the number of HTTP connections were declined to 400 connections from baseline of 3000 HTTP connections per second. It became connectionless at 20% of Smurf attack traffic. Figure 4.10 shows individual core utilization of third scenario. At 10% of Smurf attack traffic, core 1, core2, core3 and core5 were shared the

load but core5 is almost experienced 95%. While increasing attack traffic, only core5 was experienced most of the load.

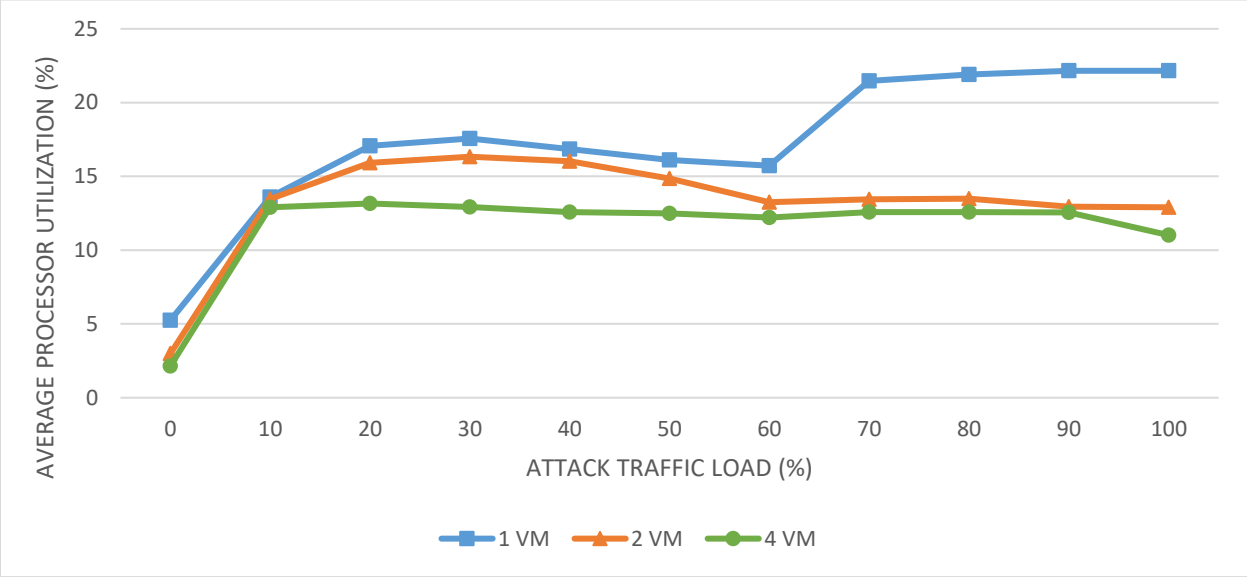

Figure 4.11Average Processor Utilization under Smurf Attack when sent to Virtual Machines

Figure 4.11 shows the average processor utilization under Smurf attack traffic in all three scenarios. It can be observed that; the average processor utilization was decreasing while increasing the number of Virtual Machines. The maximum utilization in case of one Virtual Machine was approximately by 22%. Where as in case of four Virtual Machines, the maximum utilization was only 13%. Figure 4.12 shows the number of non-paged pool allocations under Smurf attack traffic in all three scenarios. It shows, the number of non-paged pool allocations reduced by increasing the number of Virtual machines. In case of one VM, at 1000 Mbps of Smurf attack traffic, there were approximately 82500 non-paged pool allocations whereas 74000 non-paged pool allocations in case of 4 VM at 4000 Mbps of Smurf attack traffic.

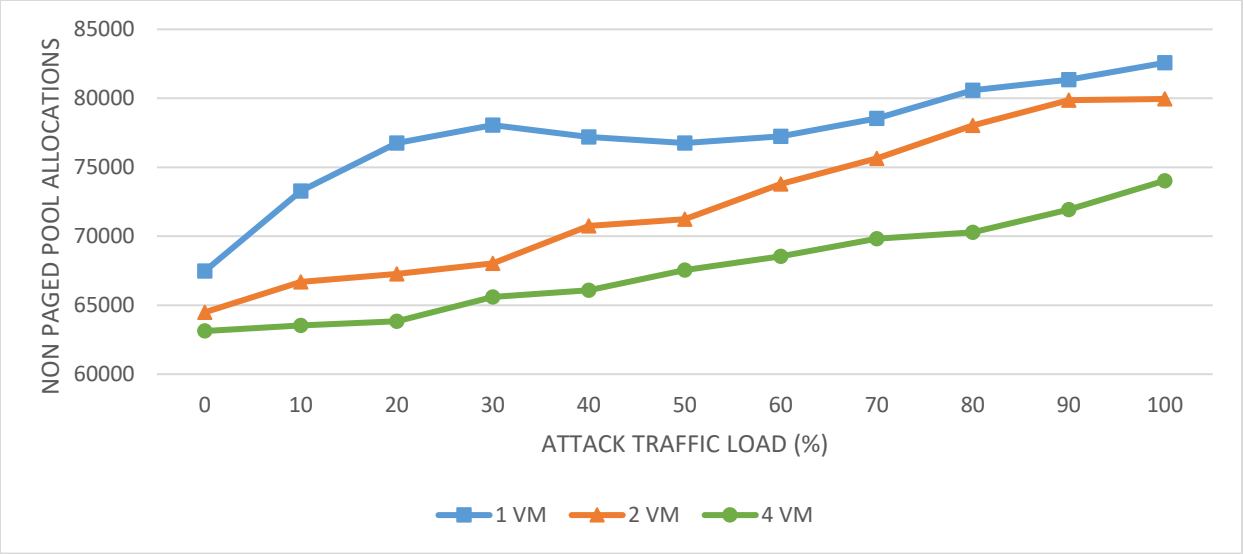

Figure 4.12Number of Non-Paged Pool Allocations under Smurf Attack when sent to Virtual Machines

I observed that, while increasing the number of Virtual Machines the number of HTTP connections per second were declined under the Smurf attack traffic. And Smurf attack can quickly decline the HTTP connections as compared to Ping Flood attack traffic. Also, the individual core utilization was also different with Ping attack. There was only one core experienced most of the load against Smurf attack traffic.

4.3.3 TCP-SYN Flood Attack

The Transfer Control Protocol (TCP) is a connection oriented protocol which is belongs to layer 4 of OSI reference model. TCP uses a three-way handshake to establish a network connection. Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open.

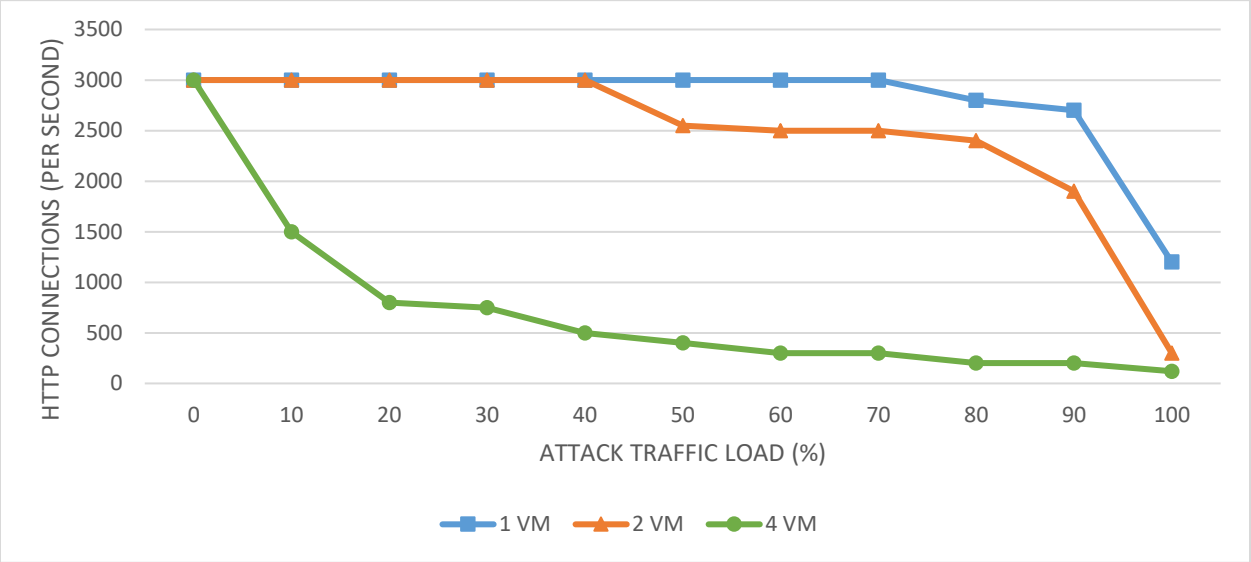

Figure 4.13 Number of HTTP connections established by the server under TCP-SYN Flood Attack when sent to Virtual Machines

Figure 4.13 shows the number of HTTP connections per second under TCP-SYN flood attack in all three different scenarios. In the first scenario, the number of HTTP connections were not affected until 700 Mbps (70% of 1 Gbps) of TCP-SYN flood attack traffic. At 800 Mbps of attack traffic, the number of HTTP connections per second were slightly declined to 2800 and later it declined to approximately 1200 HTTP connections per second at 1000 Mbps of attack traffic. Figure 4.14 shows the individual core utilization under TCP-SYN flood attack in case of

first scenario. It shows there was only one core5 experienced most of the load. The maximum utilization of core5 is 80% at 1000 Mbps of attack traffic.

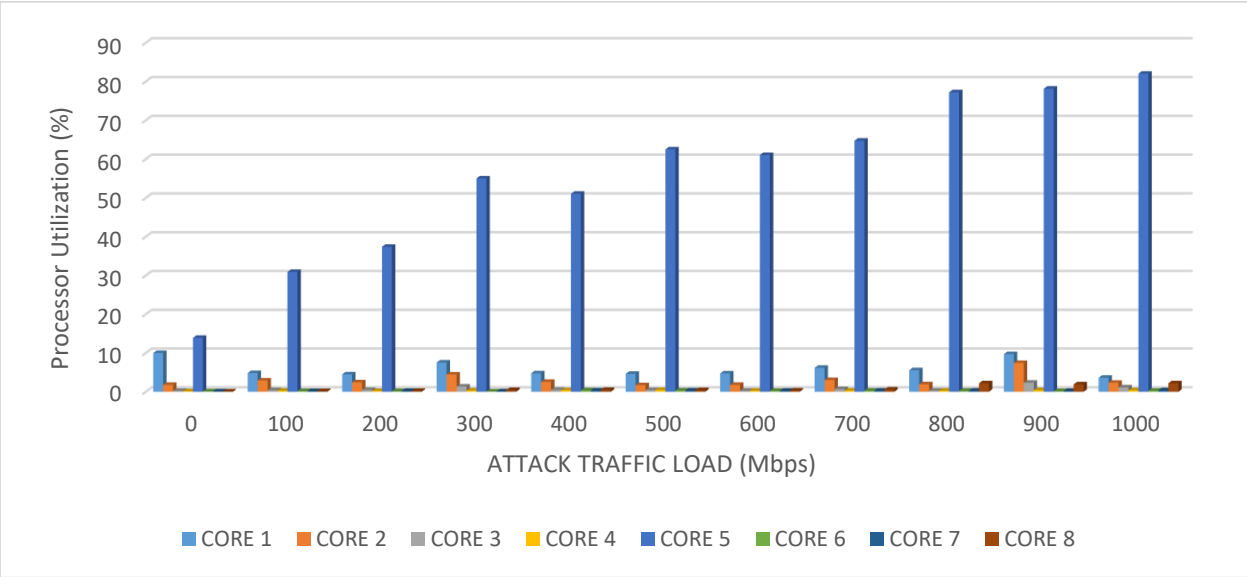

Figure 4.14 Individual Core Utilization under TCP-SYN Flood Attack when sent to one Virtual Machine

In the second scenario, the number of HTTP connections were declined at 1000 Mbps (50% of 2Gbps) of TCP-SYN flood attack traffic. A sudden declined from 2000 HTTP connections to 300 HTTP connections per second at 2000 Mbps (100% of 2Gbps) of TCP-SYN flood attack traffic. In the third scenario, the HTTP connections were declined at early stage it declined to 1500 HTTP connections per second at only 400 Mbps (10% of 4Gbps) of TCP-SYN flood attack traffic. And then, it was keep declined while increase the attack traffic. From 40% of attack traffic load, the HTTP connections were under 500. This was caused more damage than other two scenarios. And finally, it was reached approximately 100 connections at 100 % of attack traffic load.

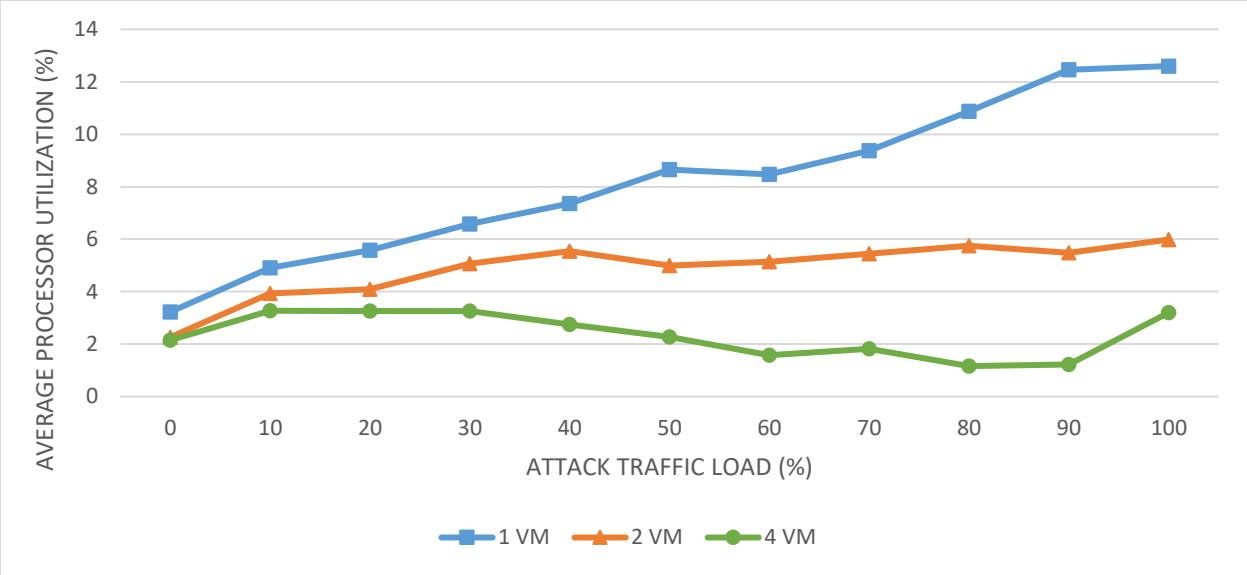

Figure 4.15Average Processor Utilization under TCP-SYN Attack when send it Virtual Machines

Figure 4.15 shows average processor utilization under TCP-SYN flood attack in all three scenarios. It shows while increasing the number of Virtual Machines the average processor utilization is decreased like all other attacks that were previously observed. The maximum average processor utilization is approximately 3% in case of four Virtual Machines. So, this attack was not processor intensive attack. Figure 4.16 shows HTTP connection latency under TCP-SYN attack in all the scenarios. The HTTP connection latency in case one VM was very less because there was no effect HTTP connection rate. In case of two VM's the HTTP connection latency was started increased at 40% of attack traffic load. Whereas in case of four VMs, the HTTP connection was increased at 20% of attack traffic load.

