12-2012

Evaluation of Security Availability of Data Components for A Renewable Energy Micro Smart Grid System

Leonel Aguilera Zambrano
University of Texas-Pan American

EVALUATION OF SECURITY AVAILABILITY OF DATA COMPONENTS FOR A RENEWABLE ENERGY MICRO SMART GRID SYSTEM

A Thesis

by

LEONEL AGUILERA ZAMBRANO

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

December 2012

Major Subject: Electrical Engineering

EVALUATION OF SECURITY AVAILABILITY OF DATA COMPONENTS FOR A RENEWABLE ENERGY MICRO SMART GRID SYSTEM

A Thesis
by
LEONEL AGUILERA ZAMBRANO

COMMITTEE MEMBERS

Dr. Sanjeev Kumar
Chair of Committee

Dr. Jaime Ramos Salas
Committee Member

Dr. Christine Reilly
Committee Member

December 2012

ABSTRACT

Aguilera Zambrano, Leonel, Evaluation of Security Availability of Data Components for a Renewable Energy Micro Smart Grid System. Master of Science (MS), December, 2012, 152 pp., 5 tables, 93 figures, references; 99 titles.

In this thesis, we study the development and security testing of photovoltaic data collection system. With the introduction of the smart grid concept, a lot of research has been done on the communication aspect of energy production and distribution throughout the power network. For Smart Grid, Internet is used as the communication medium for specific required services and for data collection. Despite all the advantages of the Smart Grid infrastructure, there is also some security concern regarding the vulnerabilities associated with internet access.

In this thesis, we consider security testing of the two most popular and globally deployed web server platforms Apache running on Red Had Linux 5 and IIS on Windows Server 2008, and their performance under Distributed Denial of Service Attacks. Furthermore we stress test the data collection services provided by MySQL running on both Windows and Linux Servers when it is also under DDoS attacks.

DEDICATION

The completion of my master studies would not have been possible without the blessings, love and support of God who enlightened me with knowledge, patience and wisdom. I would like to thank my Family for their love and trust. My wife, Gisela Quintero, for her love and patience, for being always there for me in every step of this journey and helped me to complete this work. Specially dedicated to our daughter, Ximena, by whom I have been inspired and motivated. My mother, Rosalia Zambrano, and my father, Juan A. Aguilera, wholeheartedly inspired, motivated and supported me by all means to accomplish this degree. Thank you all for your love and patience.

ACKNOWLEDGMENTS

I will always be grateful to Dr. Sanjeev Kumar, chair of my dissertation committee, for all his mentoring and advice. From NSF funding, research design, and data processing, to manuscript editing, he encouraged me to complete this process through his infinite patience and guidance. Thank you for believing in me and for the unconditional support you have provided throughout all this research work.

I would like to thank Dr. Reilly for her support in Database knowledge, which she always shared with me and helped me to develop useful skills.

Thanks to Dr. Ramos for his motivation and teachings on Power Systems. Thank you for depositing your trust in me and pushing me to go further.

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

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

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

The Internet is a network of interconnected computer systems that provides exchange of information in the form of datagrams, where sources and destinations are hosts identified by fixed length addresses, as described on [1]. Such systems are able to communicate to each other due to the protocols created formerly like TCP/IP Protocol, widely deployed on almost every computer networks nowadays.

These protocols were in first place, designed to enable communication between hosts, but they were not designed with enough security measures in mind. That is why protocols vulnerabilities have been exploited by malicious users known as hackers, who take advantage of these weaknesses in order to exhaust the web server resources and as a consequence, they bring down websites and web services, blocking legitimate traffic from accessing the desired information.

One example can be taken from [2], which shows that recently on January 19th 2012; hackers crippled some of the Justice Department websites that were down for some hours. More examples of damages caused by hackers can be seen in [3], when one of the largest attacks to government and music industry sites was announced.

Previously on February 2007, an attack brought down three of the thirteen root servers that help manage worldwide internet traffic [4], similar to the one that attacked the root servers

in October 2002 according to [5]. Compared to the first attack “The servers didn't go down this time because of the significant increase in computing power in the last four years and because the roots' defenses have been heavily beefed up since then” as mentioned in [5]. Web Servers from social networks like Tweeter and Facebook have also experienced the same problems as a result of Denial-of-Service attacks [6].

Most of these attacks are done by taking control of unprotected computers from users around the world. These computers are referred to as Botnets. A discovery of 1 million of botnets was found on December 2011 on the UK [7], and it is considered that 6% from UK computers are included in botnets. In [8] we find that most web servers available are vulnerable to DoS attacks.

The impact when a web server is attacked can vary depending on the kind of website which has been targeted, but it may be mainly monetary, social, and it may disrupt communication or have remarkable consequences.

In [9] we can see some surprising numbers regarding DoS attacks. It states that around 4000 Distributed-Denial-of-Service occur each week, and in 2004 various companies reported over $26 million loss due to this cybercrime. In [10] we can get an up to date and historical report of DDoS that have been detected.

With the ongoing implementation of the Smart Grid, a lot of interest has been shown to the security aspect of it. “The Smart Grid introduces a two way dialog where electricity and information can be exchanged between the utility and its customers; it’s a developing network of communications, controls, computers, and new technologies and tools…” [11].

