5-2015

Evaluation of security vulnerabilities of popular computer and server operating systems under cyber attacks

Rodolfo Baez Jr.
University of Texas-Pan American

EVALUATION OF SECURITY VULNERABILITIES OF POPULAR COMPUTER AND SERVER OPERATING SYSTEMS UNDER CYBER ATTACKS

A Thesis

by

RODOLFO BAEZ JR

Submitted to the Graduate School of
The University of Texas – Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2015

Major Subject: Electrical Engineering

EVALUATION OF SECURITY VULNERABILITIES OF POPULAR COMPUTER AND SERVER OPERATING SYSTEMS UNDER CYBER ATTACKS

A Thesis
by
RODOLFO BAEZ JR

COMMITTEE MEMBERS

Dr. Sanjeev Kumar
Chair of Committee

Dr. Jun Peng
Committee Member

Dr. Xiang Lian
Committee Member

May 2015

ABSTRACT

Baez Jr, Rodolfo, Evaluation of Security Vulnerabilities of Popular Computer and Server Operating Systems Under Cyber Attacks, Master of Science (MS), May, 2015, 121 pp., 6 tables, 47 figures, 78 references, 1 Appendices.

Nowadays many operating systems are including security features in order to prevent network attacks, and since one of the roles of the OS is to manage the resources as efficient as possible. It is imperative to investigate the protection that is provided. Therefore, the scientific significance of this thesis was to evaluate, what type of built-in defense mechanisms that different OS's had in place in order to mitigate these network attacks.

In this thesis, we considered the security of the following globally deployed computer OS's: Microsoft's Windows 7, Apple's OS X Lion, and Ubuntu 13.10. Furthermore, we also tested four server OS's: Microsoft's Server 2008 and 2012, Apple's OS X Lion Server, and Ubuntu Server 12.04, and their performance under DoS attacks. Our experimental results show that the OS's that were evaluated were found to have inadequate security protection and showed different degree of effectiveness in handling different DDoS attacks.

DEDICATION

The completion of my master studies would not have been possible without the blessing and support of my family. I would like to thank and dedicate my work to my parents, Rodolfo Baez and Rosalinda Gallardo, for their love and exceptional support. I would also like to thank my siblings, nieces, and nephews, by whom I have been inspired and motivated to excel in life to become an exemplary example for you all. The completion of my master would not be possible without their love and support. This thesis would be incomplete without the acknowledgement to the one person who has been there for me time and time again, CSC.

ACKNOWLEDGEMENT

I would like to formally thank:

Dr. Sanjeev Kumar, my advisor and chair of my thesis committee, for his profound help, his advice, and belief in my abilities. His research first sparked my interest back when I was an undergraduate student, and it is a pleasure to have learned so much from him. He has taught me the fundamentals of networking, to advance security analysis and security prevention techniques. I have learned so much and without you this thesis would have not been possible. Thanks you so much for your entire support and encouragement.

I would also like to thank Dr. Jun Peng and Dr. Xiang Lian, for their willingness to serve as committee members. More specifically, for the knowledge in networking that Dr. Jun Peng has provided and to Dr. Xiang Lian for the knowledge in operating systems. Without their help, this thesis would have been far more challenging. Their advice, input, and comments on my thesis helped to ensure the quality of my intellectual work. So thank you for your support and guidance.

Dr. Pearl Brazier for the initial confidence in me to become a graduate assistant for the Computer Engineering department, this allowed me to fund my Master's. Thank you for giving me that opportunity. To my friends in the Networking Research Lab, for all the technical discussions that we have had during this adventure. Thanks!

Work in this thesis is based upon the grant awarded to Dr. Kumar by the National Science Foundation (NSF) under Grant No. 0521585.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

The topic of computer and network security is of a paramount importance these days to the general welfare of the public. We live in a society that has transitioned into a web-based world that relies on the interaction of devices that are interconnected to create a complex yet sophisticated global network. As our computing and sensitive data makes its transition onto cloud and mobile platforms, the security and vulnerabilities associated with these devices tends to rely on the operating systems that are used to manage those resources. These devices include computers and servers that usually only provide a limited amount of built-in security. As a result, a single line of defense is not sufficient in providing the stability and assurance that our information and critical infrastructures are secure [1]. So there is a need to have multiple network devices that can detect and prevent malicious attacks before affecting these critical devices.

In the attempt to understand network security, one must understand how the devices in a network communicate and the inherent vulnerabilities they have. The way devices communicate with one another are through communication standards, also known as protocols which came about due to the progress that was made in the academic field and the incentives from the Department of Defense [2]. In the 1960's, the first network to implement the TCP/IP protocol suite was an early packet switching network known as Advanced Research Projects Agency Network (ARPANET) [3].

The Transmission Control Protocol and Internet Protocol (TCP/IP) suite is a collection of communication standards that define how devices communicate with each other [4-6]. In its infancy, these protocols were geared toward a standardized method for communication between network devices but with little consideration to security. Since at this time, the idea that a global communication network was unfathomable and therefore security was not a top priority. As the idea of a global communication network quickly spread and became a reality, the designers quickly realized that not everybody was using this technology in an ethical way. So enhancements or revisions were made to patch these security vulnerabilities. Unfortunately, this only alleviated some problems but not all. Hackers quickly found other vulnerabilities and exploited them to attack computers, servers and other critical infrastructures.

The equipment in a network that is used to store our sensitive data is usually servers or computers that have a finite amount of resources. When under attack if the resources on these devices are fully consumed, it can make the equipment unresponsive and can even make the equipment crash. This vulnerability has been exploited by a set of network attacks that are called a Denial-of-Service attack. A Denial-of-Service attack is an attempt to make a machine or network resource unavailable to its legitimate users. This is done so that the device can no longer provide its intended service, or to obstruct the communication media between the users and the network resources [7]. For instance, by exhausting the resources of the victim's computer or server, a hacker can effectively create a DOS attack. Some of the resources that can be exhausted are bandwidth, memory, and/or the processor. Sometimes even with the help of security systems such as firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS), we still find servers and computers becoming victims to these attacks. This compromises the

availability, confidentiality, and integrity of such critical information systems. The CIA triad promises that confidentiality, integrity, and availability are needed in order to fully secure an internet connection. These three concepts embody what is known as the fundamental security objectives for both data and for information and computing services [8].

1.1 Motivation

With the advancement of technology continuing to bring unique and unparalleled devices to the general public, it becomes imperative that security should follow the same trend and not lack or fall behind. Everyday millions of people connect their computers and servers to the internet without the necessary education on how to protect their investment. We live in a society that greatly relies on some type of electronic device that is integrated into some aspect of our lives and has access to the Internet. For instance, we can access our emails from our mobile phones or computer but we hardly think about how this is happening. In a nut shell, we are attempting to access information (i.e. our email messages) from a mail transfer agent (MTA) also known as a server [9].

According to [10], a surprising number of DoS attacks happen regularly that the Cooperative Association for Internet Data Analysis (CAIDA) estimated that 12,000 attacks occur per week. The article goes on to state that "One example is the FBI's annual report on cybercrime, based on the information that is provided by nearly 500 organizations. In the 2004 report, nearly a fifth of the respondents who suffered financial loss from an attack had experienced a DoS attack. The total reported cost of DoS attacks to these companies was over $26 million". Claiming that, a Denial of service attack was the top source of financial loss due to

cybercrime [10]. They have become so popular and powerful that DoS attacks have even been reported by several governmental agencies [10]-[12]. This was the case on Jan. 5, 2013 when the webservers of the Department of Justice, the Federal Bureau of Investigation (FBI), and some music corporation had their websites attacked. The attack that was used was a Distributed Denial of Service (DDoS) attack [11] [12]. This report proved that anybody can become a victim if an attacker is intent on disrupting their services.

It is imperative that the security built into the Internet be multi-layered so that one point of failure cannot disturb the flow of traffic. This means that there should be some type of security in the core routers and end points (desktop and servers) on the Internet. While the security devices in the core, such as routers, switches, IPS, IDS, and firewalls, are usually stored in some warehouse and are not readily available to the general public. So, we will then shift our focus to the built in security that is provided to us from several operating system that are incorporated into the computers and servers that we interact with on a daily basis when we connect to the internet. Since today's operating systems are deploying their own built-in prevention mechanisms, we intend to evaluate their performance and compare them with their respective counterparts (i.e. desktop operating systems vs desktop and server vs server). This will allow us to compare which operating system, without the aid of external security systems, is more resilient and efficient under these attacks. We will next introduce the operating systems that will be under testing.

Mac OS X 10.7 "Lion" and "Server Lion" are the eighth major release of Mac Operating System for Apple's desktop and servers and was reported to have over one million download in sales on the first day of released [14] [15]. According to Apple Inc., it claims that an iMac

computer running the latest OS X is reliable, more powerful, and safer than any other computer on the market [15].

Windows 7 is an operating system developed by Microsoft for personal computers, and is considered to be a major improvement over its predecessors. Windows 7 became generally available on October 22, 2009 and has been a major success for Microsoft [17] [18]. Microsoft reported that in just six months, over 100 million copies were sold worldwide and by July 2012 more than 630 million copies were sold.

Ubuntu 13.10 “Saucy Salamander” is an open source operating system that is compatible with a range of devices that is developed by Canonical [19]. Ubuntu provides a stylish and intuitive interface that has a built-in firewall and virus protection that powers millions of desktops PCs, laptops and servers around the world [20]. It is estimated that Ubuntu has more than 20 million users and makes up roughly 5 percent of desktop OSs in use today [21].

Microsoft has two very popular operating systems for the server platform still in use today. Windows Server 2008 R2 was released in July 2009 and Windows Server 20012 R2 was released in August 1, 2012 [22] [23]. Together, they have a combined market share of about 33 percent [24] [25]. The popularity of Windows Server 2008 R2 has grown so much that Microsoft has extended the life of support for an additional 18 months, however Redmondian Keepers says that it has to do with the support lifecycle [26] [27].

Ubuntu 12.04.5 LTS “Precise Pangolin” is an open source severs operating system that was developed by Canonical and was released on April 26, 2012 [28] [29]. This version has included multiple new features, such as a quick search, access menu and indicator action display

as known as “HUD”. Another modification was to the under lying software, which will upgrade the kernel. The major significance for this upgrade was for hardware enablement [30].

In order to evaluate the built-in security for each desktop operating system against different DDoS attacks, we used an iMac desktop computer that was capable of running Windows 7, Ubuntu 13.10, and Apple’s OS X 10.7.5 Lion. While Windows Server 2008 R2, Windows Server 2012 R2, Ubuntu Server 12.04.5, and Apple’s OS X Server Lion were installed on an iMac Pro Server. The TCP/SYN Flood and ICMP based Ping and Land Attacks were used to launch a DDoS attack on these seven operating systems. ICMP attacks are one of the most common DDoS attacks and tend to exhaust a victim’s computer computing resources by sending a flood of ICMP Echo Requests packets.

1.2 Problem Statement

Computers and servers are electronic devices that contain both hardware and software. The hardware is the underlying physical components that can be seen and felt, while the software is a program that is executing on the hardware. The most crucial software that is executing on the hardware at any given moment is the operating system. Since one of the roles of the operating system is to manage the resources as efficient as possible, it becomes apparent that the operating system can become a bottleneck in the performance of the critical services that are used every day. While there are many software companies that design and program an operating system, the most dominant companies are Apple, Microsoft, and Linux. Since numerous companies’ design and program operating systems, it becomes apparent that they are not all the same. For instance,

the way Microsoft implements security into their operating system will not be the way Apple nor Ubuntu implements theirs.

As previously mentioned, Apple Inc., has claimed that an iMac computer running the latest OS X is reliable, more powerful, and safer than any other computer on the market [14]. Claims like these sound promising, but yet suspicious. They lead to a false sense of security if the proposed claim is false. Therefore, it is imperative that outside research is performed so as to justify these claims. Therefore, we would like to evaluate the effectiveness of modern host-based intrusion prevention and resilience against cyber-attacks.

1.3 Thesis Outline

In this thesis, we consider the security of seven globally deployed computer and server operating systems while under three popular yet powerful cyber-attacks. We will study the impact that the Ping, Land Attack, and TCP/SYN Flood attacks had on an Apple's iMac computer deploying the following operating systems: Apple's OS X 10.7.5 "Lion", Microsoft's "Windows 7", and Canonicals' Ubuntu 13.10 "Saucy Salamander". The same cyberattacks will be used to study the impact on the server operating systems. However, the server operating systems: Apple's OS X 10.7.5 Server "Lion", Ubuntu's 12.04.5 LTS "Precise Pangolin", Microsoft's "Windows Server 2008 Enterprise R2" and "Windows Server 2012 Enterprise R2" will be deployed on an Apple's iMac Pro Server.

On the computer platform, we will be testing the effectiveness of the built-in security provided by each operating system by measuring the impact on the performance by the cyberattacks. As for each of the server operating systems, we will have their corresponding web

server application installed and functioning correctly so that we may evaluate the impact on the overall performance. These attacks are launched at different transmission rates starting at 100 Mbps all the way to 1000 Mbps in increments of 100 Mbps for a duration of 6 minutes. Therefore, in order to measure the effects of DDoS attacks, several tests were conducted in the Network Research Lab to evaluate the effectiveness of the selected operating systems. The experimental setup is shown in Figures 8 and 19 of chapters III and IV, respectively.

The thesis is organized as followed: Chapter I is an introduction that is oriented to give a general idea on the DDoS attacks and how the built-in security of each operating system can become a crucial bottleneck. Chapter II provides a comprehensive background on the distributed denial of service (DDoS) attacks that will be used in this thesis. In Chapters III, IV, and V we will be presenting the results gathered through experiments conducted in the Network Research Lab. More specifically, in Chapter II we evaluate and compare the desktop based operating systems under DDoS attacks. In Chapter IV, we evaluate and compare Window's based server operating systems and the impact they have on the Internet Information Service (IIS) web server application while under DDoS attack traffic. In Chapter V, we evaluate and compare the server based operating systems under DDoS attacks. In Chapter VI, we will conclude with a synopsis of our work and the potential contributions that may be included in the near future.

CHAPTER II

DISTRIBUTED DENIAL OF SERVICE ATTACKS

2.1 Background study on Different DDoS Attacks

Just about every network that is connected to the Internet can be subjected to attacks from malicious sources. As was shown in [11] attacks can even happen to government agency. This was the case on Jan. 5, 2012 when the Department of Justice, FBI, and some music corporation websites were disrupted to its user, while under a DoS attacks. In 2013, two significant DDoS attacks were behind the latest cyberattack which crippled the servers at a hosting services firm agency and the "largest" public DoS attack in history [31] [32]. In [33] – [35] are up to date and historical reports of DoS attacks that have been detected.

In general an attack can be categorizes as either passive or active. A passive attack is when a network intruder intercepts data traveling through the network. On the other hand, an active attack is when an intruder initiates commands to disrupt the networks normal operation. Each type of attack has a different weakness that it will try to exploit in the TCP/IP suite protocol or at the computers/hosts. Examples of passive attacks are wiretapping, port scanner, and idle scan. Examples of active attacks are Arp poisoning, smurf attacks (ICMP), TCP/SYN flood, teardrop attacks, permanent DoS and can be seen in Figure 1 [36].

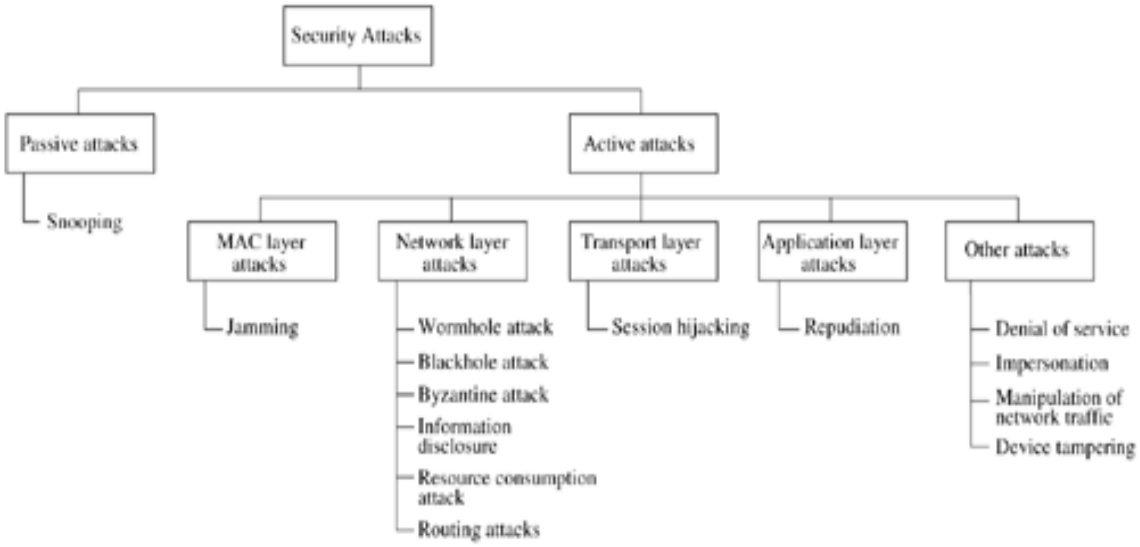


Figure 1: Classification of DDoS attacks [36]

The focus of this section will be to introduce the general idea of a DoS attack. A Denial-of-Service attack is considered an active attack and is an attempt to make a machine or network resource unavailable to its intended users. This is done so that it can no longer provide its intended service, or to obstruct the communication media between the users and the network resources. This is achieved by consuming the computational resources in such a way that communication is halted and is usually achieved by consuming the bandwidth, memory, CPU or by disrupting the configuration information for routers or switches. A Distributed Denial of Service (DDoS) attack uses many computers, also known as a botnet, to launch a coordinated DoS attack. One of the problems with DoS packets is that they are very similar to legitimate traffic, therefore, making it nearly impossible to distinguish against good and bad data.

2.1.1 ICMP Based DDoS Attacks

An attacker that uses an ICMP based attack will send a large numbers of IP packets with the source IP address faked to appear to be the address of the victim. This will cause the victim's network bandwidth to be consumed very quickly and make the computer unresponsive, which will make communication nearly impossible. The question that arises is "How is this possible?". Well to understand what is happening we must understand what ICMP is.

ICMP is an acronym that is used for Internet Control Message Protocol and is defined by Request for Comment (RFC) 792 [37]. The logic behind an ICMP message is that if an IP datagram cannot reach its destination, or if a router does not have the buffering capacity to forward a datagram, the router must send a message to the sender to inform them. Well, that type of message is an ICMP. ICMP plays other crucial role like if the time to live field in the datagram has expired source quench, timestamp, echo request, echo reply, etc. In general, an Internet Control Message Protocol (IMCP) is a layer 3 protocol that consists of error reporting and query messages. The error-reporting messages are used to report any problems in the network, like delivery error, timeout, and any other types of problems that are encountered. While the query messages are used to get specific information from a router or host in the network.

As in every protocol in the TCP/IP protocol suite, each frame consists of two sections a header and payload field. The header is where the intelligence of the packet is kept and contains crucial information for the successful delivery of the packet. While the information in the payload is usually the data that need to get delivered. Since ICMP is in the same level as IP, it is

encapsulated in an IP datagram which in turn in is encapsulated into an Ethernet packet. In Figure 2, we can see how an ICMP message is encapsulated into an Ethernet packet [38].

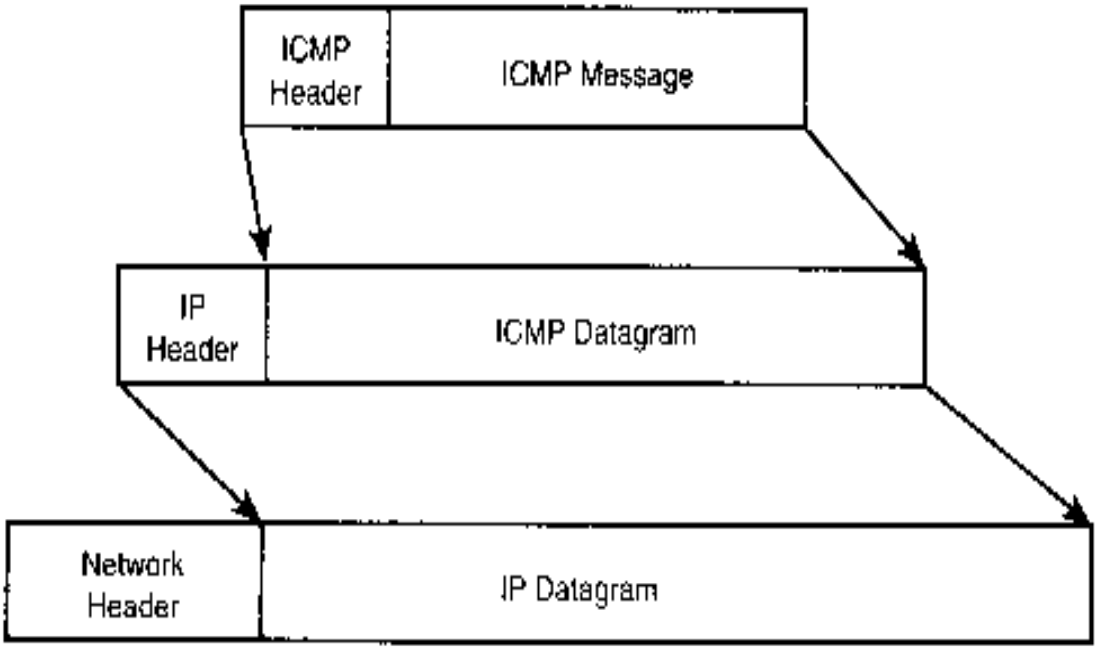


Figure 2: ICMP Message encapsulated in Ethernet Packet [38]

As mention above, there are numerous ICMP message available and can be seen in Figure 3. The ICMP packet format consists of an 8-byte header and a variable-size data section. The general format of the header is different for each ICMP message; however, the first 4 bytes are common to all. These are the TYPE, CODE, and CHECKSUM fields. Depending on how the Type and Code fields are set, will determine which ICMP message is being transmitted. The type field is 8 bits long and specifies the type of ICMP message that will be used. For instance, a 3 in the type field indicates a destination unreachable message. The code field is also 8 bits longs, in which its value will coincide with a specific problem. For instance, in a destination unreachable message, a 0 is for net unreachable, a 1 for host unreachable, a 2 for protocol unreachable, and

etc. This is then followed by a 2 byte checksum field. Checksum is an algorithm that is used for error detection.

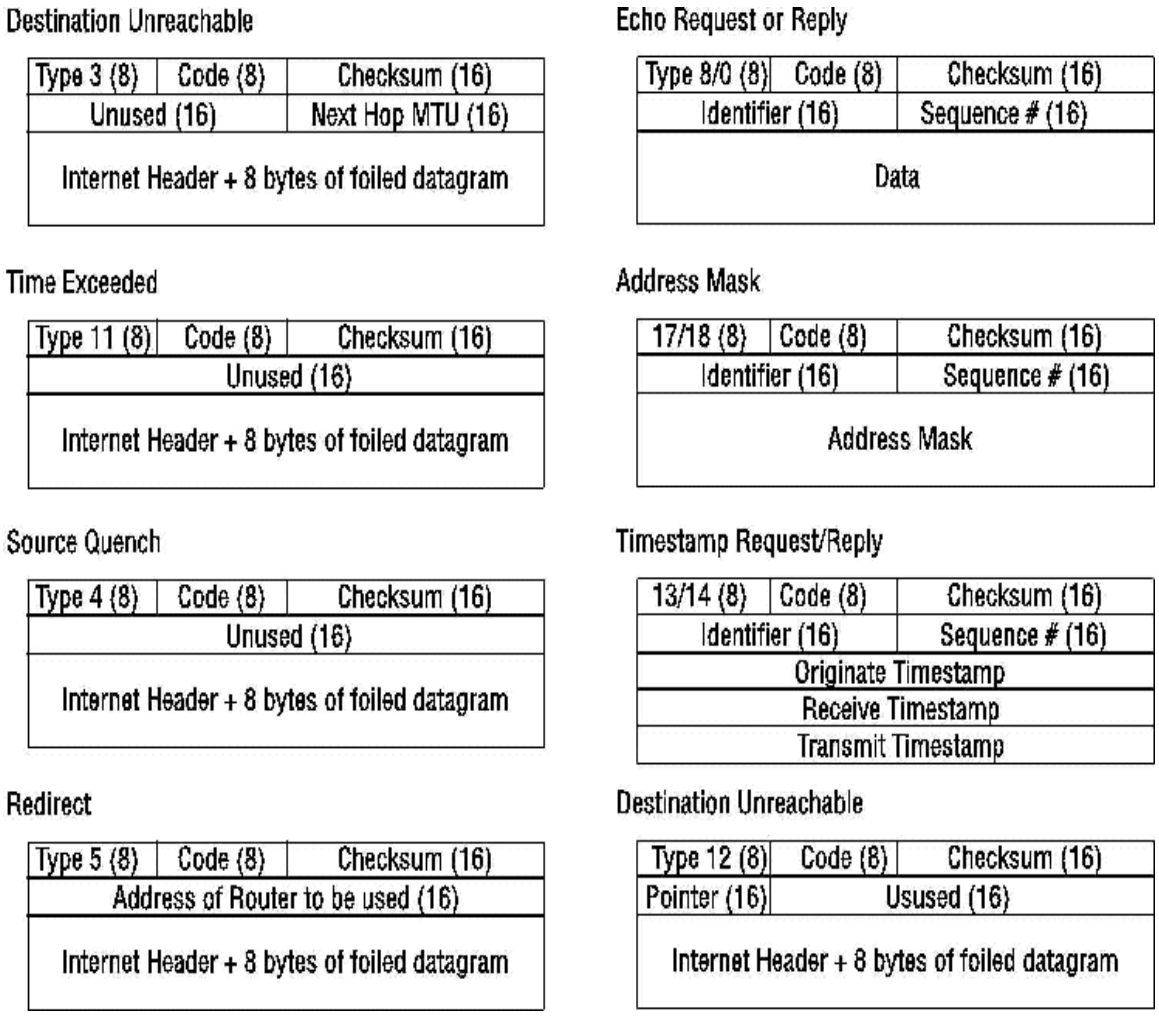


Figure 3: Various ICMP message formats [39]

Two very popular ICMP cyberattacks are the Ping and Land Attack Floods. Depending on how we set up our ICMP packet will determine if a Ping Flood or Land Attack was used. The Land and Ping Flood attacks rely on an ICMP echo request packet. Ping is a networking utility that is used to determine whether a given host is reachable [40]. An attacker will take advantage of these diagnostic packets to create a Denial of Service on a target host. Figure 4 shows the

packet format for an echo request/reply message. In order to create a Ping or Land attack, the attacker must use the ICMP Echo Request message and set the TYPE field to an 8 and the CODE field to a 0.

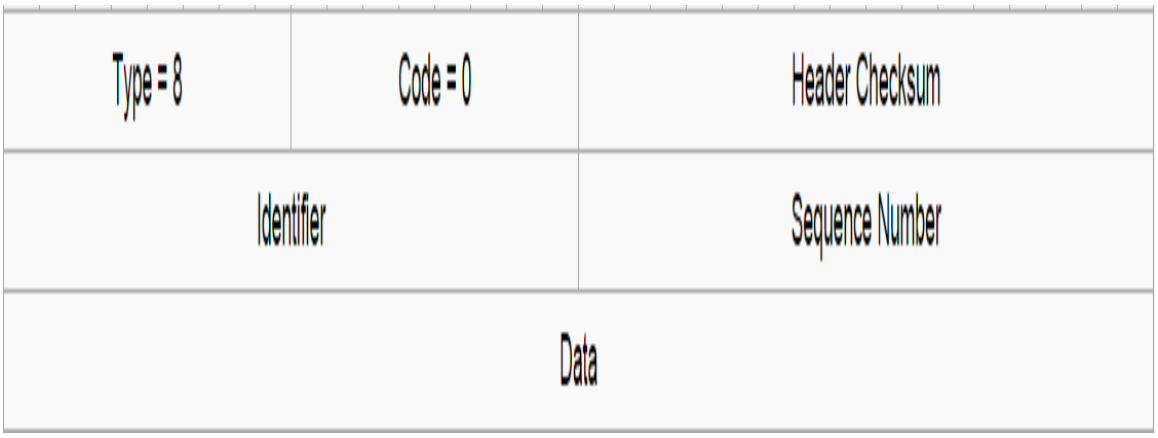


Figure 4- ICMP Echo request Header [40]

2.1.1.1 Ping Flood Attack

The Ping utility is a diagnostic tool used by network administrators to verify the end-to-end path of a host on a network. It relies on the ICMP Echo Request and Echo Reply messages to accomplish this. According to RFC 792, when a device receives an ICMP echo request, it must respond with an ICMP echo reply. Figure 5 shows how the echo request and echo reply messages work. The initiator (attacker) will flood the victims computer or server with multiple echo request messages, which in theory will keep the victims computer busy processing the flood of echo request packet that were received. This type of rudimentary flooding will cause a denial of service effect. The Ping Flood is a simple but devastating attack that has the potential of costing million if not billions of dollars in potential loss [41].

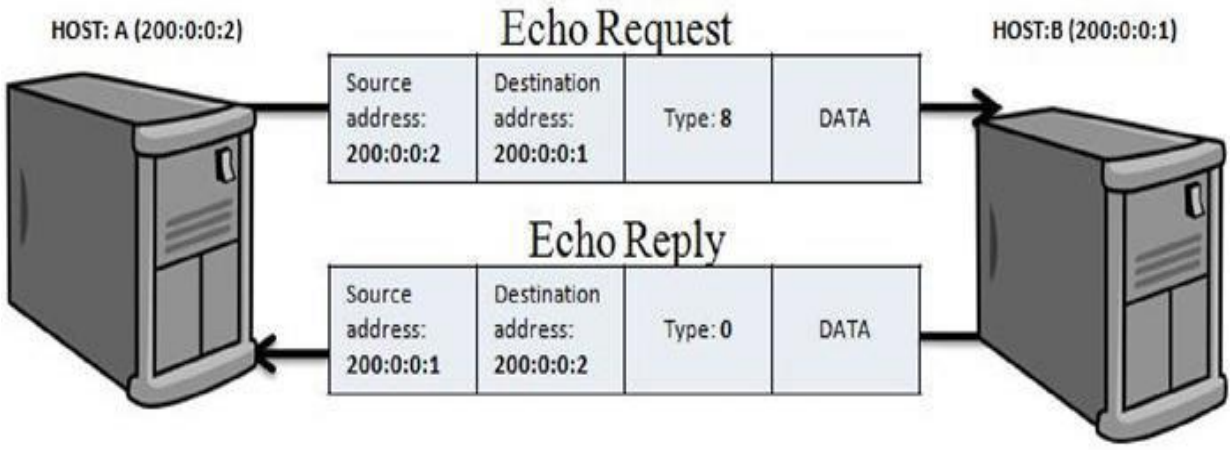


Figure 5: Ping Utility

2.1.1.2 Land Flood Attack

In an ICMP Land attack, the attacker will use the ICMP Echo Request message just like in a Ping Flood attack. However, both the destination and source IP addresses are that of the victim's computer. In doing so, it causes the victims computer to reply to itself for every echo request message that it receives. Just like in the Ping flood attack this will cause the victims computer to use up the processor to execute these request messages, but in addition to this it will have to process the reply messages as well. This in turn will cause it to use up more of the processor and bandwidth than in the ping flood attack. Figure 6 shows the principle behind the land attack. When the attacker sends a flood of echo request packets with the source and destination addresses the same, as shown in figure 6, this will cause the victims computer to send an echo reply message. However, since the source and destination were the same, the flood of echo reply message just sent by the victim's computer will return back to the victim's computer

which will eventually consume the resources of the victim's computer. In essence, creating what is called a Denial of Service.