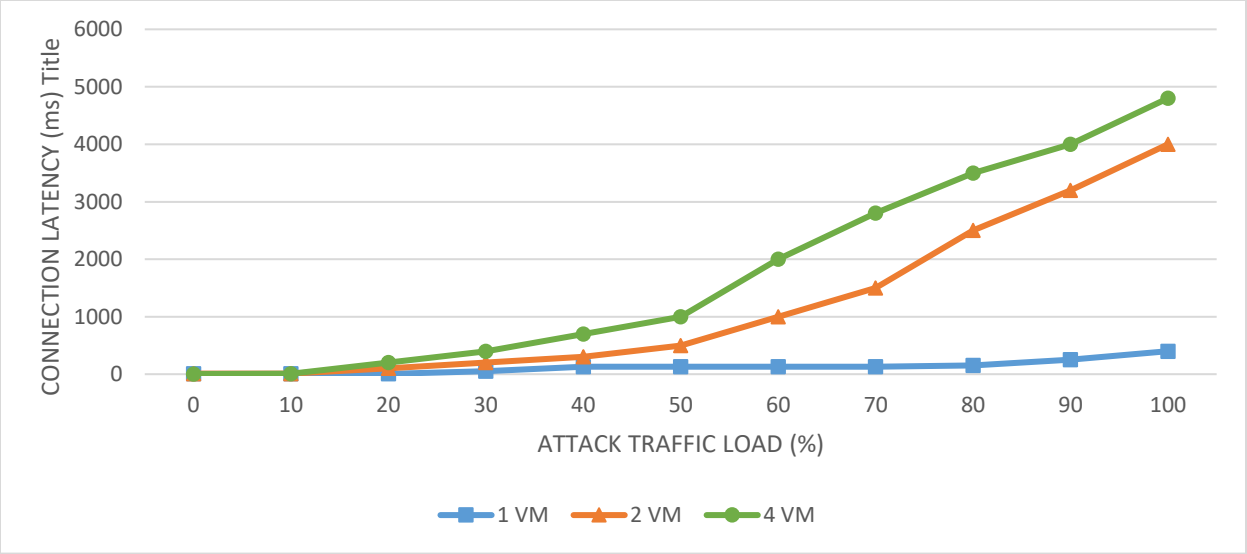

Figure 4.16HTTP Connection Latency under TCP-SYN Flood Attack when sent to Virtual Machines

Figure 4.17 shows the number of non-paged pool allocations under TCP-SYN flood attack traffic in case of one VM and four VMs. It shows the number of non-paged pool allocations were decreased while increase in number of Virtual Machines. It was directly proportional to HTTP connections per second.

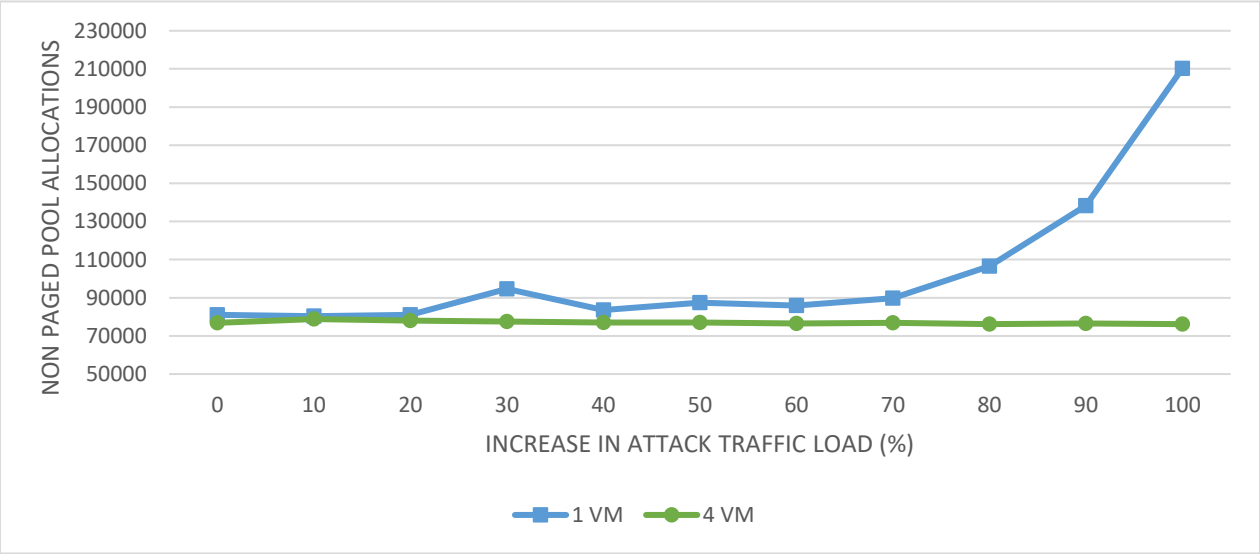

Figure 4.17Number of Non-Paged Pool Allocations under TCP-SYN flood Attack when sent to Virtual Machines

4.3.4 UDP Flood Attack

The User Datagram Protocol (UDP) is a connectionless computer networking protocol. The UDP is unlike TCP and there is no guarantee of delivering, ordering or duplicate protection. The UDP Flood Attack occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that the victim can no longer handle valid connections. The main intention of UDP Flood Attack is to freeze the internet pipe. In this UDP Flood Attack, an attacker sends UDP datagrams in IP packets with spoofed source addresses. These are all UDP datagrams targeting a DNS server. After reaching threshold limit of these datagrams, the DNS server will reject further UDP datagrams from all the addresses in the same security zone for the remainder of the current second. Because of this it will also reject legitimate UDP datagrams from an address in the same security zone.

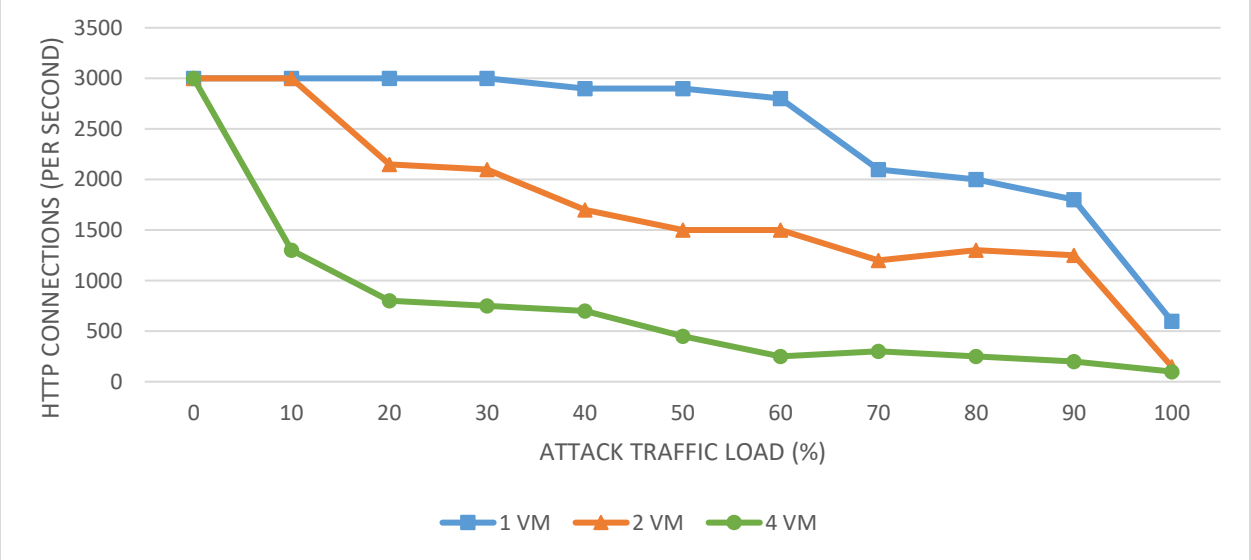

Figure 4.18 Number of HTTP connections established by the server under UDP Flood Attack when sent to Virtual Machines

Figure 4.18 shows the HTTP connection rate under UDP flood attack in all three scenarios. In the first scenario, the HTTP connection rate was not affected by the UDP flood attack until 600 Mbps. And then it declined throughout the experiment, it was reached approximately 600 HTTP connections at 1000 Mbps of attack traffic. Figure 4.19 shows individual core utilization of one Virtual Machine scenario. It shows first 5 cores were shared the load throughout the experiment after attack traffic was introduced and core5 was utilized almost 100%.

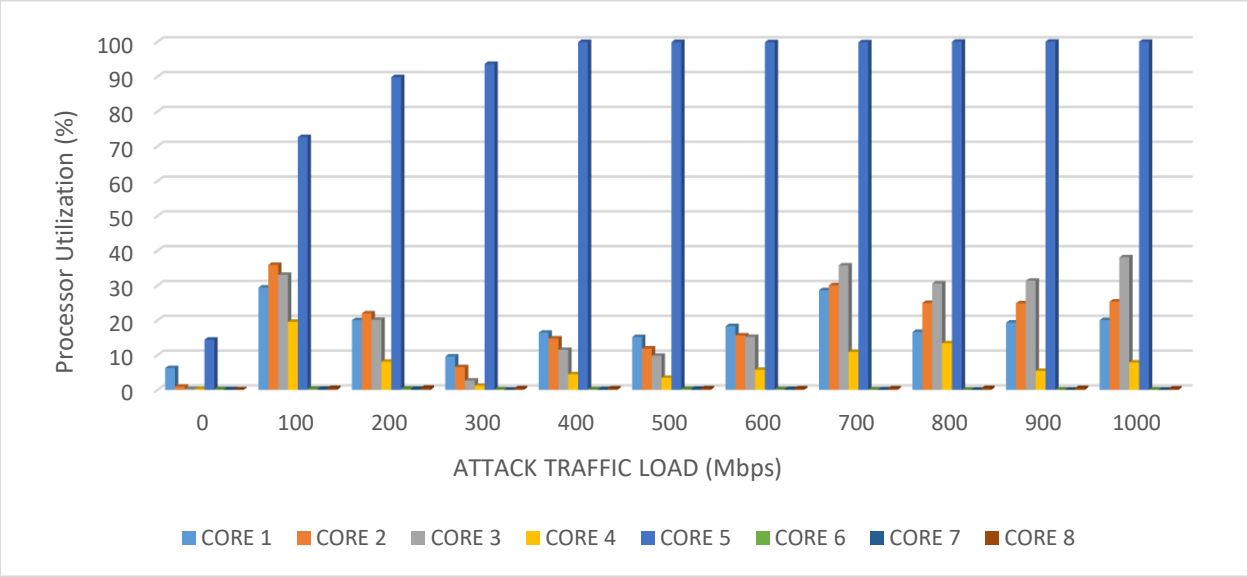

Figure 4.19 Individual Core Utilization under UDP Flood Attack when sent to one Virtual Machine

In the second scenario, the HTTP connections per second were started declining at 400 Mbps (20% of 2Gbps) of UDP flood attack traffic and at 2000 Mbps of attack traffic the number of HTTP connections were almost zero. Figure 4.20 shows individual core utilization of second scenario. It shows there was only one core (core 5) experiencing the total attack traffic load throughout the experiment.

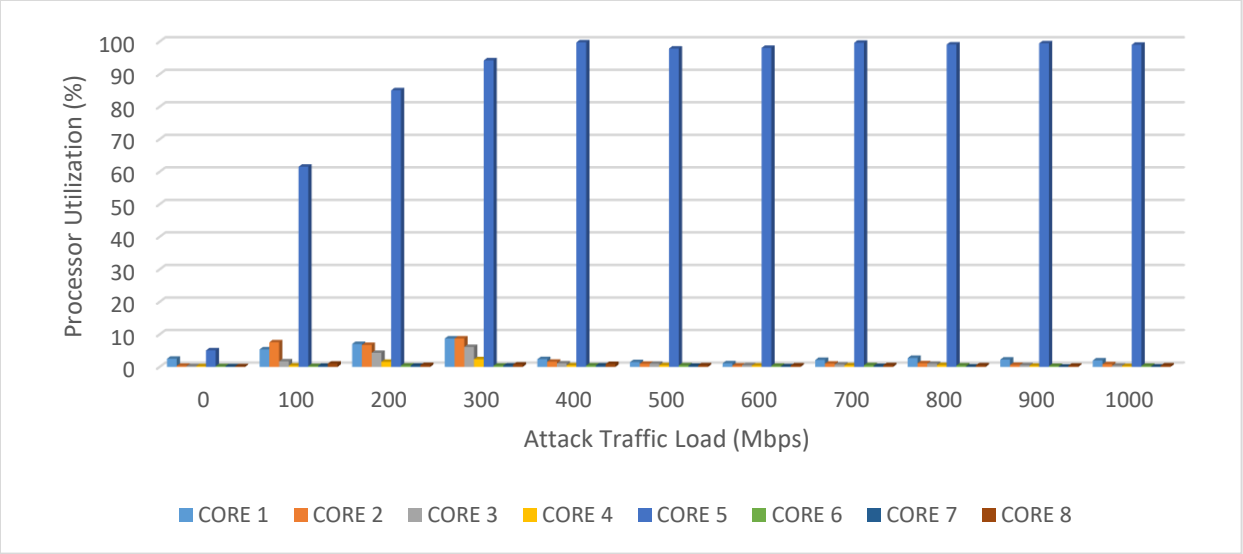

Figure 4.20 Individual Core Utilization under UDP Flood Attack of 2nd VM while using 2 Virtual Machines

In the third scenario, the number of HTTP connections per second started declining once the attack traffic was introduced. It has reached to 800 HTTP connections per second at 800 Mbps (20% of 4Gbps) of UDP flood attack to four Virtual Machines at the same time. And later it was continuously declined and there were no HTTP connections at 100% of UDP flood attack traffic. Figure 4.21 shows the average processor utilization in all three different scenarios. It shows the average processor utilization was decreasing while increasing the number of Virtual Machines. It was recorded the maximum of 25% processor utilization in case of one Virtual Machine. At 4 Gbps of attack traffic, the average processor utilization was approximately 7%.