The smart grid will use the internet as the network platform to establish its two way communication dialog. We are interested in observing its behavior under different kind of

disturbances that the smart grid is inheriting from the internet, and how it may impact data components and their performance under security attacks.

1.1 Motivation

The main goals for the Security of the Smart Grid are to protect all smart grid services from malicious attack and unintended adverse cyber and physical events that interrupt critical functions, not to allow smart grid services, networks, or technologies to be used as a stepping stone or conduit for attacks, and the system security mechanisms shouldn't provide an attack vector themselves, nor should they incorrectly respond to either malicious or benign commands in a manner that would create or worsen a security event [12].

Our main goal in this thesis is to investigate how the internet DDoS attacks can impact the data collection system for renewable solar energy we have developed in the University of Texas-Pan American. We will test the main platforms used to access and store this information. Basically its backend is compound of MySQL Server and HTTP Servers.

This Thesis has been completed in order to utilize the Data generated by the Photovoltaic Systems installed in UTPA and administered by the Power Systems Laboratory. We have also been inspired based on previous thesis work related to this topic. The previous work was done in the Network Research Laboratory testing the vulnerability of other Web Servers and diverse Operating Systems and Services when they are exposed to Distributed Denial of Service Attacks.

The impact this type of attack has proven to have on the systems under test is most of the times devastating for legitimate users who require access to the offered services. Such Services become impossible to use due to the malicious traffic and therefore, this represents a threat for

the communications systems deployed worldwide and used in everyday operations for most computer systems.

1.2 The Smart Grid Concept, and Security Concerns

In this section, we will describe the main concept and overview of the Smart Grid. We will review this new concept based on previous research and analyze what are the security concerns that pose a threat into the Smart Grid System.

The Smart Grid (SG) provides an improved electric power infrastructure serving loads and at the same time providing a continuous evolution of end use applications through encompassing the integration of power, communication, and information technologies [13]. Through the use of communications and information systems, a modern and more intelligent power system is available to be deployed.

"The Smart Grid is a complex system made up of interrelated systems" [13]. Data will be generated in large amount. "To manage, store, and effectively use this data, the power system, communications, and information technologies should be coordinated using a system of systems approach; that is, achieve interoperable communications across smart grid technologies."

From [13], we can define interoperability as "the capability of two or more network, systems, devices, applications or components to externally exchange and readily use information securely and effectively".

SG interoperability is the ability to communicate effectively and transfer meaningful data, and it is associated with Hardware/Software components, systems and platforms, Data formats, and interoperability on the content level. The SG interoperability will allow users to acquire equipment readily to be connected with other areas of the SG, and interact with other SG

components. The SG deployment involves using existing systems and protocols to establish inter-device communications like the OSI model. “Cyber security has always been a concern for utility experts” [13]. The main objectives to protect the SG are to secure Confidentiality, Integrity, and Availability. This thesis will focus on the Availability part of security, and how it is affected when the systems deployed are victims of a Distribute Denial of Service Attack. The results obtained for these tests will be presented in further chapters of this document.

In order to better understand the main systems that will be interconnected in the SG, we present the End-to-End SG Communications Model in Figure 1.1.