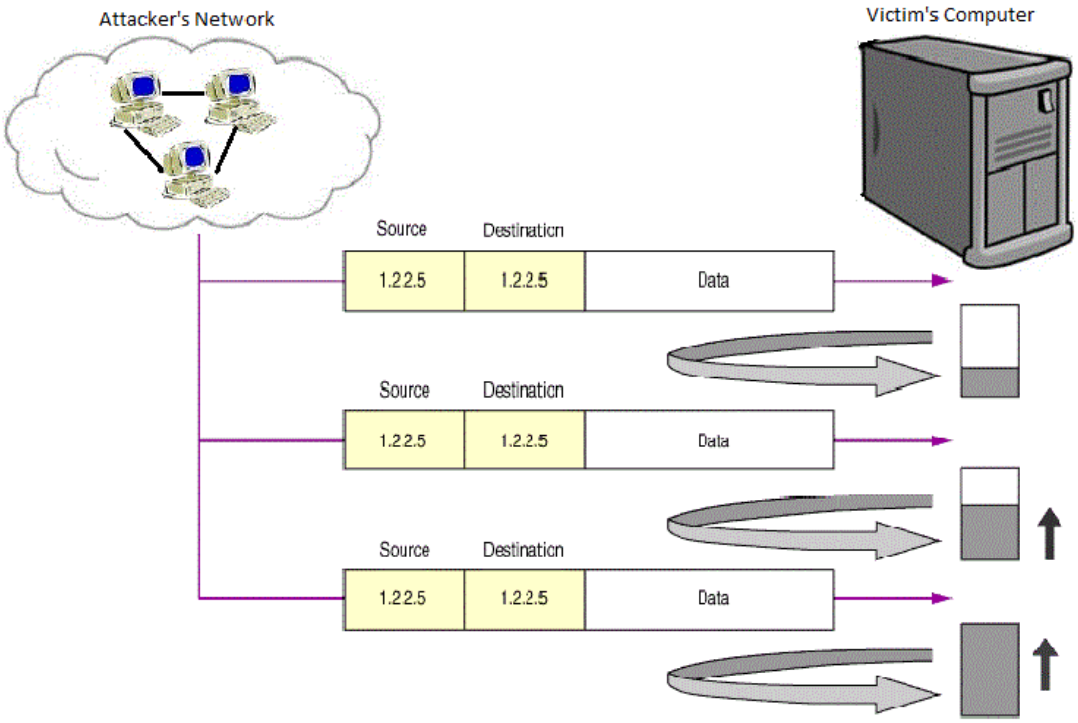


Figure 6- ICMP Land Attack

2.1.2 TCP Based DDoS Attacks

TCP stands for Transmission Control Protocol, and is defined by RFC 793[42]. TCP provides the fundamental basis for a reliable delivery system for data to travel in the internet. TCP relies on a successful connection to be established before any data is transferred. This connection is called a three-way handshake and is shown in Figure 7.

When the initiator initiates a TCP connection, it starts by sending a SYN packet to the Listener as can be seen in Figure 7. The Listener will then responds with a SYN-ACK packet and will store the requested information onto the memory stack. After receiving the SYN-ACK packet, the Initiator confirms the request by sending an ACK packet. When the Listener receives the ACK packet it checks in the memory stack to see whether this packet corresponds to a previously received SYN. If it is, then the connection is established between the client and the server/computer and data transfer can begin. An attacker that uses a TCP/SYN flood will send a flood of SYN packets, with spoofed IP addresses. A spoofed IP address is used so that it can provide an undetectable route back to the attacker and so that the connection request will result in a half open connection. Since each SYN packets is handled like a connection request, this will make the victims device reply with a SYN-ACK packet to the spoofed IP addresses computer. Since the senders IP address was forged, the packet will not get acknowledge back causing a half-open connection to remain open for some period of time. Figure 7 shows how a half open connection it possible. This attack can become very dangerous if the attacker sends numerous amounts of SYN packets to a computer or server. This will result in the saturation of the limited number of available connection that a device can handle.

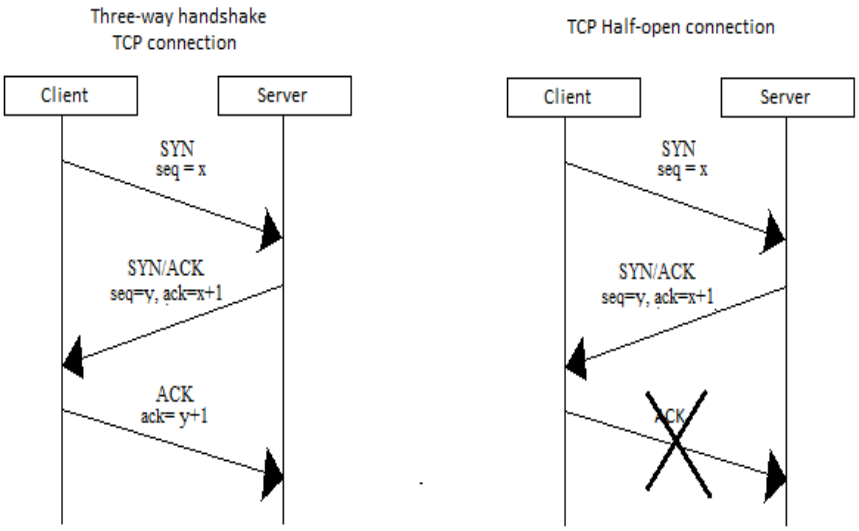


Figure 7: TCP "Three-way Handshake" Connection Procedure

2.2 Chapter Summary

In this chapter, we have discussed and reviewed the concepts behind what a Distributed Denial of Service Attack is. We have also given a small overview of the ICMP Ping Flood, ICMP Land Attack, and the TCP/SYN Flood attacks that will be used in our experimental testing.

The attack methods defined in this chapter will be used to aid us in the evaluation and comparison of the built-in security features found in today's popular operating systems.

CHAPTER III

COMPARATIVE EVALUATION OF APPLE'S MAC OS X 10.7 "LION", MICROSOFT'S WINDOWS 7, AND UBUNTU'S 13.10 "SAUCY SALAMANDER"

3.1 Experimental Setup

In this chapter, we will conduct the evaluation of device under testing using traffic simulation in the controlled environment of the Network Security Research Lab (NRL) at The University of Texas – Pan American. The performances of different operating systems were evaluated under TCP and ICMP based attack traffic up to a maximum speed of 1 Giga bit per sec (Gbps). The attack traffic was sent for six minutes for each load and was in the range of 100 to 1000 Mbps of traffic and in increments of 100 Mbps. The Operating Systems under test were Microsoft's Window 7, Apple's OS X 10.7.4 "Lion", and Canonicals' Ubuntu 13.10 "Saucy Salamander". The victim computer platform was an Apple iMac with 8 GBytes of RAM, an Intel Core i5 2.5-GHz processor. The type of testing that we plan to run are similar to the ones that are performed in [43]-[47], but with the introduction of newer modern operating systems. This approach will allow us to determine if any improvement or modification have been made in the newer operating systems to help mitigate these known cyberattacks. We start by introducing the experimental setup, followed by the specification of the hardware and software that were used during the test. Then, we will introduce the parameters that will be used to compare the performance of each operating system.

The experimental setup is shown in Figure 8. The attacker's network is connected to the victim's computer so that we can evaluate the performance of the computer platform under a Distributed Denial of Service attack. We will provide the specification of the hardware in the next section. In general, the software was used to gather the statistics on the parameters that are needed to compare and evaluate the testing that was performed. The data that was gathered will be used to plot useful graphs that can be used to compare our results in an intuitive way.

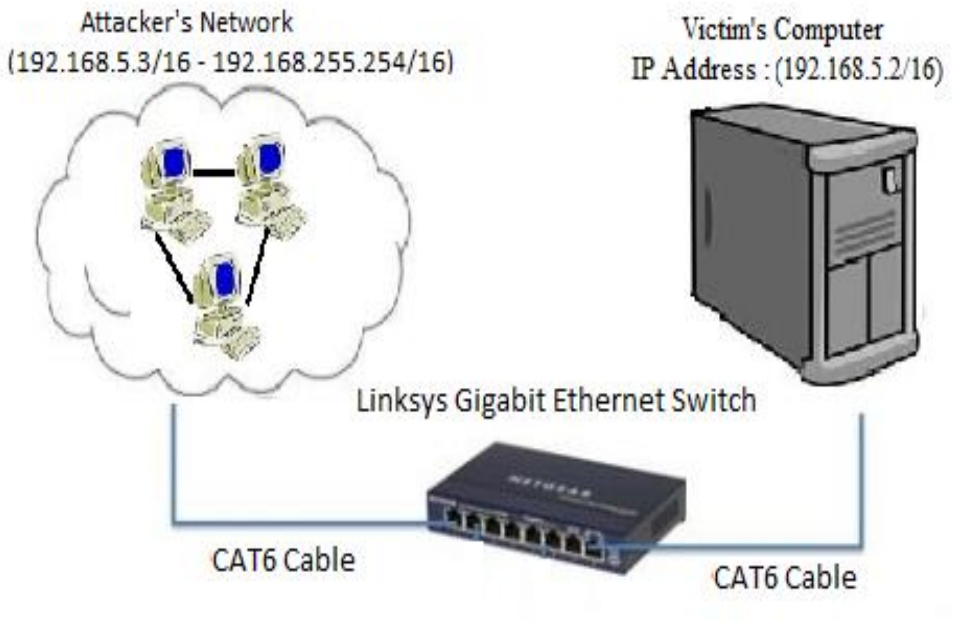


Figure 8- Experimental Setup

3.1.1 Hardware

Switch:

The type of Ethernet switch that was used in the experiment was the Cisco SRW2024 24-port Gigabit Switch. It supports gigabit networking, offering exceptional performance over

24separate connections. Since security is a priority on the SRW2024, no attached device receives information before it authenticates with the switch. If more details are needed you may refer to [48] [49].

Computer Platform:

The victim computer used was an Apple iMac equipped with an Intel Core i5 2.5-GHz quad-core processor, 8 GBytes of RAM, and a Broadcom NetXtreme Gigabit Ethernet adapter [53]. The Apple iMac has the capabilities of running the following OS: Apple's OS X 10.7 "Lion", Microsoft "Windows 7", and Canonical Ubuntu's 13.10 "Saucy Salamander".

Computer 1 has the following specification:

Operating Systems: Apple's OS X 10.7 "Lion", Microsoft's "Windows 7", and Canonicals' Ubuntu 13.10 "Saucy Salamander"

CPU: Intel Core i5 2.5 GHz

Number of Processors: 1

Number of Cores: 4

Random Access Memory (RAM): 8Giga Bytes

Network-Interface-Card (NIC): Broadcom NetXtreme Gigabit Ethernet controller

Graphics:

Chipset Model: AMD Radeon HD 6750M

Type: GPU

BUS: PCI

3.1.2 Software

The software that was used included Microsoft's excel spreadsheet [54], Microsoft's Performance Monitor [55], and other Apple and Ubuntu command line utilities. By using excel, we were able to obtain graphs with the information that was collected during testing using the performance monitor and other command line utilities.

3.2 Parameters of Performance Evaluation

For this experiment, the parameters that are being used in the evaluation of performance were the Processors utilization, the number of Echo request packets received per second, the number of Echo replies packets sent per second, and the amount of random access memory being consumed in Mbytes. These resources are being measured while the platform is being subjected to our denial of service attack traffic and are crucial for the evaluation of our system and are described below:

CPU Utilization (Usage of CPU in %) – The utilization of the Central Processing Unit (CPU) is one of the most important parameters to keep track of in any normal or abnormal situation. The utilization of the CPU informs the user of the amount of work that is being performed by the computer at that instance.

Echo Request Packets Received per Second (Echo Request /Sec) – This parameter will allow us to measure the number of echo request messages that were being received per second by the computer. When a computer received an echo request message, it must reply with an echo reply message [38]. As the value of these parameters increase, we hope to identify the impact that it has on the other parameters that we will be evaluating.

Echo Reply Packets Sent per second (Echo Reply / Sec) – This parameter will allow us to measure the number of echo reply messages that are being sent per second by the computer. The echo reply message is sent in response to the echo request message that was received.

Random Access Memory (RAM) consumed (in Mbytes) – This parameter measures the amount of finite random access memory that is being consumed. If RAM becomes completely consumed, the computer will become slow and can become unstable.

3.3 Results and Discussions

In this section, we will be reviewing the results obtained from the experiments that were performed. It should be noted that all three operating systems were running on the same Apple's hardware platform but only one operating system was running at a time.

In this experiment, we will be testing the built-in security of three very popular Operating Systems in use today: Microsoft's Windows 7, Canonicals' Ubuntu 13.1 "Saucy Salamander", and Apple's OS X 10.7 "Lion". The parameters that will be used for evaluation and comparison will be the number of echo request and echo reply messages that each operating system will receive and send, the amount of memory being consumed (RAM), and the exhaustion of the processor.

We will begin by comparing the survivability against the ICMP Ping and Land Attack Floods. Followed by, comparing the survivability against the TCP/SYN attack. For each attack, we begin by comparing the number of echo request and echo reply messages that were received and sent from 0 to 1000 Mbps of attack traffic in increments of 100 Mbps. For reference, we will

be using the 0 Mbps of attack traffic as a baseline. This baseline is very important and will demonstrate the normal operation of the system while under no attack. We will then present the graph for the utilization of the processor for each operating system, followed by the amount of memory that was consumed.

According to the results, we discovered that Ubuntu had performed better while under the ICMP Land Attack flood and the TCP/SYN attack. While Windows 7 had performed better than Ubuntu and Lion while under the Ping flood attack.

3.3.1 ICMP Ping Flood Attack

What was interesting was that all three operating systems were limiting the number of Echo Request and Echo Reply messages that were being received and sent (Table 1 and Figure 9). According to the data that was collected, we can see that Apple's Lion was limiting the number of echo request packets that it would receive to about 400 000 per second and can be seen in Table 1. We also observed that the maximum number of echo reply messages that it would send per second was 250, no matter the number of echo request messages that it would receive. This was not expected, since the ICMP protocol states that for every echo request message received the destination computer shall respond with an echo reply message [38] [40]. Table 1 will show the actual number of echo request and echo reply messages being received and sent per second for each operating system.

For Windows 7, the number of echo request messages that it would receive per second were also being limited. According to the data, the threshold for the number of echo request messages that Windows 7 would receive was limited to 600 Mbps of attack traffic and was about

820 000 packets per second and can be seen in Figure 9 and Table 1. As the attack speed was increased, the threshold seemed to decrease to below 600 000. When it came to the number of echo reply messages that were being sent, Windows 7 would only send 500 echo reply messages for the first second of the attack. Then, for the remainder of the attack it would not respond. This was not the case for Apple's Lion; every second Lion was sending 250 echo reply messages per second.

As mentioned above, Ubuntu's OS was also limiting the number of echo request and echo reply messages that it would receive and send. However, when compared to Lion and Windows 7, we can see that Ubuntu had a higher threshold limit for the number of echo request messages that it would receive. We observed that the threshold for the number of echo request packets that it was receiving was about 900 000 per second at the attack load speed of 800 Mbps and can be seen in Table 1. When it came to the number of echo reply packets being transmitted, we can see from Table 1 that after the attack load speed of 300 Mbps the number of echo reply messages begin to decrease. When compared to the other operating systems, we can see that Ubuntu had the highest threshold values for both the echo request and echo replies.

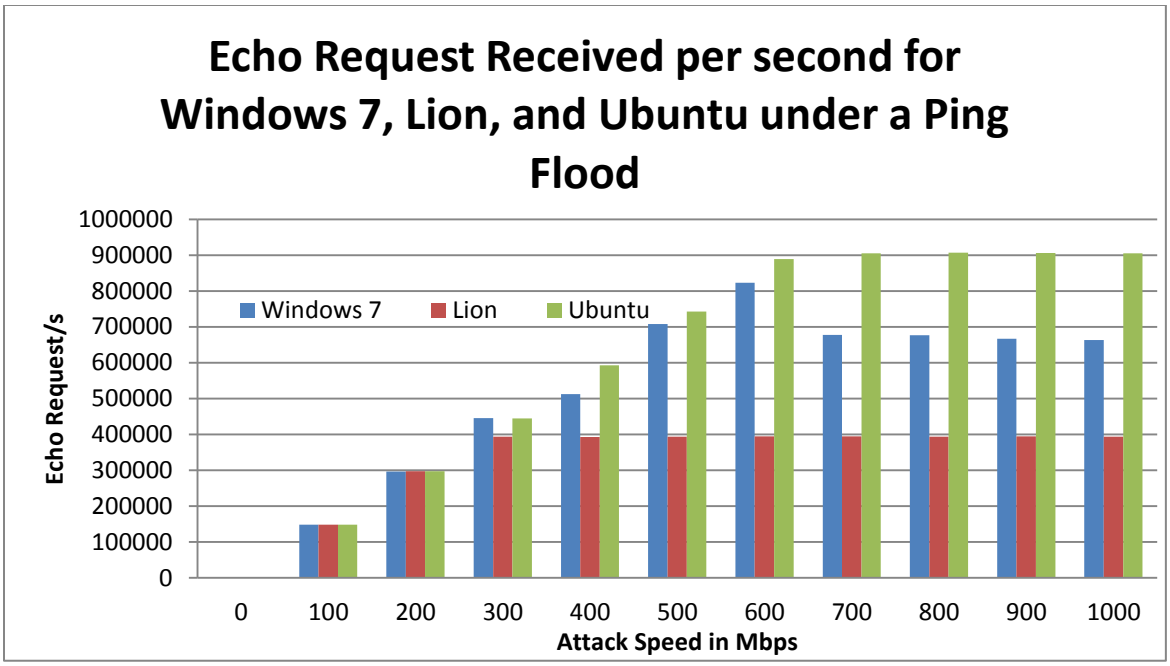


Figure 9 - Received Echo Request per second for Apple's, Canonical's, and Microsoft's OS's under an ICMP Ping Flood Attack

When it came to limiting the number of echo request and echo reply message being received and sent, all three operating systems were able to do so. However, some performed better than others. According to Table 1, it seems that Apple's OS Lion was able to have the lowest threshold limit for the number of echo request messages it would receive. However, this was not the case for the number of Echo reply that was sent. In fact our data shows that Windows 7 was only responding to 500 echo reply messages for the first second of the attack, then it stopped responding. This is an amazing observation, since we have verified that each company is implementing a unique solution to this type of DDoS attack. This fact introduces a new question "How does having different threshold values affect the utilization of the processor"?

Table 1:

Echo Request and Echo Reply per second for Lion, Windows 7, and Ubuntu under a Ping Flood attack

	Lion		Windows 7		Ubuntu	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	MAX: Echo Reply	Echo Request/sec	Echo Reply/sec
0	0	0	0	0	0	0
100	148830	250	148249	500	148422	148422
200	297620	250	296788	500	297002	297002
300	393442	250	445683	500	444683	351776
400	393200	250	512613	500	592928	348706
500	393680	250	707809	500	742469	326691
600	394483	250	822832	500	889358	265275
700	394287	250	677178	500	904736	258401
800	394170	250	676774	500	906854	261616
900	394320	250	667195	500	905826	263212
1000	393440	250	662857	500	904731	261131

When it came to the utilization of the processor, we observed some unexpected results. From Table 1, we can see that Lion has the lowest threshold limit for the number of Echo Request packets received per second stats at about 400 000 packets per second and Ubuntu had the highest threshold limit with about 904 000 Echo Request packets per second. We therefore concluded that Lion was going to have the smallest CPU utilization and Ubuntu would have the highest. However, we discovered that this was just the opposite. We saw that Lion had a maximum CPU utilization of 32 %. While Ubuntu's and Windows 7 each had a much lower CPU utilization of 13 % and 15 %, respectively, and can be seen in Figure 10.

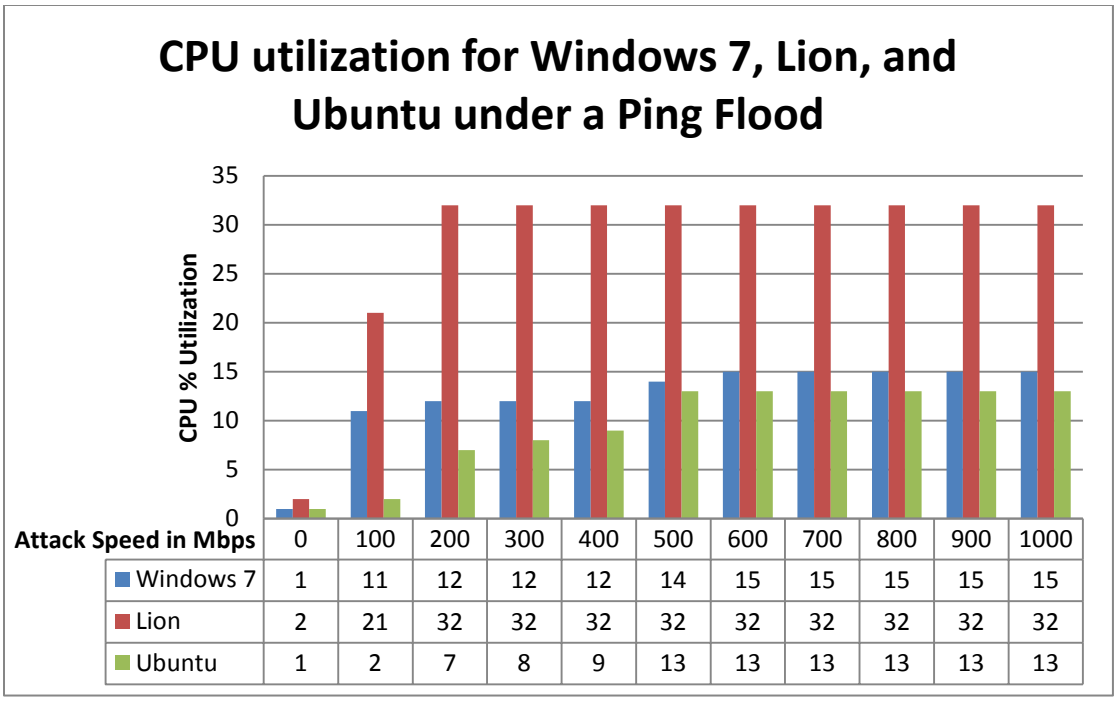


Figure 10 - CPU Exhaustion for Apple's, Canonicals, and Microsoft's OS's under an ICMP Ping Flood Attack

This discovery was unintuitive at first, but when one tries to understand the underlying aspect of a computer system and how it functions it becomes clear. When a packet is received by a computer, there is some processing done by the network interface controller before the packet is allowed into the computer system. If this packet is allowed in, then the computer will run the appropriate procedure which in turn will be executed by the CPU. If not, then this packet is dismissed and no further processing in allowed [56]. Therefore, we can clearly see that the procedures that the software engineers at Apple have developed to process the incoming packets are less efficient than Microsoft and Canonical such as polling or interrupt-driven I/O processing [57].

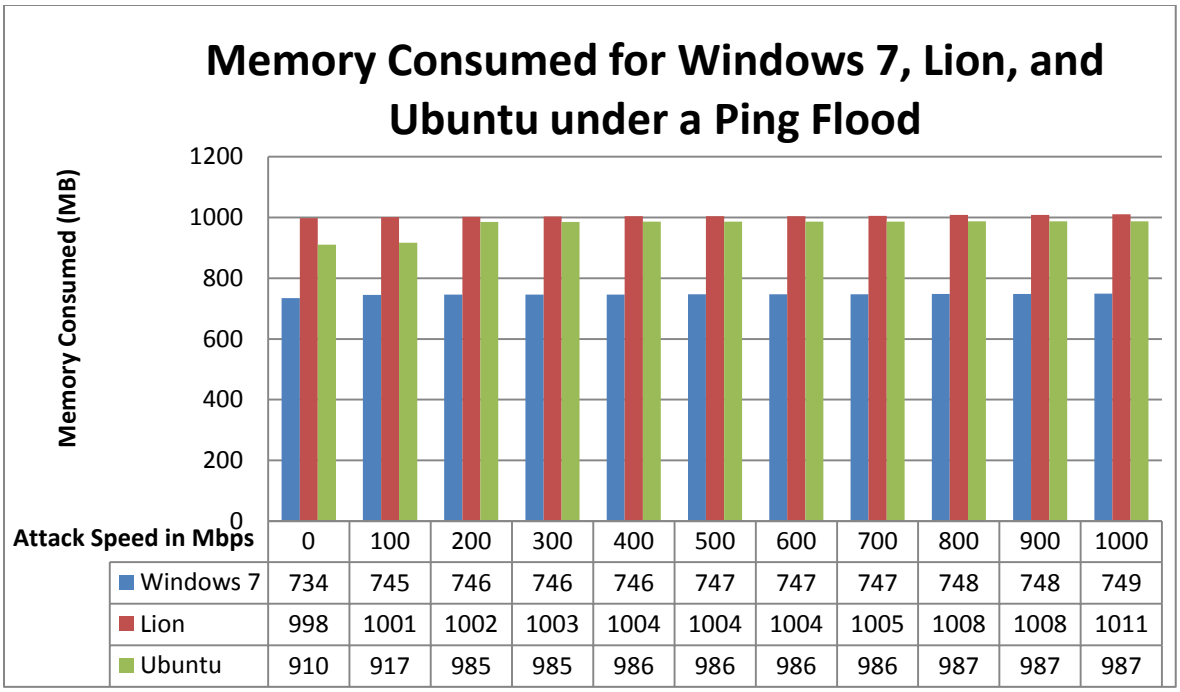
This gave us an understanding as to why the exhaustion of the processors might be different when comparing one operating system to another (Figure 10). We believe that this contributed to Apple's Lion having a maximum CPU exhaustion of 32%, while Windows 7 and Ubuntu had a maximum CPU utilization of 15% and13% respectively. We observed that the exhaustion of the iMac processors when running it native OS was at its maximum under the relatively low attack load of 200 Mbps of attack traffic. This was not the case when the iMac desktop was running Windows 7 or Ubuntu. Windows 7 was only utilizing 12% of the CPU while Ubuntu was utilizing 7 % of the CPU to process 200 Mbps of attack traffic. It was not until the attack loads of 600 Mbps and 500 Mbps that Windows 7 and Ubuntu, respectively, reached their maximum for the exhaustion of the processor. With all three operating systems, we discovered that once the maximum CPU utilization was reached it stayed the same.

Even though it seemed that Apples Lion was limiting the number of echo request and echo replies packets that it would receive to about 400 000 per second and 250 per second, which seems to be more efficient, this did not really contribute to the overall performance. However, what can be seen is that all three operating systems were only using one of the four cores to execute the attack. This was the same technique that Windows XP was using when the computer's architecture had multiple logical processors [45]. This technique helped since the attack was being contained to only one of the cores, leaving the other cores free to process any application request from the user.

Under normal operating conditions, 0 Mbps of attack traffic, we observed that the amount of RAM that was being consumed was higher for Apple's Lion than Microsoft's Windows 7 and Ubuntu. This was partially due to the fact the most of Apple's operating systems seem to be

executing unnecessary application at start up. For instance, some of the applications that Apple's Lion would start up were Safari, finder, docks, and other unnecessary applications. Due to this fact, we believe that this contributed to Lion having the highest memory and CPU consumption at start up (Figure 11).

As for the amount of memory that was being consumed during the Ping Flood attack, it seemed to be consistent for all three of the operating systems. The amount of memory that was being consumed by this attack was minimal. At start up, the amount of memory that was being consumed by Lion was 998 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1011Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu was consuming 910 Mbytes of memory at 0 Mbps of attack traffic. As the attack traffic was increased, we saw a steady but small increase in the consumption of memory. The biggest jump in memory occurred at 200 Mbps of attack traffic. At 100 Mbps, the amount of memory being consumed was 917 Mbytes and once the attack increased to 200 Mbps the consumption increased to 985 Mbytes. This was a 52 Mbyte increase in memory. Overall, Ubuntu had a maximum consumption of memory of 987 Mbytes and was reached at 1000 Mbps of attack traffic. At start up, the amount of memory that was being consumed by Windows 7 was 734 Mbytes and as a baseline was the lowest. As we increased the attack traffic, we observed that Windows 7 had a maximum memory consumption of 749 Mbytes and occurred at 1000 Mbps of attack traffic.



Attack Speed in Mbps	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	734	745	746	746	746	747	747	747	748	748	749
Lion	998	1001	1002	1003	1004	1004	1004	1005	1008	1008	1011
Ubuntu	910	917	985	985	986	986	986	986	987	987	987

Figure 11- Memory Consumed for Apple's, Ubuntu, and Microsoft's OS's under an ICMP Ping Flood Attack

3.3.2 ICMP Land Flood Attack

In the case of the land attack, the graphs that we obtained were a little different than what was expected. We expected that all three of the operating systems would use most of the CPU and memory processing both the echo request and echo reply messages, as in the case with Apple's Leopard and Windows Vista [45].

According to the data that was collected for the Land Attack, we observed that Lion was exhibiting the same performance for the number of echo request and echo reply messages that it would receive and send while under a Ping Flood attack. We can see from table 2 and Figure 12 that Lion was limiting the number of echo request messages that it would receive to 400 000 per second at 300 Mbps of attack traffic. As the attack traffic was increased, the threshold limit was

able to limit this parameter. Regardless of the number of echo request messages that Lion was receiving, we found that Lion was again limiting the sending of echo reply messages to 250 per second.

The same can be said about Microsoft's Windows 7, which was also using the same threshold limiting techniques for the number of echo request received and echo replies sent per second that was implemented for the Ping Flood attack. By looking at Table 2 and Figure 12, we can see that Windows 7 was limiting the number of echo request that it was receiving to around 710 000 packets per second and occurred at 500 Mbps of attack traffic. However, unlike the Operating System Lion, Windows 7 was implemented to not respond to a Land Attack echo request packet. This was sort of expected, since Windows 7 was only acknowledging a maximum of 500 echo request packet while under the Ping Flood attack.

What we discovered was that Ubuntu's operating system "Saucy Salamander", like Lion, had the same limiting threshold values for the Ping and Land Attack Floods. Saucy Salamander was receiving a maximum of 906 000 echo request packets per second at the attack traffic of 800 Mbps. When it came to the number of echo reply packets that Ubuntu was replying to, we observed that the threshold was at about 350 000 echo reply packets per second. The limiting threshold set for Ubuntu was the highest, but consumed the CPU the least which had little contribution to the overall utilization of the CPU.