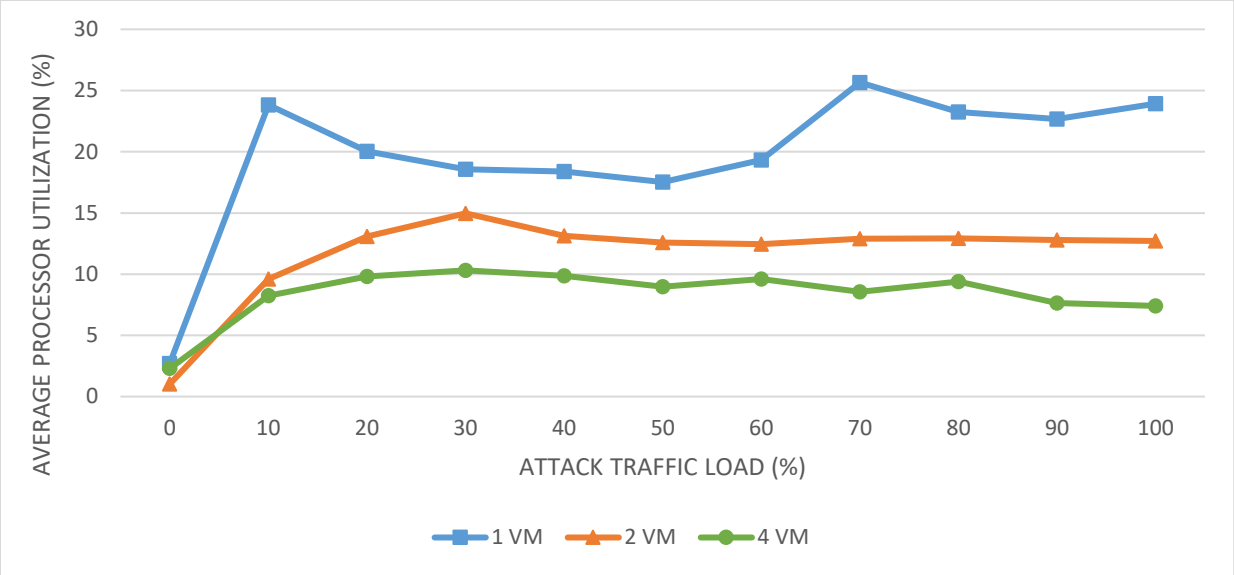

Figure 4.21Average Processor Utilization under UDP Flood Attack when sent it Virtual Machines

It observed that, the HTTP connection latency was increasing while increasing the number of Virtual Machines as shown in Figure 4.22 under UDP flood attack traffic. At 4 VM's scenario, it took 6 seconds to react to the client at 4 Gbps of UDP flood attack traffic.

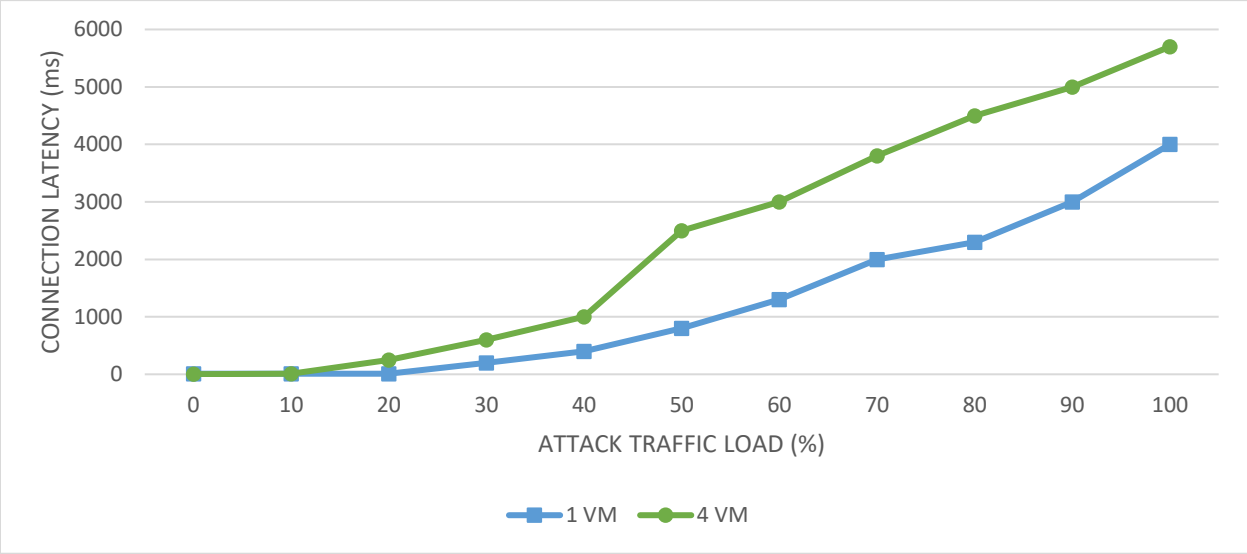

Figure 4.22HTTP Connection Latency under UDP Flood Attack when sent to Virtual Machines

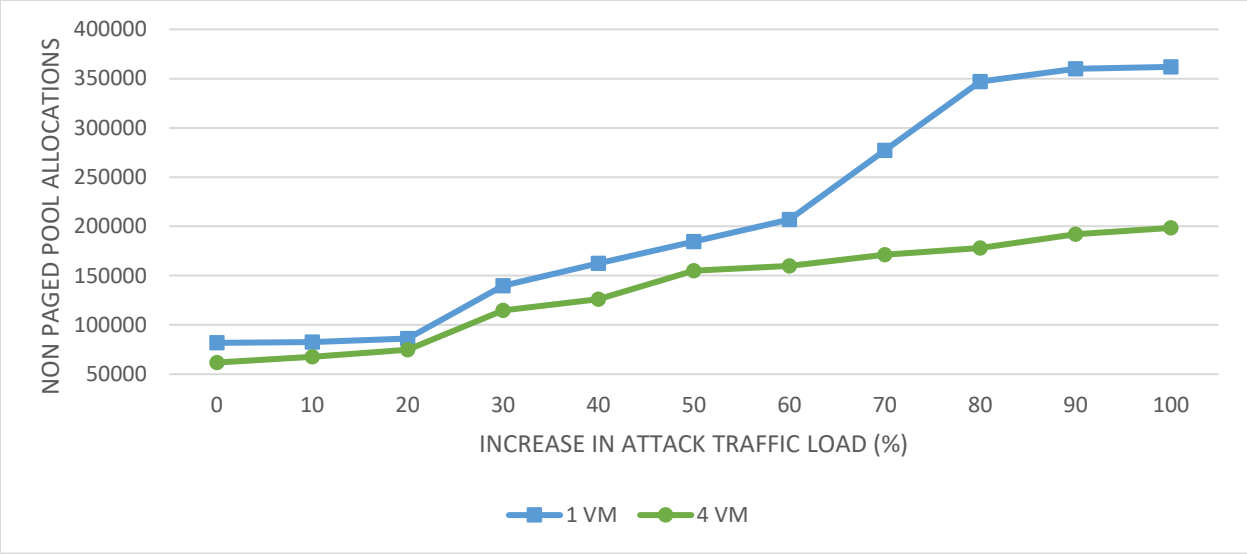

Figure 4.23Number of Non-Paged Pool Allocations under UDP Flood Attack when sent to Virtual Machines

Figure 4.23 shows the number of non-paged pool allocations under UDP flood attack traffic in case of one VM and four VM's. In the first scenario, the number of non-paged pool allocations were increasing while increasing the attack traffic. Once the attack traffic was sent to four Virtual Machines the number of non-paged pool allocations were reduced compared to one Virtual Machine.

4.4 Chapter summary

In this chapter, I virtualized the Windows Server 2012 R2 operating system on MAC hardware platform. I installed Hyper-V on Windows Server 2012 R2 operating system and I installed four Virtual Machines with same Windows Server 2012 R2 operating system on it. Later I sent different Distributed Denial of Service attacks to those Virtual Machines to evaluate the efficiency in terms of HTTP connection rate, memory utilization, processor utilization, Non-paged pool allocations and HTTP connection latency. I observed that under Smurf attack, the Virtual Machine is affected and crashed. If one Virtual Machine was attacked by security attack like DDoS attack, there will be more chance of getting attack traffic or vulnerability to other Virtual Machines that were installed on the same server. I observed that, when two or more Virtual Machines were affected by DDoS attacks there will be more impact on server as compared to one Virtual Machine alone.

CHAPTER V

COMPARISION BETWEEN VIRTUALIZED AND NON-VIRTUALIZED WINDOWS SERVER 2012 R2 OS ON MAC HARDWARE PLATFORM

In this chapter, I compared Virtualized Apple MAC PRO server having Windows Server 2012 R2 operating system with non-virtualized Apple MAC PRO server having same Windows Server 2012 R2 operating system.

5.1 Experimental Setup

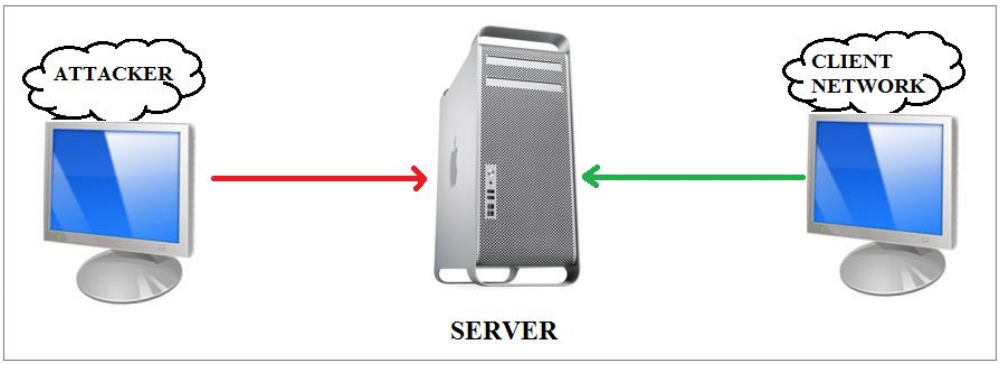

Figure 5.1 Experimental Setup for Non-Virtualization

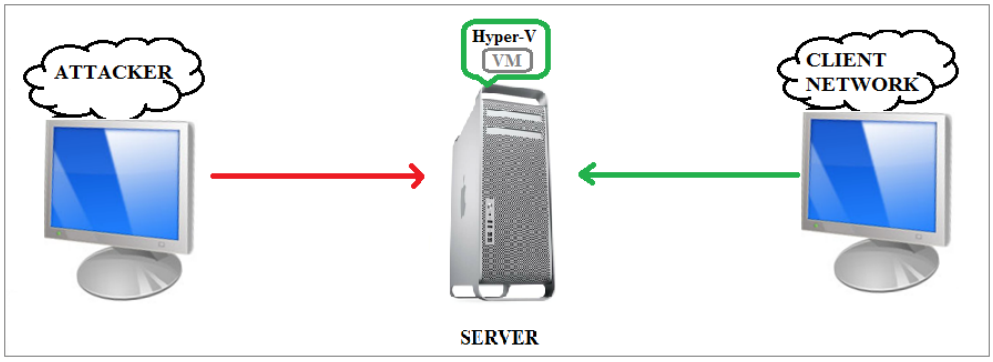

Figure 5.2 Experimental Setup for Virtualization

Initially, the Apple MAC PRO server having Windows Server 2012 R2 operating system was evaluated for its performance against different DDoS attacks which was sent to one external port of victim server. And the victim server with the same operating system was virtualized and evaluated for its performance against same DDoS attacks that were sent to one Virtual Machine installed on the victim server using Hyper-V application. The Non-Virtualization and Virtualization setups were shown in Figure 5.1 and Figure 5.2 respectively.

5.2 Comparison between Virtualized and Non-Virtualized Server

In this Chapter, I sent 3000 HTTP connection requests per second to the victim server in the both cases with and without virtualization and this was considered as baseline for the experiment and Once the baseline of the experiment established, the attack traffic was introduced to the victim server. Initially, attack traffic intensity of 100 Mbps was applied from simulated attack network for five minutes. Later, it was increased to 200 Mbps of attack traffic for five minutes and this process was continued until reached 1000 Mbps maximum bandwidth of Gigabit Ethernet adapter by increasing regular interval of 100 Mbps of attack traffic and run for regular interval of five minutes. This total experiment took fifty-five minutes to run.

5.2.1 Ping Flood Attack

Under Ping flood attack, the non-virtualized server started declining its HTTP connection rate at 600 Mbps of attack traffic load as shown Figure 5.3. Whereas for Virtualized server connections started at 900 Mbps of Ping attack traffic. At 1000 Mbps of Ping flood attack traffic, there were approximately 400 HTTP connections per second in case of non-virtualized server and 1000 HTTP connections in case of virtualized server. This shows that connection performance of Virtual server was better than non-virtualized server.

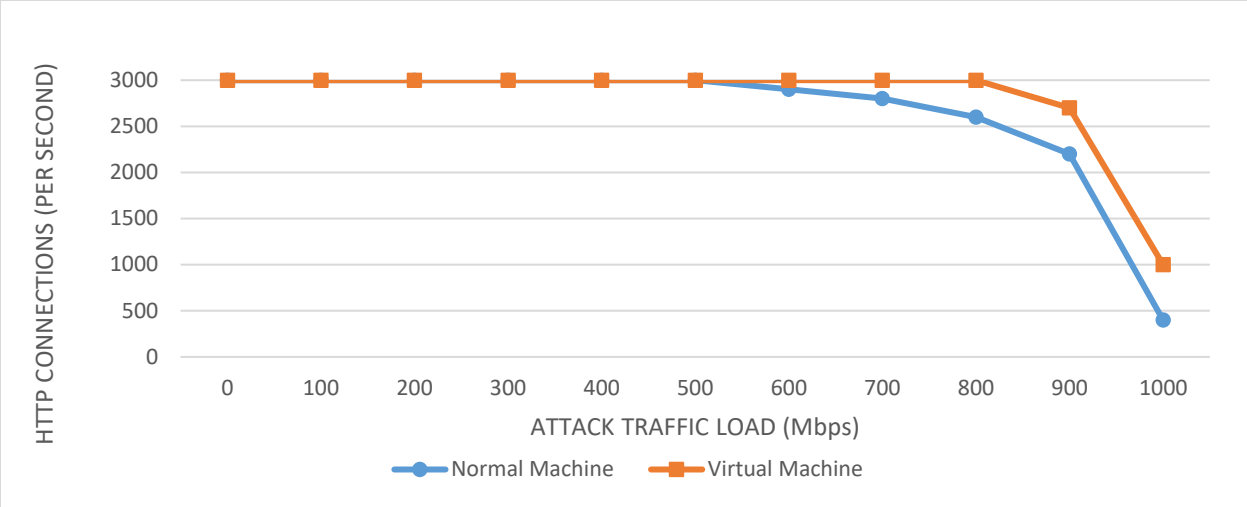

Figure 5.3 Comparison of HTTP connection rate in Virtualized and Non-Virtualized Server under Ping Flood Attack Traffic.