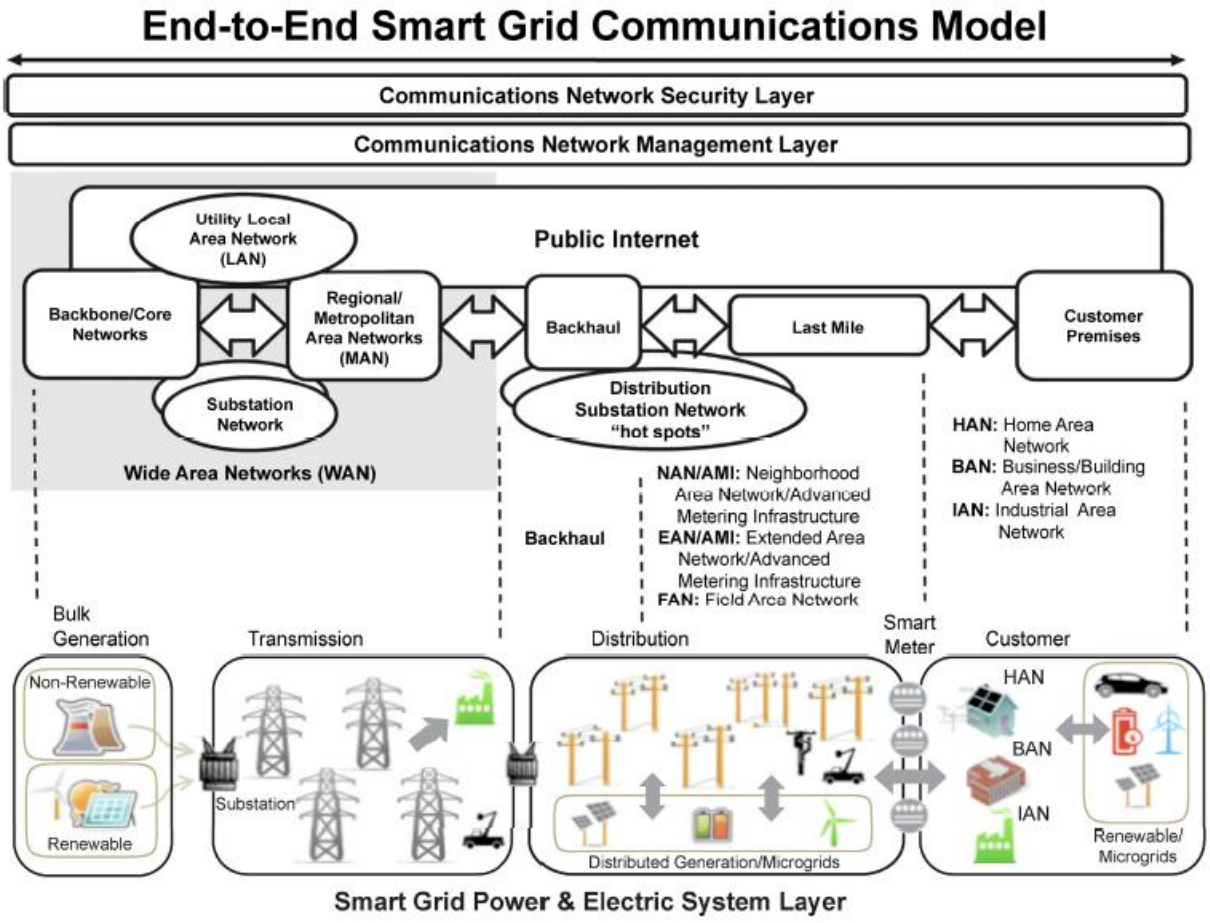


Figure 1.1 - End-to-end smart grid communications model [13]

Diverse systems require being interconnected and exchanging valuable data in order to achieve successful SG interoperability, starting from Bulk and Renewable generation sources, going through Transmission and Distribution phases, and finally being delivered to end users as loads which can go from Home, Business and Industrial Area Networks.

According to [14], smart metering technology will be the foundation of any SG. Through Advanced Metering Infrastructure (AMI), the end users are able to generate measurements and communicate with other stages of the model. Smart meters are a two-way communication interface that will help to achieve SG interoperability. "A smart grid integrates advanced sensing technologies, control methods, and integrated communications into the current electricity grid" [15].

In [16] we find that the SG deployment is an event taking place worldwide, and will continue for several years before it is complete. The power grid should increase their flexibility and compatibility to improve their security and defense capability and self-healing ability continuously. "The SG is the inevitable trend of grid development" [17].

It is important to consider all aspects of data management: collect, store, organize, analyze and share [18]. [19] Presents the characteristics of a real time and historical database in SG technologies, which represent features implemented in the database system we have developed for this thesis, and will be described in chapter III.

With the addition of communication capability to the power grid, the SG becomes vulnerable to security attacks. The existence of cyber vulnerabilities is already present in the power grid [20]. Security of the power grid is a big concern, and the SG is facing a large risk from the cyber-attacks. A ten degree Human-Automation Interaction Framework has been proposed for the SG cyber security [21]. A need to identify SG vulnerabilities has been recognized by Federal

agencies like NIST and the US Department of Commerce [22]. Some additional risks in the grid may be that when the complexity of the grid is increased, new vulnerabilities might be introduced, interconnected networks can introduce common vulnerabilities, the increase of vulnerabilities of communications and the increase of malicious software could result in Denial of Service Attack [20].

One of the major vendors for network technologies presents the schema displayed in figure 1.2, and proposes a security alternative for the successful communication dialog between components in the SG [23]. The main security principles they address are: Access control, Data integrity, confidentiality, and privacy, Threat detection and mitigation, Device and platform integrity.