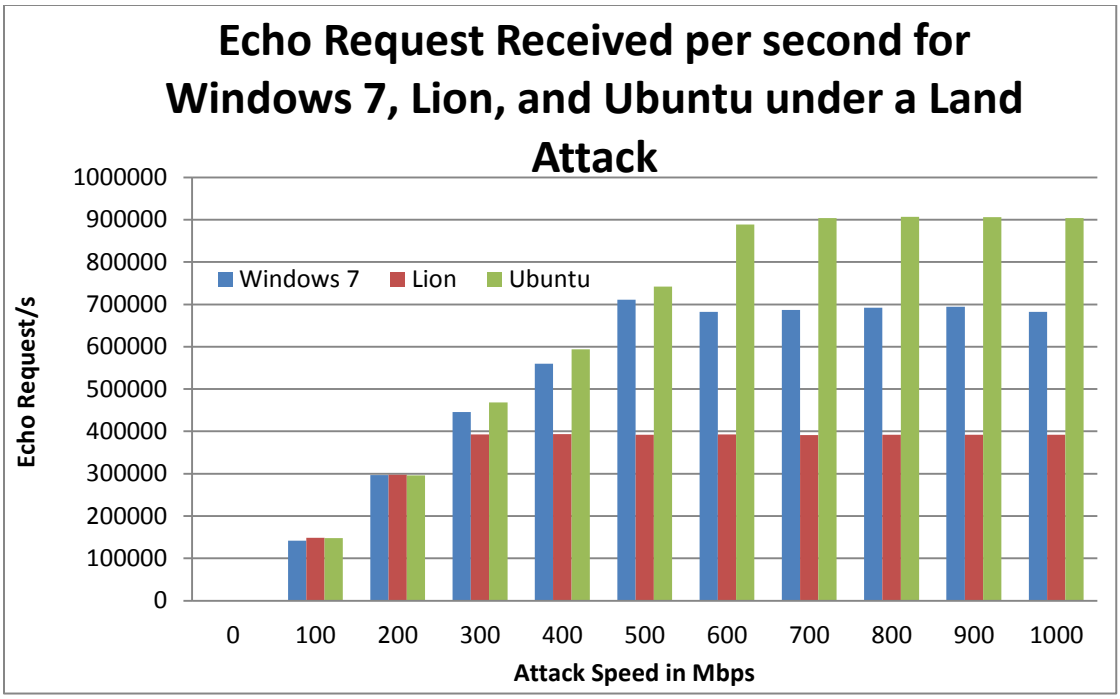
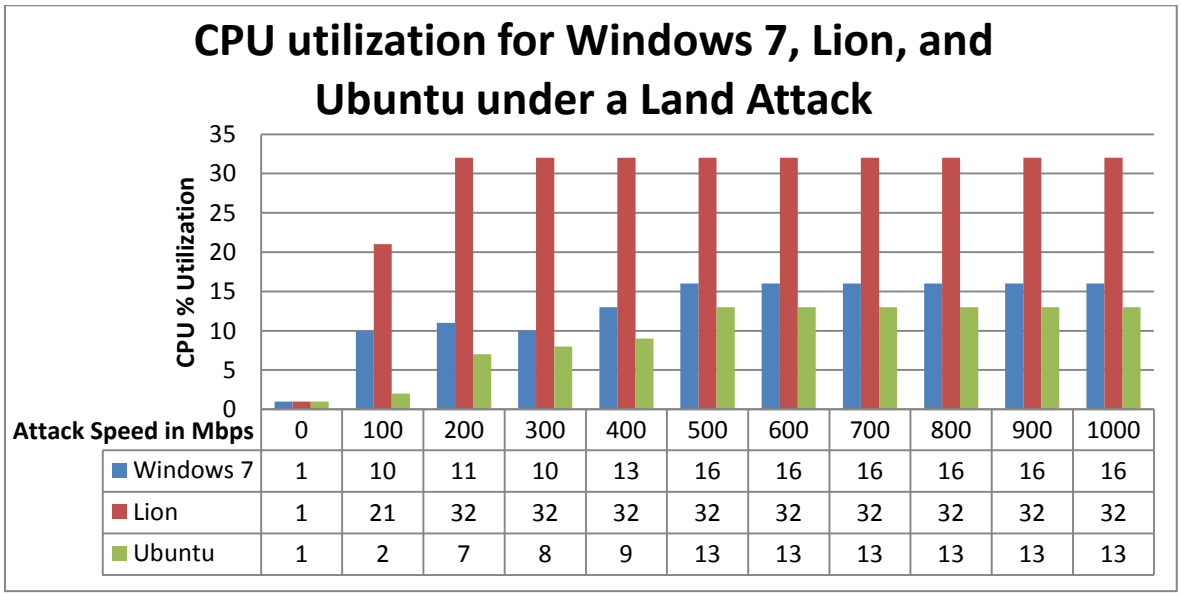


Figure 12- Received Echo Request for Apple's, Canonical's, and Microsoft's OS's under an ICMP Land Attack

Table 2:

Echo Request and Echo Reply per second for Lion, Windows 7, and Ubuntu under a Land Attak Flood						
	Lion		Windows 7		Ubuntu	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec
0	0	0	0	0	0	0
100	148755	250	147502	0	147522	147522
200	297653	250	296449	0	296053	296053
300	392761	250	445522	0	468546	352653
400	393240	250	559820	0	593652	350253
500	391979	250	711266	0	742153	325658
600	392585	250	682639	0	888995	266356
700	391560	250	686729	0	903785	257562
800	392160	250	692254	0	906542	260259
900	392182	250	694162	0	905845	260250
1000	392254	250	682542	0	903895	260305

From Figure 13, we can see that Lion reached a maximum CPU consumption of 32% at the relatively low attack speed of 200 Mbps. While both Windows 7 and Ubuntu had a maximum CPU consumption of 16% and 13%, respectively, at the same attack load speed of 500 Mbps. We saw that Lion, Windows 7, and Ubuntu each had a baseline reading of 1 % for CPU utilization. From Figure 13, we can conclude that the exhaustion of the iMac processor when running it native OS was at its maximum under the attack load of 200 Mbps and as the attack traffic was increased, the consumption of the processor stayed the same. We can see that Windows 7 had a maximum CPU exhaustion of 16% at the attack speed of 500 Mbps. From Figure 6, we can see that Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 13% at the attack speed of 500 Mbps.

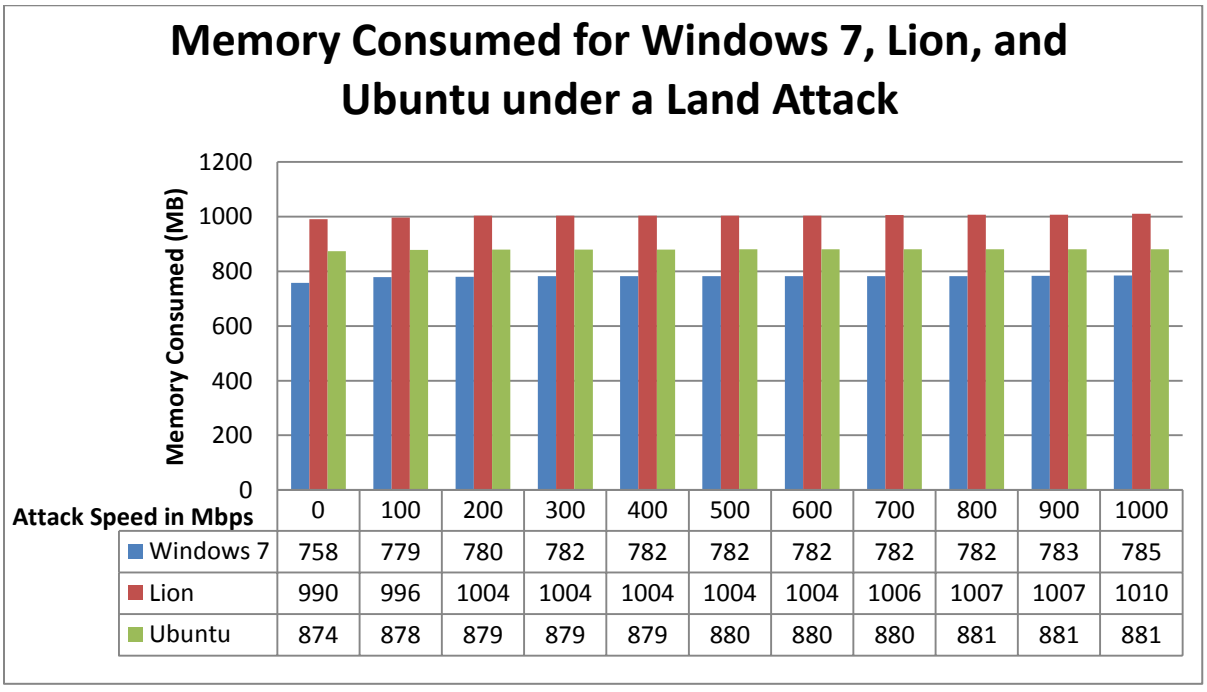


Attack Speed in Mbps	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	1	10	11	10	13	16	16	16	16	16	16
Lion	1	21	32	32	32	32	32	32	32	32	32
Ubuntu	1	2	7	8	9	13	13	13	13	13	13

Figure 13- CPU utilization for Apple's, Canonical's, and Microsoft's OS's under an ICMP Land Attack Flood

As for the amount of memory that was being consumed during the Land Attack Flood, we observed a similar pattern during the Ping Flood attack. The amount of memory that was

being consumed by this attack was minimal. At start up, the amount of memory that was being consumed by Lion was 990 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1010 Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu was consuming 874 Mbytes of memory at 0 Mbps of attack traffic. As we increased the attack traffic, we saw a steady but small increase in the consumption of memory. The maximum consumption of memory was 881 Mbytes as was reached at 1000 Mbps of attack traffic. At start up, the amount of memory that was being consumed by Windows 7 was 758 Mbytes and as a baseline was the lowest. As the attack traffic was increased, we observed that Windows 7 had a maximum memory consumption of 785 Mbytes and occurred at 1000 Mbps of attack traffic.



Attack Speed in Mbps	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	758	779	780	782	782	782	782	782	782	783	785
Lion	990	996	1004	1004	1004	1004	1004	1006	1007	1007	1010
Ubuntu	874	878	879	879	879	880	880	880	881	881	881

Figure 14- Memory Consumed for Apple's, Canonical's, and Microsoft's OS's under an ICMP Ping Flood Attack

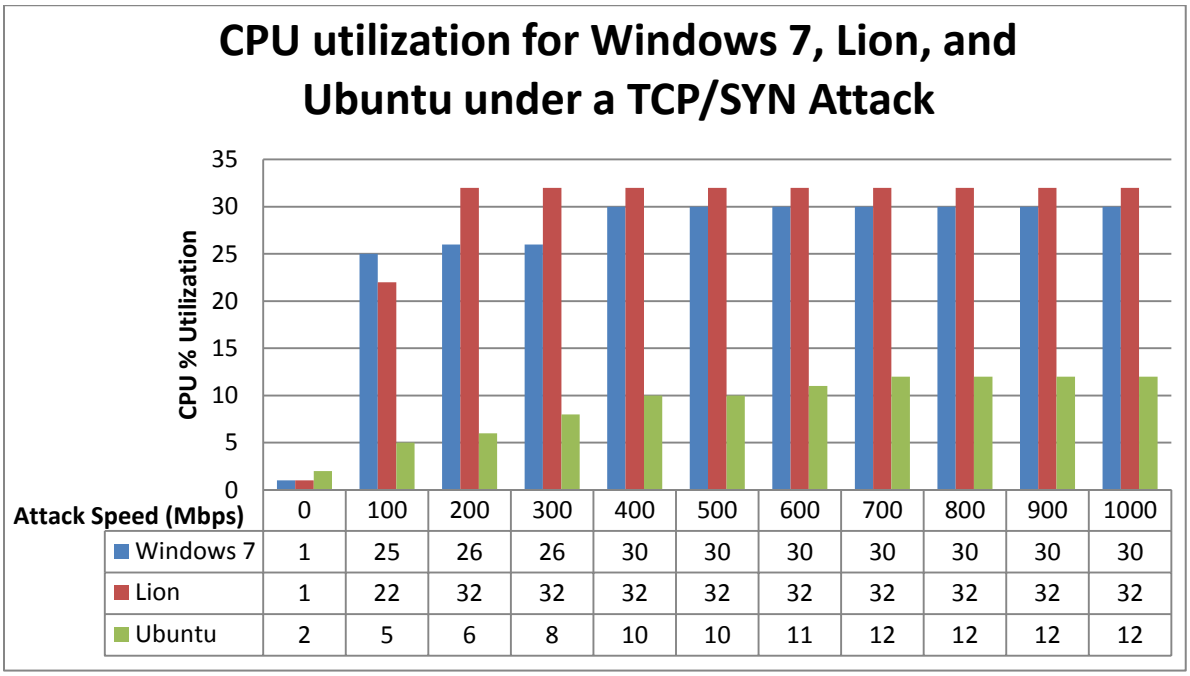
3.3.3 TCP-SYN Flood Attack

In this section, we will begin by presenting the data for the closed port TCP-SYN attack followed by the open port TCP-SYN attack. One thing to note is that we are trying to establish numerous amounts of half-open TCP connections. We will attempt to establish half-open connections to two different types of TCP ports, one that is closed and non-listening and another that will be opened and listening. The idea behind this variation is simple and can be explain by an analogy. Consider a home that is empty, if someone comes knocking at the door we do not expect to receive a response. However, if there is someone at home we expect to be greeted and a response is expected. That is the idea behind a closed and open port. A closed and non-listening port is an empty home, while an open and listening port is a home with someone inside it. By attacking a closed port, we assume that the overall performance of the computer system should not get affected. We believe that attacking an open TCP port will result in a more devastating attack, when compared to a closed port attack. Overall, we wanted to see the effects of each attack and if all three operating systems had taken precautions by implementing security for them.

We will begin by presenting the data for the closed port attack, followed by the data for the open port TCP/SYN attack. In each attack, we will be presenting the utilization of the CPU followed by the amount of memory that was consumed during the attack. For both of these attacks, we will be trying to create half-open TCP connections by sending a TCP packet with the SYN flag set. We will be increasing the attack traffic from 0 to 1000 Mbps for a duration of six minutes and a cool down period of two minutes in between each trial.

Closed Port

From Figure 15, we can see that Lion reached a maximum CPU consumption of 32% at the relatively low attack speed of 200 Mbps. While both Windows 7 and Ubuntu had a maximum CPU consumption of 30 % and 12%, respectively, at the attack load speed of 400 and 700 Mbps. We can see that for the baseline CPU utilization, Lion and Windows 7 were using 1 % of the CPU while Ubuntu had a 2 % utilization.



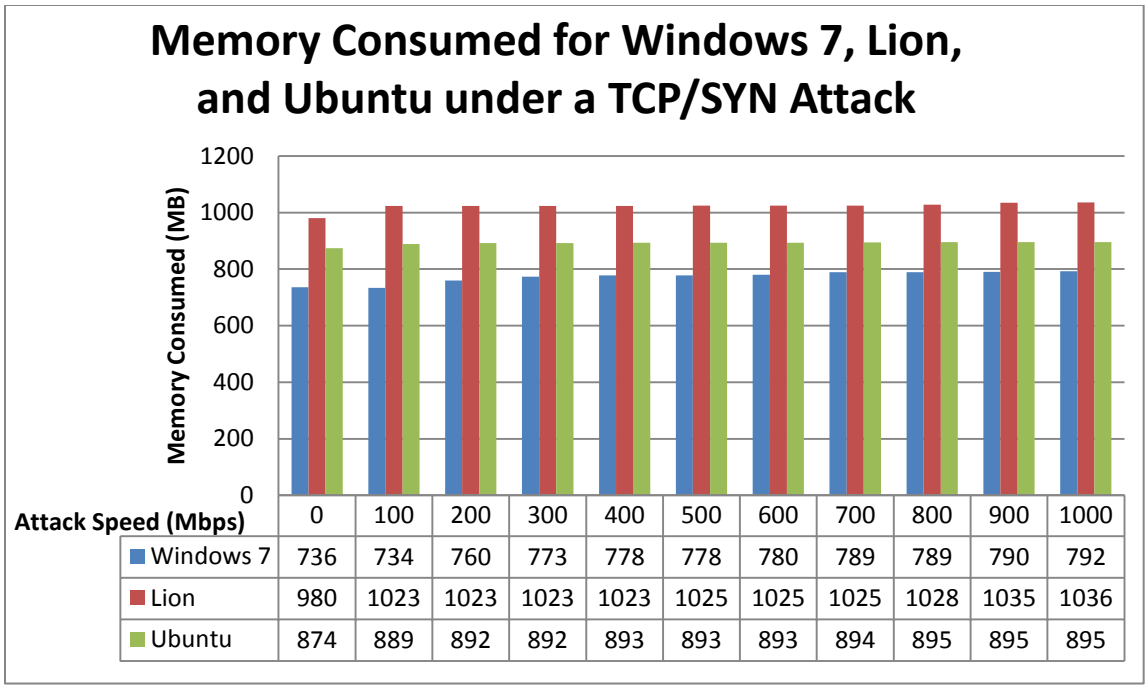
Attack Speed (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	1	25	26	26	30	30	30	30	30	30	30
Lion	1	22	32	32	32	32	32	32	32	32	32
Ubuntu	2	5	6	8	10	10	11	12	12	12	12

Figure 15- CPU utilization for Apple's, Canonical's, and Microsoft's OS's under a TCP/SYN Flood

Based on the data, we concluded that the exhaustion of the iMac processor when running it native operating system, Lion, had a maximum CPU utilization of 32 % and was reached under the attack load of 200 Mbps. When we increased the attack traffic, we notice that the consumption of the processor stayed the same throughout the rest of the test cases. We can see

that Windows 7 had a maximum CPU utilization of 30 % at the attack speed of 400 Mbps. When Ubuntu was introduced to the TCP/SYN attack, we observed that overall Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 12% at the attack speed of 700 Mbps.

As for the amount of memory that was being consumed during the closed TCP/SYN attack, the amount of memory that was being consumed by this attack was minimal. At start up, the amount of memory that was being consumed by Lion was 980 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1086 Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu was consuming 874 Mbytes of memory at 0 Mbps of attack traffic. As the attack traffic was increased, we saw a steady but small increase in the consumption of memory. The maximum consumption of memory was 895 Mbytes as was reached at 1000 Mbps of attack traffic. At start up, the amount of memory that was being consumed by Windows 7 was 736 Mbytes and as a baseline was the lowest. As the attack traffic was increased, we observed that Windows 7 had a maximum memory consumption of 792 Mbytes and occurred at 1000 Mbps of attack traffic.



Attack Speed (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	736	734	760	773	778	778	780	789	789	790	792
Lion	980	1023	1023	1023	1023	1025	1025	1025	1028	1035	1036
Ubuntu	874	889	892	892	893	893	893	894	895	895	895

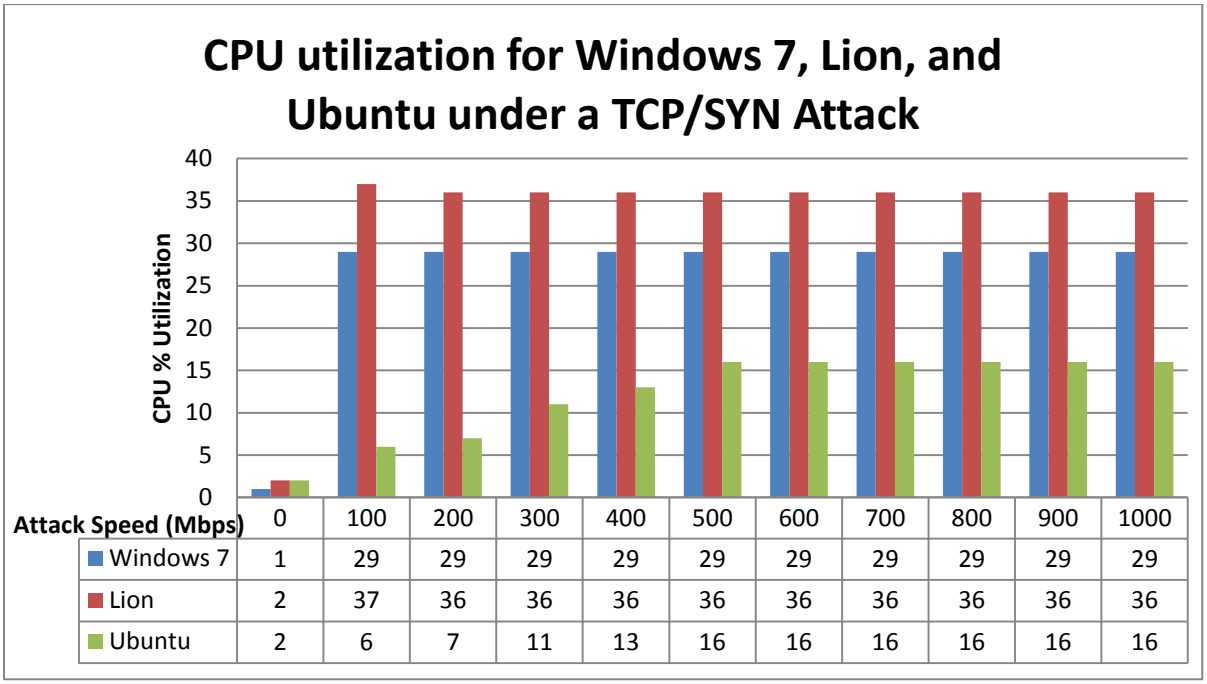
Figure 16 - Memory Consumed for Apple's, Canonical's, and Microsoft's OS's under a TCP/SYN Flood

Open Port

By altering the attack and sending a connection request packet to an open port, rather than to a closed port, we discovered that the attack affected both Lion and Ubuntu differently when compared to the closed port attack. From Figure 17, we can see that Lion reached a maximum CPU consumption of 37% at the attack speed of 100 Mbps. While both Windows 7 and Ubuntu had a maximum CPU consumption of 29 % and 16%, respectively, at the attack load speed of 100 and 500 Mbps. We can see that for the baseline CPU utilization, Lion and Ubuntu were using 1 % of the CPU while Windows 7 had a 2 % utilization.

As mentioned above, we observed that Lion reached its maximum CPU utilization at the attack load of 100 Mbps and reached 37 %. As we increased the attack traffic, we noticed that

the consumption of the processor decreased by 1 %, but we believe that this small change is negligible. We can see that Windows 7 had a maximum CPU utilization of 29 % and was reached at the attack speed of 100 Mbps. This behavior was new and when compared to the closed port attack, we observed that the max was reached at 400 Mbps. When Ubuntu was introduced to the opened port TCP/SYN attack, we observed that overall Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 16 % at the attack speed of 500 Mbps. As with the other two operating systems, when the maximum utilization occurred we did not see a decrease in the exhaustion of the CPU as the attack traffic was increased.

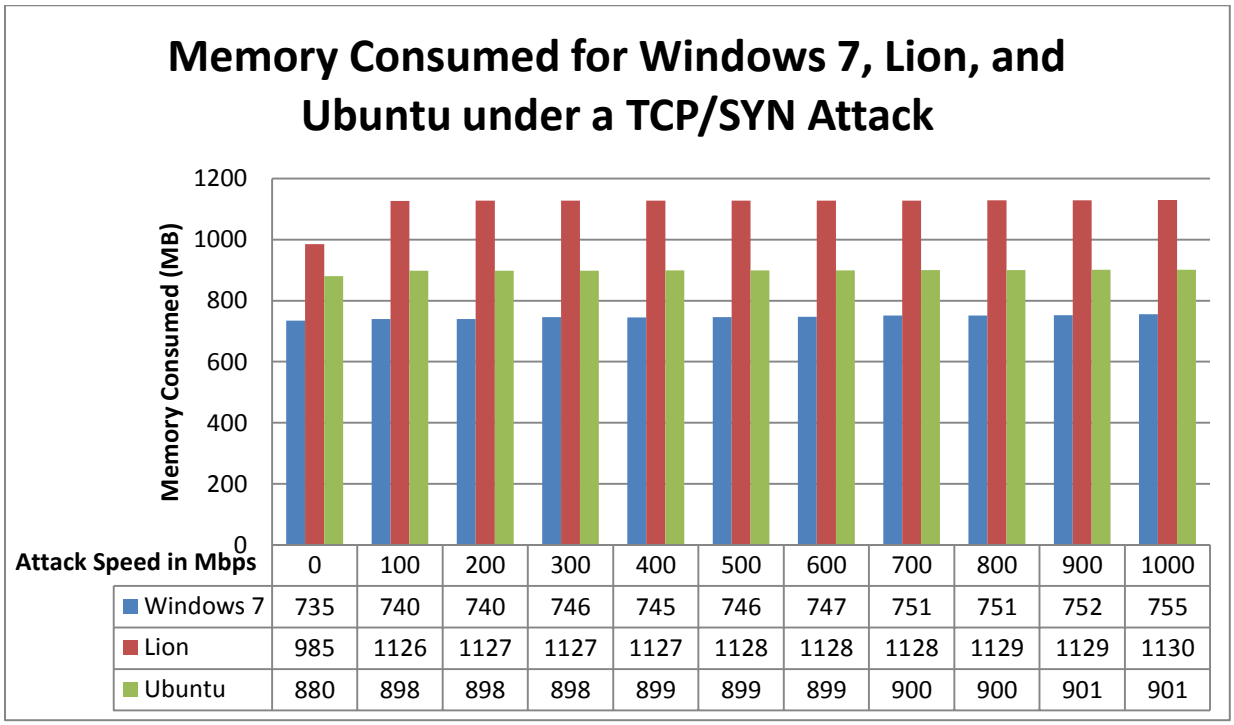


Attack Speed (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	1	29	29	29	29	29	29	29	29	29	29
Lion	2	37	36	36	36	36	36	36	36	36	36
Ubuntu	2	6	7	11	13	16	16	16	16	16	16

Figure 17 - CPU utilization for Apple's, Canonical's, and Microsoft's OS's under a TCP/SYN Flood

As for the amount of memory that was being consumed during the Land Attack Flood, we observed a similar pattern during the Ping Flood attack. The amount of memory that was

being consumed by this attack was minimal. At start up, the amount of memory that was being consumed by Lion was 985Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1130 Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu was consuming 880 Mbytes of memory at 0 Mbps of attack traffic. As the attack traffic was increased, we again observed a steady but small increase in the consumption of memory. The maximum consumption of memory was 901 Mbytes as was reached at 1000 Mbps of attack traffic. At start up, the amount of memory that was being consumed by Windows 7 was 735 Mbytes and again as a baseline reading was the lowest. As we increased the attack traffic, we observed that Windows 7 had a maximum memory consumption of 755 Mbytes and occurred at 1000 Mbps of attack traffic.



Attack Speed in Mbps	0	100	200	300	400	500	600	700	800	900	1000
Windows 7	735	740	740	746	745	746	747	751	751	752	755
Lion	985	1126	1127	1127	1127	1128	1128	1128	1129	1129	1130
Ubuntu	880	898	898	898	899	899	899	900	900	901	901

Figure 18 - Memory Consumed for Apple's, Canonical's, and Microsoft's OS's under a TCP/SYN Flood

3.4 Chapter Summary

With the utilities provided by each operating system, we were able to gather information about the performance of the computer while under the ICMP Ping Flood, Land Attack, and TCP/SYN Flood attacks [58] – [62]. Then, by using Microsoft's Excel, we were able to translate our data in graphs. By graphing our data, we are able to intuitively compare our data.

Overall, all three operating systems were implementing similar techniques when under each attack. For instance, Lion, Windows 7, and Ubuntu were limiting the number of Echo Request and Echo Reply messages that they would receive and send. Even though it seemed that Apples Lion was limiting the number of echo request messages more efficiently and was only sending 250 echo reply messages every second, this did not contribute to an overall better performance. We can think about this as a tradeoff situation. For instance, Apple chose a lower threshold limit for the number of echo request message it would receive. By having a lower threshold limit than its competitors, it seemed to have led to a higher CPU consumption. We believe that this led to more CPU cycles needed to execute the procedure that was design by Apple to handle this threshold, or maybe it required extra memory to accomplish the limiting threshold task than to process the ping messages.

Even with a lower threshold limit, Lion was outperformed by Windows 7 and Ubuntu when we compared the amount of exhaustion of the CPU. According to the simulations, Windows 7 had a maximum CPU consumption of 15%, while under the Ping Flood attack. Lion had a maximum CPU consumption of 32% while Ubuntu was using 13 %.

When it came to subjecting our operating systems to the Land Attack flood, we again observed the same limiting technique as in the Ping Flood Attack. Lion was limiting the number of echo request packet to a maximum of 400 000 packets per second. Windows 7 and Ubuntu were limiting theirs to about 710 000 and 906 000 packets per second respectively. When it came to replying to these packets, we observed that Windows 7 was not acknowledging any of the echo request packets that it was receiving. With Lion, we observed that it was replying with 250 echo reply packet per second for the duration of the attack. According to our test results, we found that Ubuntu again had the lowest CPU utilization. It had a maximum CPU utilization of 13 %. The highest was Lion with 32 % and Windows with 15%. One thing to mention is that Windows 7 seemed to be the only operating system that was able to detect the subtle difference between the Ping and Land Attacks.

When introducing our operating systems to the closed TCP/SYN attack, Lion, had a maximum CPU utilization of 32 % and was reached under the attack load of 200 Mbps. We also saw that Windows 7 had a maximum CPU utilization of 30 % at the attack speed of 400 Mbps and Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 12% at the attack speed of 700 Mbps.

When introducing our operating systems to the open TCP/SYN attack, Lion, had a maximum CPU utilization of 37%. We also saw that Windows 7 had a maximum CPU utilization of 29 % at the attack speed of 400 Mbps and Ubuntu had a maximum CPU exhaustion of 16 % at the attack speed of 700 Mbps.

Overall, we saw that all three operating systems were able to survive these attacks. The transition that the developers have made is astounding. We saw that in previous version of Windows operating system, Windows XP, the Ping Flood was able to disable the operating system. Now, Windows 7 is able to handle the attack and is still able to function. This transition in protection shows that security has been a major role in the software development cycle. Although Lion was able to survive the attack and still functioned correctly, we feel that since Lion was outperformed by both operating systems the claim that an iMac computer running the latest Apple operating system is the most secure system was false. Claims like this that are made by big corporations are example of statements that need to be researched and verified by the public domain before they are allowed to make claims like this and market their software to the general public.

CHAPTER IV

COMPARATIVE EVALUATION OF WINDOWS BASED SERVERS UNDER DDOS ATTACKS

According to a survey in November of 2013, Windows Server 2008 and Windows Server 2008 R2 were named as one of the top three most reliable, mainstream server operating systems in used today [63]. Currently, Microsoft-IIS is being used by approximately 13.2 % of the websites in the world according to a survey by w3techs.com [64]. With multiple versions of IIS in the market, Version 6 and Version 7 are considered the most dominate versions and hold 26.2 % and 57.9 % of the market respectively [65]. Now with the release of Windows Server 2012, we would like to evaluate and compare the built-in security that is provided by Windows Server 2012 with its predecessor while under a DDoS attack.

4.1 Experimental Setup

In the controlled environment of the Network Research Lab (NRL) at The University of Texas-Pan American, the performance of Microsoft's Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 were evaluated under ICMP and TCP based attack traffic up to a maximum speed of 1 Giga bit per second (Gbps). This time around, we used an iMac Pro server that will be setup as a web server, and will be tested to see how many successful TCP connections per second it is able to handle while running the mentioned operating systems. By

using the setup guide provided by Microsoft, we are able to install IIS-8 on Windows Server Enterprise 2012 R2 and IIS-7 on Windows Server Enterprise 2008 R2 [66].

The victim server platform was an Apple iMac Pro Server equipped with an Intel Xeon 2.8-GHz quad-core processor, 12 GBytes of RAM, and a Broadcom NetXtreme Gigabit Ethernet adapter. The Apple iMac Pro Server has the capabilities of running the following OS: Microsoft "Windows Server Enterprise 2008 R2" and Microsoft "Windows Server Enterprise 2012 R2". The test that we plan to run are similar to the ones in [43] – [47], but with the operating systems mentioned above. By doing this, we can see if the software engineers at Microsoft have introduced or modified any security features in Windows Server 2012 that its predecessor did not have.

The type of evaluation method that was used was experimental testing. We first established a baseline of successful legitimate HTTP requests to Windows IIS web server while under no attack. Then, while sending the legitimate HTTP traffic we introduced a barrage of attack traffic in increments of 100 Mbps of traffic for 10 minutes. This was done to study the impact or effects that the denial of service traffic would have on the server's resources and the legitimate traffic.