Figure 5.4 shows the comparison of virtualized and non-virtualized server of average processor utilization under ping flood attack. It shows normal machine used little bit more processor utilization as compared to virtual machine and normal used its maximum of 25% at 200 Mbps of ping attack traffic. The connection latency of normal server was more when compared to virtual server as shown in Figure 5.5.

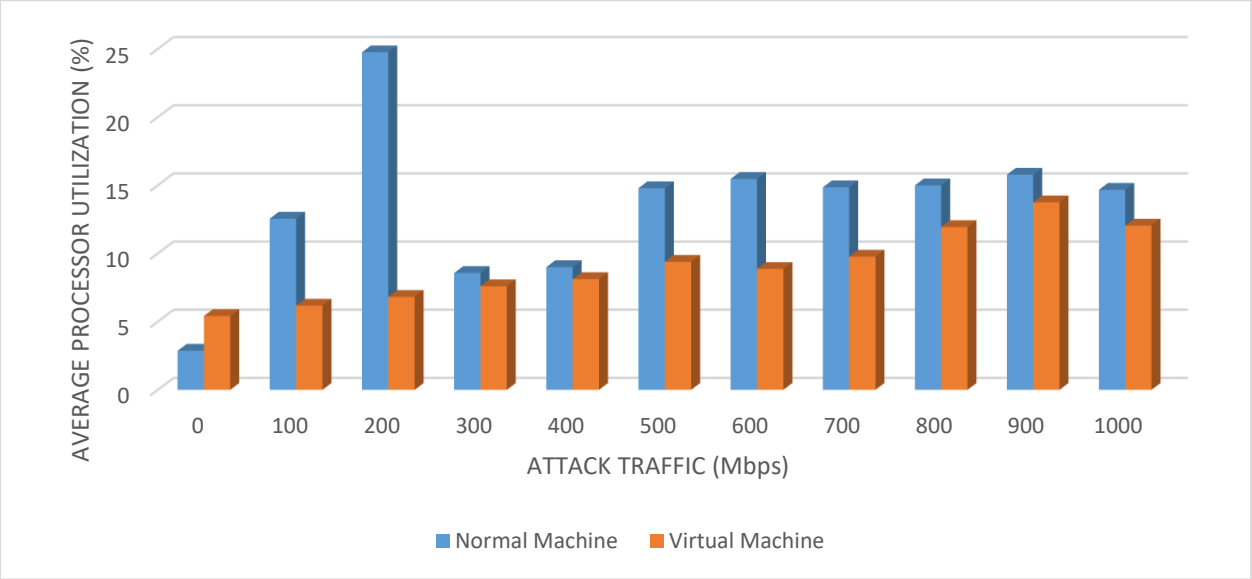

Figure 5.4. The Average processor utilization of Virtualized and Non-Virtualized Server under Ping Attack Traffic

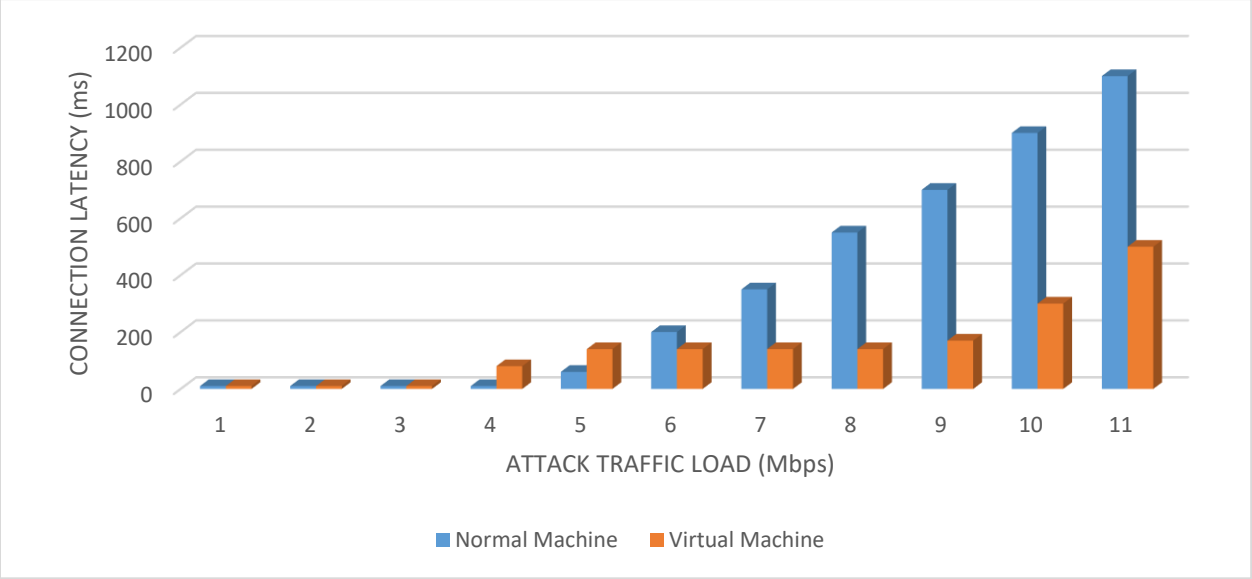

Figure 5.5. Comparison of HTTP connection latency in Virtualized and Non-Virtualized Server under Ping Attack Traffic.

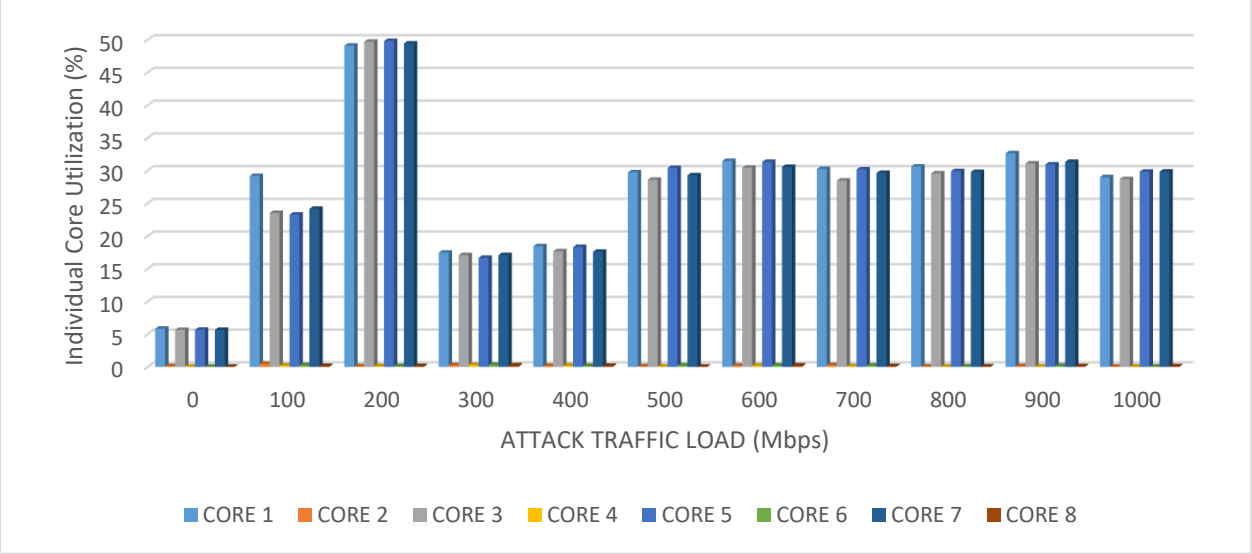

Figure 5.6. The Individual core utilization of Virtualized Server under Ping Attack Traffic

Figure 5.6 and figure 5.7 shows the individual core utilization of non-virtualized server and virtualized server respectively under Ping attack traffic. In the non-virtualized server, four cores

(Core1, Core3, Core5, Core7) were sharing the traffic load. Whereas in virtualized server, one core (Core5) was experienced most of the load. once attack traffic was introduced and all other cores were using below 40% of their respected cores.

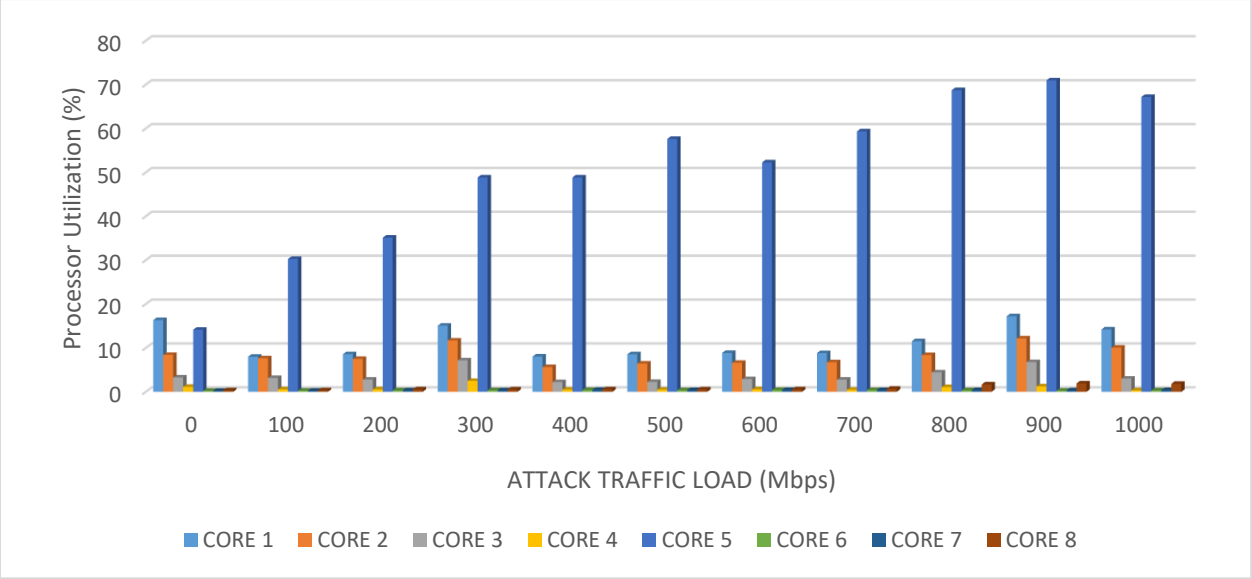

Figure 5.7. The Individual core utilization of Non-Virtualized Server under Ping Attack Traffic

5.2.2 Smurf Attack

Under Smurf attack traffic, the HTTP connection rate was started declining at 100 Mbps of attack traffic load in case of virtual server and there were approximately 900 HTTP connections per second established by the virtual server at 200 Mbps of attack traffic load. And at higher attack traffic, the connection rate was keep declining in the virtual server. Whereas in case of non-virtualized server, the HTTP connection rate was started declining at 300 Mbps of attack traffic load. Approximately there were 1750 HTTP connections per second established by the non-virtual server and 400 HTTP connections per second at 1 Gbps of attack traffic load. Figure 5.8 shows that; the virtual server has affected more by Smurf attack traffic.

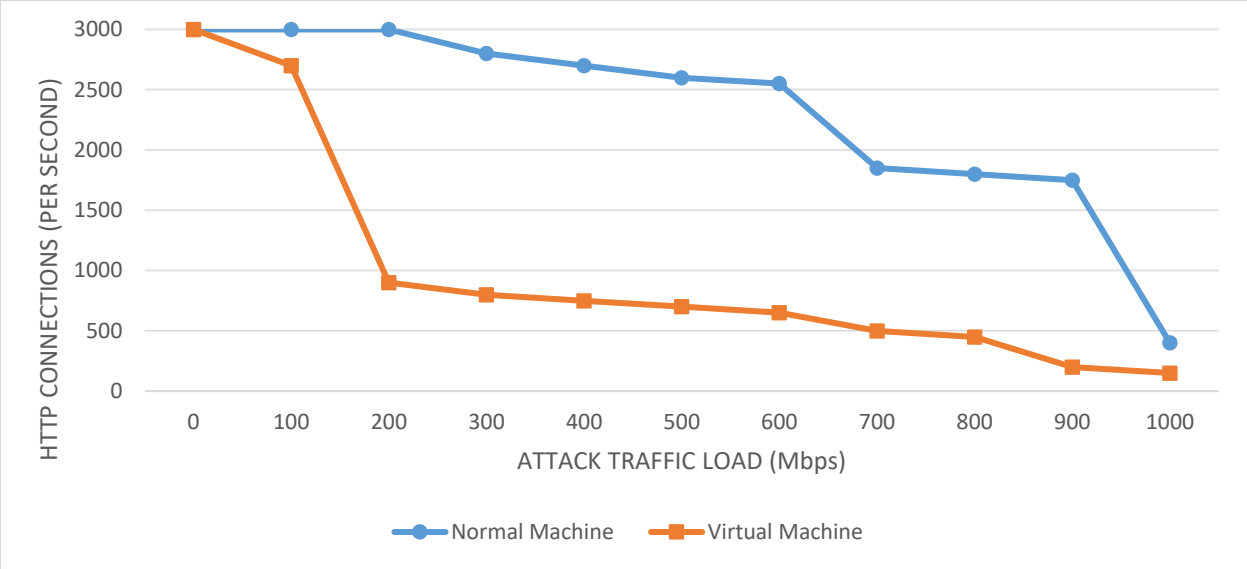

Figure 5.8 Comparison of HTTP connection rate in Virtualized and Non-Virtualized Server under Smurf Attack Traffic.

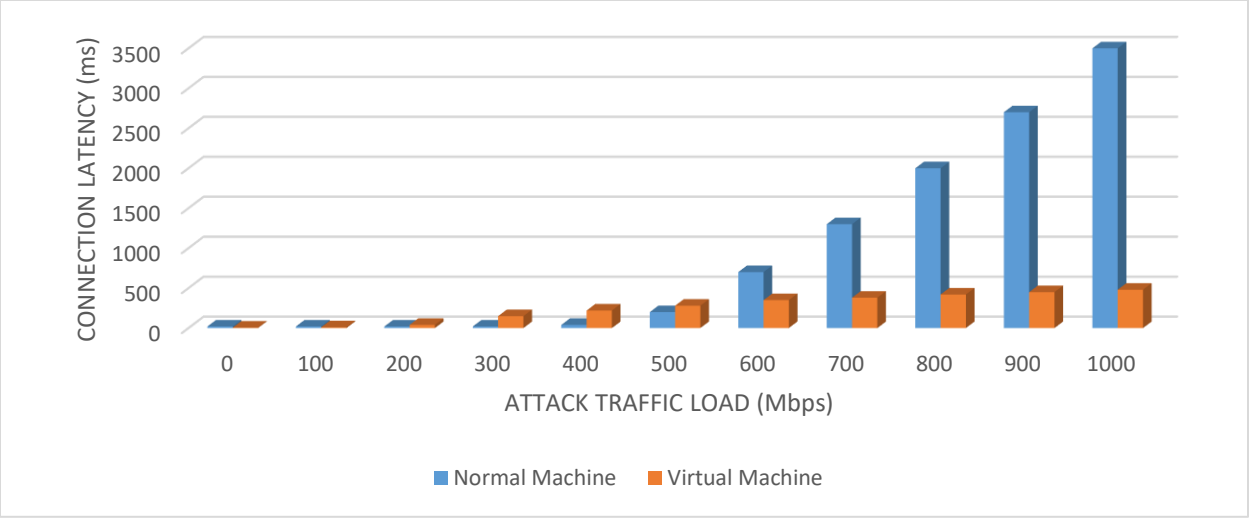

Figure 5.9 Comparison of HTTP connection latency in Virtualized and Non-Virtualized Server under Smurf Attack Traffic.