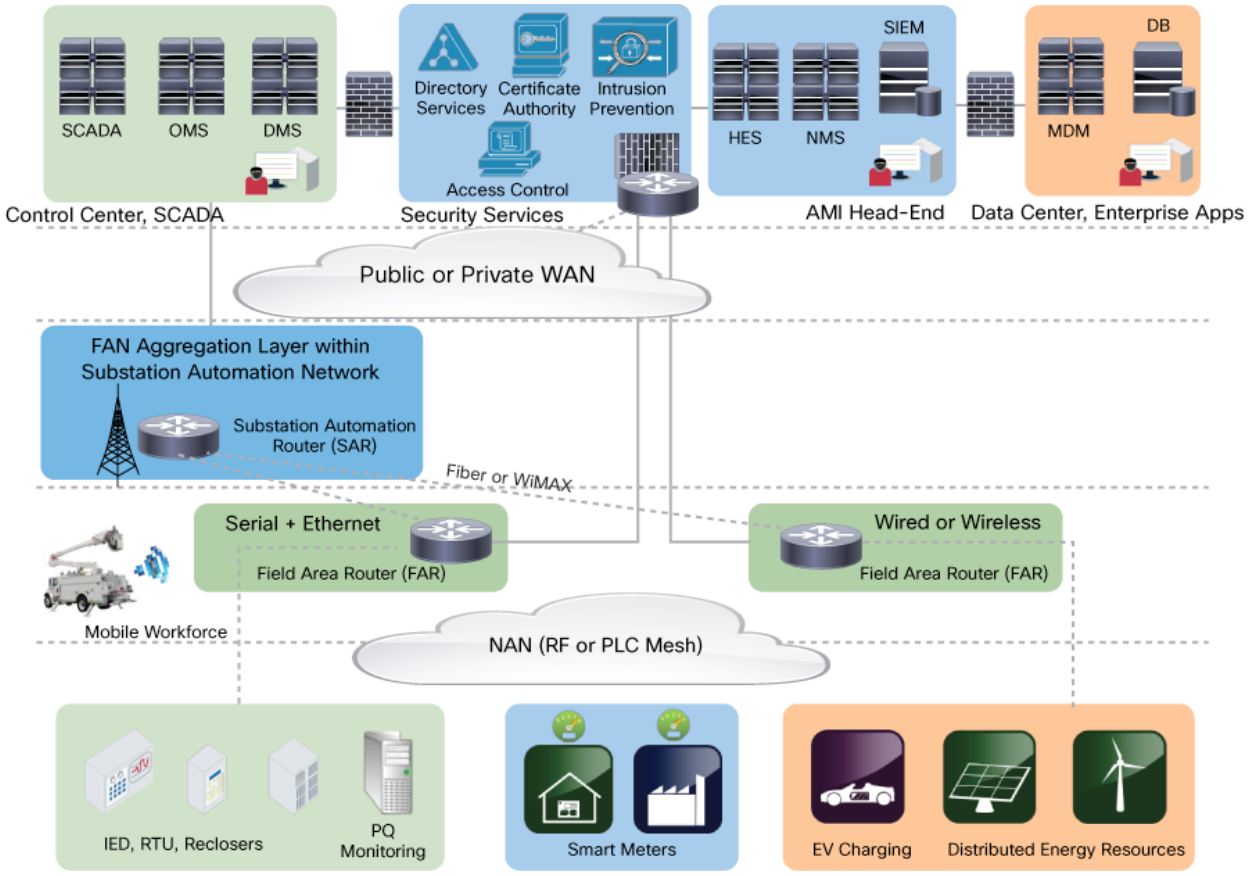


Figure 1.2 - Tiered Communications Network with multi-service FAN and Head-End Components [23]

The same company recommends the use of IP technologies as the SG communication foundation, but they recognize that with this implementation, the smart grid becomes a target for acts of terrorism and cyber-attacks [24].

In our setup we are testing a Micro Smart Grid system and we consider the photovoltaic system we are using in the University of Texas-Pan American as a Micro Smart Grid due to the fact that we have two Renewable Energy sources that are generating data continuously. This information is later processed by our Database System and stored for being accessed on demand basis. We also have a Solar Radiation Lab, which gives us detailed information of the solar resource obtained in our site. A Smart Grid System will be a very large scale system, and have other sources of power generation, smart meters and load monitoring and control phases, thus the System we have at UTPA can be considered as a Micro Smart Grid.

1.3 Statement of Problem

This thesis has its focus on analyzing the impact of Distributed Denial of Service (DDoS) attacks on some services required for the operation of the SG. Vulnerabilities to DDoS are present due to the fact that the SG uses TCP/IP base protocols. Work done in [25-33] make an emphasis on the impact of DDoS attacks to the SG, and how DDoS can affect the information transmission and successful communication of devices required for proper SG interoperability.

Web servers and database servers will be one of the important data components that will be used in deployment of Smart Grid infrastructure, serving millions of subscribers on mass scale. Internet connectivity will play an important role, however it will also bring the security issues inherent in legacy TCP/IP protocols. DDoS attacks can affect availability of data components in

the Smart Grid, making the entire infrastructure unreliable for proper operation. That's why in this thesis we are testing the robustness of Web Servers and Database Servers under DDoS.

Distributed-Denial-of-Service attacks aim on directing a huge amount of traffic coming from diverse sources in different locations to the victim web servers, processing of these packets overwhelms the victim computer resources like CPU, Memory and Bandwidth allocated to host legitimate users. The victim is not able to answer to client requests or it might delay, making the communication slow or completely stopping it from happening, and the web service becomes unavailable to the legitimate users.

Even though DDoS attacks usually do not create a permanent damage to the equipment under attack, they pose a very large threat, and can bring down network communications for the duration of the attack, which as a consequence can cause revenue loss of millions of dollars. In the smart grid systems, this loss of Internet communication can bring down the power grid enabling the attackers to cause even more damage by creating blackouts on the affected areas.

When reviewing DDoS, the seriousness of this threat is further increased by the ease of use of tools available on the Internet, which can be accessed and executed by unsophisticated users without advanced computing skills. The magnitude of attacks is increased due to the large number of unprotected computers available in the network that can be controlled as botnets, and by the increasing speed of communication links that has been deployed nowadays, available to most users, which allows more traffic exchange between hosts.

A lot of research has been done to detect DDoS attacks by using algorithms that analyze the traffic, similarity of the packets received, or by analyzing the source IP addresses, which may be either slow rate or high intensity attacks [34-38]. More work has been done on trace back [39, 40], defense and mitigation mechanisms of DDoS attacks [41-45], despite these, DDoS attacks

are still reaching target hosts and crashing computer systems. Since DDoS traffic is very similar to legitimate traffic, and also because DDoS attacks distribute the traffic from sources all around the world coming from diverse IP address converging onto a single targeted system, it is quite challenging to completely prevent such attacks.