The experimental setup consisted of an Ethernet switch, a traffic generator, and an iMac Pro Server and can be seen in Figure 19. We will use the CAT6 Ethernet cable to connect the traffic generator that will simulate the attack traffic to the Ethernet switch. We then used another CAT6 cable to connect the switch to the iMac Pro Server. We will give the specification of the hardware in the next section. In general, the software was used to gather the statistics on the

parameters that are needed to compare and evaluate the testing that was performed. The data that was gathered will be used to plot useful graphs that we can used to compare our results in an intuitive way.

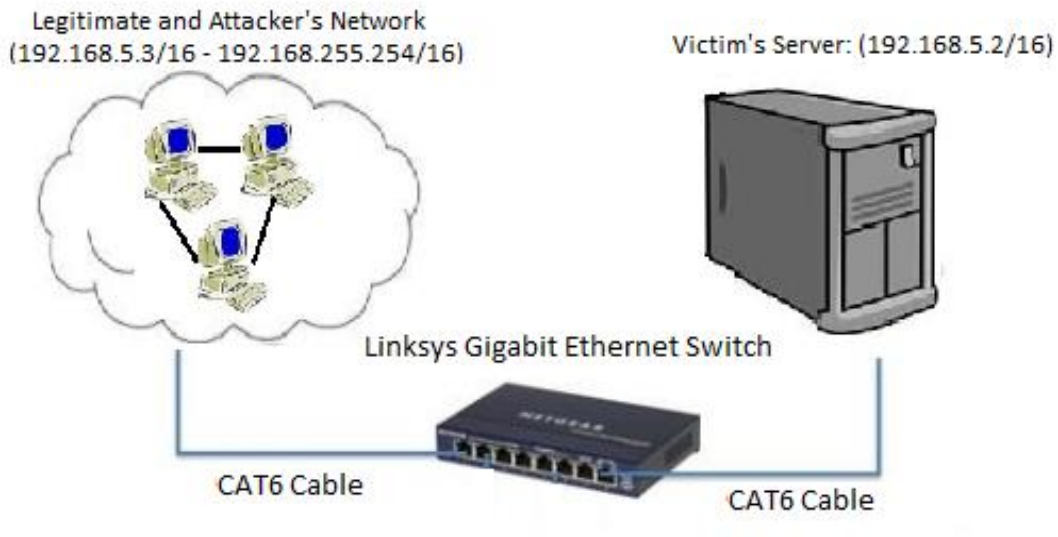


Figure 19 - Experimental Set-up

With the utilities provided by each operating system, we were able to gather information about the performance of the computer while under the TCP and ICMP attack traffic. We will then use Microsoft's Excel to obtain graphs for the information that was collected to aid us in the comparison of the two systems.

4.1.1 Hardware

Switch:

The type of Ethernet switch that will be used in the experiment is the Cisco SRW2024 24-port Gigabit Switch. It supports gigabit networking, offering exceptional performance over

24separate connections. Since security is a priority on the SRW2024, no attached device receives information before it authenticates with the switch. If more details are needed you may refer to [48] [49].

Computer Platform:

The victim server used was an Apple iMac Pro Server equipped with an Intel Xeon 2.8-GHz quad-core processor, 12 GBytes of RAM, and a Broadcom NetXtreme Gigabit Ethernet adapter [67]. The Apple iMac Pro has the capabilities of running the following operating systems: Microsoft's "Windows Server Enterprise 2008 R2" and "Windows Server Enterprise 2012 R2".

Computer 1 has the following specification:

Operating Systems: Microsoft "Windows Server Enterprise 2008 R2" and Microsoft "Windows Server Enterprise 2012 R2".

CPU: Intel Xeon 2.8 GHz

Number of Processors: 1

Number of Cores: 4

Random Access Memory (RAM): 12 Giga Bytes

Network-Interface-Card (NIC): Broadcom NetXtreme Gigabit Ethernet controller

Graphics:

Chipset Model:

Type:

BUS: PCI

4.1.2 Software

The software that was used includes Microsoft's excel spreadsheet [54], and Microsoft's Performance Monitor [55]. By using excel, we were able to obtain graphs with the information that was collected during testing using the performance monitor.

4.2 Parameters of Performance Evaluation

For this experiment, the parameters that will be used to evaluate the performance of each operating system are the Connection Rate, the Processors utilization, the number of Echo request packets received per second, the number of Echo reply packets sent per second, and the amount of random access memory being consumed in Mbytes. These resources are being measured while the platform is being subjected to our denial of service attack traffic and are crucial for the evaluation of our system and are described below:

Connection Rate (TCP connections/second) – Whenever a webpage gets requested from a webserver, a connection will be established using the Transmission Control Protocol (TCP) between the client and server. If the connection is unsuccessful, then the webpage will not get fetched and an error will be displayed. Therefore, it is imperative to verify the effects that these attacks have on the number of successful TCP connection.

CPU Utilization (Usage of CPU in %) – The utilization of the Central Processing Unit (CPU) is one of the most important parameters to keep track of in any normal or abnormal situation. The utilization of the CPU informs the user of the amount of work that is being performed by the server at that instance.

Echo Request Received per second (Echo Request /Sec) – This parameter will allow us to measure the number of echo request messages that were being received per second by the server. When a computer received an echo request message, it must reply with an echo reply message [38]. As the value of these parameters increase, we hope to identify the impact that it has on the other parameters that we will be evaluating.

Echo Reply Sent per second (Echo Reply / Sec) – This parameter will allow us to measure the number of echo reply messages that are being sent per second by the server. These Echo Reply messages are being sent in response to the echo request messages that were received.

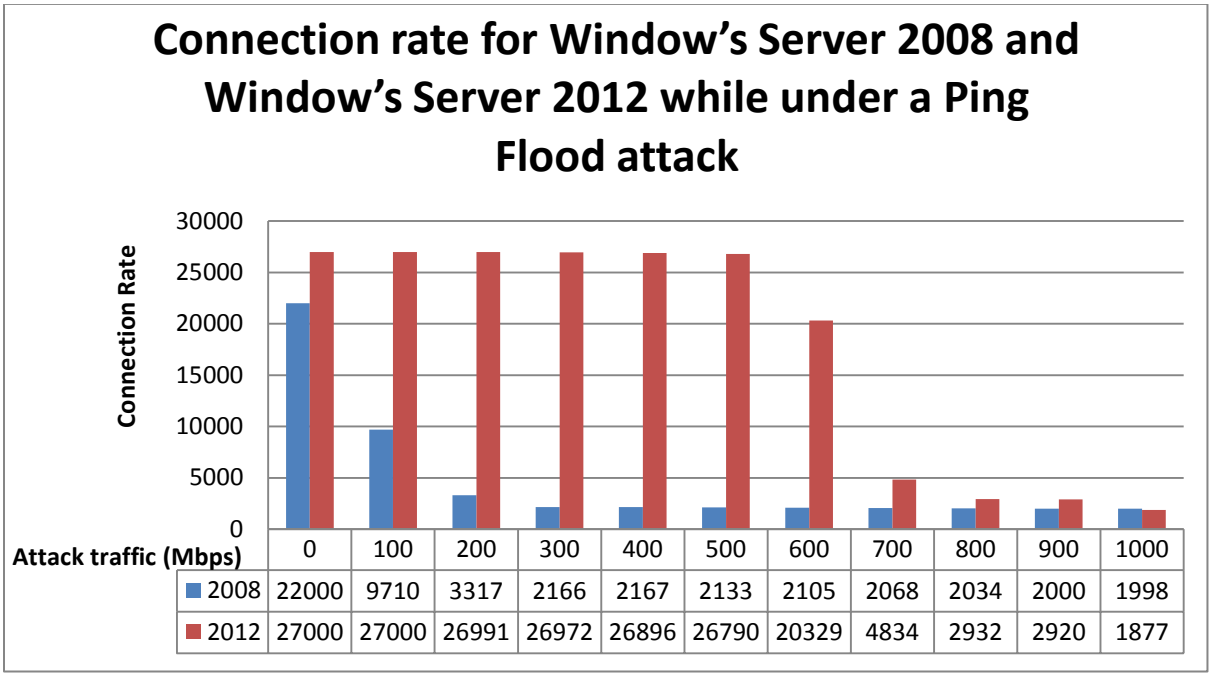
Random Access Memory (RAM) consumed (in Mbytes) – This parameter measures the amount of finite random access memory that is being consumed by the attack. If the RAM becomes completely consumed, the computer will become slow and can become unstable.

4.3 Results and Discussions

In this experiment, we will be testing the built-in security of two popular Microsoft Operating Systems in use in today's servers: Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2. The parameters used for comparison are the number of successful TCP connections per second, the number of echo request and echo reply messages that each operating system receives and sends, the utilization of the processor, and the amount of memory being consumed.

4.3.1 ICMP Ping Flood Attack

We found that when the iMac Pro Server was running Windows Server 2008, it was able to successfully handle 22,000 connections per second under normal operating conditions, while Windows Server 2012 was able to handle 27,000 connections per second (see Figure 20). Once the ICMP Ping flood traffic was introduced, we can see from Figure 20 that a significant drop in the connection rate for Windows Server 2008 occurred. We saw that the connection rate drop from 22,000 to 9,710 connections per second while under 100 Mbps of Ping Flood attack traffic.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	22000	9710	3317	2166	2167	2133	2105	2068	2034	2000	1998
2012	27000	27000	26991	26972	26896	26790	20329	4834	2932	2920	1877

Figure 20 - Connection rate for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Ping Flood Attack

As the attack traffic was increased, the connection rate continued to decrease until 300 Mbps of attack traffic. After 300 Mbps, the connection rate stayed steady at about 2,000 connections per second. This was not the case for Windows Server 2012, we observed that it was not until the attack traffic of 600 Mbps that the server was affected and the connection rate

decreased to 20,300 connections per second. However, at the attack traffic of 700 Mbps and beyond, we observed that the connection rate was less than 5,000 connections per second.

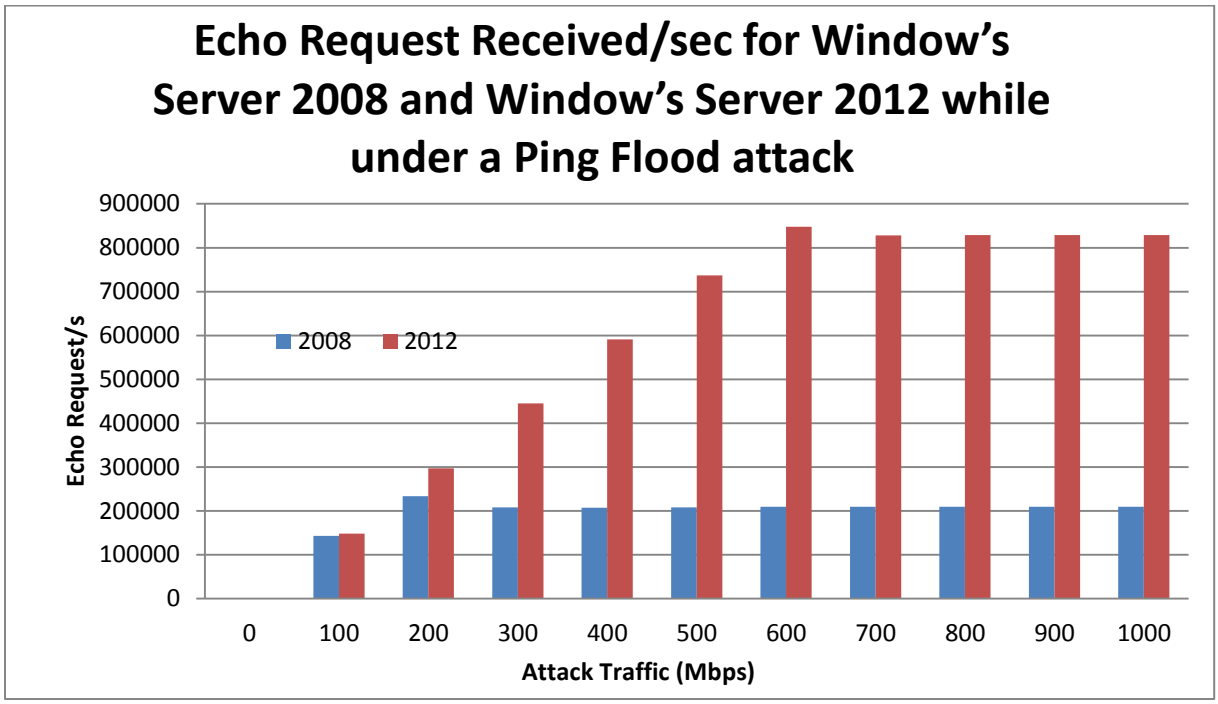


Figure 21 - Received Echo Request per second for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Ping Flood Attack

We also discovered that both operating systems were limiting the number of Echo Request and Echo Reply messages that were being received and sent (see Figure 21 and Table 3). According to the data that was collected, we can see that Windows Server 2008 was limiting the number of echo request packets that it would receive to about 233,000 per second. At 100 Mbps, the number of echo request messages that were received was about 143,000 per second and at 200 Mbps the limiting threshold was reached. What was surprising was that Windows Server 2012 was using the same limiting technique, but had a higher threshold limit when

compared to Windows Server 2008. As shown in Figure 21, we observed that the threshold limit was not reached until the attack traffic of 600 Mbps and reached as high as 847,000 echo request messages received per second. This is almost four times as high as Windows Server 2008.

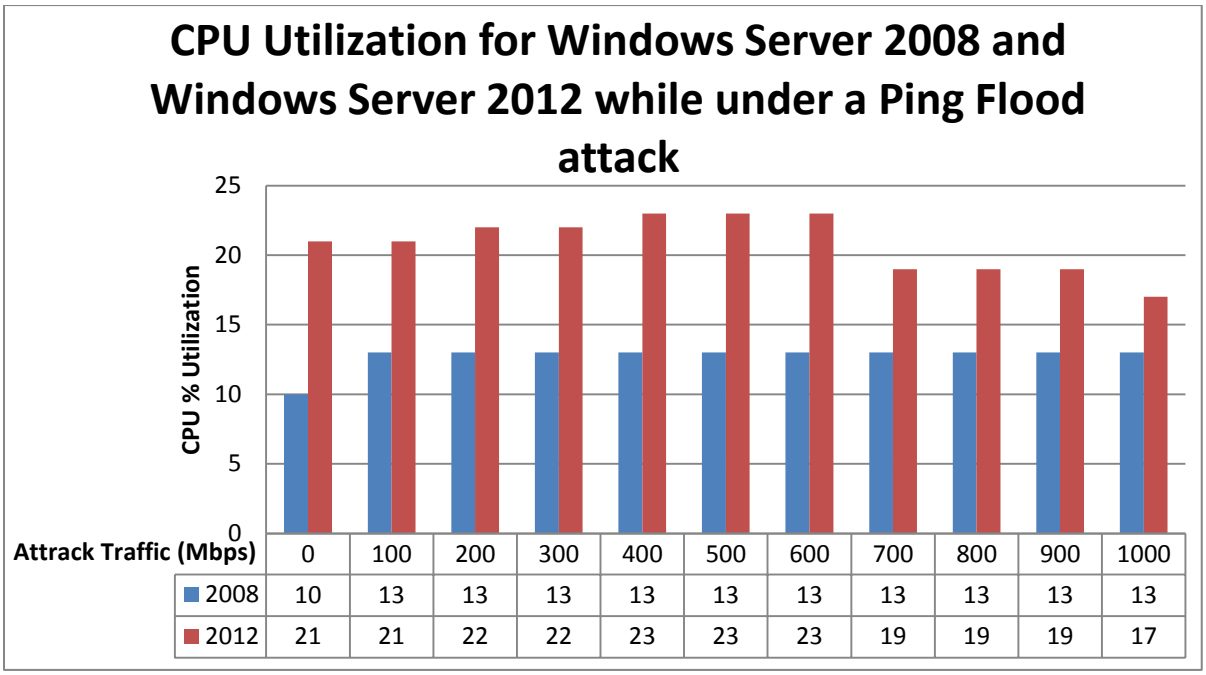
Table 3:

Echo Request and Echo Replies per second for Windows Server 2008 and Windows Server 2012 under a Ping Flood attack

	Windows Server 2008		Windows Server 2012	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec
0	0	0	0	0
100	143317	0	148660	0
200	233694	0	297063	0
300	207966	0	444893	0
400	207449	0	590926	0
500	208219	0	736678	0
600	209313	0	847571	0
700	209486	0	828369	0
800	209486	0	828791	0
900	209963	0	828514	0
1000	209958	0	828517	0

From Figure 22, we can see that Windows Server 2008 reached a maximum CPU consumption of 13 % at 100 Mbps while Windows Server 2012 had a maximum CPU consumption of 23 % at 400 Mbps. While only receiving legitimate traffic, 0 Mbps of attack traffic, the server while running Windows Server 2008 have a CPU consumption of 10 % and 21 % when running Windows Server 2012. As we introduced 100 Mbps of attack traffic, we observed that Windows Server 2008 CPU consumption increased to 13 % and remained the same throughout the other trials. When we introduced Windows Server 2012 to the attack traffic, we observed that at 100 Mbps the CPU consumption did not increased. In fact we observed that from 200 to 600 Mbps, the increase to the CPU was only 2 %. But if we look at 700 Mbps, we

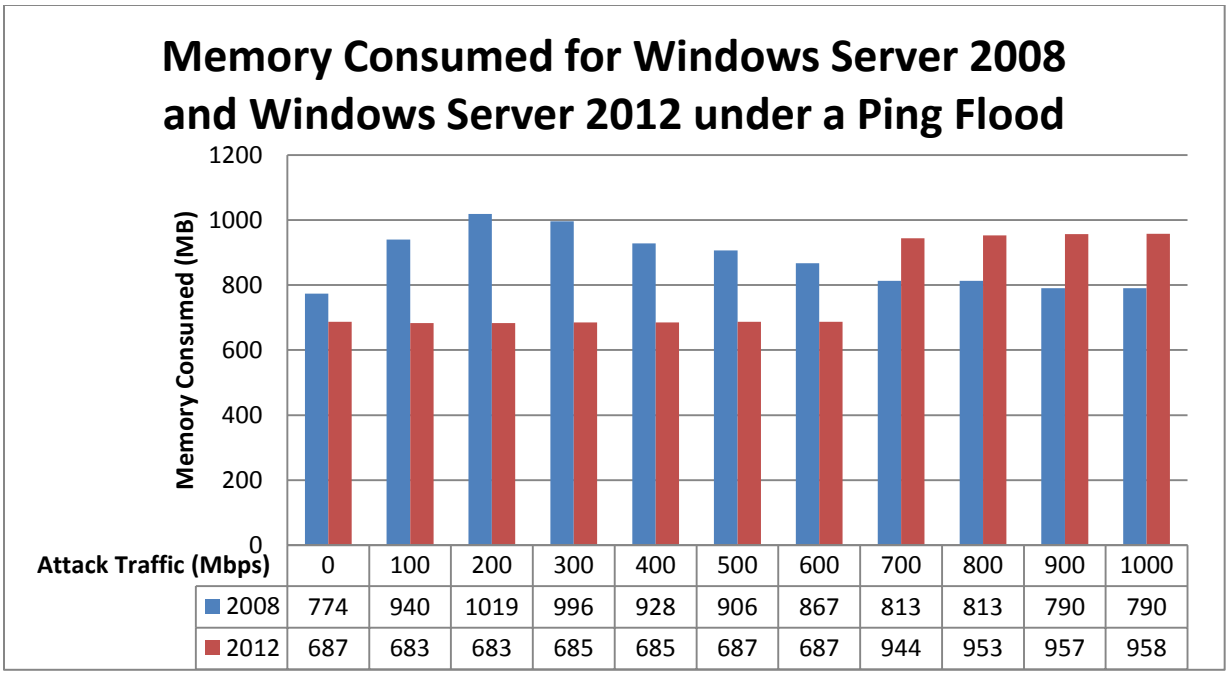
saw a drop in the CPU consumption when the server was running 2012. We believe that this was a side effect from the connection rate which experienced a significant decreased at 700 Mbps and can be seen in Figure 20.



Attrack Traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	10	13	13	13	13	13	13	13	13	13	13
2012	21	21	22	22	23	23	23	19	19	19	17

Figure 22 - CPU Exhaustion for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Ping Flood Attack

Although Windows Server 2008 was limiting the number of echo request messages that it was receiving to about 233,000 per second, which seems to be more efficient than Windows Server 2012. This did not really contribute to a better overall performance. Since a server must be able to successfully handle numerous HTTP connections request per second and Windows Server 2008 failed at 200 Mbps. We concluded that Windows Server 2012 had better performance while under the Ping Flood attack.



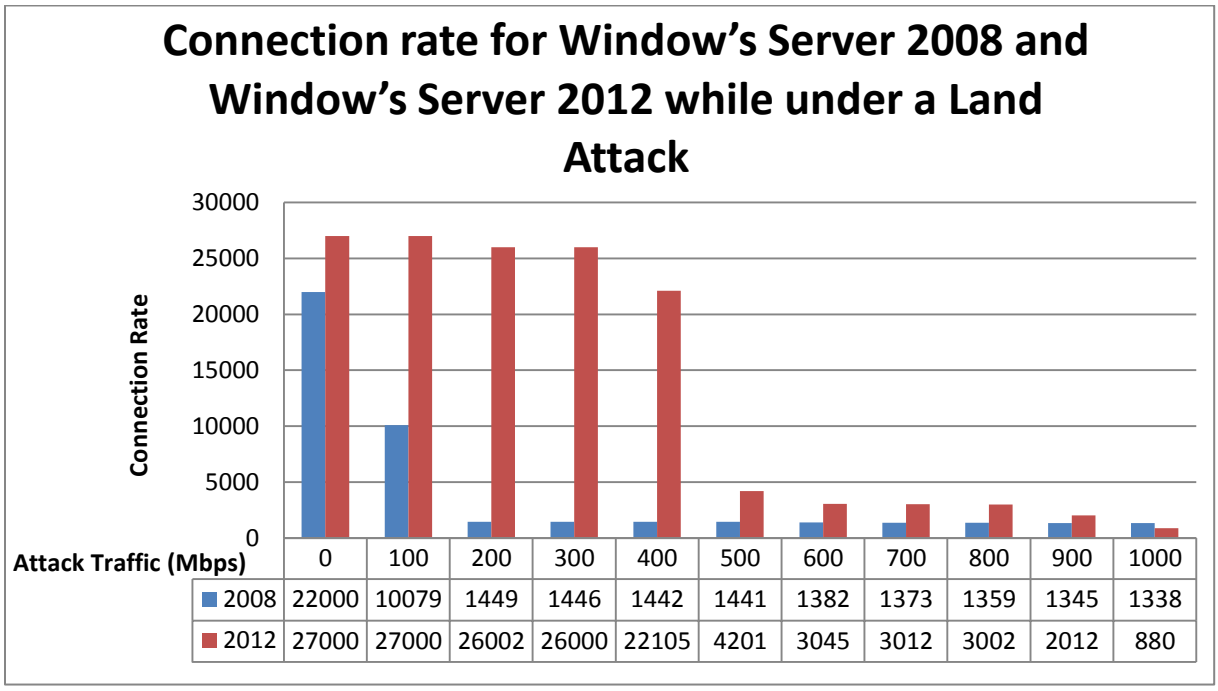
Attack Traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	774	940	1019	996	928	906	867	813	813	790	790
2012	687	683	683	685	685	687	687	944	953	957	958

Figure 23 - Memory Consumed for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Ping Flood Attack

The amount of memory that was being consumed by this attack was surprising. At start up, the amount of memory that was being consumed by Windows Server 2008 was 774 Mbytes and 687 Mbytes for Windows Server 2012. As the attack traffic was introduced, we observed that Windows Server 2008 had an increase in memory consumption from 100 to 200 Mbps. At 200 Mbps the consumption of memory was 1019 Mbytes. From this point on, we observed that the memory that was being consumed started to decrease. As for Windows Server 2012, the consumption of memory was around 685 Mbytes up until 600 Mbps. Then, an increase of 257 Mbytes occurred. This coincides with the decrease in the connection rate, the CPU consumption, and the threshold limit for the number of echo request being received at 700 Mbps.

4.3.2 ICMP Land Flood Attack

When introduced to the Land Attack, we found that when the iMac Pro Server was running Windows Server 2008 it was able to successfully handle 22,000 connections per second under normal operating conditions while Windows Server 2012 was able to handle 27,000 connections per second (see Figure 24). Once the ICMP Land Attack traffic was introduced, we can see from Figure 24 that a significant drop in the connection rate for Windows Server 2008 occurred. We saw that the connection rate drop from 22,000 to about 10,000 connections per second.



Attack Traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	22000	10079	1449	1446	1442	1441	1382	1373	1359	1345	1338
2012	27000	27000	26002	26000	22105	4201	3045	3012	3002	2012	880

Figure 24 - Connection rate for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Land Attack Flood

This was slightly more when compared to the Ping Flood attack. As the Land Attack traffic was increased, the connection rate continued to decrease until 200 Mbps of attack traffic.

At this point, the number of connections per second was about 1449 connections per second. We observed that the connection rate was slightly decreasing from 200 Mbps to 1000 Mbps, where it reached 1338 connections per second. This was not the case for Windows Server 2012, we observed that it was not until the attack traffic of 200 Mbps that the server was affected and the connection rate decreased from 27,000 to 26,000 connections per second. We then observed another decrease at 400 Mbps of attack traffic. One thing to note is that the server was still able to handle 22,000 successful connections per second. However, at the attack traffic of 500 Mbps and beyond, we observed that the connection rate was less than 5,000 connections per second. Despite this behavior, it was not until 1000 Mbps that Windows Server 2012 was out performed by Windows Server 2008.

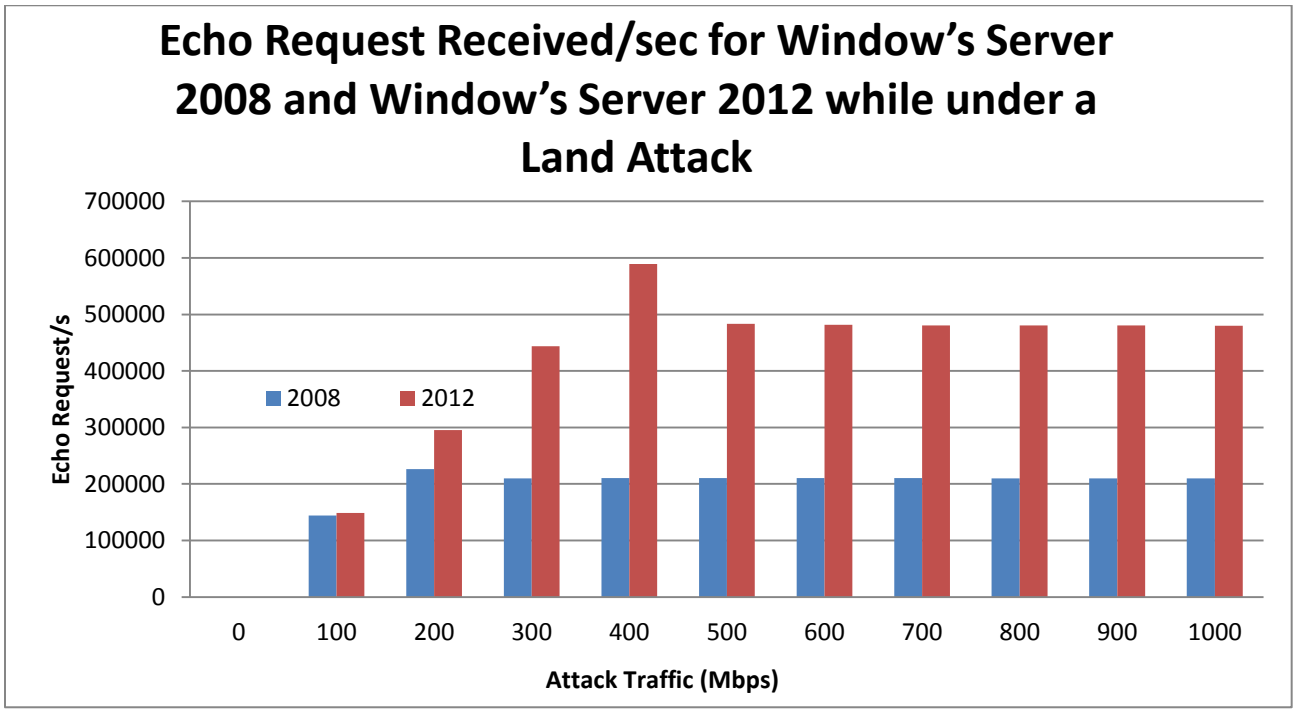


Figure 25 - Received Echo Request per second for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Land Attack Flood

Table 4:

Echo Request and Echo Replies per second for Windows Server 2008 and Windows Server 2012 under a Land Attack Flood				
	Windows Server 2008		Windows Server 2012	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec
0	0	0	0	0
100	144168	0	148569	0
200	226090	0	295360	0
300	210122	0	443947	0
400	210224	0	589107	0
500	210568	0	483188	0
600	210512	0	481682	0
700	210531	0	480532	0
800	209895	0	480612	0
900	209875	0	480562	0
1000	209954	0	480132	0

Just like in the previous experiment, we also discovered that both operating systems were limiting the number of Echo Request and Echo Reply messages that were being received and sent (see Figure 25 and Table 4). According to the data that was collected, we can see that Windows Server 2008 was limiting the number of echo request packets that it was receiving to about 226,000 per second. At 100 Mbps, the number of echo request messages that were received was about 144,000 per second and at 200 Mbps the limiting threshold was reached. With the increase in attack traffic, the received packets were about 210,000 per second for the rest of the trials. We can see that Windows Server 2012 reached its threshold for the maximum number of echo request packets per second that it would receive at 400 Mbps and reached 589,000 messages per second. Just like in every other case, once the threshold was reached the number of packets received decreased until the attack traffic reached 1000 Mbps. In the case for the Land Attack, this number was about 210,000 packets per second.