Above Figure 5.9 shows the HTTP connection latency in virtualized server and non-virtualized server under Smurf attack traffic. The connection rate was increasing with the attack traffic load in case of non-virtualized server and virtualized server as well. It was observed that, the connection latency was 3500 milliseconds in case of non-virtualized server and 480 milliseconds in case of virtualized server.

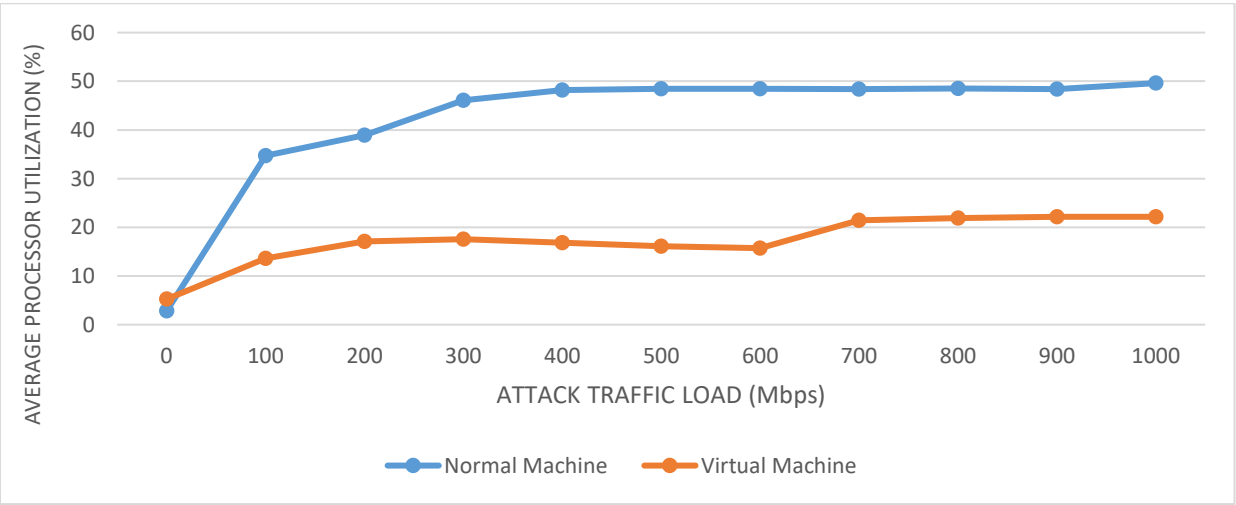

Figure 5.10 The Average processor utilization under Smurf attack by Virtualized and Non-Virtualized Server

The average processor utilization of virtualized server and non-virtualized server under Smurf attack traffic was shown in above Figure 5.10. The average processor utilization was increasing with the attack traffic load in both the cases. Approximately 20% of average processor utilized by the virtualized server and 48% was utilized by the non-virtualized server at 1000 Mbps of Smurf attack traffic load.

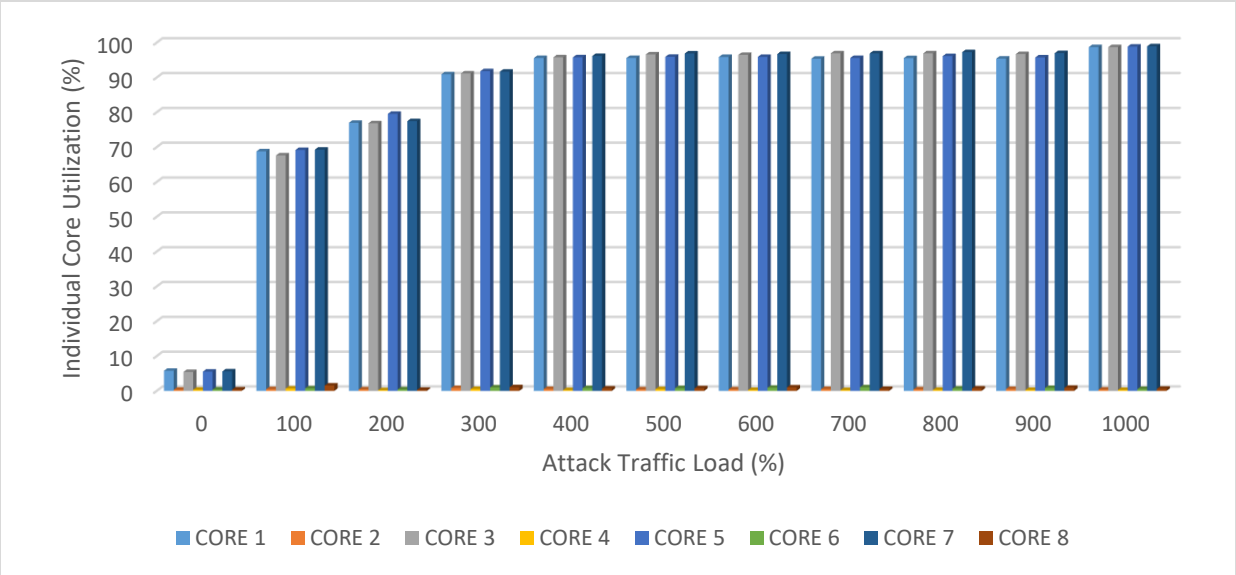

Figure 5.11 The Individual core utilization of Non-Virtualized Server under Smurf attack

Figure 5.11 and Figure 5.12 shows the individual core utilization of non-virtualized server and virtualized server respectively under Smurf attack traffic. In the non-virtualized server, four cores (Core1, Core3, Core5, Core7) were sharing the traffic load. Whereas in virtualized server, one core (Core5) was experiencing 100% once attack traffic was introduced and all other cores were using below 40% of their respected cores.

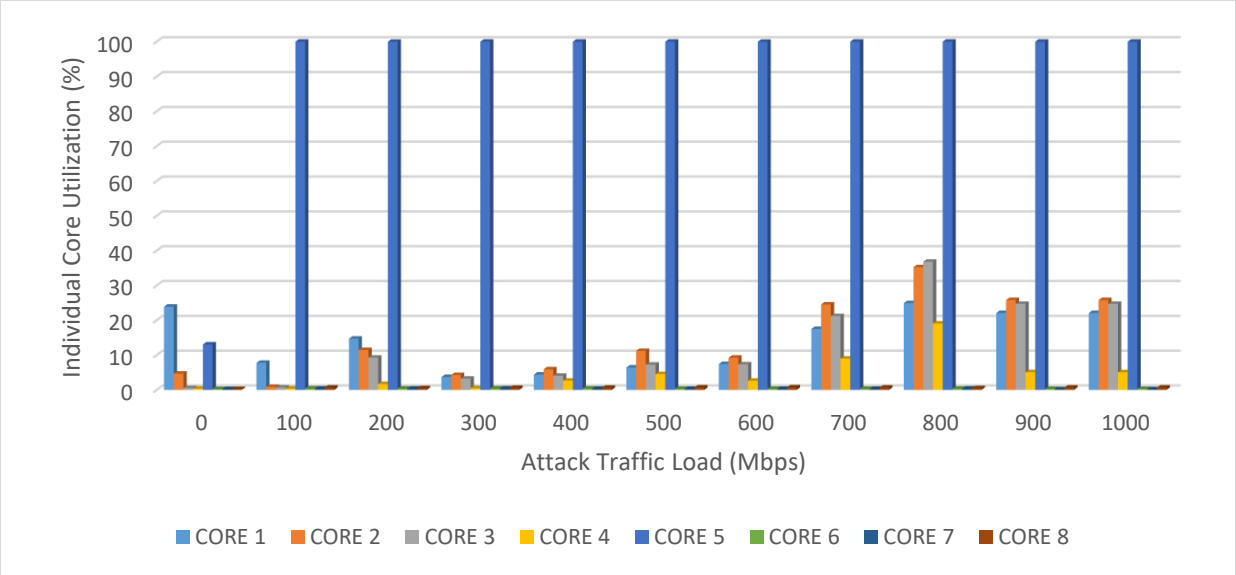

Figure 5.12 The Individual core utilization of Virtualized Server under Smurf attack

5.2.3 TCP-SYN Attack

Under TCP-SYN flood attack traffic, the number of HTTP connections per second were 3000 in both virtualized and non-virtualized server until 400 Mbps of attack traffic load as shown in Figure 5.13. At 500 Mbps of TCP-SYN attack traffic, the number of HTTP connections per second were started declining in case of non-virtualized server. Whereas there were 3000 HTTP connections per second established by the virtualized server until 700 Mbps of TCP-SYN flood attack traffic. At 1 Gbps of attack traffic load, approximately there were no HTTP connections established by the on-virtualized server and 1200 HTTP connections per second were established by the virtualized server. It shows the security against TCP-SYN flood attack traffic was better in the virtual server as compared to non-virtual machine.

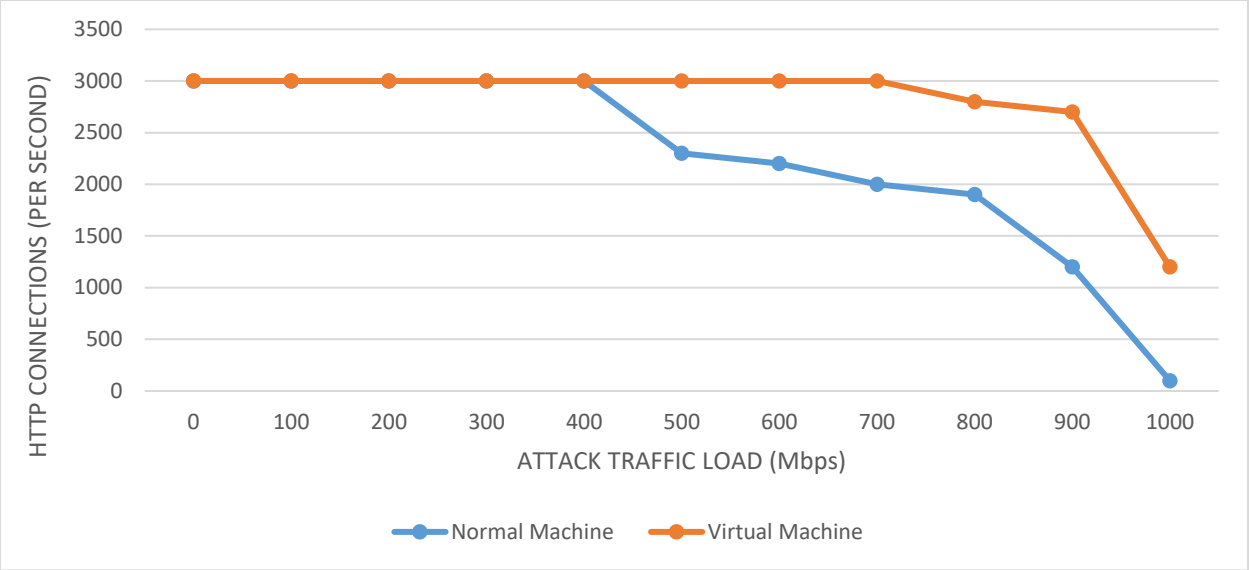

Figure 5.13 Comparison of HTTP connection rate in Virtualized and Non-Virtualized Server under TCP-SYN Flood Attack Traffic.

5.2.4 UDP Flood Attack

Figure 5.14 shows the HTTP connections per second were established by the Virtualized and non-virtualized server under UDP flood attack traffic. Approximately there were full connections established by the server in the both cases until 500 Mbps of UDP flood attack traffic load. And later, the HTP connection rate was declined gradually in both cases. At 1 Gbps of attack traffic there were 250 HTTP connections established by the non-virtualized server and 600 HTTP connections established by the virtual server. It shows under UDP flood attack traffic, there was similar impact on HTTP connection in both the cases.

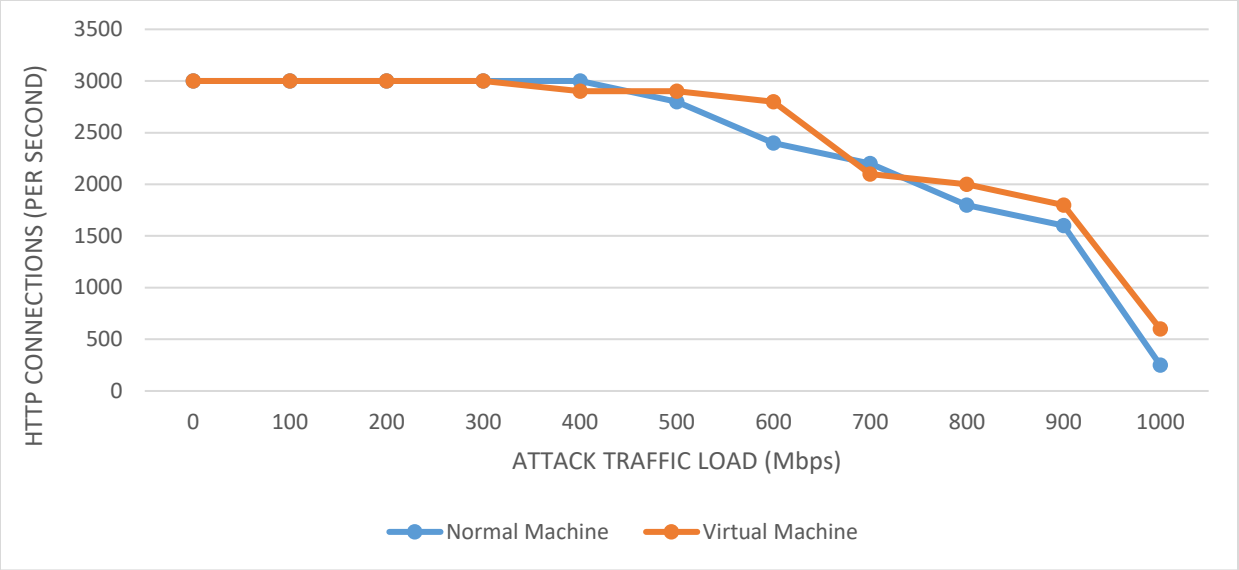

Figure 5.14 Comparison of HTTP connection rate in Virtualized and Non-Virtualized Server under UDP Flood Attack Traffic.