1.4 Thesis Outline

In this thesis we test and evaluate two of the most commonly deployed web server platforms around the world [46] under DDoS attacks. Apache web server is used on 66%, and Microsoft IIS on 18% of all websites, ranking 1st and 2nd respectively, compared to other web servers [46].

For this purpose, our evaluation is focused on testing server platforms like Microsoft Windows Server 2008 using IIS 7, and Red Hat Linux 5 using Apache web server. We will also test and evaluate the performance of the MySQL database used for data collection from the solar systems located in our facilities when they are under DDoS.

These attacks are launched at different transmission rates starting at 10Mbps going up to 1000 Mbps (1Gbps). Therefore, to measure the effects of DDoS attacks, several tests were conducted in Network Research Lab to test the Operating systems. The experimental setup is shown in Figure 4.1 of chapter IV.

This thesis is organized in six chapters. Chapter I is an introduction oriented to give a general idea on the DDoS attacks and Smart Grid systems. Chapter II provides a comprehensive background on some of the most popular DDoS attacks used by malicious attackers. In Chapter III, we present the description of the Data Collection System that we developed, its functions and capabilities for solar information storage and presentation. Chapter IV presents the results gathered through experiments conducted on the Network Research Lab for Windows 2008

Server and Red Hat Linux 5 Sever, when they were under DDoS Attack. The results in this section reflect performance of operating systems and server platforms under DDoS Attack. In Chapter V, we evaluate the MySQL database performance when it is exposed to the same network attacks. In Chapter VI, we conclude this thesis and suggest possible future work.

CHAPTER II

DISTRIBUTED DENIAL OF SERVICE ATTACKS

In this chapter, we first give a brief description of what a Distributed Denial of Service Attack (DDoS) is. Then we will review the types of DDoS attacks and methodology used in what is considered a kind of Cyber-terrorism [47].

A "Denial-of-Service" (DoS) attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Some examples may include: attempts to "flood" a network, attempts to disrupt connections between two machines, attempts to prevent a particular individual from accessing a service, and attempts to disrupt service to a specific system or person [48].

A Distributed Denial of Service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets [49].

A DDoS Attack is composed of three main steps: Design a piece of mobile code, which attack(s) one or more preset targets upon receiving a trigger. This code then ‘infects’ a number of weakly secured locations throughout the Internet. When the code is triggered it then starts an attack from all or some of these infected locations [47].

We can now see that scanning is the first step to exploit any system. The attacker first recruits the machines that have some vulnerability. As the second step we can say propagation deals with recruiting further machines with the help of already compromised machines. Finally,

as third step the communication channel is important for coordinating an attack. In Agent-Handler Modal communication can be done by using TCP/ICMP/UDP protocol between attacker to handler, handler to agent and vice versa [50].

DoS attacks can be classified into five categories based on the attacked protocol level, as illustrated in Figure 2.1 [51].

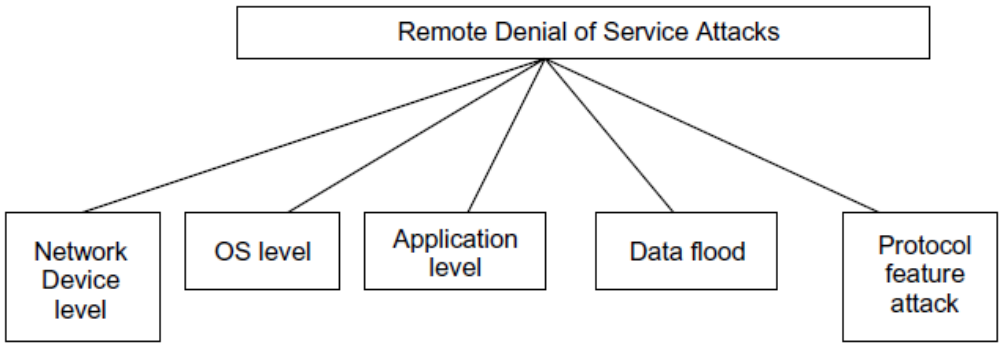


Figure 2.1 - Classification of Remote DoS Attacks [51]

DoS attacks at the *Network Device Level* include attacks that might be caused either by taking advantage of faults or weaknesses in software or by trying to exhaust the hardware resources of network devices. One example of a network device exploit is the one that is caused by a buffer overrun error in the password checking routine. The *OS level* DoS attacks take advantage of the operating systems implementation of the protocols. *Application-based attacks* try to set a machine or a service out of order either by taking advantage of specific faults in network applications that is running on the target host or by using such applications to drain the resources of their victim. In *data flooding attacks*, an attacker attempts to use the bandwidth available to a network, host or device at its greatest extent, by sending massive quantities of data and thereby causing it to process extremely large amounts of data. An attacker could attempt to use up the available bandwidth of a network by simply bombarding the targeted victim with

normal, but meaningless packets with spoofed source addresses. DoS *attacks based on protocol features* take advantage of certain standard protocol features. For example, several attacks exploit the fact that IP source addresses can be spoofed. Several types of DoS attacks have focused on DNS, and many of these involve attacking DNS cache on name servers [51].