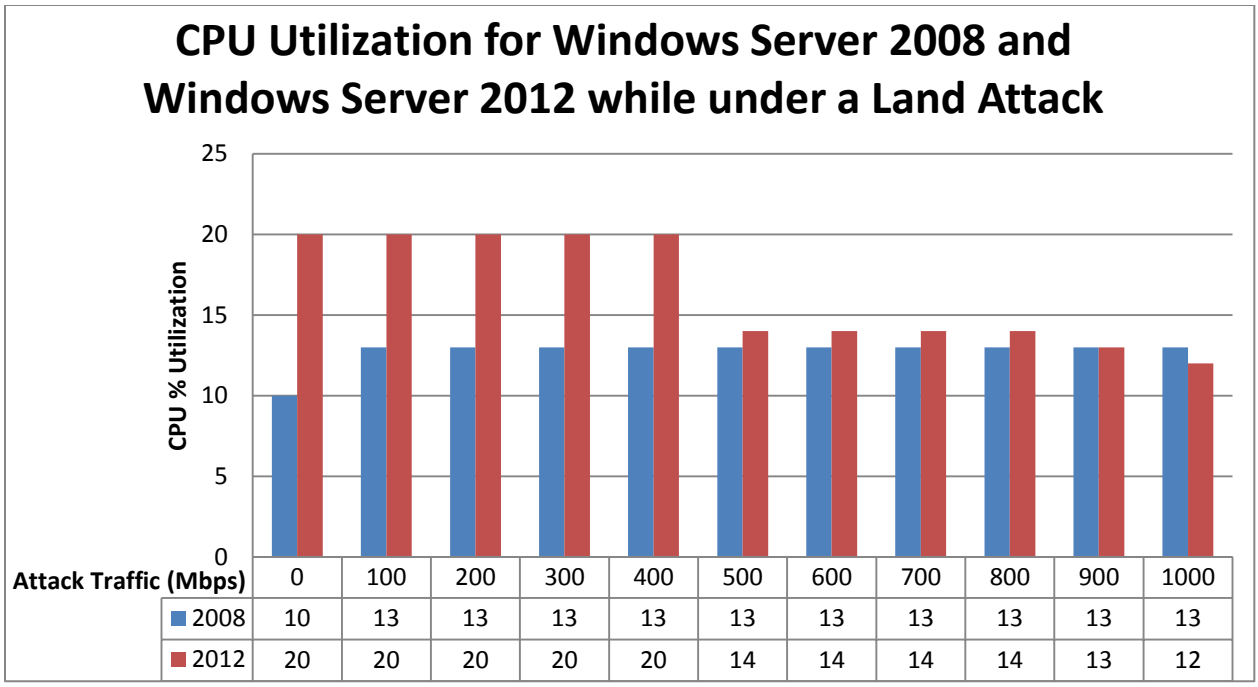


Figure 26 - CPU utilization for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Land Attack Flood

From Figure 26, we can see that Windows Server 2008 reached a maximum CPU consumption of 13% at the relatively low attack speed of 100 Mbps and stayed the same throughout the whole attack range. One thing that was interesting was that Windows Server 2012 had a baseline CPU consumption of 20 % and stayed the same when the attack traffic was introduced up until 400 Mbps of attack traffic. Once the attack reached 500 Mbps, we observed that the CPU utilization decreased to 14 %. This was very interesting and surprising. We contribute this behavior to the effects of the limiting mechanism previously described. If we look at Figures 24 and 25, we can see that both the connection rate and the number of echo received packets decrease after 400 Mbps. We believe that with the decrease in the number of successful connection coupled with the threshold limit, the utilization of the CPU was able to be reduced.

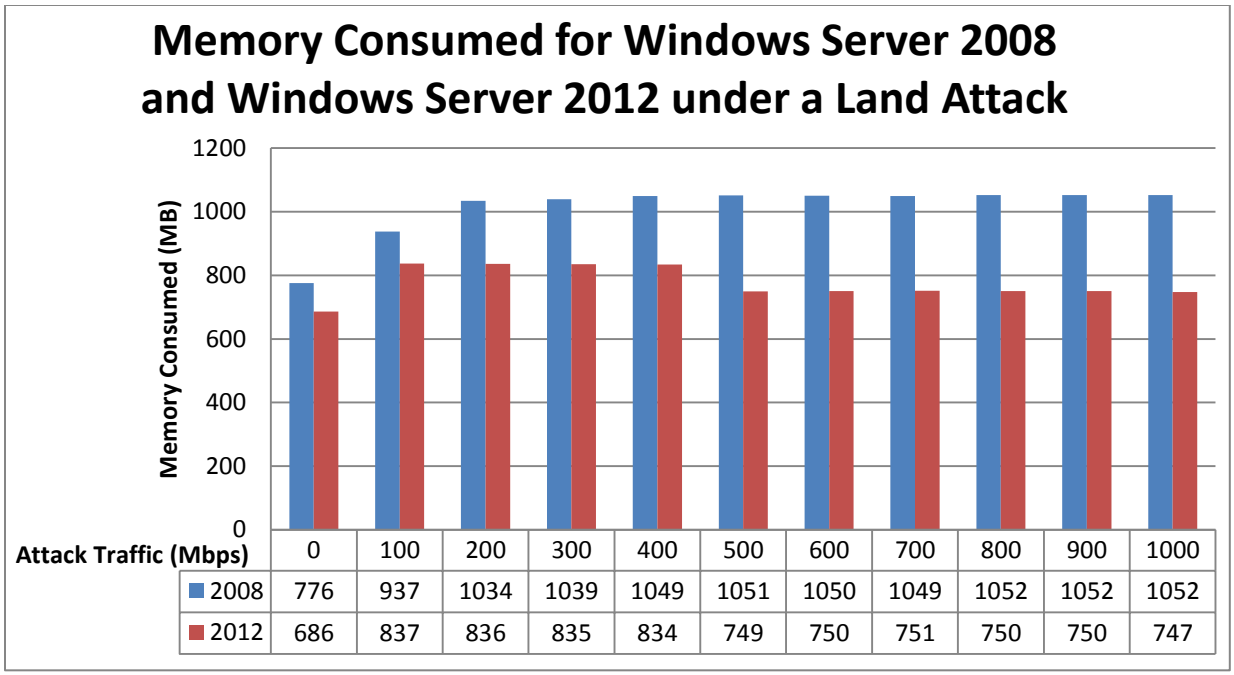


Figure 27 - Memory Consumed for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under an ICMP Land Attack Flood

At start up, the amount of memory that was being consumed by Windows Server 2008 was 776 Mbytes and 686 Mbytes for Windows Server 2012. As the attack traffic was introduced, we observed that Windows Server 2008 had an increase in memory consumption from 100 to 200 Mbps. At 200 Mbps the consumption of memory was 1034 Mbytes. Unlike in the Ping Flood attack, from this point on we observed the amount of memory that was being consumed stayed the same throughout the rest of the trails. As for Windows Server 2012, once the attack was introduced we immediately saw a jump in consumption of about 150 Mbytes. We then observed another decrease from 400 to 500 Mbps of about 100 Mbytes. This again coincides with the decrease in the connection rate, the CPU consumption, and the threshold limit for the number of echo request being received at 500 Mbps.

Even though Windows Server 2008 was limiting the number of echo request that it was receiving to about 220,000 per second, which seems to be more efficient then Windows Server 2012, this did not really contribute to a better overall performance. This has to do with the fact that a server must be able to successfully handle numerous connections request per second and Windows Server 2008 failed at 200 Mbps while, Windows Server 2012 had a more successful connection rate then Windows Server 2008 until 1000 Mbps.

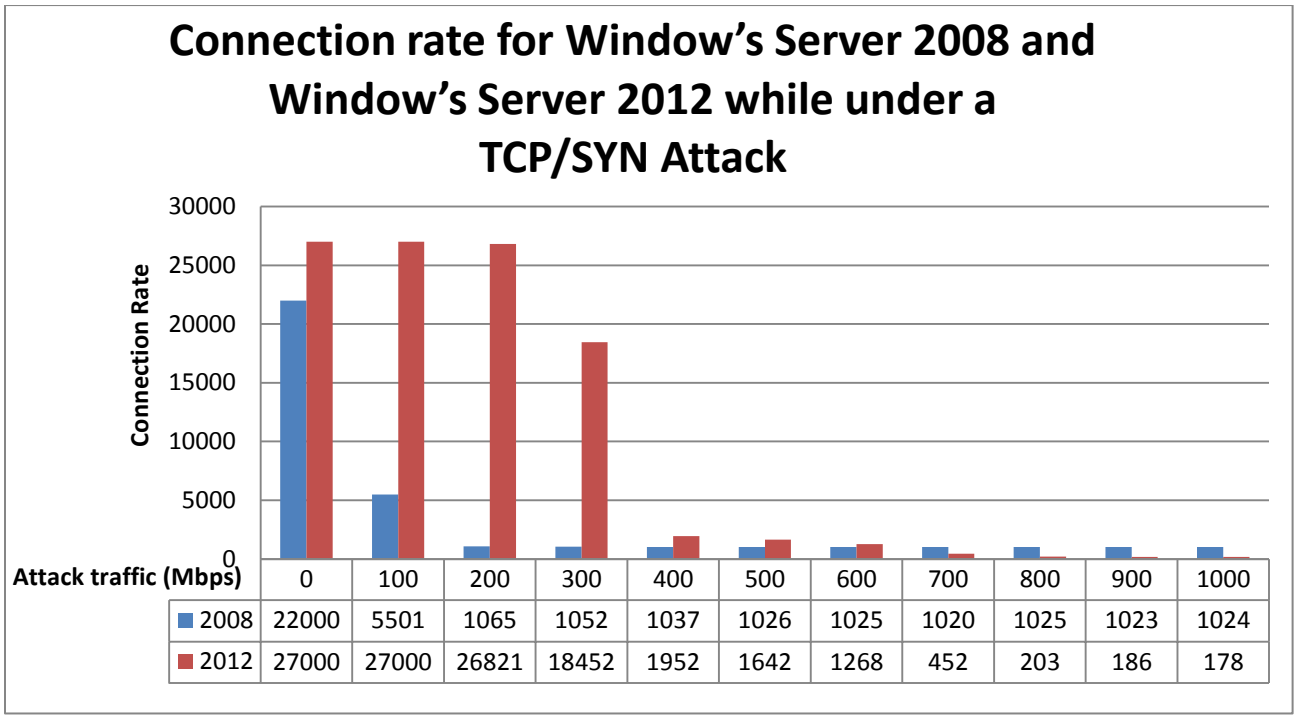
We can think about this as a tradeoff situation. For instance, Windows Server 2008 chose a lower threshold limit for the number of echo request messages that it would allow the operating system to receive. By having a lower threshold limit than Windows Server 2012, it seemed to have led to a lower connection rate and CPU consumption. On the contrary, Windows Server 2012 had a higher threshold limit and CPU consumption, but was able to sustain the connection rate for a lot longer. So the tradeoff seems to be with the limiting threshold, connection rate, and CPU consumption.

4.3.3 TCP-SYN Flood Attack

Closed Port

While under no attack traffic, the baseline for the connection rate for Windows Server 2008 was 22,000 connections per second and 27,000 connections per second for Windows Server 2012 and can be seen in Figure 28. As we introduced the TCP/SYN traffic, we observed the largest drop so far for Windows Server 2008. At 100 Mbps, the connection rate was 5,501 connections per second. At 200 Mbps, another drop occurred but not as drastic. After this drop, the connection rate leveled off and Windows Server 2008 was able to handle about 1,040

connections per second. When the TCP/SYN was introduced to Windows Server 2012, we did not see an impact to the connection rate. At 100 Mbps, the connection was the same as the baseline case of 27,000 connections per second. At 200 Mbps, the connection rate was slightly affected and was only able to establish about 26,000 connections per second. However, between the attack traffic of 400 and 600 Mbps, we observed that the connection rate was less than 2,000 connections per second but still higher than Windows Server 2008. Then at 700 Mbps and beyond, the connection rate dropped below Windows Server 2008 and was less than 500 connections per second.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	22000	5501	1065	1052	1037	1026	1025	1020	1025	1023	1024
2012	27000	27000	26821	18452	1952	1642	1268	452	203	186	178

Figure 28 - Connection rate for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN Flood attack

From Figure 29, we can see that both Windows Server 2008 and Windows Server 2012 reached their maximum CPU consumption at the attack speed of 100 Mbps. Windows Server 2008 had a maximum CPU consumption of 14 % and 28 % for Windows Server 2012. Just like in the other attacks, once Windows Server 2008 reached its peak CPU consumption it stayed the same for the other trials. Between 100 and 300 Mbps, the CPU consumption for Windows Server 2012 decreased by one percent. At 400 Mbps and beyond, the utilization of the CPU decreased to 19 %.

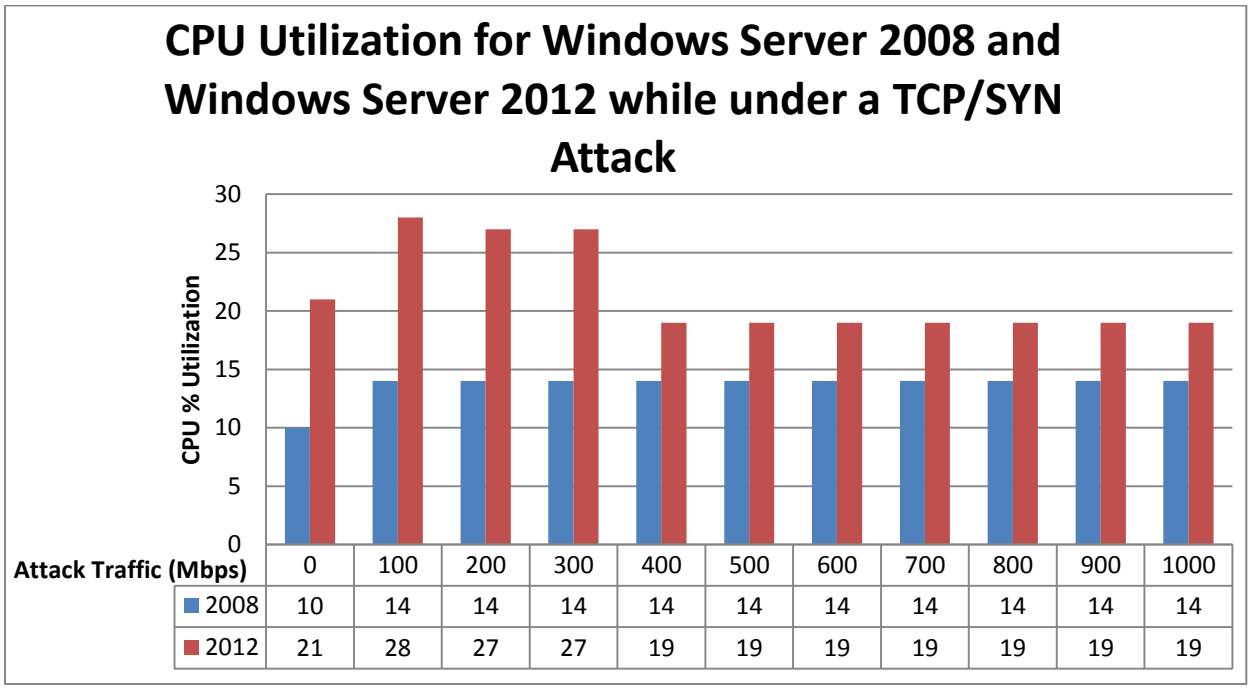
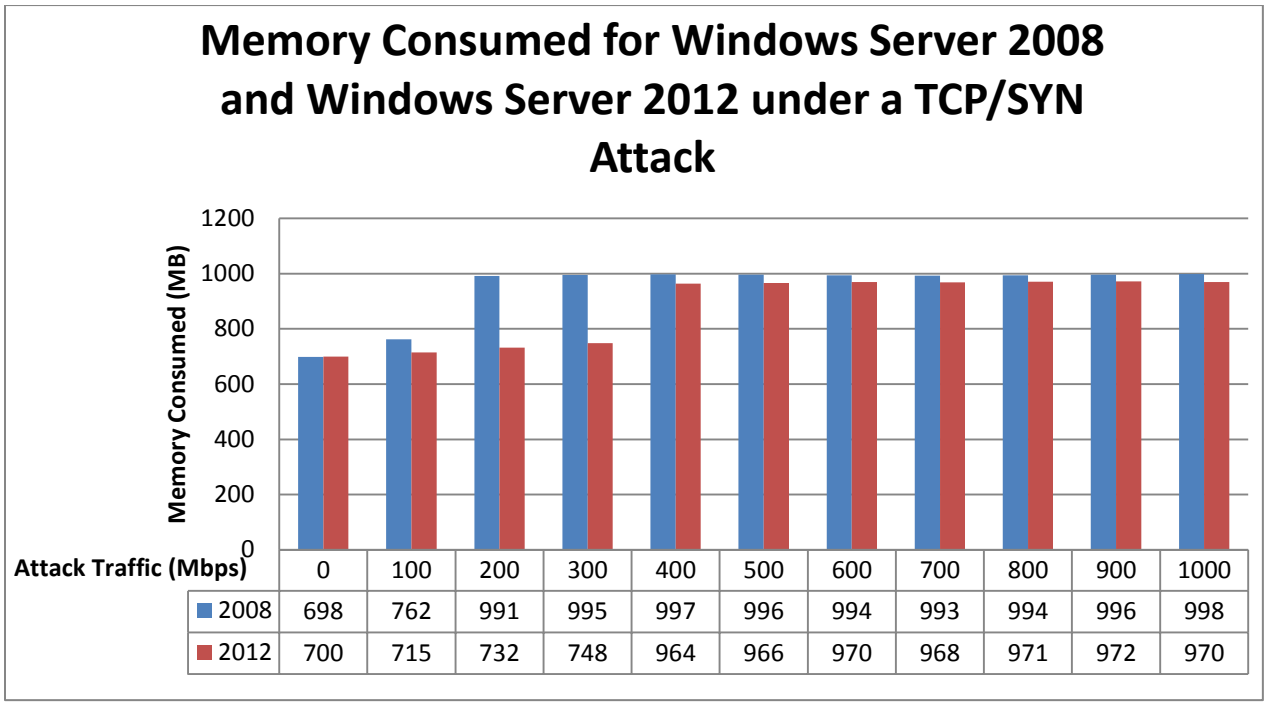


Figure 29 - CPU utilization for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN attack

At start up, the amount of memory that was being consumed by Windows Server 2008 was 700 Mbytes and 698 Mbytes for Windows Server 2012 and is shown in Figure 30. As the

attack traffic was introduced, we observed that Windows Server 2008 had an increase in memory consumption of 64 Bytes which resulted in a consumption of 762 Mbytes. At 200 Mbps the consumption of memory was 991 Mbytes, which was an increase of 229 Mbytes. After 200 Mbps, the consumption of memory stayed around 995 Mbytes for the remainder of the attack trials. As for Windows Server 2012, once the attack was introduced we saw a 15 Mbyte increase from 100 to 300 Mbps. Then at 400 Mbps we observed an increase of 216 Mbytes, which resulted in a consumption of 964 Mbytes of memory. Just like in Windows Server 2008, once the maximum amount memory was achieved it fluctuated slightly.

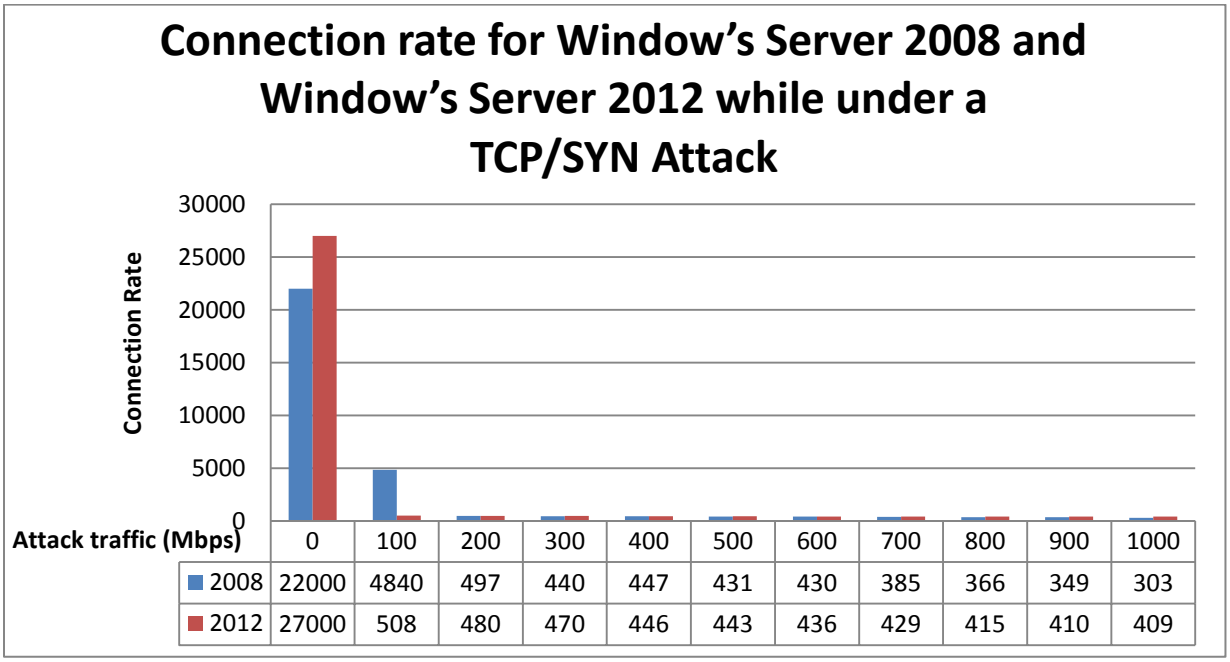


Attack Traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	698	762	991	995	997	996	994	993	994	996	998
2012	700	715	732	748	964	966	970	968	971	972	970

Figure 30 - Memory Consumed for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN attack

Open Port

While under no attack traffic, the baseline for the connection rate for Windows Server 2008 was 22,000 connections per second and 27,000 connections per second for Windows Server 2012 and can be seen in Figure 31. As we introduced the TCP/SYN traffic to an open port, we observed that at 100 Mbps the connection rate dropped to 4,480 connections per second. At 200 Mbps, another drop occurred which resulted in a connection rate of 497. After this, the connection rate continued to decrease as the attack speed was increased. When the TCP/SYN attack was introduced to Windows Server 2012, we did not expect what we saw. At 100 Mbps, Windows Server 2012 was only able to establish 508 connections per second. As the attack continued, the connection rate eventually stayed at 409.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	22000	4840	497	440	447	431	430	385	366	349	303
2012	27000	508	480	470	446	443	436	429	415	410	409

Figure 31 - Connection rate for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN Flood attack

From Figure 32, we can see that both Windows Server 2008 and Windows Server 2012 reached their maximum CPU consumption at the attack speed of 100 Mbps. Windows Server 2008 had a maximum CPU consumption of 14 % and 23 % for Windows Server 2012. Just like in the other attacks, once Windows Server 2008 reached its peak CPU consumption it stayed the same for the other trials. Unlike in the closed port attack, Windows Server 2012 had the same CPU consumption throughout the whole attack range.

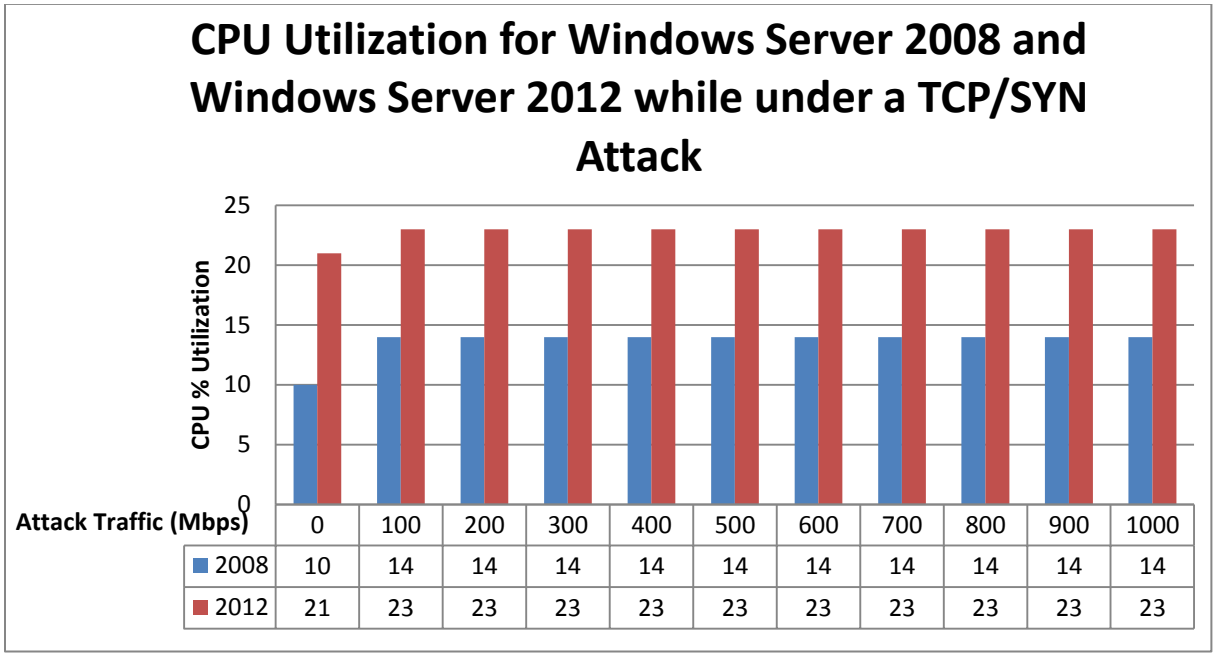
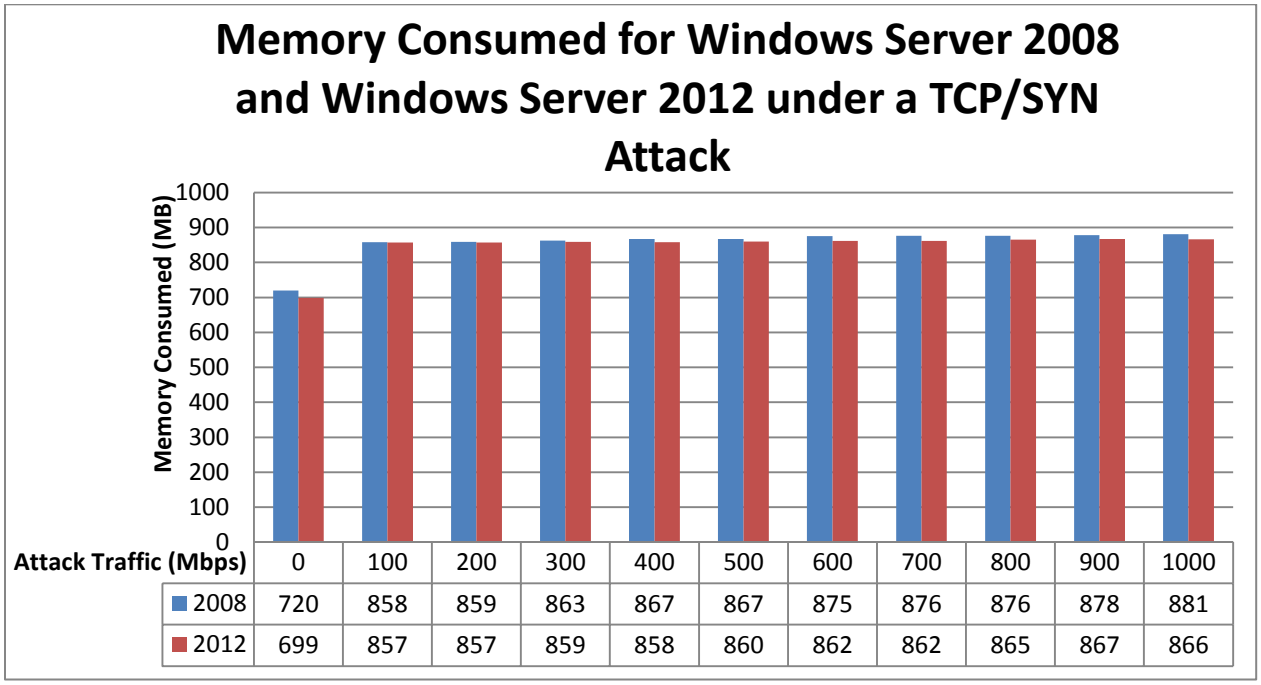


Figure 32 - CPU utilization for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN Flood attack

At start up, the amount of memory that was being consumed by Windows Server 2008 was 720 Mbytes and 699 Mbytes for Windows Server 2012 and is shown in Figure 33. As the attack traffic was introduced, we observed that Windows Server 2008 had an increase in memory consumption of 138 Mbytes which resulted in a consumption of 858 Mbytes. From the attack

range of 100 to 1000 Mbps, the memory that was consumed went from 858 to 881 Mbytes. Just like with Windows Server 2008, once the attack was introduced to Windows Server 2012 we saw a 158 Mbyte increase which resulted in a consumption of 857 Mbytes. Just like in Windows Server 2008, once the maximum amount of memory that was being consumed was achieved it increased slightly.



Attack Traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2008	720	858	859	863	867	867	875	876	876	878	881
2012	699	857	857	859	858	860	862	862	865	867	866

Figure 33 - Memory Consumed for Windows Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 under a TCP/SYN Flood attack

4.4 Chapter Summary

What we set out to do was to get an understanding if both Window's Server Enterprise 2008 R2 and Windows Server Enterprise 2012 R2 provided the same protection from the TCP/SYN, Ping, and Land Attack Floods. If they did, then we could say that the built-in security

of Windows Server 2012 was not improved and there is not really a reason to upgrade to the latest operating from Microsoft.

While under the ICMP Ping Flood attack, Windows Server Enterprise 2008 R2 had a maximum CPU consumption of 13% while Windows Server Enterprise 2012 R2 had a maximum CPU consumption of 23%. However, the more valuable parameter to measure is the number of successful connections per second that each operating system was able to handle. In this case, Windows Server 2012 was able to handle the attack up to the speed of 600 Mbps while Windows Server 2008 was only able to handle the attack up to 200 Mbps. While under the ICMP Land Attack, Windows Server Enterprise 2008 R2 had a maximum CPU consumption of 13% while Windows Server 2012 had a maximum CPU consumption of 20%. When we focused on the connection rate, Windows Server 2012 was able to handle the attack up to the speed of 400 Mbps while Windows Server 2008 was only able to handle the attack up to 200 Mbps.

While under the closed TCP/SYN attack, Windows Server Enterprise 2008 R2 had a maximum CPU consumption of 14% while Windows Server 2012 had a maximum CPU consumption of 28%. When we focused on the connection rate, Windows Server 2012 was able to handle the attack up to the speed of 300 Mbps while Windows Server 2008 was not able to handle the attack at all. While under the closed TCP/SYN attack, both operating systems were not able to handle this attack. We should mention that Windows Server Enterprise 2008 R2 was able to have a higher connection rate at 100 Mbps, but then dropped below Window Server Enterprise 2012 R2 after 100 Mbps. This information shows that Windows Server 2012 was able to prolong the effect of the DDoS attack, but was still not able to mitigate the attack.

According to our test results, we found that both operating systems were not able to handle the attacks passed a certain attack speed. Once this attack speed was passed, the

webservers were not able to handle a significant amount of connections per second making them inefficient and a waste of money and resources.

CHAPTER V

COMPARATIVE EVALUATION OF APPLE'S OS X LION SERVER, MICOROSOFT'S WINDOWS 2012 ENTERPRISE R2, AND UBUNTU'S 12.04 LTS "PRECISE PANGOLIN"

The servers on today's market are design to be stable yet dynamic machines, but with several operating systems on the market it becomes a difficult decision to choose one. Currently, the two most popular server operating systems on the market are Linux and Microsoft [68]. Now with the introduction of an Apple operating system into the market, this is a new and great opportunity to compare its performance with Linux and Microsoft. Linux and Microsoft have established a reputation that has made them extremely popular, so Apple has an uphill battle to establish a dominant presence in this market [68].

Internet Information Services (IIS) is a web server software that has been developed and used by Microsoft for many years and is considered to be the most reliable HTTP application since 2011 [63]. IIS is currently being used by approximately 13.2 % of the websites in the world according to a survey by w3techs.com [65]. Apache HTTP Server is the world's most widely used web server software and is being used by approximately 58.3 % of the websites in the world [69]. Since Apple's OS X Server application is new to the web server market, it currently holds less than .1 % of the market share.

Apple has the reputation for being the leader in innovation when it comes to hardware design and simplistic user-interfaces. With the introduction of OS X Server, Apple has designed a user friendly web server application that has bundled the services and security that are crucial in today's computing world [70]. So this becomes a great opportunity to compare the built-in security that is provided by these three operating systems.

5.1 Experimental Setup

In the controlled environment of the Network Research Lab (NRL) at The University of Texas-Pan American, the performance of Microsoft's Windows Server Enterprise 2012 R2 was evaluated and compared to Canonicals Ubuntu "Precise Pangolin" and Apples OS X "Lion Server". By using the setup guide provided by each company, we are able to install IIS-8 on Windows Server Enterprise 2012 R2 [66], OS X Server on Lion Server [71], and Apache on Ubuntu [72] [73].