Figure 5.15 shows the HTTP connection latency against different DDoS attacks in both virtualized and non-virtualized server. The connection latency was more in case of virtual server and approximately virtualized server took 4 seconds to respond the client request at 1 Gbps of UDP attack traffic load.

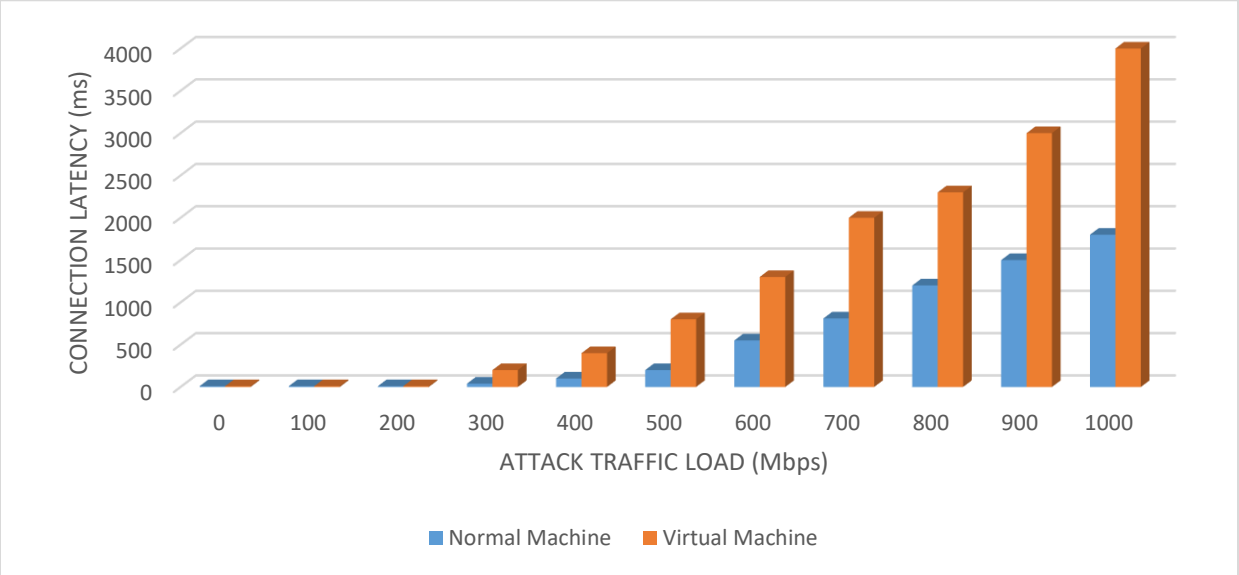

Figure 5.15 Comparison of HTTP connection latency in Virtualized and Non-Virtualized Server under UDP Flood Attack Traffic

5.3 Comparison of Virtual Machines

Here I compared the one Virtual Machine results with four Virtual Machines results against different Distributed Denial of Service (DDoS) attacks in terms of HTTP connection rate, memory utilization, average processor utilization and non-paged pool allocations.

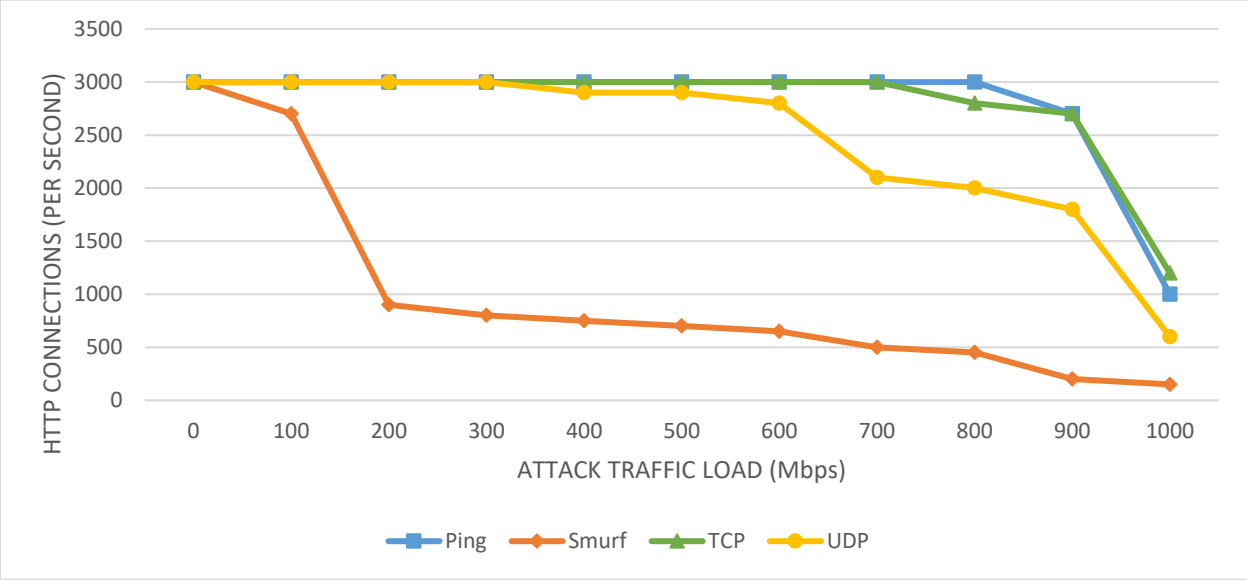

Figure 5.16 Number of HTTP connections under different DDoS attacks when sent to one Virtual Machine

The Figure 5.16 shows the number of HTTP connections were established under different Distributed Denial of Service attacks when sent to one Virtual Machine. There was no much impact on HTTP connection rate in case of Ping Flood attack and TCP-SYN flood attack. Where as in case of UDP flood attack, the connection rate was started declining at 600 Mbps of attack traffic load. In case of Smurf attack, the connection rate was declined to approximately 900 at only 200 Mbps of attack traffic load And later it was keep declining while increasing the attack traffic load and it was reached to 150 HTTP connections per second at 1000 Mbps of traffic. It seems there was no built-in protection for Smurf based DDoS attack on virtualized windows server 2012 R2 Operating system on MAC hardware platform.

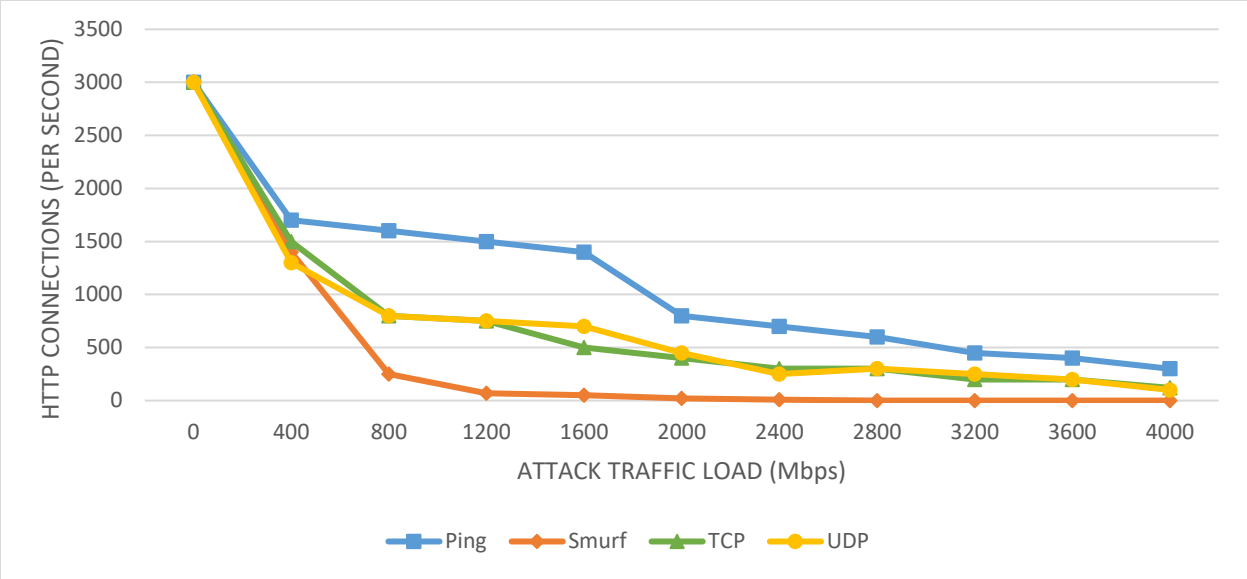

Figure 5.17 Number of HTTP connections under different DDoS attacks when sent to four Virtual Machine

Figure 5.17 shows the number of HTTP connections established under different DDoS attacks sent to four Virtual Machines. It shows the number of HTTP connections per second were dropped under all DDoS attacks once the attack traffic was introduced. Under Ping attack traffic, there was still 400 connections at 4000 Mbps of attack traffic load whereas under other attacks there were no HTTP connections at 4000 Mbps traffic load. At 1600 Mbps of Smurf attack traffic, the connection rate was almost zero. If compared this results with one Virtual Machine results, the Smurf attack crashed the server very quickly in case of four Virtual Machines. And also, there was huge change in all other DDoS attacks. I observed, if the more number of Virtual Machines were affected by cyber-attack, there will be a more impact on HTTP connection rate and even virtual machine crashed in some of attacks.

The Figure 5.18 shows the average processor utilization of one VM under four different DDoS attacks. It can be observed that, Smurf and UDP flood attacks were more used processor utilization, the average of 20% was used under both the attacks. Whereas under Ping and TCP flood attack traffic, the average processor utilization was under 10% even after attack traffic was

introduced. In case of four VM's the average processor utilization was less as compared to one VM. Under Smurf attack, the average processor utilization of 4th virtualized server while using 4 VMs was approximately 12% throughout the experiment because of one core out of 8 cores was completely utilized. Like one VM, under ping and TCP attack there was not much impact on processor usage as shown in Figure 5.19.

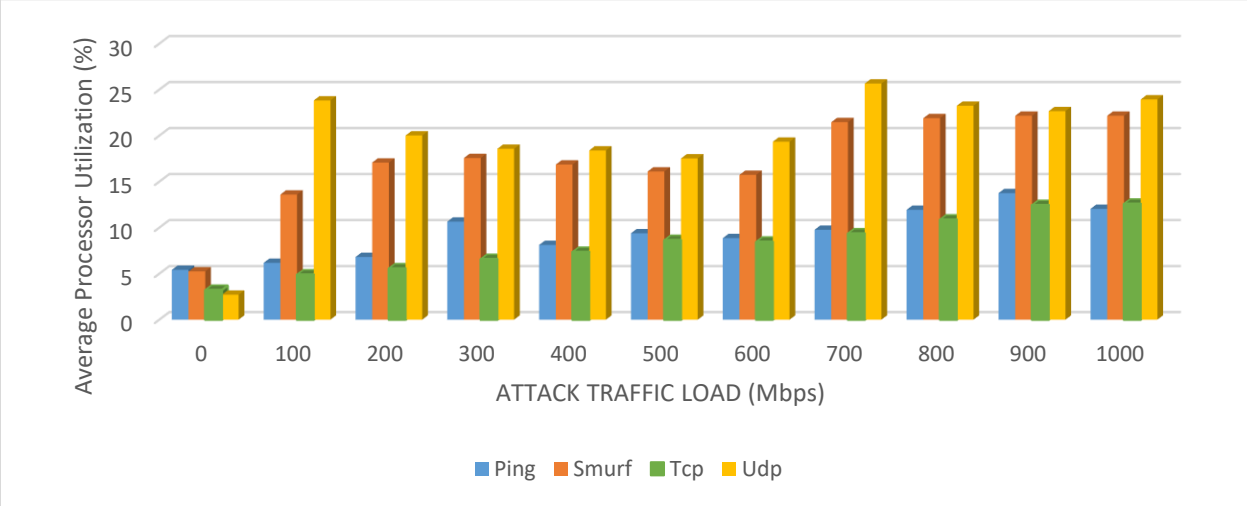

Figure 5.18 Average processor utilization of one Virtual Machine under different DDoS attacks

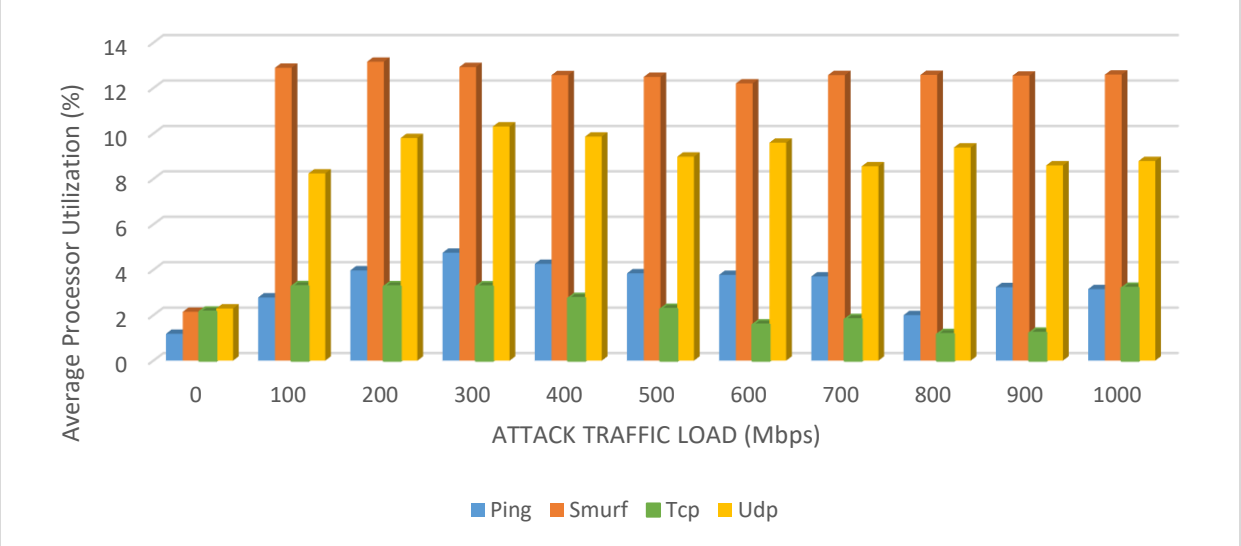

Figure 5.19 Average processor utilization of 4th VM while using four Virtual Machines under different DDoS attacks

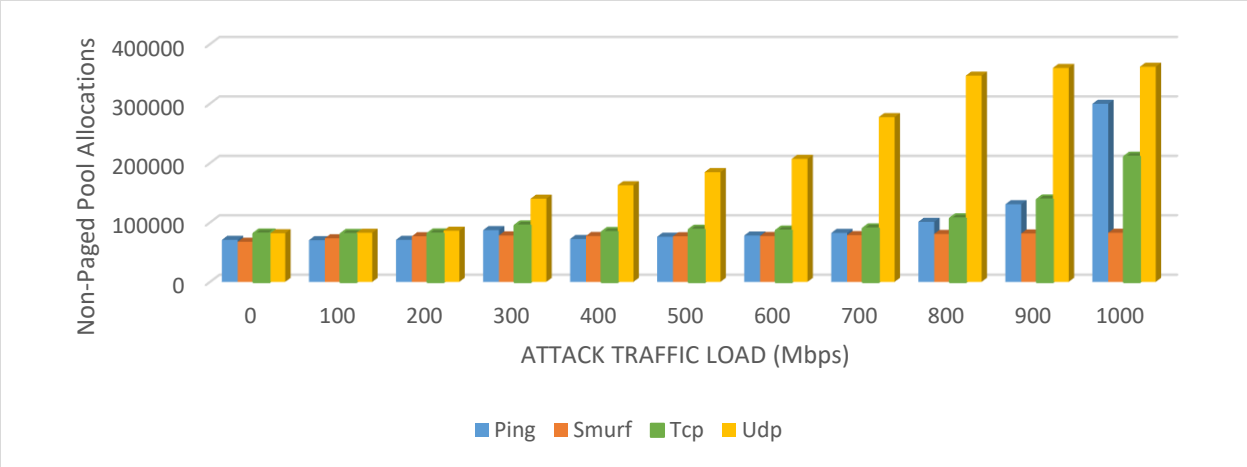

Figure 5.20 Number of Non-Paged Pool Allocations of one Virtual Machine under different DDoS attacks

Figure 5.20 shows the number of non-paged pool allocations of one VM under different DDoS attacks. It can be observed that, only UDP has major impact on server with respect to number of non-paged pool allocations. Whereas in 4 VM scenario, both TCP and UDP got affected by non-paged pool allocations as shown inn figure 5.21.