A variety of DDoS attacks and a comprehensive classification has been provided in [52]. In the first level, attacks are classified according to their degree of automation, exploited vulnerability, attack rate dynamics and their impact. In the second level, specific characteristics of each first level category are recognized as illustrated in Figure 2.2. Many efforts have also been made to defend against DDoS attacks and some of the approaches include IP Trace back methods [41], hop count filtering [53], multiple data sources analysis [54], interpretation of behavioral models [55], egress filtering [56, 57], ingress filtering [58], disabling unused services [59], anomaly detection using intrusion prevention system (IPS) [60], and many others.

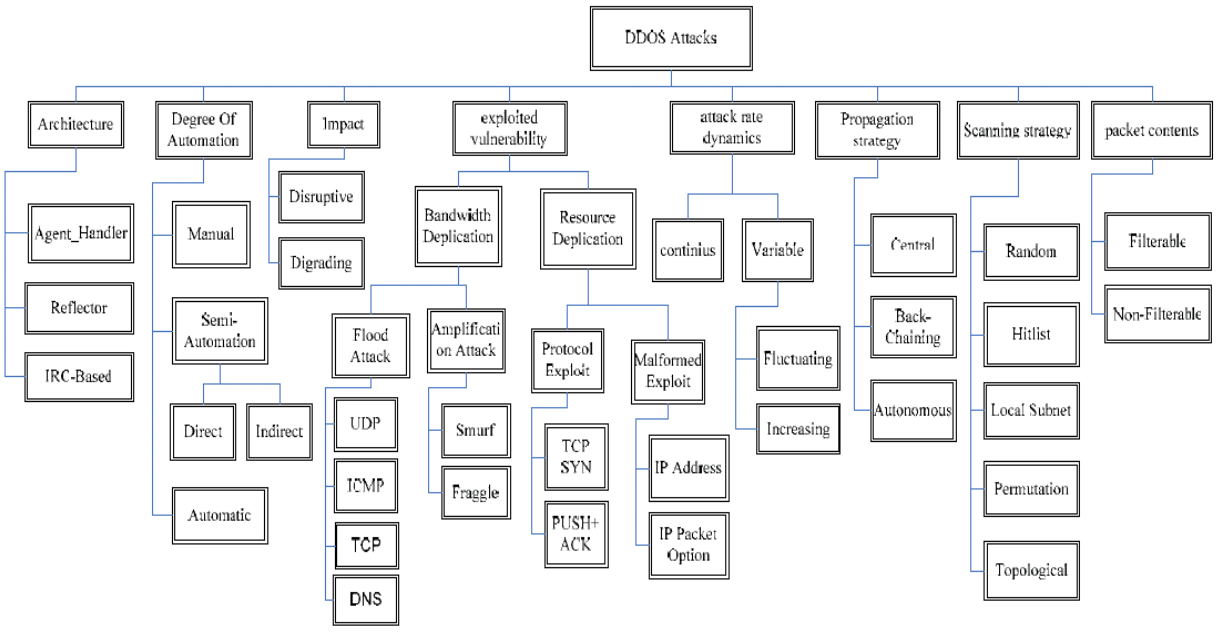


Figure 2.2 - DDoS Attacks Comprehensive Classification [52]

In Figure 2.3, an illustration is shown representing how the attacker issues a DDoS [61].