The victim server platform was an Apple iMac Pro Server, just like in chapter 4, which was equipped with an Intel Xeon 2.8-GHz quad-core processor, 12 GBytes of RAM, and a Broadcom NetXtreme Gigabit Ethernet adapter [67]. Unlike in chapter 4, the Apple iMac Pro Server will be running the operating systems mentioned above. For each attack, we will first establish a baseline for the number of legitimate and successful TCP connections that each web server can handle. Then, while still sending the legitimate TCP traffic we will introduce the attack traffic to the system in increments of 100 Mbps for a duration of 10 minutes. By introducing the attack while the server is receiving legitimate traffic, we can study the impact that the denial of service attack has on the server's resources and legitimate traffic.

The experimental setup is exactly the same as in Chapter IV and can be seen in Figure 19. We will not discuss how the experimental setup is connected in detail, since it is described in Chapter IV and can be quickly referenced in section 4.1.1.. With the utilities provided by each operating system, we are able to collect the critical information that is required to compare the performances of each operating system while under the TCP and ICMP attacks. We will then use Microsoft's Excel to plot graphs from the information that was collected to aid us in our evaluation.

5.2 Parameters of Performance Evaluation

For this experiment, the parameters that will be used to evaluate the performance of each operating system are: the Connection Rate, the Processors utilization, the number of Echo request packets received per second, the number of Echo reply packets sent per second, and the amount of random access memory being consumed in Mbytes. These resources are being measured while the platform is being subjected to our denial of service attack traffic and are crucial for the evaluation of our system and are described below:

Connection Rate (TCP connections/second) – Whenever a webpage gets requested from a webserver, a connection will be established between the client and server by using the Transmission Control Protocol (TCP). If the connection is unsuccessful, then the webpage will not get fetched and an error will be displayed. Therefore, it is imperative to verify the effects that these attacks will have on the number of successful TCP connection.

CPU Utilization (Usage of CPU in %) – The utilization of the Central Processing Unit (CPU) is one of the most important parameters to keep track of in any normal or abnormal situation. The

utilization of the CPU informs the user of the amount of work that is being performed by the server at that instance.

Echo Request Received per second (Echo Request /Sec) – This parameter allows us to measure the number of echo request messages that were being received per second by the server. When a computer received an echo request message, it must send an echo reply message [38]. As the values of these parameters increase, we hope to identify the impact that it has on the other parameters that we will be evaluating.

Echo Reply Packets Sent per second (Echo Reply / Sec) – This parameter will allow us to measure the number of echo reply messages that are being sent per second by the server. These Echo Reply messages are being sent in response to the echo request messages that were received.

Random Access Memory (RAM) consumed (in Mbytes) – This parameter measures the amount of finite random access memory that is being consumed by the attack. If the random access memory becomes completely consumed, the computer will become slow and can become unstable.

5.3 Results and Discussions

In this section, we will be presenting the results that were collected from the experiments that were performed. It should be noted that the three operating systems were running on the same iMac Pro Server but with only one operating system running at a time.

For the final experiment, we will be testing the built-in security features that Windows Server 2012 R2, Lion Server, and Ubuntu's 12.04 LTS provided for the web server application

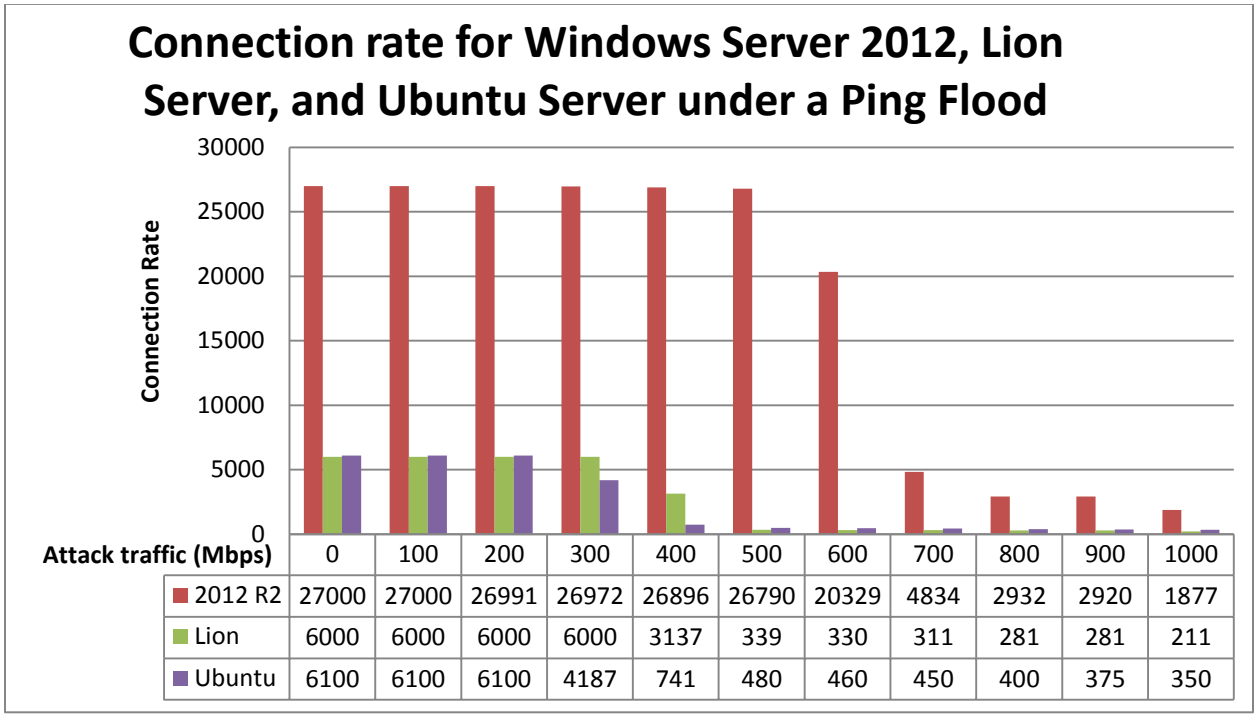
that is running in each one. We will begin by comparing the survivability against the ICMP Ping and Land Attack Floods. Followed by, comparing the survivability against the TCP/SYN attack. For each attack, we will begin by comparing the number of successful TCP connections per second that are established, the number of echo request and echo reply messages that each operating system receives and sends, the utilization of the processor, and the amount of memory being consumed.

5.3.1 ICMP Ping Flood Attack

We found that when the iMac Pro Server was running its native operating system, Lion Server was able to successfully handle 6,000 connections per second under normal operating conditions, while Ubuntu had a connection rate of 6,100. At first, we believed that these connection rates were the ideal values that would ensure optimized performance. However, when we installed Windows Server 2012 R2 we discovered that a connection rate of 27,000 connections per second was achieved. This left us puzzled and we hypothesized that Windows Server 2012 would have the worst overall performance.

Once we introduced 100 Mbps of attack traffic, we can see from Figure 34 that all three operating systems had the same connection rate as in the baseline trial. As we continued to increase the traffic, it was not until 300 Mbps that we saw a decrease in the connection rate for Ubuntu. The connect rate drop from 6,100 to 4,187 connections per second, that was almost a drop of 2000 connections per second. Then for the rest of the attack traffic range, Ubuntu's connection rate drop to below 742.The next operating system that experienced issues with the connection rate was Lion Server. Lion Server was able to keep the baseline connection of 6000

up to 300 Mbps of attack traffic. Then at 400 Mbps, we observed a drop of about 3,000 connections per second. From 500 Mbps and beyond, the connection rate decreased below 340. What was surprising was that Windows Server 2012 was able to sustain the baseline connection rate or near the baseline the longest. We observed that it was not until the attack traffic of 600 Mbps that the connection rate was greatly affected and drop to 20,300 connections per second. Then for the rest of the attack traffic range, the connection rate dropped to below 5000.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	27000	27000	26991	26972	26896	26790	20329	4834	2932	2920	1877
Lion	6000	6000	6000	6000	3137	339	330	311	281	281	211
Ubuntu	6100	6100	6100	4187	741	480	460	450	400	375	350

Figure 34 – Connection rate for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under a Ping Flood attack

Just like in the desktop operating systems, we observed that all three operating systems were limiting the number of Echo Request and Echo Reply messages that were being received and sent (Table 5 and Figure 35). According to the data that was collected, we can see that

Apple's Lion Server was limiting the number of echo request packets that it would receive to about 650,000 per second. Just like the Apple's desktop operating system, Lion, we observed that the maximum number of echo reply messages that it would send per second was 250. In Table 1 we show the actually number of echo request and echo reply messages being received and sent per second for all three operating systems.

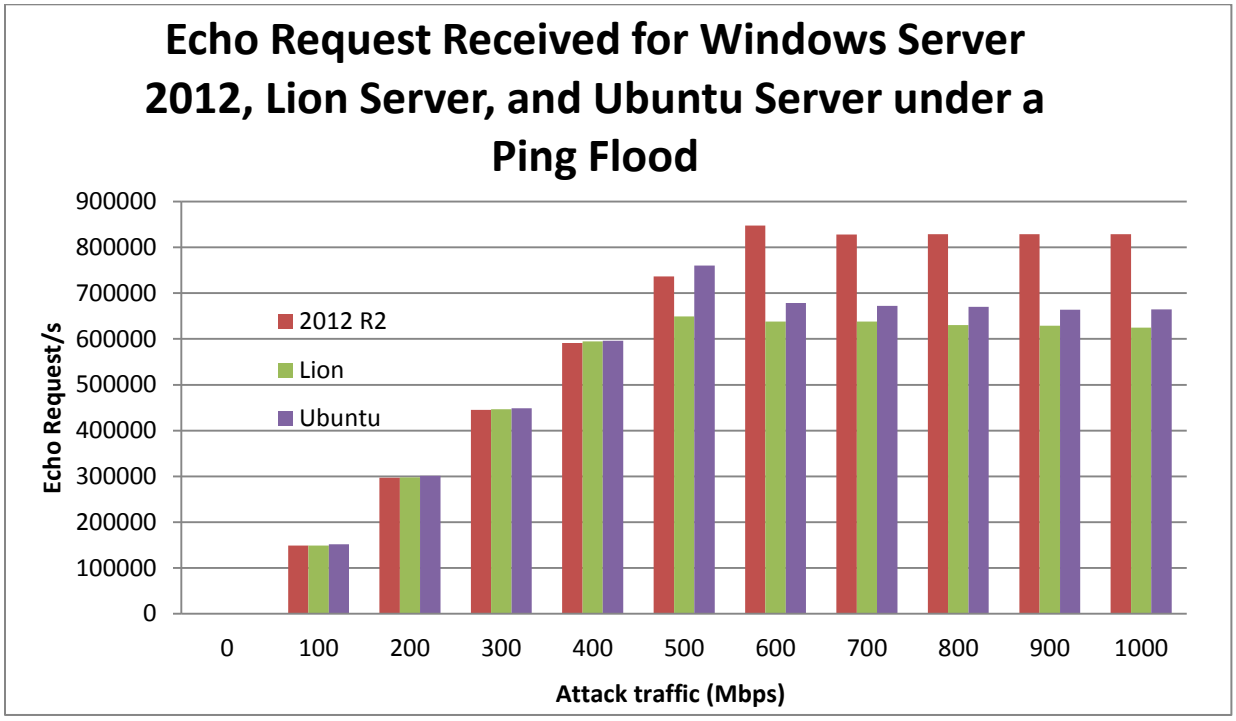


Figure 35 - Received Echo Request per second for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under a Ping Flood attack

According to the data, the threshold for the number of echo request messages that Windows Server 2012 R2 would receive was reached at 600 Mbps of attack traffic and was about 848,000 packets per second and can be seen in Figure 35 and Table 5. As we increased that speed of the attack traffic, the threshold seemed to decrease to about 830,000 for the rest of the

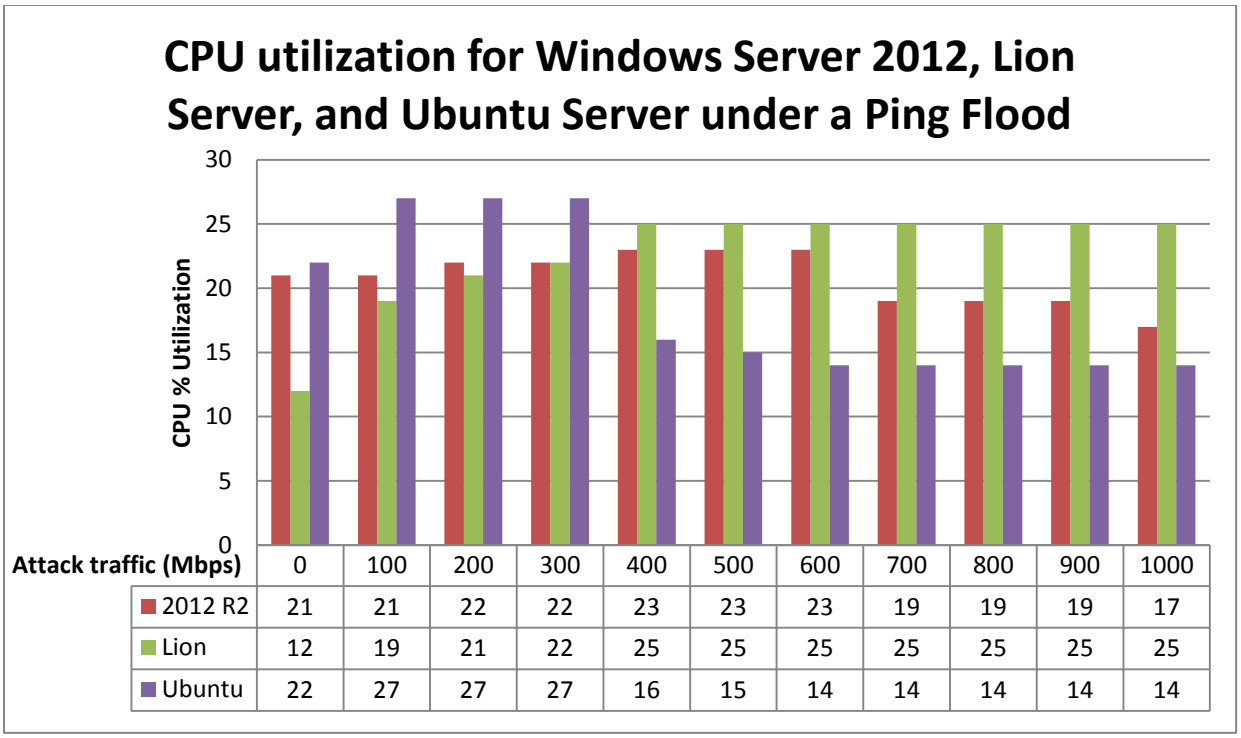
trials. When it came to the number of echo reply messages that were being sent, we detected that Windows Server 2012 was not replying to any of the echo request packets that were received.

As mentioned above, Ubuntu Server was also limiting the number of echo request and echo reply messages that it would receive and send. We observed that the threshold for the number of echo request packets that it was receiving was about 760,000 per second at the attack load speed of 500 Mbps and can be seen in Table 5. From the attack traffic speed of 600 Mbps and beyond, we observed the threshold drop to about 672,000. One thing that has change from the desktop version of Ubuntu was the number of echo reply messages that were being sent per second. For the whole attack range of 100 to 1000 Mbps, Ubuntu Server was only sending 70 echo reply packets per second.

Table 5:

Echo Request and Echo Reply per second for Lion Server, Windows Server Enterprise 2012 R2, and Ubuntu Server under a Ping Flood attack						
	Lion Server		Windows Server Enterprise 2012 R2		Ubuntu Server	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec
0	0	0	0	0	0	0
100	148906	250	148660	0	151928	70
200	297742	250	297063	0	301388	70
300	446360	250	444893	0	448496	70
400	594794	250	590926	0	596356	70
500	649424	250	736678	0	760666	70
600	638326	250	847571	0	678592	70
700	638291	250	828369	0	672539	70
800	630562	250	828791	0	670514	70
900	628990	250	828514	0	663854	70
1000	624739	250	828517	0	664845	70

When it came to the baseline reading for the utilization of the CPU, Lion Server had the lowest at 12 %. While, Windows Server 2012 R2 and Ubuntu had a baseline CPU utilization of 21 % and 22 % respectively. When Ubuntu was introduced to 100 Mbps of attack traffic, we observed that the CPU Utilization jumped from 22 to 27 %. From 100 Mbps to 300 Mbps, the utilization stayed at 27 %. Then, from 300 Mbps to 400 Mbps the utilization dropped to 16 %. This was interesting; since we also observed that the connection rate dropped from about 4200 to 740 at these speeds and shortly after the threshold was reached for the number of echo request packets received per second.

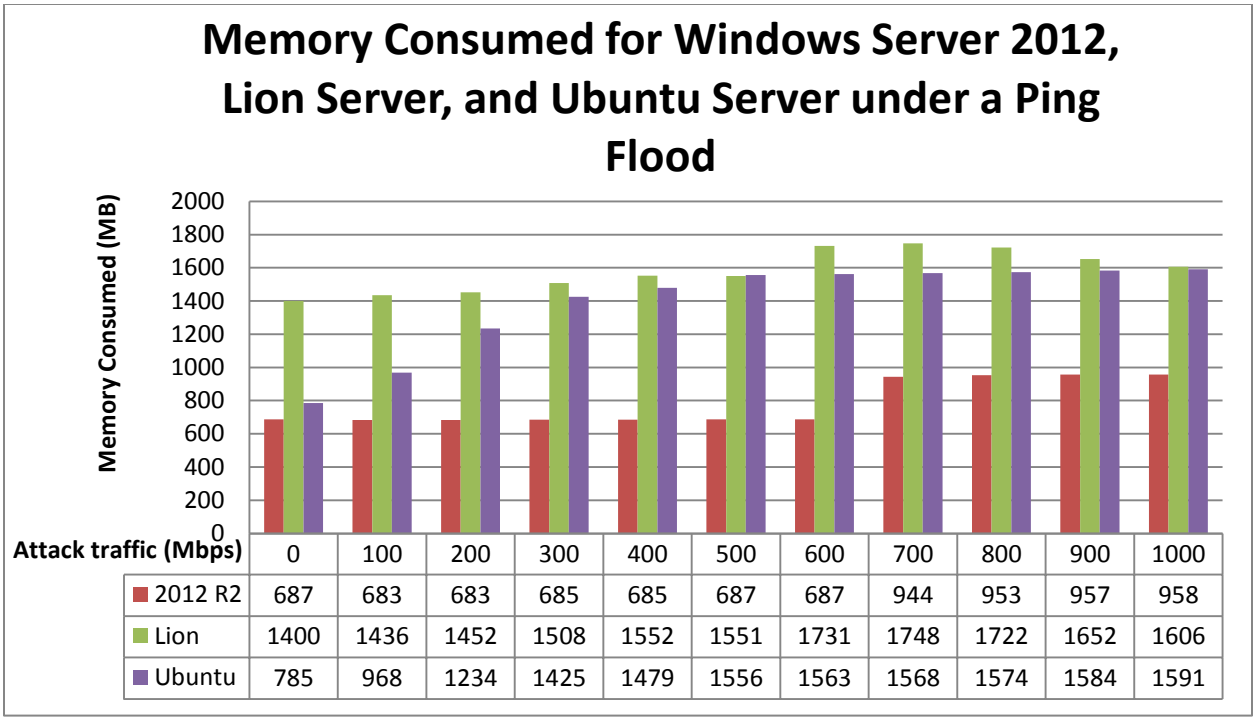


Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	21	21	22	22	23	23	23	19	19	19	17
Lion	12	19	21	22	25	25	25	25	25	25	25
Ubuntu	22	27	27	27	16	15	14	14	14	14	14

Figure 36 – CPU Utilization for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under a Ping Flood attack

When the iMac Pro Server was running its native operating system, Lion Server was able to prolong the effects to the CPU from the Ping Flood attack when compared to Ubuntu Server. We saw that the maximum CPU utilization was 25 % under the attack load of 400 Mbps of attack traffic. Once the maximum CPU utilization was reached, it stayed the same for the rest of the attack trails. When Windows Server 2012 R2 was introduced to the attack traffic, we did not really see a big impact to the CPU. From 200 Mbps to 300 Mbps, the utilization went from 21 % to 22 %. Then from 400 Mbps to 600 Mbps, the CPU utilization was 23 % and was at its maximum for this attack. For the remainder of the attack trials we observed that the CPU utilization dropped to 19 %.

Under normal operating conditions, 0 Mbps of attack traffic, we observed that the amount of RAM that was being consumed was higher for Lion Server than its competitors. At start up, the amount of memory that was being consumed by Lion Server was 1400 Mbytes and as a baseline was the highest. As we increased the attack traffic, we observed that at 600 Mbps the memory being consumed jumped from 1552 Mbytes to 1731 Mbytes. This almost coincides with the threshold for the number of echo request packets received per second, the maximum CPU consumption, and the connection rate decreasing. Ubuntu Server was consuming 785 Mbytes of memory at 0 Mbps of attack traffic. As the attack traffic was increased, we saw about a 210 Mbytes of memory being consumed from 100 Mbps to 300 Mbps. Overall, Ubuntu Server had a maximum consumption of memory of 1591 Mbytes. We calculated that about 800 Mbytes of memory were getting consumed. At start up, the amount of memory that was being consumed by Windows Server 2012 R2 was 687 Mbytes and as a baseline was the lowest. As we increased the attack traffic, we observed that Windows Server 2012 had a maximum memory consumption of 958 Mbytes and occurred at 1000 Mbps of attack traffic. Just like with Ubuntu Server and Lion Server, we discovered that Windows Server 2012 R2 had a significant jump in memory consumption when the threshold and connection rate were affected.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	687	683	683	685	685	687	687	944	953	957	958
Lion	1400	1436	1452	1508	1552	1551	1731	1748	1722	1652	1606
Ubuntu	785	968	1234	1425	1479	1556	1563	1568	1574	1584	1591

Figure 37 - Memory Consumed for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under a Ping Flood attack

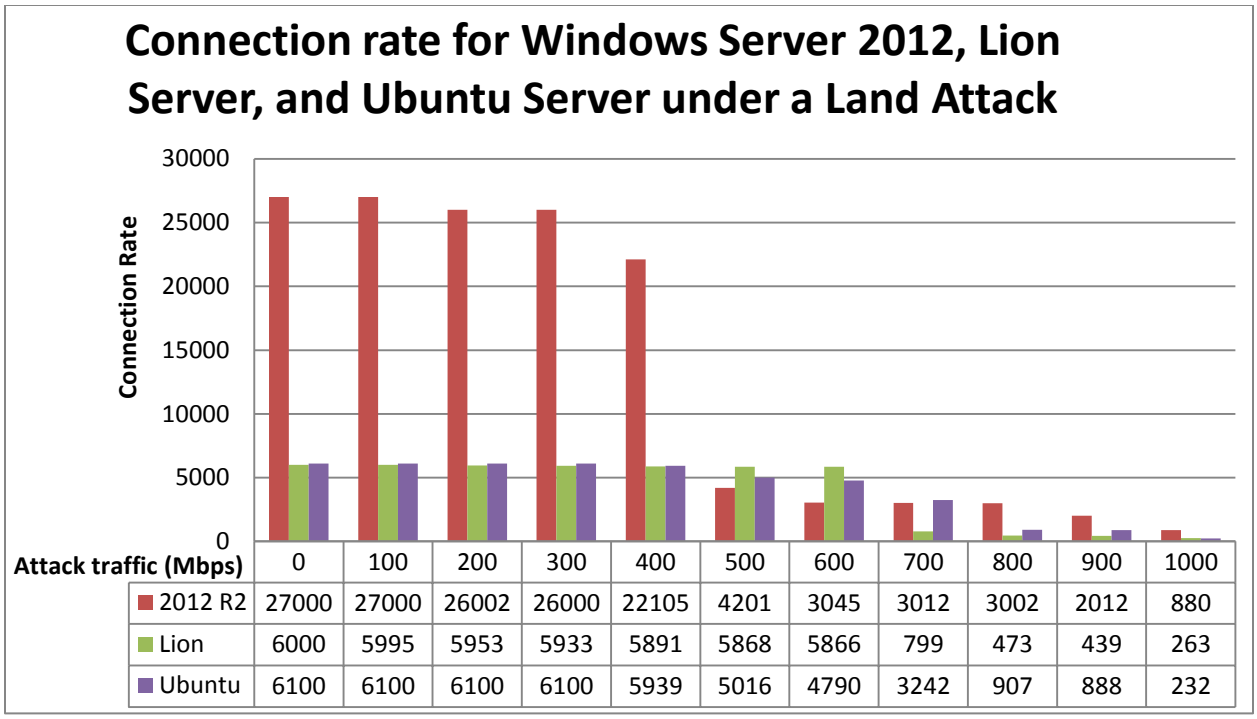
5.3.2 ICMP Land Flood Attack

When we introduced the iMac Pro Server to the traffic from the land attack, we expected that the operating systems would use most of the CPU and memory processing both the echo request and echo reply messages, as in the case with Apple's Leopard and Windows Vista [45].

We found that when the iMac Pro Server was running its native operating system, Lion Server was able to successfully handle 6,000 connections per second under normal operating conditions, while Ubuntu Server and Windows Server 2012 R2 had a connection rate of 6,100 and 27,000 respectively.

Once we introduced 100 Mbps of attack traffic, we can see from Figure 38 that all three operating systems had or were close to the same connection rate as in the baseline trial. As we continued to increase the traffic, it was not until 300 Mbps that we saw that Windows Server 2012 R2 had the biggest decrease of all three in the connection rate. At 400 Mbps, the connection rate for Windows Server 2012 R2 was about 22,000 and at 500 Mbps it dropped to 4,200 connections per second. As we will see later, this correlates to the threshold for the number of echo request packets received per second. When it came to Lion and Ubuntu, we discovered that they were not greatly affected until 700 Mbps and 800 Mbps respectively. For Lion, the connection rate at 600 Mbps was 5,866 and at 700 Mbps it dropped to 799. At 700 Mbps, the connection rate was 3,242 and at 800 Mbps it dropped to 907. Overall, the lowest connection rate for all three operating systems was at 1000 Mbps.

What we discovered was that all three operating systems had a drastic decrease to their connection rate from one sample point to the next. This sample point is different for all three operating systems and correlates to the threshold for the number of echo request packets received per second.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	27000	27000	26002	26000	22105	4201	3045	3012	3002	2012	880
Lion	6000	5995	5953	5933	5891	5868	5866	799	473	439	263
Ubuntu	6100	6100	6100	6100	5939	5016	4790	3242	907	888	232

Figure 38 - Connection rate for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under an ICMP Land Attack

According to the data that was collected for the Land Attack, we observed that Windows Server 2012 and Lion Server were exhibiting the same limiting strategy for the number of echo request and echo reply messages that they would receive and send while under a Ping Flood attack. However, the threshold values were different. This time around, we can see from Table 6 and Figure 39 that Lion Server was limiting the number of echo request messages that it would receive to about 1,049,000 packets per second at 700 Mbps of attack traffic. When compared to the threshold limit during the Ping Flood, we can see that during the Land Attack the threshold increased from 400,000 (Table 2) to 1,049,000 (Table 6). This was not expected and was really quite shocking. As the attack traffic increased, the threshold limit was able to limit this parameter which in turn caused a decrease in the connection rate. Regardless of the number of echo request

messages that Lion Server was receiving per second, we found that Lion Server was again sending only 250 echo reply messages per second.

By looking at Table 6 and Figure 39, we can see that Windows Server 2012 R2 was limiting the number of echo request packets that it was receiving to around 590,000 packets per second and occurred at 400 Mbps of attack traffic. However, unlike the Lion Server, Windows 7 was implemented to not respond to a Land Attack echo request packet. This was expected, since Windows Server 2012 R2 was not acknowledging any of the echo request packets that it received while under the Ping Flood attack. As we increased the attack traffic, we saw that the limiting threshold got lower as we increased the attack speed. At 1000 Mbps, the limiting threshold was at 480,000.

When we introduced the Land Attack flood traffic to Ubuntu Server, we discovered that Ubuntu was not receiving any of the echo request messages that were being sent to it even as we increased the speed to the attack traffic. Since Ubuntu was not receiving any packets, therefore it could not respond with an echo reply. So for this parameter, it was also zero throughout the whole attack range. We do not know why this is exactly happening, but we can conclude that Ubuntu is doing a little more processing than Windows Server 2012 and Lion Server when a packet is received. For instance, Ubuntu might be looking at the source IP address of each packet that was received so that it can mitigate the Land Attack. As we will see in Figure 40, this might be one of the reasons that Ubuntu has such a high CPU Utilization.