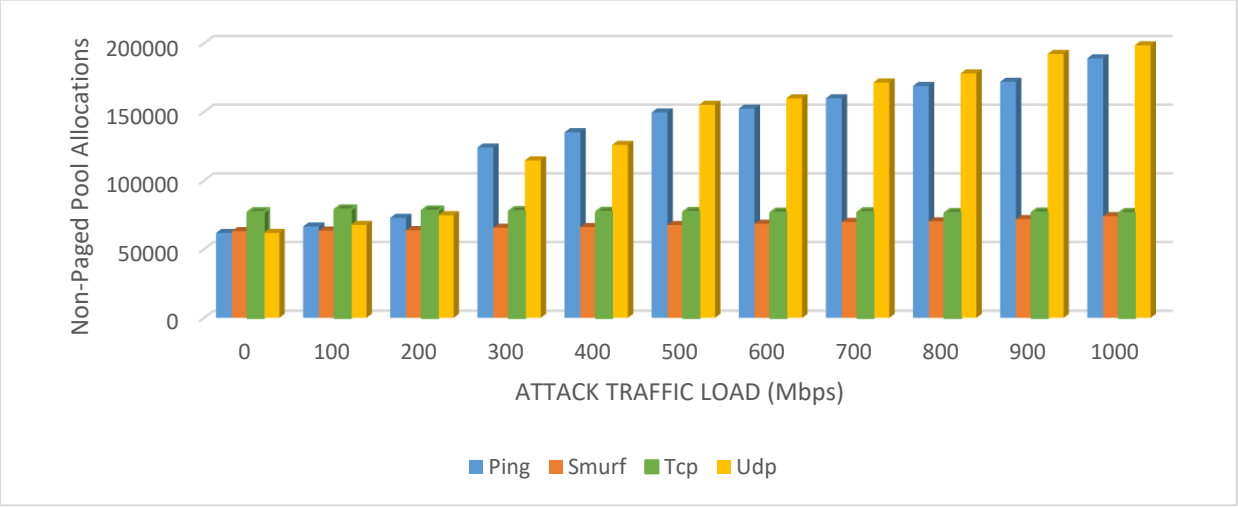

Figure 5.21 Number of Non-Paged Pool Allocations of 4th VM while using four Virtual Machines under different DDoS attacks

5.4 Mixed DDoS Attacks

Here, I sent four different DDoS attack traffic to non-virtualized server using four Broadcom gigabit ethernet ports. And HTTP clients were shared to all four ports by the equal amount i.e., 750 connections per each port. In this experiment I used Ping Flood attack, Smurf attack, TCP-SYN Flood attack, UDP Flood attacks.

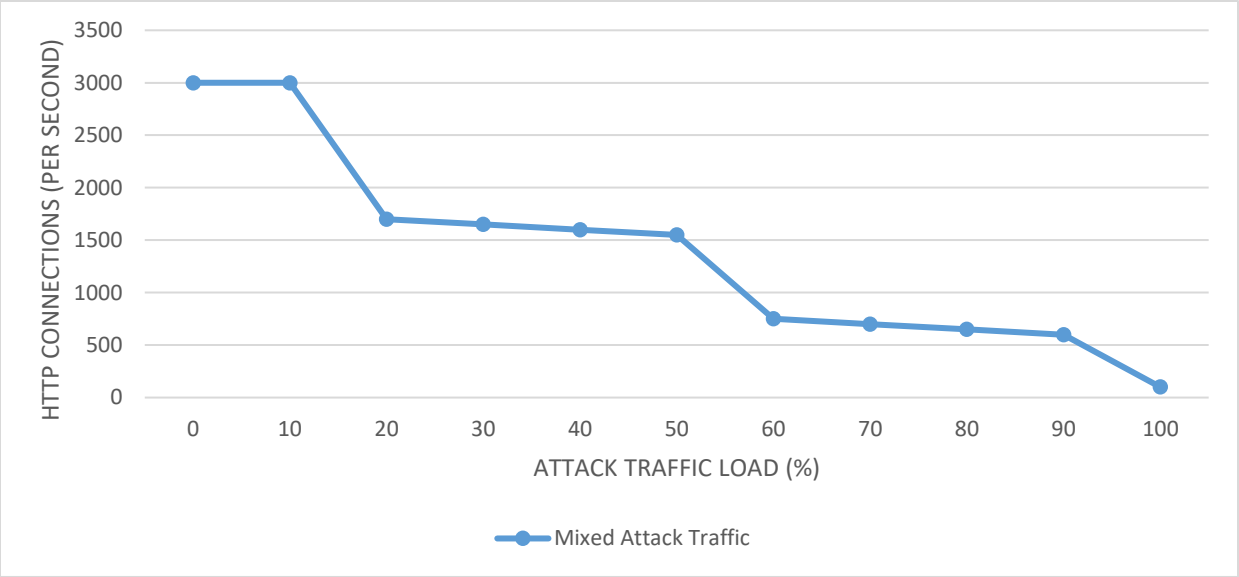

Figure 5.22 The number of HTTP connections per second established by the Non-Virtualized Server under four Mixed DDoS attacks

The HTTP connection rate was started declining at 30% of mixed attack traffic was shown in Figure 5.22. At 100% of mixed attack traffic load, there were only 100 HTTP connections per second established by the non-virtualized victim server. Maybe because of Smurf attack traffic, the connection rate was declined at 20% of attack load because we saw that Smurf attack caused more damage than other DDoS attacks. The individual core utilization was also similar to the behavior of Smurf attack traffic as shown in Figure 5.23. Once the attack traffic was introduced, the four cores (core1, core3, core5, core7) were experienced the load of 100% and remaining four cores were not participated in the core utilization.

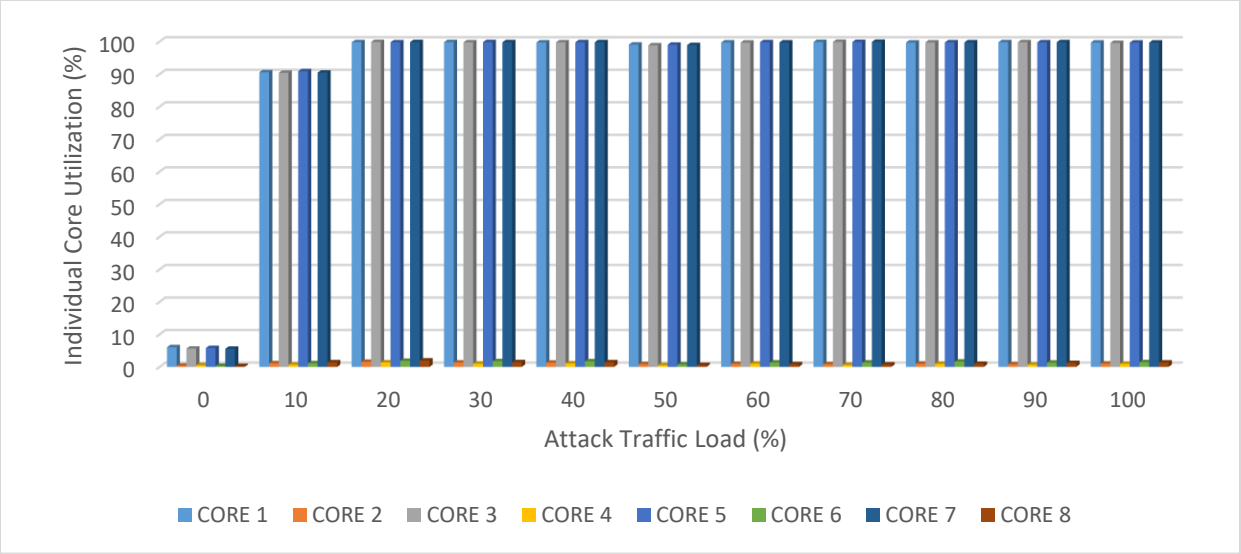

Figure 5.23 The Individual core utilization of Non-Virtualized Server under four Mixed DDoS attacks

5.5 Chapter Summary

In this chapter I observed, the security in the Virtualized server against Smurf attack traffic needs to be improved. Except Smurf attack traffic, the Virtualized server having better security than non-virtualized server. If one Virtual machine was compromised and there will be more probability of affecting other virtual machines that were installed on the same server. In this case if more virtual machines affected by the attack traffic, the HTTP connection rate was declining very quickly. In case of mixed attacks, the server was almost getting crashed at the 100 % of attack traffic load.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this thesis, the security performance of Apple MAC PRO server hardware platform with two leading operating systems were evaluated, which were Windows Server 2012 R2 operating system and MAC OS X SERVER LION 10.7.5 operating system against four different DDoS attacks. It was observed that under the Smurf attack traffic of 150Mbps, the Windows Server 2012 R2 OS on Apple MAC PRO server hardware crashed whereas the same didn't happen with other DDoS traffic types. It seemed there was not enough prevention mechanism deployed against Smurf based security attacks in Windows Server 2012 R2 OS on MAC PRO server hardware. It was also found that in the case of Smurf traffic, there was only one processor core that was handling total attack traffic load, which caused it to be fully utilized, resulting in the reduction of the overall HTTP connection rate. Whereas in the case, MAC OS on MAC PRO server platform, it performed worse than Windows Server 2012 R2, and its connection rate declined to zero under TCP-SYN flood attack traffic of 200 Mbps which is rather a small bandwidth for the server to handle.

In this thesis, we also evaluated two different NIC cards, the built in NIC from Apple Inc, and an external NIC card from Broadcom. Unlike the previous experiments, the Window Server OS 2012 R2 didn't crash under Smurf attack traffic when the external NIC card from Broadcom was used. The performance of Windows Server 2012 R2 operating system on Apple MAC PRO

server hardware having 4-port Broadcom gigabit ethernet NIC adapter was evaluated. It was observed that the effect of class C and class B networks was not similar even though the intensity of attack traffic was equal. This could have been due to more hosts in class B which could have affected victim server more than that of the class C network.

Additionally, In this thesis, we evaluated the performance of virtualized the Windows Server 2012 R2 operating system on Apple MAC PRO server hardware platform which used Hyper-V for virtualization along with the external Broadcom gigabit Ethernet NIC card adapter. For virtualization, up to four VMs were used for experiments. In this experimental evaluation, the performance of Virtualized and Non-Virtualized Windows Server 2012 R2 on Apple MAC PRO server hardware platform was measured. For Smurf attack traffic, virtualized server couldn't support any connections whereas for other traffic types, the virtual server provided better connections compared to non-virtualized server.

REFERENCES

[1]. US Issues First Government Guide on Responding to Cyber-Attacks, Available online at http://gadgets.ndtv.com/internet/news/us-issues-first-government-guide-on-responding-to-cyber-attacks-866040 last access on: , 2017.

[2]. Possible Data Exposure for Hunting and Fishing License Online Sales System, Available online at http://www.vtfishandwildlife.com/cms/One.aspx?portalId=73163&page-Id=4198627 last access on: 2017.

[3]. Massive DDoS Attack on a Chinese gambling site could be the largest recorded assault ever, Available online at http://www.ibtimes.co.uk/largest-ddos-attack-ever-massive-470gbps-assault-hits-chinese-gambling-site-1568511 last access on: 2017.

[4]. Lizard Squad hacker mocks games after alleged DDoS attack on Blizzard servers, Available online at http://www.ibtimes.co.uk/lizard-squad-hacker-mocks-gamers-after-alleged-ddos-attack-blizzard-servers-1566580 last access on: 2017.

[5]. Major DNS provider hit by mysterious, focused DDoS attack, Available online at http://arstechnica.com/information-technology/2016/05/major-dns-provider-hit-by-mysterious-focused-ddos-attack/ last access on: 2017.

[6]. Statement from Randy Livingston regarding compromised tax data, Available online at http://news.stanford.edu/2016/04/08/tax-issue-announce-040816/ last access on: 2017.

[7]. Ransomware attacks on Hospitals put patients at risk, Available online at http://thehackersnews.com/2016/04hospital-ransomware.html last access on: 2017.

[8]. Opeyemi Osanaiye, Kim-Kwang Raymond Choo, Mqhele Dlodlo, "Distributed Denial of service resilience in cloud: Review and conceptual cloud DDoS mitigation framework" Journal of Network and Computer Applications, 2016.

[9]. Denial of Service Attacks, Available online at https://www.cnet.com/news/how-a-denial-of- -service-attack-works/ last access on: 2017.

[10]. Server Virtualization, Available online at http://searchservervirtualization.techtarget.com/definition/server-virtualization last access on: 2017.

[11]. DDoS Attacks, Available online at https://burmabit.wordpress.com/2014/04/22/dos-attack/ last access on: 2017.

[12]. Understanding Man-in-the-Middle Attacks - ARP Cache Poisoning, Available online at http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html last access on: 2017.

[13]. Gade, R.S.R., Vellalacheruvu, H. and Kumar, S. (2010) Performance of Windows XP, Windows Vista and Apple's Leopard Systems under a DDoS Attack. International Conference on Digital Society (ICDS'10).

[14]. Kumar, S. (2007) Smurf-Based Distributed Denial of Service (DDoS) Attack Amplification in Internet. 2nd International Conference on Internet Monitoring and Protection (ICIMP), San Jose, 1-5 July 2007, 25.