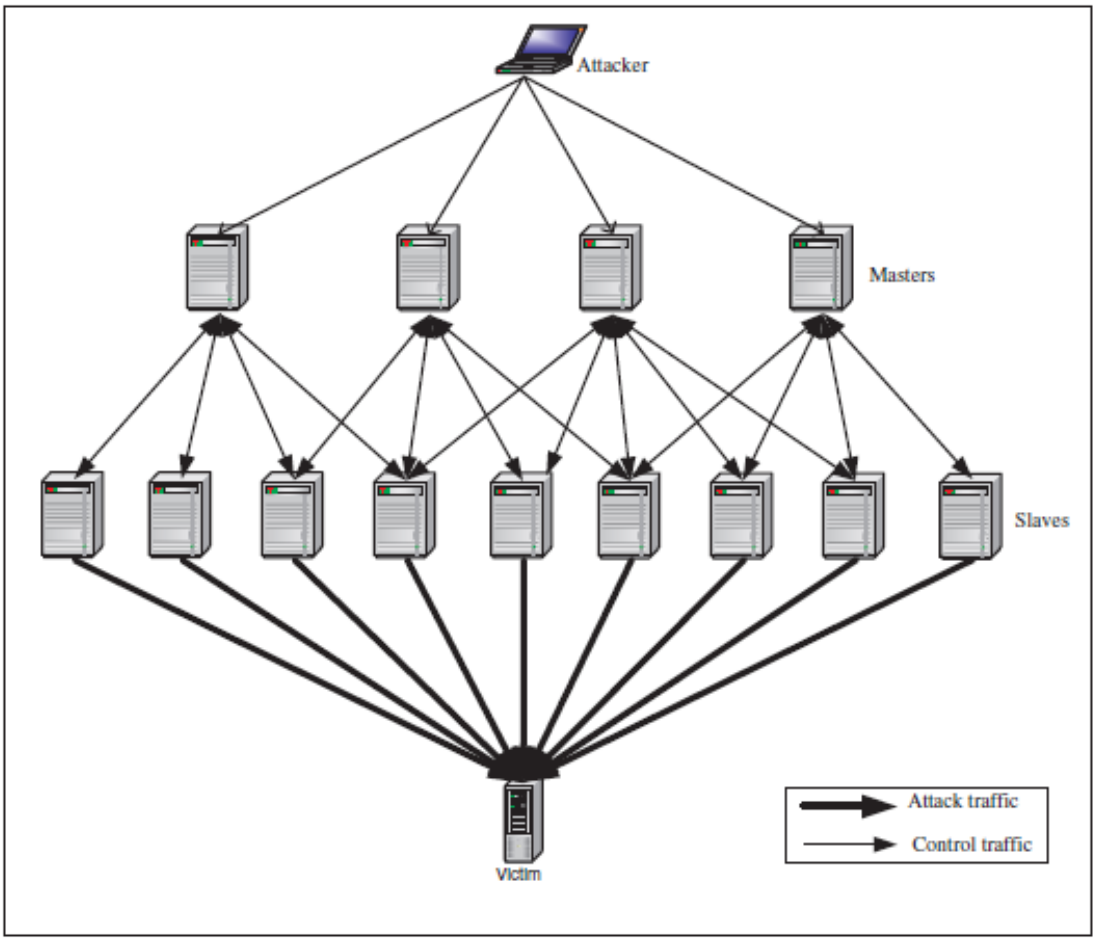


Figure 2.3 - DDoS Attack Architecture [61]

The attacker starts a session with the master host. The daemon process, launched in the master host, offers many commands facilities to the attacker in order to launch the desired flooding attacks. After establishing a connection with the master, the attacker may launch a flooding attack against one (or many) victim(s) using the different commands offered by the master. The master receives the commands through the established connection with the attacker and sends the corresponding commands to the slave(s) using the specified communication with

the slave. Hence, the slave floods the victim with attack traffic using the specifications received from the master.

For the research experiments presented in this thesis, we will be using mainly *Flood Attacks* using protocols such as UDP, ICMP, and TCP, *Amplification Attacks* like Smurf, and the *Protocol Exploitation Attacks* using TCP-SYN as attack mechanisms to evaluate the performance of the victim when it is under attack.

2.1 Background Study on Different DDoS Attacks

In this section, background information about different Distributed Denial of Service attacks is considered and explained according to the Protocol Layer in TCP/IP suite.

2.1.1 ARP Flood Attack

The ARP-based flooding is a Layer-2 attack as it usually happens in the Local Area Network. Address Resolution Protocol (ARP) is used in Local Area networks to resolve IP addresses into hardware MAC (Media Access Control) addresses. The ARP request message consists of the IP address of the host, IP and hardware MAC address of the initiator who wish to communicate and broadcasts within the LAN, the packet format of which is shown in Figure 2.4. All the hosts in the LAN receives the ARP request but only the host who has the Target IP in that packet will respond and unicast the initiator its hardware MAC address. Also the ARP cache table of receiver host will be updated with the corresponding IP-MAC addresses for further communication with the initiator [61]. Attackers take advantage of this protocol and try to flood the end host with ARP Requests and the host becomes busy as it replies to those requests and updates its cache table. With a flood of such requests, resource starvation usually happens on the

host computer. Those resources can be either processor consumption or memory. One general way of DDoS is to storm the host with a barrage of ARP requests thereby incurring a DDoS attack on the host while being consumed in replying to all the requests it receives and exhausts the system resources. The impact of such ARP storm was studied in [62] where it exhausted the resources of the victim system under attack.

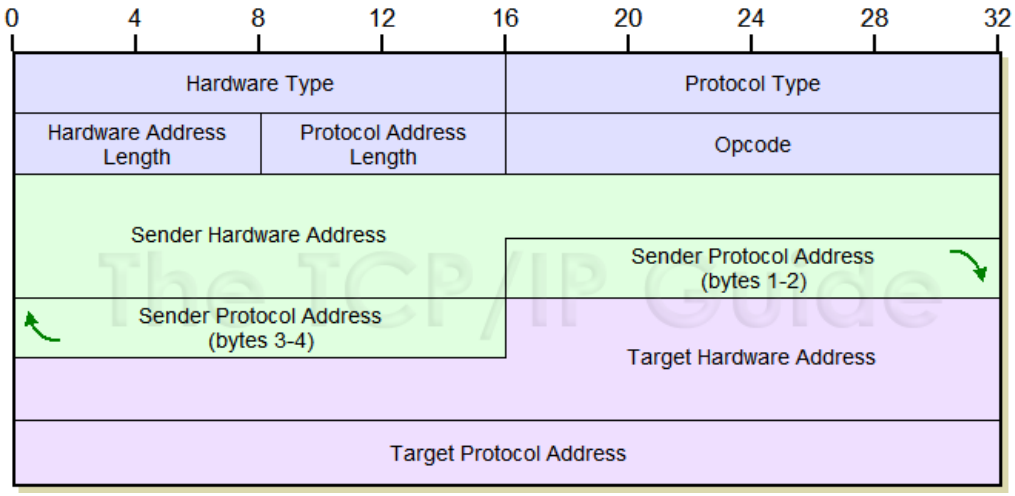


Figure 2.4 - Address Resolution Protocol (ARP) Message Format [63]