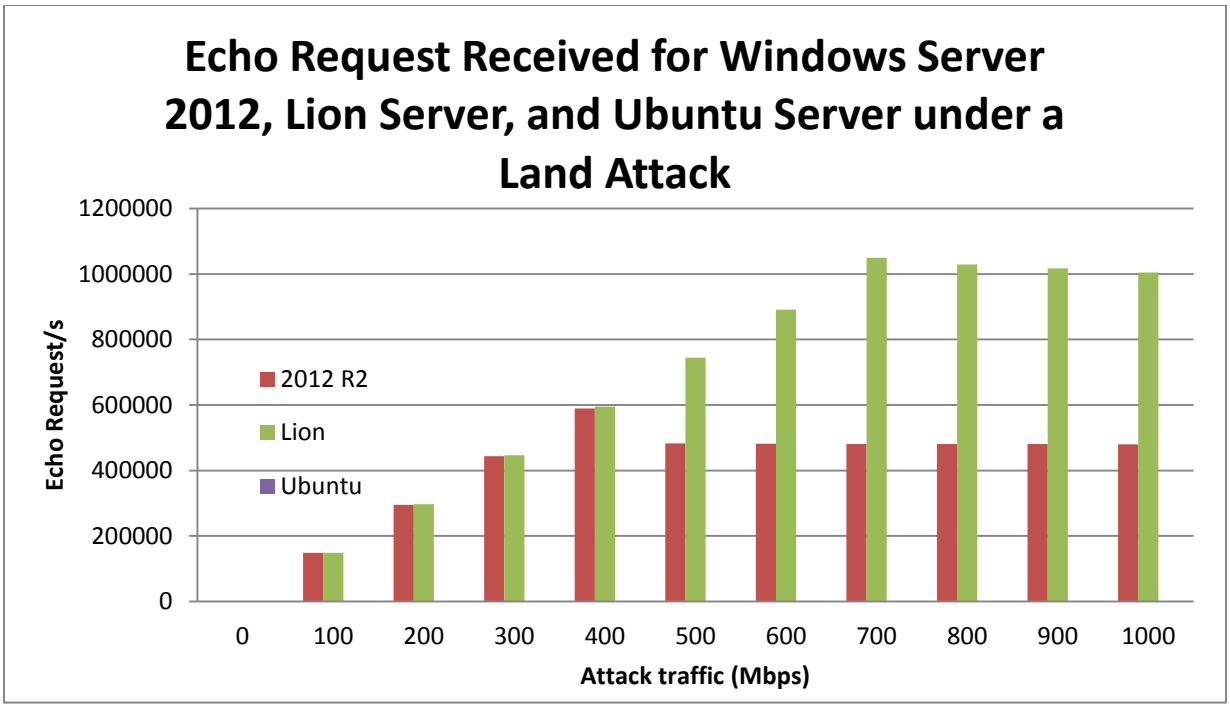


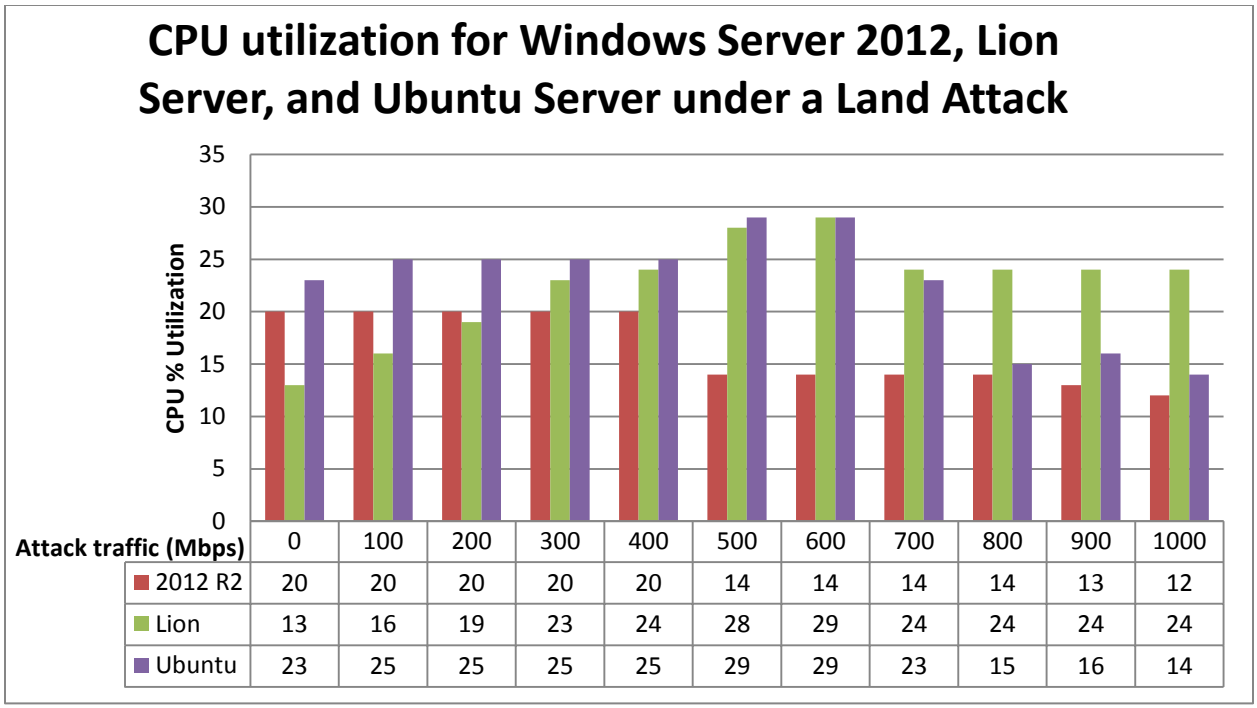
Figure 39 - Received Echo Request per second for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under an ICMP Land Attack

Table 6:

Echo Request and Echo Reply per second for Lion Server, Windows Server Enterprise 2012 R2, and Ubuntu Server under a Land Attak Flood						
	Lion Server		Windows Server Enterprise 2012 R2		Ubuntu Server	
Attack Load in Mbps	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec	Echo Request/sec	Echo Reply/sec
0	0	0	0	0	0	0
100	148792	250	148569	0	0	0
200	297582	250	295360	0	0	0
300	446220	250	443947	0	0	0
400	594979	250	589107	0	0	0
500	744578	250	483188	0	0	0
600	891264	250	481682	0	0	0
700	1048944	250	480532	0	0	0
800	1028809	250	480612	0	0	0
900	1016782	250	480562	0	0	0
1000	1004735	250	480132	0	0	0

From Figure 40, we can see that Lion Server reached a maximum CPU consumption of 29 % at the attack speed of 600 Mbps. While Windows Server 2012 R2 and Ubuntu Server had a maximum CPU consumption of 20 % and 29 %, respectively. When Lion Server was introduced to 100 Mbps of attack traffic, we saw that the CPU utilization went up to 16 %. As we continued to increase the speed of the traffic, the CPU utilization increased to 29 % and was reached at 600 Mbps. Then for the rest of the attack trials, the CPU utilization decreased to 24 %. As previously mentioned, in Figures 38 and 39 we can see that 600 Mbps is the sample point right before the connection rate for Lion Server was greatly affected and when the limiting threshold was reached. This behavior continues to strengthen our idea of a correlation between these parameters.

When Windows Server 2012 R2 was subjected to the Land Attack traffic, we observed that from 100 Mbps to 400 Mbps the CPU utilization remained the same as in the baseline reading of 20 %. From Figure 39, we can see that the limiting threshold for the number of echo request packets received was reached at 400 Mbps and coincides with the maximum CPU utilization. After this, the CPU utilization decreased for the rest of the trials. The CPU utilization for Ubuntu Server from 100 Mbps to 400 Mbps was 25 %, which was a 2 % increase from the baseline reading. Then, from 500 Mbps to 600 Mbps the maximum CPU utilization of 29 % was reached. Since Ubuntu was recording that it was not receiving any echo request packets, all we can say is that the decrease in the connection rate at 700 Mbps is the cause for the drop in the consumption of CPU.

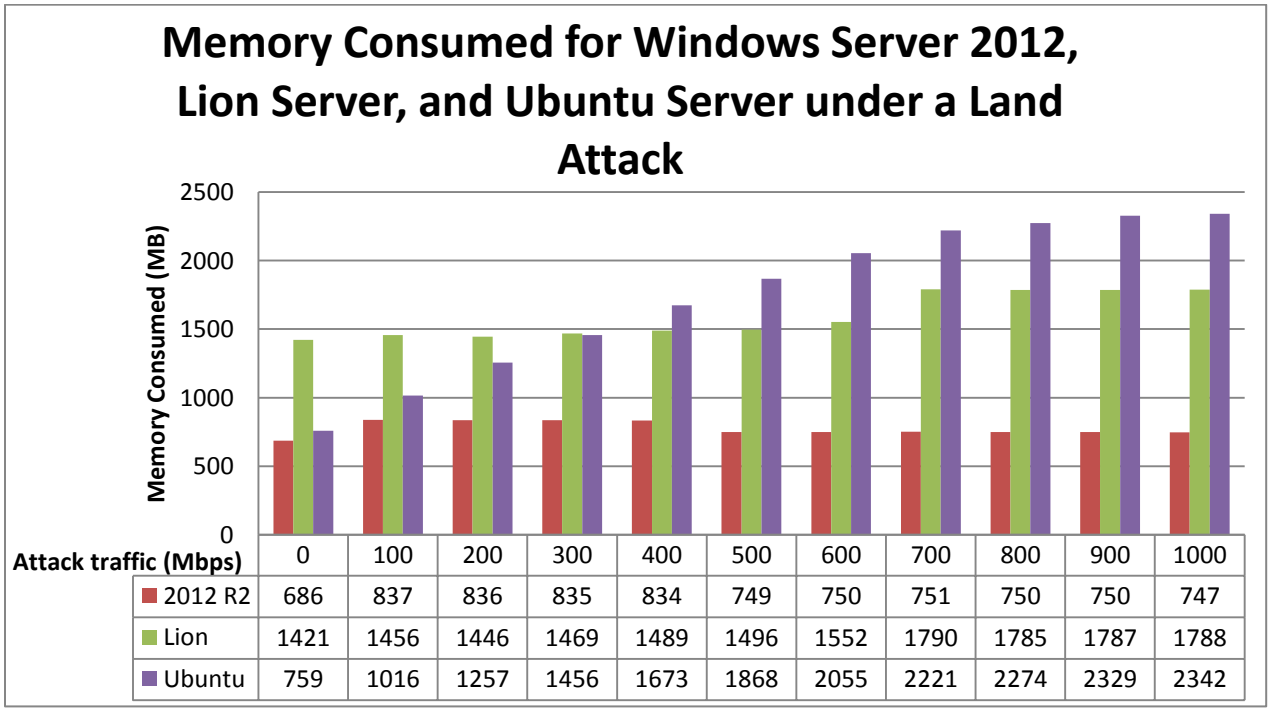


Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	20	20	20	20	20	14	14	14	14	13	12
Lion	13	16	19	23	24	28	29	24	24	24	24
Ubuntu	23	25	25	25	25	29	29	23	15	16	14

Figure 40 - CPU Utilization for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under an ICMP Land Attack

The amount of memory that was being consumed by this attack was minimal. At start up, the amount of memory that was being consumed by Lion Server was 1421 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1788 Mbytes and occurred at 1000 Mbps of attack traffic. At start up, the amount of memory that was being consumed by Windows Server 2012 R2 was 686 Mbytes and as a baseline was the lowest. As the attack traffic was increased, we observed that Windows Server 2012 R2 had a maximum memory consumption of 837 Mbytes and occurred at 100 Mbps of attack traffic. What was interesting was the inverse affect that the connection rate and the limiting threshold had to the memory consumption. For instance, when the connection rate was close to the baseline reading and the attack traffic was between 100 Mbps and 400

Mbps, we can see from Figure 41 that the consumption of memory was at its highest. However, once the limiting threshold was reached and the connection rate dropped the amount of memory that was being consumed decreased. Which is the opposite of what we observed when Lion Server and Ubuntu Server reached their failure limit. Ubuntu was consuming 759 Mbytes of memory at 0 Mbps of attack traffic. As we increased the attack traffic, we saw a steady increase of 200 Mbytes in the consumption of memory from 100 Mbps to 700 Mbps. Then, the failure point was reached and a slight change in the consumption of memory was observed for the rest of the trials. In the end, the maximum consumption of memory was 2342 Mbytes and was the highest of all three operating systems.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	686	837	836	835	834	749	750	751	750	750	747
Lion	1421	1456	1446	1469	1489	1496	1552	1790	1785	1787	1788
Ubuntu	759	1016	1257	1456	1673	1868	2055	2221	2274	2329	2342

Figure 41 - Memory Consumed for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under an ICMP Land Attack

5.3.3 TCP-SYN Flood Attack

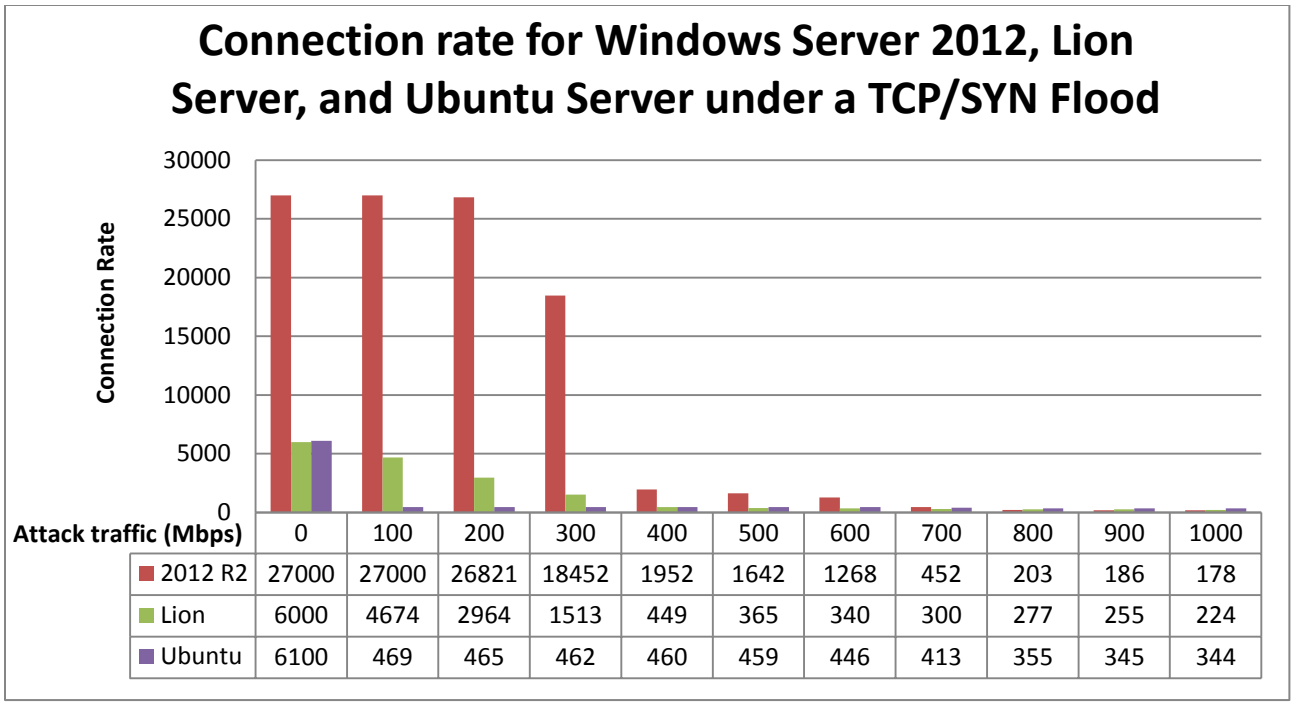
In this section, we will begin by presenting the data for the closed port TCP-SYN attack followed by the open port TCP-SYN attack. One thing to note is that we are trying to establish numerous amounts of half-open TCP connections. We will attempt to establish half-open connections to two different types of TCP ports, one that is closed and non-listening and another that will be opened and listening. The idea behind this variation is simple and can be explain by an analogy. Consider a home that is empty, if someone comes knocking at the door we do not expect to receive a response. However, if there is someone at home we expect to be greeted and a response is expected. That is the idea behind a closed and open port. A closed and non-listening port is an empty home, while an open and listening port is a home with someone inside it. By attacking a closed port, we assume that the overall performance of the computer system should not get affected. We believe that attacking an open TCP port will result in a more devastating attack, when compared to a closed port attack. Overall, we wanted to see the effects of each attack and if all three operating systems had taken precautions by implementing security for them.

We will begin by presenting the data for the closed port attack, followed by the data for the open port TCP/SYN attack. In each attack, we will be presenting the utilization of the CPU followed by the amount of memory that was being consumed during the attack. For both of these attacks, we will be trying to create half-open TCP connections by sending a TCP packet with the SYN flag set. We will be increasing the attack traffic from 0 to 1000 Mbps for a duration of six minutes and a cool down period of two minutes in between each trial.

Closed Port

While under no attack traffic, the connection rates for Lion Server, Ubuntu Server, and Windows Server 2012 R2 were 6,000, 6,100, and 27,000 connections per second, respectively and can be seen in Figure 42. As we introduced the TCP/SYN traffic to Lion Server, we observed that Lion was able to handle up to 300 Mbps of attack traffic before we determined that the connection rate is too low to be considered a working web server. At 300 Mbps, the connection rate was 1513 and at 400 Mbps it dropped to 449. For the remainder of the trails the lowest that the connection rate fell to was 224 and occurred at 1000 Mbps. When Ubuntu Server was subjected to the TCP/SYN attack traffic, we discovered that it was not even able to handle 100 Mbps of traffic. The connection rate went for 6,100 to 469 at 100 Mbps. This was about the connection rate that Lion Server reached at 400 Mbps. This showed that Lion was able to prolong the effects when compared to Ubuntu.

When the TCP/SYN was introduced to Windows Server 2012, we did not see an impact to the connection rate until 300 Mbps. At 300 Mbps, the connection rate went from 26,821 to 18,452. However, between the attack traffic of 400 and 600 Mbps, we observed that the connection rate was less than 2,000 connections per second but still about 3 to 4 times higher than Lion Server and Ubuntu Server. It was not until 700 Mbps that the connection rate dropped to 450 and we could no longer say that Windows Server 2008 was efficient.



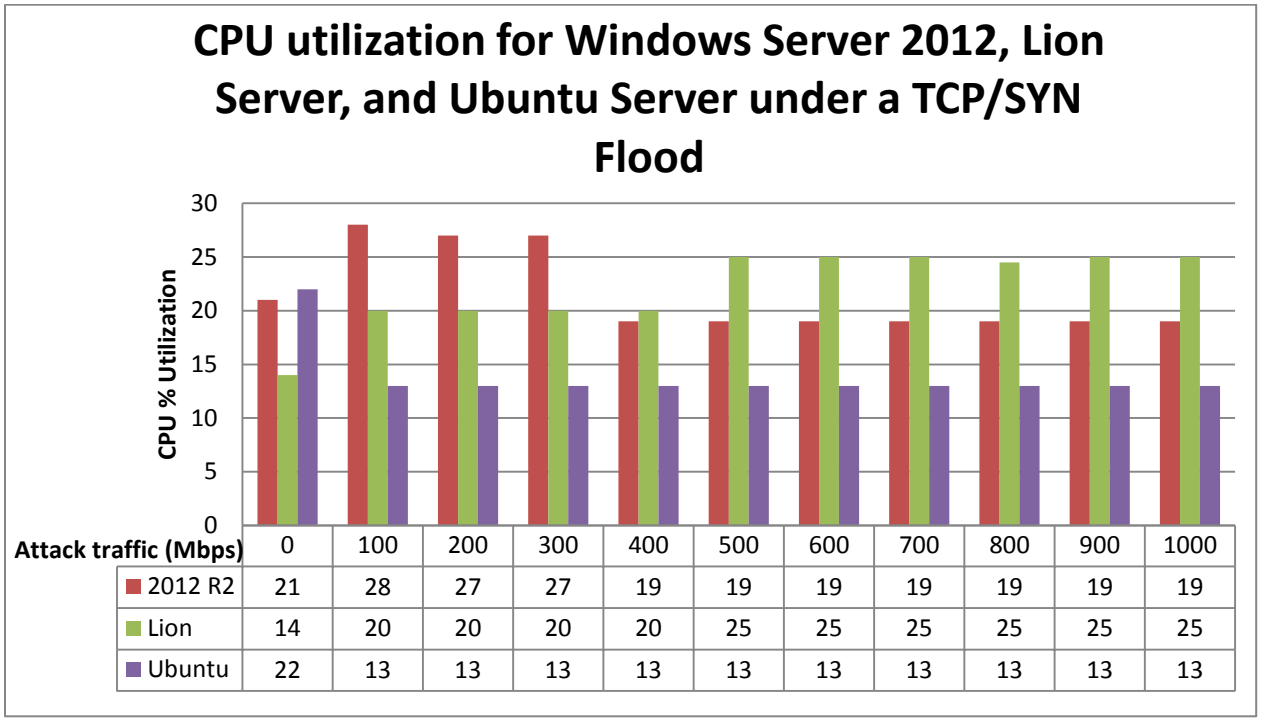
Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	27000	27000	26821	18452	1952	1642	1268	452	203	186	178
Lion	6000	4674	2964	1513	449	365	340	300	277	255	224
Ubuntu	6100	469	465	462	460	459	446	413	355	345	344

Figure 42 - Connection rate for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under a TCP/SYN Flood attack

When it came to the baseline reading for the utilization of the CPU, Lion Server had the lowest at 14 %. While, Windows Server 2012 R2 and Ubuntu had a baseline CPU utilization of 21 % and 22 % respectively. When Ubuntu was introduced to 100 Mbps of attack traffic, we observed that the CPU Utilization decreased from 22 % to 13 %. This was sort of expected since we discovered that when the connection rate drops the CPU Utilization also drops. So for the rest of the attack trials, the CPU utilization stayed at 13 %.

From Figure 43, we can see that when Lion Server was introduced to the attack traffic, we saw that the CPU utilization went up to 20 % from 100 Mbps to 400 Mbps. Then for the rest of the trials, the CPU utilization went up to 25 %. So far, we have seen that when the connection

rate decreases the CPU utilization usually drops, but this time around the CPU utilization increased when the connection rate dropped. When Windows Server 2012 was introduced to the attack traffic, the CPU utilization increased to 28 % for 100 Mbps to 300 Mbps. Then, for the remainder of the trials the utilization drops to 19 %.

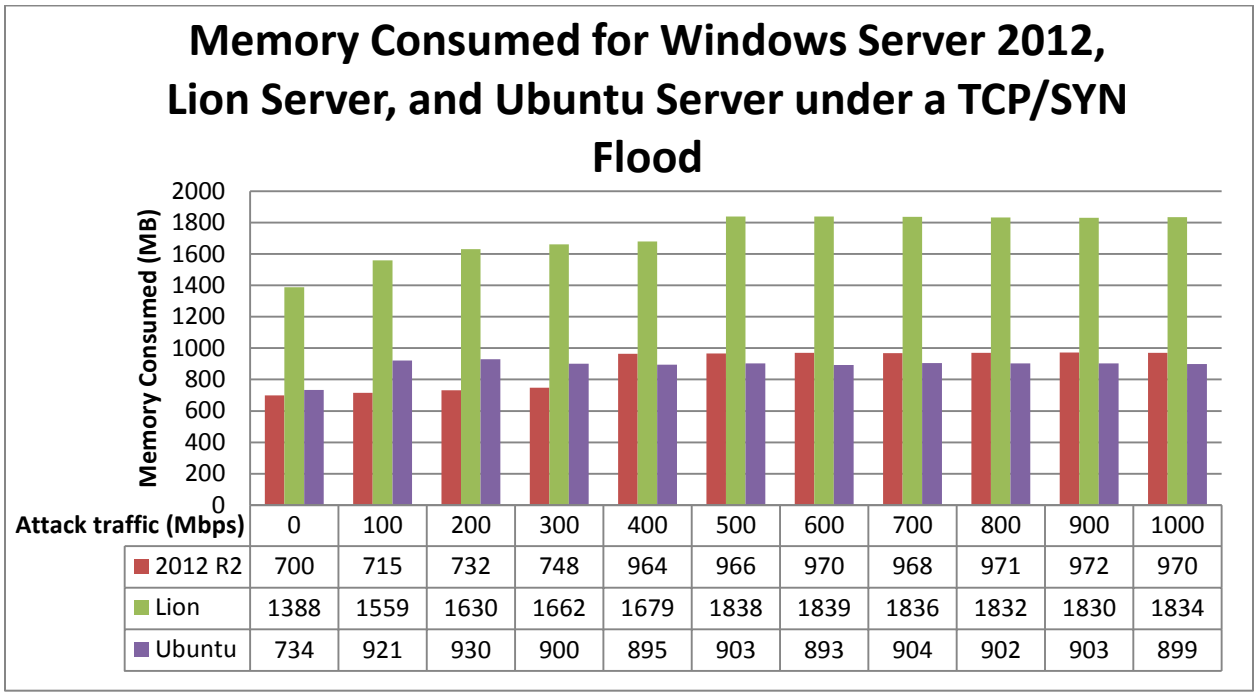


Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	21	28	27	27	19	19	19	19	19	19	19
Lion	14	20	20	20	20	25	25	25	25	25	25
Ubuntu	22	13	13	13	13	13	13	13	13	13	13

Figure 43 - CPU Utilization for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under a TCP/SYN Flood attack

At start up, the amount of memory that was being consumed by Lion Server was 1388 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1836 Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu Server was consuming 734 Mbytes of memory at 0 Mbps of attack traffic. As we increased the attack traffic, we saw an immediate increase in the consumption of memory. The

maximum consumption of memory was 930 Mbytes and occurred at 200 Mbps. However, at 300 Mbps it dropped to 900 Mbps and seemed to fluctuate around there for the rest of the trials. At start up, the amount of memory that was being consumed by Windows Server 2012 R2 was 700 Mbytes and as a baseline was the lowest. As we increased the attack traffic, we observed that Windows Server 2012 R2 also had a drastic increase in memory consumption and occurred at 400 Mbps. This coincides with the decrease in the connection rate. Once this occurred, the consumption of memory was fluctuating around 965 Mbytes for the rest of the trails.

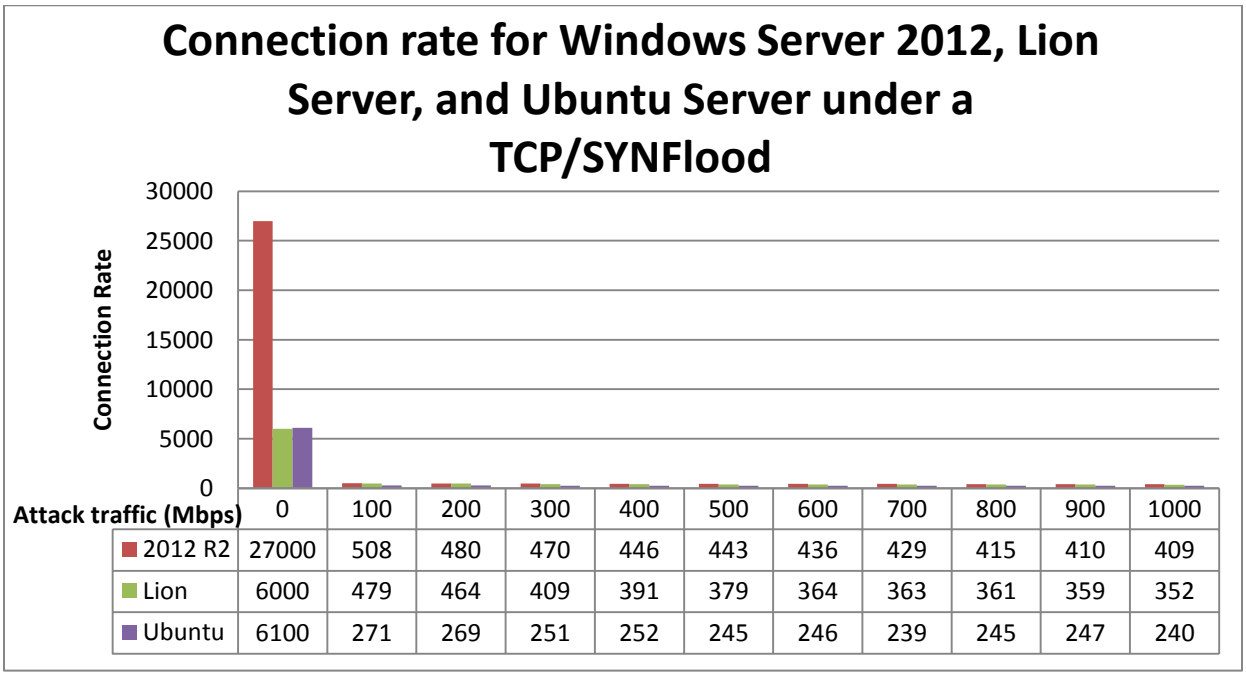


Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	700	715	732	748	964	966	970	968	971	972	970
Lion	1388	1559	1630	1662	1679	1838	1839	1836	1832	1830	1834
Ubuntu	734	921	930	900	895	903	893	904	902	903	899

Figure 44 - Memory Consumed for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under a TCP/SYN Flood attack

Open Port

Just like in the other cases, the baseline reading for the connection rate was the same. While under no attack traffic, the connection rates for Lion Server, Ubuntu Server, and Windows Server 2012 R2 were 6,000, 6,100, and 27,000 connections per second, respectively and can be seen in Figure 45. What was interesting was that when we introduced the attack traffic, none of the operating systems had good performance. At 100 Mbps, the connection rate for Lion Server was 479 and by 1000 Mbps was at 352. Windows Server 2012 R2 had a connection rate of 508 at 100 Mbps and by 1000 Mbps was at 409. When Ubuntu Server was subjected to the TCP/SYN attack traffic, we discovered that it had the lowest connection rate. At 100 Mbps, the connection rate was 271 and eventually dropped to 240 at 1000 Mbps of attack traffic. While under the open port TCP/SYN attack the connection rate was at or below 500 for the entire attack traffic range.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	27000	508	480	470	446	443	436	429	415	410	409
Lion	6000	479	464	409	391	379	364	363	361	359	352
Ubuntu	6100	271	269	251	252	245	246	239	245	247	240

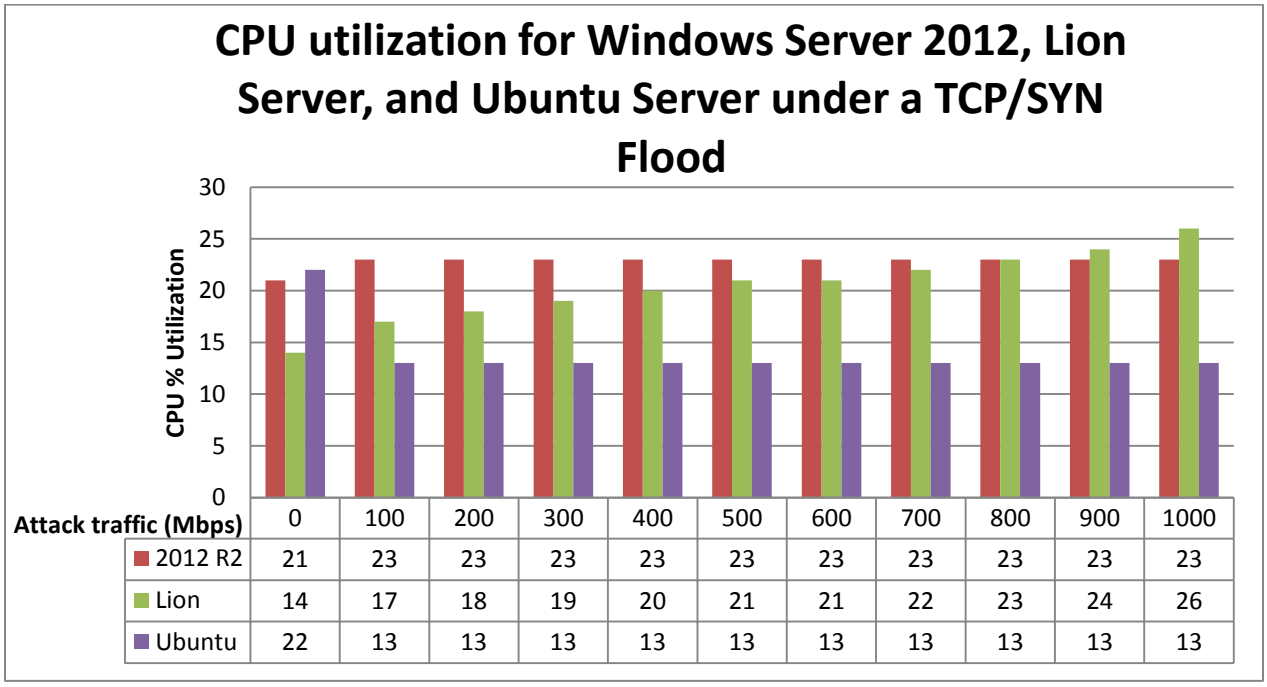
Figure 45 - Connection rate for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under a TCP/SYN Flood attack

As with all hardware components, there is a finite amount of resources in any given system. That is the case here, the TCP protocol was designed to allocate or reserve a small chunk of memory for each connection or half-open connection that is established. Therefore, the computer has established a limit on the number of simultaneous full and half open connections that can be established before a new connection can be requested. So when a computer system receives a packet that is trying to establish a TCP connection the operating system should make sure that the computer has not reached the threshold for this parameter. It is a little more complicated than just a comparison or lookup. It also involves the numbers of retries that the server has tried to reestablish the half-open connection, the amount of time left on the connections timer, etc. before a connection is dropped. It should be noted that this threshold is set to a default value for each operating system, but can be modified if needed.

Since all three operating systems were not able to withstand the open port TCP/SYN attack traffic, the CPU utilization for Ubuntu Server and Lion Server are easier to explain than Windows Server 2012 R2. When it came to the baseline reading for the utilization of the CPU, Lion Server had the lowest at 14 %. While, Windows Server 2012 R2 and Ubuntu had a baseline CPU utilization of 21 % and 22 % respectively. When Ubuntu was introduced to the attack traffic, we observed that the CPU Utilization decreased from 22 % to 13 %. This was sort of expected, since we discovered that when the connection rate drops the CPU utilization also drops. So for the rest of the attack trials, the CPU utilization stayed at 13 %. This was the same behavior that was exhibited during the closed TCP/SYN attack. From Figure 46, we can see that when Lion Server was introduced to the attack traffic, we saw that for each sample point the utilization was increasing. The maximum CPU was 26 % and was achieved at 1000 Mbps. Again

this is expected, since we noticed that the CPU utilization was increasing as the connection rate decrease during the closed port TCP/SYN attack.