[15]. S. Kumar, "PING Attack – How Bad Is It?" Elsevier *Computers & Security Journal*, vol. 25, no. 5, July 2006, pp. 332-337, DOI: https://doi.org/10.1016/j.cose.2005.11.004 ,

[16]. Smurf Attack, Available online at http://en.wikipedia.org/wiki/Smurf_attack last access on: December 2017.

[17]. Ferguson, P. and Senie, D. (2000) Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. RFC 2827, BCP 38.

[18]. Vellalacheruvu, H.K. and Kumar, S. (2011) Effectiveness of Built-In Security Protection of Microsoft's Windows Server 2003 against TCP SYN Based DDoS Attacks. Journal of Information Security, 2, 131-138. https://doi.org/10.4236/jis.2011.23013

[19]. S. Kumar and E. Petana, "TCP protocol attacks on Microsoft's Windows XP-based Computers," International Conference on Networking, pp. 238-242, April 2008. Available from IEEE online library Xplore

[20]. Ganesh Reddy Gunnam and Sanjeev Kumar, (2017) Do ICMP Security Attacks Have Same Impact on Servers? Journal of Information Security, 8, 274-283. https://doi.org/10.4236/jis.2017.83018

[21] Einar Petana and S. Kumar, "TCP SYN-based DDoS attack on EKG signals monitored via a wireless sensor network," Wiley Journal of Security and Communication Networks, vol.4, no.12, Dec 2011, pp. 1448-1460, Online ISSN: 1939-0122; Online DOI: 10.1002/sec.275

[22]. Transmission Control Protocol, TCP, Available online at https://en.wikipedia.org/wiki/Transmission_Control_Protocol last access on: 2017.

[23]. User Datagram Protocol, UDP, Available online at https://en.wikipedia.org/wiki/User_Datagram_Protocol last access on: 2017.

[24]. Understanding UDP Flood Attacks, Available online at http://www.juniper.net/documentation/en_US/junos15.1x49/topics/concept/denial-of-service-network-udp-flood-attack-understanding.html last access on: 2017.

[25]. Virtualization, Available online at https://en.wikipedia.org/wiki/Virtualization last access on: 2017.

[26]. Text book: Negus software solutions series, Practical Virtualization Solutions by Kenneth Hess and Amy Newman, 2009

[27]. Text book: Microsoft Virtualization Secrets by John Savill , John Wiley & Sons, Inc. , 2012

[28]. Cisco's Cloud Strategy for Cloud Providers, Available online at http://www.cisco.com/c/dam/global/es_mx/solutions/cloud/at-a-glance-c45-730761.pdf last access on: 2017.

[29]. Cisco Cloud Web Security, Available online at http://www.cisco.com/c/dam/en_us/about/ciscoitatwork/borderless_networks/docs/Cloud_Web_Security_IT_Methods.pdf last access on: 2017.

[30]. Text book: VIRTUALIZATION ESSENTIALS by Matthew Portnoy, 2012

[31]. vSphere Hypervisor, Available online at https://www.vmware.com/products/vsphere-hypervisor.html last access on: 2017.

[32]. Virtualization Security, INFOSEC Institute, Available online at, http://resources.infosecinstitute.com/virtualization-security-2/#gref last access on: 2017.

[33]. Jordi Mongay Batalla, Miroslaw Kantor, Constandinos X. Mavromoustakis, Georgios Skourletopoulos, George Mastorakis, " A Novel Methodology for Efficient Throughput Evaluation in Virtualized Routers", 2015 IEEE International Conference on Communications (ICC), Pages 6899-6905, 2015.

[34]. XenServer 7.0 Standard Edition, Available online at https://www.citrix.com/downloads/xenserver/product-software/xenserver-70-standard-edition.html last access on: 2017.

[35]. vSphere ESXi Hypervisor Features, Available online at http://www.vmware.com/products/esxi-and-esx.html last access on: 2017.

[36]. VMware ESXi, Available online at https://en.wikipedia.org/wiki/VMware_ESXi

[37]. Varun kumar Manik, Deepak Arora, "Performance Comparison of Commercial VMM: ESXI, XEN, HYPER-V & KVM", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pages 1771-1775.

[38]. *Snover, Jeffrey (August 1, 2012).* "Windows Server 2012 released to manufacturing!". *Windows Server Blog. Microsoft.* TechNet blogs. Retrieved January 29, 2013.

[39]. Windows Server 2012 R2, Wikipedia, Available online at https://en.wikipedia.org/wiki-/Windows_Server_2012#cite_note-ZDNet1-4 last access on: 2017.

[40]. K. Sundar and Sanjeev Kumar, "BlueScreen of Death observed for the Microsoft's Server 2012 R2 under Denial of Service Attacks," *Journal of Information Security*, vol. 7, pp. 225-231, July 2016, DOI: 10.4236/jis.2016.74018

[41]. Sanjeev Kumar, Raja Gade, "Windows 2008 Vs. Windows 2003: Evaluation of Microsoft's Windows Servers under Cyber Attacks," *Journal of Information Security*, April 2015, DOI: 10.4236/jis.2015.62016,

[42]. S. Kumar, Ricardo Valdez, Orifiel Gomez "Survivability Evaluation of Wireless Sensor Networks Under DDoS Attack," *International Conference on Networking*, April 2006.

[43]. S. Kumar, “Impact of Distributed Denial of Service (DDoS) attack due to ARP-storm,” published in *The Lecture Notes in Computer Science -Book Series*- LNCS-3421 – Networking-ICN 2005, part-II, vol. 3421, pp. 997-1002, April 2005, Publisher – Springer-Verlag

[44]. Sanjeev Kumar and Raja Sekhar, “Experimental Evaluation of Juniper Network's Netscreen-5GT Security Device against Layer4 Flood Attacks,” *Journal of Information Security*,vol. 2, no.3, pp. 50-58, Jan. 2011, DOI: 10.4236/jis.2011.21005

[45]. Sirisha Surisetty and Sanjeev Kumar, “Microsoft’s Windows7 Vs. Apple’s Snow Leopard: An Experimental Evaluation of Resilience against Distributed Denial of Service (DDoS) Attacks,” *IEEE Security and Privacy*, Vol.10, Issue 2, pp. 60-64, April 2012, DOI: 10.1109/MSP.2011.147

[46]. Rodolfo Baez Jr., 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, pp. 123-135, July 2014, DOI: 10.4236/jis.2014.53012

[47]. Surisetty, S, Dr. S. Kumar, “Is Apple's iMac Leopard Operating System Secure under ARP-Based Flooding Attacks?” Second International Conference on Internet Monitoring and Protection (ICIMP 2010).

[48]. 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, DOI: 10.1080/19393555.2011.569908

[49]. R.Aishwarya, Dr.S.Malliga, “intrusion Detection System- an Efficient way to Thwart against Dos/DDoS Attack in the Cloud Environment” 2014 International Conference on Recent Trends in Information Technology.

[50]. S. Kumar and D.P. Agrawal, “Performance of the class of Sliding-Window ATM Switch Architectures for Broadband Communications Networks," Proceedings of *IEEE SOUTHEASTCON*, March 1995.

[51] S. Surisetty and S. Kumar, “Evaluation of a Security Vulnerability in Apple’s Leopard Operating System,” International Conference on Internet Monitoring and Protection, May 2010. Available from IEEE online library Xplore

[52] S. Surisetty and S. Kumar, “Is McAfee SecurityCenter/Firewall Software Providing Complete Security for your Computer?” International Conference on Digital Society (ICDS’10), Feb. 2010.

[53]. S. Kumar and Orifiel Gomez, “Denial of Service due to direct and indirect ARP storm attacks in LAN environment,” – *Journal of Information Security*, vol. 2, no.3, pp. 88-94, Oct. 2010, DOI: 10.4236/jis.2010.12010; ISSN Print: 2153-1234

[54] S. Kumar, M. Azad, O. Gomez, and R. Valdez, “Can Microsoft’s Service Pack 2 (SP2) Security Software Prevent Smurf Attacks?” Proceedings of the Advanced International Conference on Telecommunications (AICT’06), Feb 2006. Available from IEEE online library Xplore.

[55] S. Kumar and T. Doganer, “Effect of Scan-Planes on the Memory Bandwidth of Sliding-Window Switch Architecture,” - Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR05), May 2005.

[56] S. Kumar "On Impact of Distributed Denial of Service (DDoS) attack due to ARP storm," – Lecture Notes in Computer Science – Book Series, LNCS-3421, Networking - ICN 2005, Part-II, Publisher: Springer-Verlag, April 2005.

[57] S. Kumar and T. Doganer, “Memory-Bandwidth Performance of the Sliding-Window based Internet Routers/Switches,” Proceedings of the IEEE Workshop on Local and Metropolitan Area Networks, San Francisco, CA, April 2004.

[58] S. Kumar, T. Doganer, A. Munoz, “Effect of Traffic Burstiness on Memory-Bandwidth of the Sliding-Window Switch Architecture,” Proceedings of the International Conference on Networking, March 2004.

[59] S. Kumar, A. Munoz, T. Doganer, “Performance Comparison of Memory-Sharing Schemes for Internet Switching Architecture,” Proceedings of the International Conference on Networking, March 2004.

[60]. Mac Pro (Mid 2010) - Technical Specifications, Available online at https://support.apple.com/kb/SP589?locale=en_US last access on: 2017.

[61]. Solid-State Drive Replacement Instructions for Mac Pro, Available online at https://manuals.info.apple.com/MANUALS/1000/MA1548/en_US/Mac_Pro_SSD_DIY.pdf last access on: 2017.

[62]. Apple statement on core distribution, Available online at https://support.apple.com/en-us/HT201838. last access on: 2017.

[63]. Installing IIS 8.0 on Windows Server 2012, Available online at http://www.iis.net/learn/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012 last access on: 2017.

[64]. Hyper-V Network Virtualization technical details, Microsoft, 2014, Available online at https://technet.microsoft.com/library/jj134174.aspx last access on: 2017.

[65]. MidhunBabu Tharayanil, Gill Whitney, Mahdi Aiash, ChafikaBenzaid, Virtualization and Cyber Security: Arming Future Security Practitioners, IEEE Computer Society, 2015, DOI 10.1109/Trustcom.2015.537

[66]. Install Hyper-V and create a Virtual Machine, TechNet Library, 2016, Available online at https://technet.microsoft.com/en-us/library/hh846766.aspx

[67]. 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

[68]. Windows Platform Design Notes, Design Information for the Microsoft® Windows® Family of Operating Systems, White Paper, 2002.

[69]. Michael Perez and Sanjeev Kumar, "A Quick Survey on Cloud Computing and Associated Security, Mobility and IoT Security Issues," *Journal of Computer Communications*, Vol.5, no 12, pp. 80-95, Oct 2017 DOI: 10.4236/jcc.2017.512009

[70]. Edni Del Rosal and Sanjeev Kumar, "A Fast FPGA Implementation for Triple DES Encryption Scheme," Journal *of Circuits and Systems*, Vol.8, pp. 237-246, September 2017, DOI: 10.4236/cs.2017.89016

[71]. A. Munoz, S. Kumar "Buffer Management in the Sliding-Window Packet Switch," *Int'l Journal of Communications, Network and System Sciences*, vol.7, no. 7, pp. 448-255, July 2014 DOI: 10.4236/ijcns.2014.77027

[72]. Raja Sekhar Gade, and Sanjeev Kumar, "Experimental Evaluation of Cisco ASA-5510 Intrusion Prevention System against Denial of Service attacks," *Journal of Information Security*, Vol.3, No.2, ISSN Print: 2153-1234, pp.122-137, April 2012, DOI: 10.4236/jis.2012.32015

[73]. S. Kumar and Alvaro Munoz, "Comparison of Memory Assignment Schemes for Switch Architectures with Shareable Parallel Memory Modules," *Journal of Electrical and Computer Engineering*, June 2010, DOI:10.1155/2010/126591

[74]. S. Kumar and Alvaro Munoz, "Performance Comparison of Switch Architectures with Shareable Parallel Memory Modules,"*IEEE Communications Letters*, vol. 9, no. 11, November 2005, pp. 1015-1017. DOI: 10.1109/LCOMM.2005.11021 ISSN: 10897798

75]. S. Kumar and Alvaro Munoz, "Memory Bandwidth Performance of Shared Multiple Buffer Switch Versus Sliding-Window Switch," *IEEE* Electronics *Letters*, vol.41, no.18, pp. 1036-1037, September 2005. DOI: 10.1049/el:20052155 , ISSN:00135194

[76]. S. Kumar, "The Sliding-Window Packet Switch: A New Class of Packet Switch Architecture with Parallel Memory Modules and Decentralized Control," *IEEE Journal on Selected Areas in Communications*, vol.21, no.4, pp. 656-673, May 2003

[77]. S. Kumar and D.P. Agrawal, "Design and Performance of Shared-Buffer Direct-Access (SBDA) ATM Switch Architecture for Broadband Networks," *International Journal of Computer Systems and Engineering*, vol. 12, no. 2, pp. 69-79, March 1997.

[78]. S. Kumar and R. Dantu, "The Era of the Full-Service Networks," *America's Network*, March 1997.

[79]. S. Kumar and D.P. Agrawal, "On multicast support for shared-memory-based ATM switch architecture," *IEEE Network*, vol. 10, no.1, January 1996. DOI: 10.1109/65.484230 Journal ISSN: 0890-8044

[80]. S. Kumar, “Are Packet-Switches Delivering Bandwidth Capacity as Advertised?” *8th Annual the University of Texas System ITDE conference*, May 2002.

[81]. S. Kumar and D.P. Agrawal, “The Sliding-Window Approach to High Performance ATM Switching for Broadband Networks,” Proceedings of *IEEE GlobeCom*, November 1996.

BIOGRAPHICAL SKETCH

Ganesh Reddy Gunnam was born on August 31, 1992. He has completed his Bachelors of Technology in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in May 2014. He has completed his Master of Science in Electrical Engineering from University of Texas Rio Grande Valley, Texas, USA in December 2017. He worked as Supplemental Instruction Leader in Learning Center at UTRGV from October 2015 to December 2015. He also worked as a Teaching Assistant in Computer Engineering department at UTRGV from January 2016 to December 2017. He also worked as a Research Assistant in Network Research Lab at UTRGV. Currently, he is doing his PhD in Electrical Engineering at University of Texas San Antonio.

Permanent Address:

H.No: 2-50, Yetdharpally, Tadoor, Mahaboobnagar, Telangana, India - 509209

Email: gunnamganeshreddy@gmail.com

His Publications:

1) Ganesh Reddy Gunnam, Sanjeev Kumar, (2017) Do ICMP Security Attacks Have Same Impact on Servers? *Journal of Information Security*, 8, 274-283. https://doi.org/10.4236/jis.2017.83018

2) Ganesh Reddy Gunnam, Sanjeev Kumar, Jaime Ramos, “Cyber Attack on Smart Electric Meter”, Poster Presentation, HESTEC, University of Texas Rio Grande Valley, 2017.