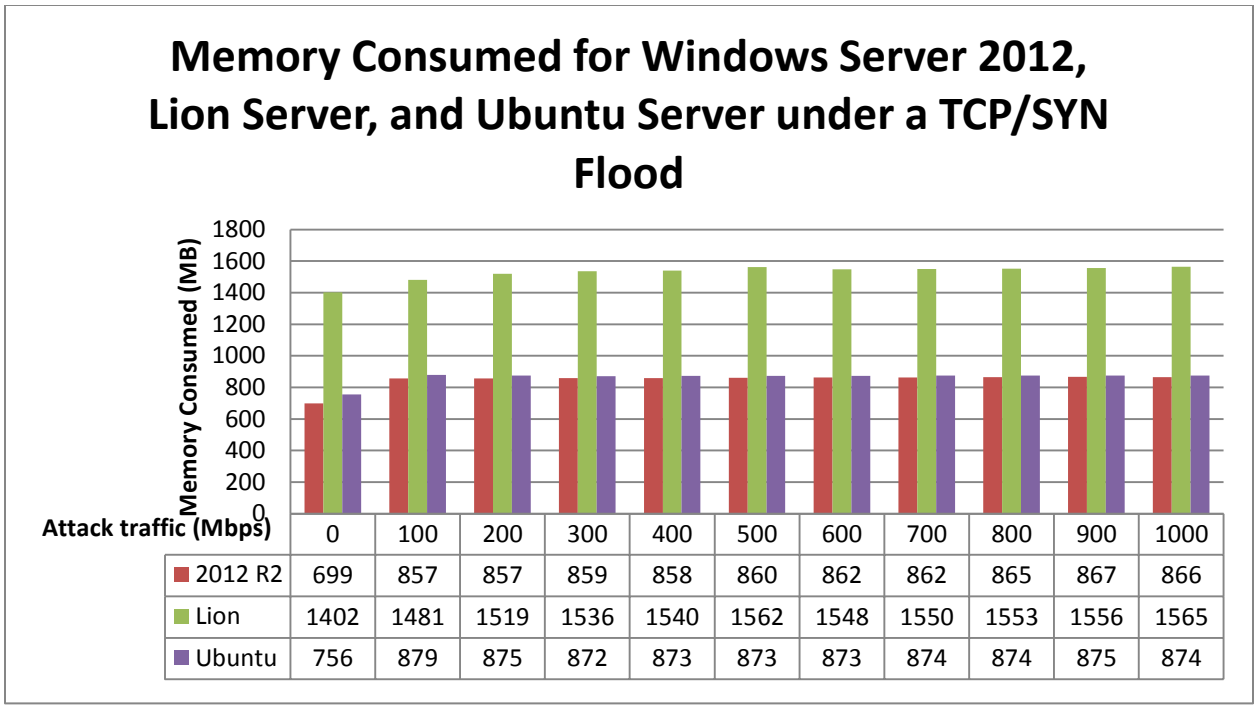
However, when Windows Server 2012 R2 was introduced to the attack traffic we saw that the CPU utilization increased to 23 % for each of the trials. This is a little harder to explain, since this was not the correlation that we have associated with Windows in the previous experiments and is the first time that we have seen this behavior. Usually, there is a proportional relationship between the connection rate and the utilization of the CPU. To the best of our understanding, we believe that the procedure that is being executed by the closed port attack is different than the procedure that is being executed by the open port attack. The added security that is being implemented in the procedure has translated to extra CPU cycles.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	21	23	23	23	23	23	23	23	23	23	23
Lion	14	17	18	19	20	21	21	22	23	24	26
Ubuntu	22	13	13	13	13	13	13	13	13	13	13

Figure 46 - CPU Utilization for Windows Server Enterprise 2012 R2, Lion Server and Ubuntu Server under a TCP/SYN Flood attack

At start up, the amount of memory that was being consumed by Lion Server was 1402 Mbytes and as a baseline was the highest. As the attack traffic was increased, we observed that the maximum consumption of memory was 1565 Mbytes and occurred at 1000 Mbps of attack traffic. Ubuntu Server was consuming 756 Mbytes of memory at 0 Mbps of attack traffic. As we increased the attack traffic, we saw an immediate increase of about 120 Mbytes in the consumption of memory. The maximum consumption of memory was 879Mbytes and occurred at 100 Mbps. As we increased the attack traffic we discovered that the consumption of memory was oscillating around 873 Mbytes for the remainder of the trials. The amount of memory that was being consumed by Windows Server 2012 R2 was 699 Mbytes and as a baseline was the lowest. As we introduced the attack traffic, we observed that Windows Server 2012 R2 also had a drastic increase in the consumption of memory occurred at 100 Mbps. Although at 100 Mbps the memory consumption was not at its maximum, the memory increased a total of 9 Mbytes from 100 Mbps to 1000 Mbps.



Attack traffic (Mbps)	0	100	200	300	400	500	600	700	800	900	1000
2012 R2	699	857	857	859	858	860	862	862	865	867	866
Lion	1402	1481	1519	1536	1540	1562	1548	1550	1553	1556	1565
Ubuntu	756	879	875	872	873	873	873	874	874	875	874

Figure 47 - Memory Consumed for Windows Server Enterprise 2012 R2, Lion Server, and Ubuntu Server under a TCP/SYN Flood attack

5.4 Chapter Summary

With the utilities provided by each operating system, we were able to gather information about the performance of the computer while under the ICMP Ping Flood, Land Attack, and TCP/SYN Flood attacks [61] – [65]. Then, by using Microsoft's Excel, we were able to translate our data into graphs. Using graphs and tables allows us to present our data in a way that helps readers visualized and compare the performance of each system with minimal effect.

Overall, all three operating systems were implementing similar techniques when under each attack. For instance, all three server operating systems were limiting the number of Echo Request and Echo Reply messages that they would receive and send while under the Ping Flood

attack. Just like in the desktop version of Lion, Lion Server had the lowest limiting threshold value for the number of echo request messages received per second. However, this did not contribute to an overall better performance. Just like in programming and electronic design, engineers are faced with constraints in their hardware and software designs that requires' them to choose between sacrificing performance or efficiency. These tradeoffs decision happen every day and depend on the specifications of the system and the requirements of the project. For instance, Apple chose a lower threshold limit for the number of echo request messages it would allow by modifying the size of memory that is allocated to the queue on the network interface card. By utilizing a smaller queue size, lead to a lower threshold limit than its competitors but a higher CPU utilization.

Even with a lower threshold limit, Lion was outperformed by Windows Server 2012 R2 and Ubuntu Server when we compared the utilization of the CPU. According to the simulations, Windows Server 2012 had a maximum CPU consumption of 23%, while under the Ping Flood attack. Lion had a maximum CPU consumption of 25% while Ubuntu was using 27 %. However, the more valuable parameter to measure is the number of successful connections per second that each operating system was able to establish before they were affected by the attack traffic. In this case, Windows Server 2012 was able to handle the attack up to the speed of 700 Mbps. Lion Server was able to sustain a significant number of connection up to 400 Mbps, while Ubuntu Server was only able to handle the attack up to 300 Mbps.

When it came to subjecting our operating systems to the Land Attack flood, we again observed the same limiting technique as in the Ping Flood Attack. Lion Server was limiting the number of echo request packets that it would receive, however the threshold value was a lot

higher and reached a maximum of 1,048,944 packets per second. Windows Server 2012 R2 and Ubuntu Server were limiting theirs to about 589,107 and 0 packets per second respectively. When it came to replying to these packets, we observed that Windows Server 2012 was not acknowledging any of the echo request packets that it was receiving. Lion Server was replying with 250 echo reply packets per second for each attack trail. According to our test results, we found that Windows Server 2012 had the lowest maximum CPU utilization. It had a maximum CPU utilization of 20 %. Both Lion Server and Ubuntu Server had a maximum CPU utilization of 29 %. When we focused on the connection rate, Windows Server 2012 was able to handle the attack up to the speed of 900 Mbps, while Lion Server and Ubuntu Server were able to handle the attack up to 600 Mbps and 700 Mbps respectively.

When Lion Server was introduced to the closed TCP/SYN attack, we found that the maximum CPU utilization was 25 % and was reached under the attack load of 600 Mbps. We also saw that Windows Server 2012 had a maximum CPU utilization of 28 % at the attack speed of 100 Mbps and Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 13% at the attack speed of 100 Mbps. When the connection rate parameter was graphed, we discovered that Windows Server 2012 was able to sustain a high connection rate until 300 Mbps. Lion Server was able to prolong the effects until 200 Mbps of attack traffic, while Ubuntu was not even able to withstand 100 Mbps. Ubuntu server had a connection rate of 469 at 100 Mbps.

When Lion Server was introduced to the open TCP/SYN attack, we found that the maximum CPU utilization was 26 % and is the highest of all three. When Windows Server 2012 was subjected to the attack traffic, we discovered that for the whole attack range the CPU utilization

stayed at 23 %. When Ubuntu Server was under attack, the CPU exhaustion decreased to 13 % and stayed there for the whole attack range. When the connection rate parameter was graphed, we discovered that all three operating systems were not able to sustain a high connection rate at 100 Mbps. Each operating system had a connection rate of less than 500.

Overall, we saw that all three server operating systems were able to keep the utilization of the CPU and the amount of memory to a safe percentage; however, we still believe that none were able to successfully survive these attacks at the higher transmission rates. Since, we are testing the security that is provided by each operating system to the webserver application running on the iMac Pro Server. The connection rate is the parameter that has greater significance to the overall efficiency and performance of the server.

CHAPTER VI

CONCLUSION AND FUTURE WORK

The problem of network security is one that has received much consideration from experts in the field. However, this has not stop attackers from trying to bring down devices such as computers and servers. As previously mentioned, Apple Inc., has claimed that an iMac computer running the latest OS X operating system is reliable, more powerful, and safer than any other computer on the market [14]. Claims like these sound promising, but yet suspicious. They lead to a false sense of security if the proposed claim is false. Therefore, it is imperative that outside research is performed so as to justify these claims.

In this thesis, we evaluated the built-in security provided by seven popular and globally deployed computer and server operating systems while under three popular yet powerful cyber-attacks. We discovered the impact that the popular Ping, Land Attack, and TCP/SYN Flood attacks had on an Apple's iMac computer deploying the following Operating Systems: Apple's OSX 10.7.5 "Lion", Microsoft's "Windows 7", and Canonicals' Ubuntu 13.10 "Saucy Salamander". The same cyberattacks were used to study the impact on a iMac Pro Server deploying the following Operating Systems: Apple's OS X 10.7.5 Server "Lion", Ubuntu's 12.04.5 LTS "Precise Pangolin", Microsoft's "Windows Server 2008 Enterprise R2" and "Windows Server 2012 Enterprise R2".

Overall, we saw that all three desktop operating systems were able to survive these attacks. Although Lion was able to survive the attack and still functioned correctly, we feel that since Lion was outperformed by both operating systems the claim that an iMac computer running the latest Apple operating system is the most secure system was false. Claims like this that are made by big corporations are example of statements that need to be researched and verified by the public domain before they are allowed to make claims like this and market their software to the general public.

According to the simulations, during the Ping Flood attack Lion was limiting the number of echo request packet to a maximum of 400, 000 packets per second. Windows 7 and Ubuntu were limiting theirs to about 820,000 and 906,000 packets per second respectively. When it came to replying to these packets, we observed that Windows 7 would acknowledge only the first 500 echo request packets that it would receive and then would not reply after that. With Lion, we observed that it was replying with 250 echo reply packet per second for the duration of the attack. According to our test results, we found that Ubuntu had the lowest CPU utilization. It had a maximum CPU utilization of 13 %. The highest was Lion with 32 % and Windows with 15%.

When it came to subjecting our operating systems to the Land Attack flood, we again observed the same limiting technique as in the Ping Flood Attack. Lion was limiting the number of echo request packet to a maximum of 400,000 packets per second. Windows 7 and Ubuntu were limiting theirs to about 710,000 and 906,000 packets per second respectively. When it came to replying to these packets, we observed that Windows 7 was not acknowledging any of the echo request packets that it was receiving. With Lion, we observed that it was replying with 250 echo reply packet per second for the duration of the attack. According to our test results, we

found that Ubuntu again had the lowest CPU utilization. It had a maximum CPU utilization of 13 %. The highest was Lion with 32 % and Windows with 16%. One thing to mention is that Windows 7 seemed to be the only operating system that was able to detect the subtle difference between the Ping and Land Attacks.

When introducing our operating systems to the closed TCP/SYN attack, Lion, had a maximum CPU utilization of 32 % and was reached under the attack load of 200 Mbps. We also saw that Windows 7 had a maximum CPU utilization of 30 % at the attack speed of 400 Mbps and Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 12% at the attack speed of 700 Mbps. When introduced to the open TCP/SYN attack, Lion had a maximum CPU utilization of 37 %. We saw that Windows 7 had a maximum CPU utilization of 29 % at the attack speed of 400 Mbps and Ubuntu had a maximum CPU exhaustion of 16 % at the attack speed of 700 Mbps.

When we exposed the server operating systems to the cyber-attacks, we discovered that they were implementing similar techniques like the ones implemented in the desktop versions. For instance, all three server operating systems were limiting the number of Echo Request and Echo Reply messages that they would receive and send.

Even with a lower threshold limit, Lion Server was outperformed by Windows Server 2012 R2 and Ubuntu Server when we compared the utilization of the CPU. According to the simulations, Windows Server 2012 had a maximum CPU consumption of 23 %, while under the Ping Flood attack. Lion had a maximum CPU consumption of 25 % while Ubuntu had 27 %. However, the more valuable parameter to measure is the number of successful connections per

second that each operating system was able to establish before they were affected by the attack traffic. In this case, Windows Server 2012 was able to handle the attack up to the speed of 700 Mbps. Lion Server was able to sustain a significant number of connection up to 400 Mbps, while Ubuntu Server was only able to handle the attack up to 300 Mbps.

When it came to subjecting our operating systems to the Land Attack flood, Lion Server was limiting the number of echo request packets that it would receive to 1,048,944 packets per second. Windows Server 2012 R2 and Ubuntu Server were limiting theirs to about 589,107 and 0 packets per second, respectively. When it came to replying to these packets, we observed that Windows Server 2012 was not acknowledging any of the echo request packets that it was receiving. Lion Server was replying with 250 echo reply packets per second for each attack trail. According to our test results, we found that Windows Server 2012 had the lowest maximum CPU utilization. It had a maximum CPU utilization of 20 %. Both Lion Server and Ubuntu Server had a maximum CPU utilization of 29 %. When we focused on the connection rate, Windows Server 2012 was able to handle the attack up to the speed of 900 Mbps, while Lion Server and Ubuntu Server were able to handle the attack up to 600 Mbps and 700 Mbps respectively.

When Lion Server was introduced to the closed TCP/SYN attack, we found that the maximum CPU utilization was 25 % and was reached under the attack load of 600 Mbps. We also saw that Windows Server 2012 had a maximum CPU utilization of 28 % at the attack speed of 100 Mbps and Ubuntu had the lowest CPU exhaustion of all three. It had a maximum CPU exhaustion of 13% at the attack speed of 100 Mbps. When the connection rate parameter was graphed, we discovered that Windows Server 2012 was able to sustain a high connection rate

until 300 Mbps. Lion Server was able to prolong the effects until 200 Mbps of attack traffic, while Ubuntu was not even able to withstand 100 Mbps. Ubuntu server had a connection rate of 469 at 100 Mbps.

When Lion Server was introduced to the open TCP/SYN attack, we found that the maximum CPU utilization was 26 % and is the highest of all three. When Windows Server 2012 was subjected to the attack traffic, we discovered that for the whole attack range the CPU utilization stayed at 23 %. When Ubuntu Server was under attack, the CPU exhaustion decreased to 13 % and stayed there for the whole attack range. When the connection rate parameter was graphed, we discovered that all three operating systems were not able to sustain a high connection rate at 100 Mbps. Each operating system had a connection rate of less than 500.

Overall, we saw that all three server operating systems were able to keep the utilization of the CPU and the amount of memory to a safe percentage; however, we still believe that none were able to successfully survive these attacks at the higher transmission rates. Since, we are testing the security that is provided by each operating system to the webserver application running on the iMac Pro Server.

REFERENCES

[1] Living in a web-based world

(http://www.connect-world.com/~cwiml/index.php/magazines/north-america/item/1094-living-in-a-web-based-world)

[2] Brief History of the Internet

(http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet)

[3] ARPANET

(http://en.wikipedia.org/wiki/ARPANET)

[4] System Administration Guide: IP Services

(http://docs.oracle.com/cd/E23823_01/html/816-4554/ipov-6.html)

[5] The TCP/IP Guide
(http://www.tcpipguide.com/free/t_TCPIPInternetArchitectureandProtocolSuite.html)

[6] TCP/IP Reference Page

(http://www.protocols.com/pbook/tcpip1.htm)

[7] What is a DDoS Attack? (2013)

http://www.digitalattackmap.com/understanding-ddos/

[8] William Stallings (2011). Network Security Essentials. United States of America: Pearson Education. p4-5.

[9] HTG Explains: How Does Email Work? (2011)

http://www.howtogeek.com/56002/htg-explains-how-does-email-work/

[10] Mirkovic, J., Dietrich, S., Dittrich, D. and Reiher, P. (2013) Understanding a Denial of Service Attack. (http://www.informit.com/articles/article.aspx?p=386163&seqNum=5)

[11] DDoS Attacks Against Government and Entertainment Websites Escalate (2013) (http://www.infosecisland.com/blogview/19543-DDoS-Attacks-Against-Government-and-Entertainment-Websites-Escalate.html)

[12] Extra Life charity comes under DDoS attack.

(http://www.joystiq.com/2013/11/02/extra-life-charity-comes-under-ddos-attack/)

[13] Arbor Networks (2014) Worldwide Infrastructure Security Report. http://www.arbornetworks.com/research/infrastructure-security-report

[14] (2014) OS X Lion – The World's Most Advanced Desktop Operating Systems. http://web.archive.org/web/20110806091718/http:/www.apple.com/macosx/

[15] (2014) Apple's Lion Roars onto Computers with 1 Million Downloads in a Day. The Independent (UK).

http://www.independent.co.uk/life-style/gadgets-and-tech/apples-lion-roars-onto-computers-with-1-million-downloads-in-a-day-2318755.html

[16] (2013) Why you'll love a MAC

(http://www.apple.com/getamac/whymac/)

[17] Nash, M. (2014) Why 7? The Windows Blog. Microsoft.

http://blogs.windows.com/windows/archive/b/windowsvista/archive/2008/10/14/why-7.aspx

[18] Wikipedia (2014) Windows 7.

http://en.wikipedia.org/wiki/Windows_7

[19] Ubuntu 13.10 “Saucy Salamander”

http://releases.ubuntu.com/13.10/

[20] Ubuntu 13.10

http://www.techspot.com/downloads/5068-ubuntu.html

[21] Nayes, Katherine., Diertrich, Schmitz., (2012) 10 Reasons to choose Ubuntu 12.10 over Windows 8.

http://www.pcworld.com/article/2013431/10-reasons-to-choose-ubuntu-12-10-over-windows-8.html

[22] Windows Server 2008 R2

http://www.microsoft.com/en-us/download/details.aspx?id=11093

[23] Windows Server 2012 R2

http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/explore.aspx

[24] Wikipedia (2014) Usage share of operating Systems

http://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Desktop_and_laptop_computers

[25] w3Techs (2014) Historical Trends in the Usage of operating systems (2014)

http://w3techs.com/technologies/history_overview/operating_system

[26] Foley, Mary Jo., (2012) Microsoft Extends Windows Server 2008 Support Cut-Off Date.

http://www.zdnet.com/article/microsoft-extends-windows-server-2008-support-cut-off-date/

[27] Keizer, Gregg. (2012) Microsoft Extends Windows Server 2008 Support by 18 Months

http://www.computerworld.com/article/2491621/operating-systems/microsoft-extends-windows-server-2008-support-by-18-months.html

[28] Ubuntu 12.04.5 LTS "Precise Pangolin"

http://releases.ubuntu.com/12.04/

[29] Precise Upgrades (2014)

https://help.ubuntu.com/community/PreciseUpgrades

[30] Features in Ubuntu 12.04.4 (2014)

https://wiki.ubuntu.com/PrecisePangolin/ReleaseNotes/UbuntuDesktop

[31] Westervelt, Robert. DDoS Attack Behind Latest Network Solutions Outage (2013)

http://www.crn.com/news/security/240158492/ddos-attack-behind-latest-network-solutions-outage.htm

[32] Franzen, Carl (2013) 'Largest' public denial of service attack in internet history linked to Europe span dispute

http://www.theverge.com/2013/3/27/4152540/largest-ddos-attack-spamhaus-linked-to-cyberbunker-spam

[33] DDoS Historical Achives (2015)

https://www.shadowserver.org/wiki/pmwiki.php/Stats/DDoSHistorical

[34] Your Resource for DDoS Protection (2014)

http://www.ddosattacks.biz/tags/tags/attacks/

[35] Hoffman, Stefanie. (2013) DDoS: A Brief history

http://blog.fortinet.com/post/ddos-a-brief-history

[36] Siva Ram Murthy, C. and Manoj, B.S. Transport Layer and Security Protocols for Ad Hoc Wireless Networks (2005)

http://www.informit.com/articles/article.aspx?p=361984&seqNum=10

[37] Report for comments for ICMP

http://www.ietf.org/rfc/rfc792.txt

[38] ICMP Encapsulation

http://home.mira.net/~marcop/even_more_tcpip.htm

[39] Pix For Icmp Header

http://pixgood.com/icmp-header.html

[40] Ping (networking utility)

http://en.wikipedia.org/wiki/Ping_(networking_utility)

[41] Couts, Andrew. (2011) Wordpress Suffers Devastating DDoS Attack

http://www.digitaltrends.com/computing/wordpress-suffers-devastating-ddos-attack/#skip-video

[42] Report for comments for TCP

http://www.ietf.org/rfc/rfc793.txt

[43] S. Surisety and S. Kumar, “Is Apple’s iMac Leopard Operating System Secure under ARP-Based Flooding Att acks?,” Proc. 5th Int’l Conf. Internet Monitoring and Protection, IEEE, 2010, pp. 60–64.

[44] S. Surisetty and S. Kumar, “Is McAfee Security Center/Firewall Software Providing Complete Security for your Computer?”, 2010 Fourth International Conference on Digital Society

[45] Raja Sekhar Reddy Gade, Hari Vellalacheruvu and Sanjeev Kumar, “Performance of Windows Xp, Windows Vista and Apple's, Leopard Computers under a Denial of Service Attack”, Digital Society, 2010. ICDS '10. Fourth International Conference on, vol., no., pp.188-191.

[46] S. Surisetty and S. Kumar,” Microsoft vs. Apple: Resilience against Distributed Denial-of-Service Attacks”, IEEE Security & Privacy, March/April 2012

[47] Junior, R. and Kumar, S. (2014) Apple’s Lion vs Microsoft’s Windows 7: Comparing Built-In Protection against ICMP Flood Attacks. Journal of Information Security, 5, 123-135. doi: 10.4236/jis.2014.53012.

[48] Website of datasheet for Cisco SRW2024 24-port Gigabit Switch

http://www.cisco.com/en/US/products/ps9989/index.html

[49] WebView Switches User Guide

http://www.cisco.com/c/dam/en/us/td/docs/switches/lan/csbms/srw2048/administration/guide/SRW-US_v10_UG_A-Web.pdf

[50]Website of datasheet for IXIA Optixia XM2

http://www.ixiacom.com/products/network_test/chassis/index.php?skey=ch_optixia_xm2

[51] IXIA XM2 Chasis

http://www.ixiacom.com/products/xm2-chassis

[52] XM2 Portable Chassis

http://legacy-www.ixiacom.com/products/network_test/chassis/display?skey=ch_optixia_xm2

[53] iMac User Guide

[54] Microsoft Excel 2010

http://www.microsoft.com/en-us/download/details.aspx?id=16642

[55] How to: Use Perfmon in Windows 7

http://blogs.msdn.com/b/securitytools/archive/2009/11/04/how-to-use-perfmon-in-windows-7.aspx

[56] The Network Interface Card

http://pluto.ksi.edu/~cyh/cis370/ebook/ch02c.htm

[57] Network interface controller

http://en.wikipedia.org/wiki/Network_interface_controller

[58] Overview of Windows Performance (2015)

https://technet.microsoft.com/en-us/library/cc749154.aspx

[59] Using the Terminal

https://help.ubuntu.com/community/UsingTheTerminal

[60] Netstat utility

http://linux-ip.net/html/tools-netstat.html

[61] Use Activity Monitor to read system memory and determine how much RAM is being used (OS X Mountain Lion and earlier)

https://support.apple.com/en-us/HT201538

[62] How to use the Activity Monitor

https://support.apple.com/en-us/HT201464

[63] Damico, Shane Nov. 2013Windows Servers Continue to Shine in Reliability Surveys, Downtime No Longer a Concern, available online at

http://www.corepointhealth.com/geni/windows-servers-continue-to-shine-in-reliability-surveys

[64] Usage of web servers for websites

http://w3techs.com/technologies/overview/web_server/all

[65] Usage statistics and market share of Microsoft websites

http://w3techs.com/technologies/details/ws-microsoftiis/all/all

[66] Installing IIS 8 on Windows Server 2012

http://www.iis.net/learn/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012

[67] iMac Pro Server User Guide

[68] Usage statistics and market share of OS X for websites

http://w3techs.com/technologies/details/os-macos/all/all

[69] Apache HTTP Server

http://en.wikipedia.org/wiki/Apache_HTTP_Server

[70] Apple's OS X SERVER Features

https://www.apple.com/osx/server/features/

[71] Apple OS X Server: How to configure websites

http://www.techrepublic.com/article/apple-os-x-server-how-to-configure-websites/

[72] HTTPD – APACHE2 Web Server

https://help.ubuntu.com/12.04/serverguide/httpd.html

[73] Ubuntu Server Guide

https://help.ubuntu.com/12.04/serverguide/serverguide.pdf

[74] Sar Command Manual

http://linux.die.net/man/1/sar

[75] top Command Manual

https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/top.1.html

[76] Netstat Command Manual

https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/netstat.1.html

[77] vmstat Administrator's Manual

http://linuxcommand.org/man_pages/vmstat8.html

[78] Atsar Ubuntu Manuals

http://manpages.ubuntu.com/manpages/karmic/man1/atsar.1.html

APPENDIX A

APPENDIX A

TOOLS USED DURING PERFORMANCE MEASUREMENT

Microsoft has designed a built-in tool that can be used to measure the system performance and it is called Performance Monitor. The performance monitor allows us to keep track of different parameters called counters. Overall, the performance monitor is a versatile networking tool that can be used to monitor the health of the system. The following list of parameters, or performance attributes, that were measured in Performance Monitor. The Performance Monitor also has the ability to present the data being collected in a number of views like, Chart view, Counter and Trace Log, Report View, and Alert View [61].

We used the Counter and Trace Log View to record the following parameters: \ProcessorUsage, \MemoryConsumption, \RecievedEchoRequest, \SentEchoReplies, and were saved into a log file (.csv) at a sample interval of 1 sample/sec for all the attacks that were simulated on Windows 7, Windows Server 2008 Enterprise R2, and Windows Server 2012 Enterprise R2.

To measure the performance on Apple's operating systems, Mac OS X 10.7, we had to use the command line interface to execute a few commands. The SAR (System Activity Reporter) is a command that will create a log file that will record the processor and memory utilization. The command format to log the processor behavior was [74]:

sar –u 1 360 >> CPU0.txt

The ‘-u’ parameter indicates that the resource to be monitored is the processor, the ‘1’ parameter is the number of seconds that will be waited before taking an additional measurement, and the ‘360’ is the number of samples before the command terminates. The ‘>>’ indicates that the output will be stored in a log file, “CPU0.txt”, and should not get displayed to the screen.

The TOP command was used to display and update sorted information about the general health of the system. We mainly collected the memory consumption from this command. The output of this command was also stored into a log file [75]. The format is similar to the sar command.

top –i 1 –l 360 >> “mem0.txt”

The ‘-i’ parameter indicates that the resource to be monitored is the memory, the ‘1’ parameter is the number of seconds that will be waited before taking an additional measurement, and the ‘360’ is the number of samples before the command terminates. The ‘>>’ indicates that the output will be stored in a log file, “mem0.txt”, and should not get displayed to the screen.

The last command that was to collected relevant performance parameters in Apple was the netstat (Network status) command. The netstat command displays the contents of various network-related data structures. The output will again be stored in a log file and has the format the follows [76]:

Netstat –s –w 1 –p icmp >> “echoReceived0.txt”

The ‘-s’ parameter is used to show the per-protocol statistics, the ‘-w 1’ parameter is the number of seconds that will be waited before taking an additional measurement, and the ‘-p icmp’ is used to indicate which protocol should be collected. The ‘>>’ indicates that the output will be stored in a log file, “echoReceived0.txt”, and should not get displayed to the screen.

For Ubuntu, the command that were used to collect the CPU and Memory utilization was the vmstat (virtual memory statistics) command [77]:

Vmstat -1 360 >> "CPU0.txt"

The '-1' parameter is the number of seconds that will be waited before taking an additional measurement, and the '360' is the number of samples before the command terminates. The '>>' indicates that the output will be stored in a log file, "CPU0.txt", and should not get displayed to the screen.

The atsar (system activity report) command is similar to the netstat command in Apple. The atsar command can be used to displays statistics, such as protocols being used, CPU and Memory utilization, etc.. . The output will again be stored in a log file and has the format the follows [78]:

Atsar –t 1 360 >> "TCP0.txt"

The '-t' parameter is used to show the TCP protocol statistics, the '1' parameter is the number of seconds that will be waited before taking an additional measurement, and the '360' is used to indicate which protocol should be collected. To collect data for the icmp protocol, we would use the '–w' parameter instead of '-t'. The '>>' indicates that the output will be stored in a log file, "TCP0.txt", and should not get displayed to the screen.

BIOGRAPHICAL SKETCH

Rodolfo Baez Jr was born on August 31, 1988. He finished his undergraduate studies at the University of Texas Pan American on 2012 and earned his BS in Computer Engineering. He has also earned his MS in Electrical Engineering from the University of Texas Pan American in May 2015. His permanent mailing address is,

P.O. Box 23

Hebbronville, Texas 78361

His Publications and Poster Presentations achieved during his Masters are:

- Baez Jr., R. and Kumar, S. (2014) "Apple's Lion vs Microsoft's Windows 7: Comparing Built-In Protection against ICMP Flood Attacks". Journal of Information Security, 5, 123-135.
- Baez Jr., R. and Kumar, S. (Pending) "Windows Server 2008 vs Windows Server 2012: Comparing Built-in Protection against ICMP Flood Attacks"
- Baez Jr., R. and Kumar, S. (In Progress) "Evaluation of Window's 7, Ubuntu's "Saucy Salamander", and Apple's "Lion" Operating Systems under a Denial of Service Attack"
- Baez Jr., R. and Kumar, S. "Evaluating the Computers of Tomorrow, by Using Denial-of-Service Attacks Today" Poster for HESTEC 2013 competition, UTPA.