2.1.2 ICMP based DDoS Attacks

Internet Control Message Protocol (ICMP), RFC 792 [64], based DDoS attack is launched by sending a barrage of corresponding attack request packets by the compromised systems to the target computer. ICMP messages are used when a gateway or a host wants to communicate with another host to report to the problems like delivery error, connectivity problems, address mask discovery, diagnostics etc. The purpose of such diagnostic messages is to provide information about the problems in the network or communication environment. There are different types of ICMP message formats and each is defined for a specific purpose. Attackers take the advantage

of these diagnostic packets to create a Denial of Service on a target host. Depending on which ICMP packets are used to create the Denial of Service on the target we have those many types of DDoS attacks, but those that are used here are the Ping Flood, Smurf and ICMP Land Attacks.

2.1.2.1 Ping Flood Attack. Ping is a type of ICMP message that is used to know the availability of a host across the network. Based on RFC 792 [64], ICMP Echo request must be replied with an ICMP Echo Reply message. Attackers take advantage of this protocol and try to flood the end host with Ping Requests and the host ultimately replies to those requests and hence consumes the computer resources. With a flood of such requests, resource starvation usually happens on the host computer. The attacker, generally, spoofs the source IP and sends a barrage of Ping requests to the victim computer. Flooding of echo request messages with spoofed source IP address to the victim causes denial of service which is called ping based denial of service attack. The victim computer tries to respond with echo reply message for every echo request received, which requires significant Processing, Storage and Bandwidth. The victim computer incurs Denial of Service while being consumed in replying to all the requests it receives. This Ping Flood Attack is a Layer-3 attack in the TCP/IP suite. Ping attack is reported in [4] to bring down the whole Internetwork by attacking the root DNS servers. One of the earlier work shows that a simple Ping attack can make the target host busy in processing the ping requests consuming 100% of the CPU utilization [65].

Different methods for detecting the DDoS attacks are proposed in [66 - 72], where most of them depend on some selected nodes in the network to detect the attack. So the defense of DDoS attack in the network using these methods depends significantly on end systems capability of withstanding to DDoS attacks.

2.1.2.2 Smurf Attack. Smurf is another Layer-3 attack. Smurf attack is a type of DDoS attack where an attacker exploits unprotected computers on Internet to direct a flood of ICMP echo-reply messages towards the victim computer. Smurf attack primarily exploits the ICMP messages that are diagnostics tools frequently used to troubleshoot the problems in the network. In Smurf attack both the ICMP echo request and ICMP echo reply messages are used. While the perpetrator sends ICMP echo request messages to an unprotected broadcast domain for amplifying the attack, the victim computer actually receives amplified attack traffic that comprises mainly of ICMP echo reply messages. In Smurf based flooding attack, a large amount of ICMP echo messages i.e., Ping messages are sent to broadcast addresses, and where the Ping messages contain the spoofed source address of the victim computer. Each host of the broadcast domain receives an ICMP echo message, and responds to it by sending ICMP echo reply. In effect, the broadcast domain helps amplify and direct the DDoS attack traffic towards a victim computer and it is flooded with a large number of ICMP echo reply messages resulting in bandwidth exhaustion and also the resource exhaustion of the victim computer. The amplification of such attacks in the internet is considered to be very dangerous on the victim's computer and effect of such amplification can be seen in [73]. According to the work in [74], service packs and patches released by Microsoft are also not able to mitigate the Smurf attack completely.

2.1.2.3 ICMP Land Attack. This is another Layer-3 attack where the ICMP ping request packet is spoofed with destination IP host/port address same as victim IP address. When a barrage of such Land attack packets are sent, the host becomes busy replying to its own IP address and results in system lockup. This vulnerability was found in Windows XP with SP2 service pack and also Windows Server 2003 with firewall turned off. These systems are found

vulnerable for the LAND attack, which caused a temporary Denial of Service (DoS) that lasts for 15 to 30 seconds. In case of windows Server 2003 not only the server but also all workstations on the network froze [75]. A similar testing was done on Windows XP, Vista and Apple's Leopard OS, where it was found that the Windows Vista crashed under ICMP Land attack load of 30Mbps [76].

2.1.3 TCP-SYN Flood Attack

TCP flood attack is Layer-4 attack, which is one of the popular Denial of Service attacks that exhausts the system resources and brings many serious threats to the entire network. The July 4th attack on U.S. and South Korean government websites was a TCP-SYN attack [77-78]. The host retains many half open connections and there by exhausts its memory and processor utilization. The Transmission Control Protocol (TCP) layer provides a three-way handshake process for any connection establishment. When a client initiates the TCP connection, it sends a SYN packet to the server and then the server responds with an SYN-ACK packet and stores the requested information in memory stack. After receiving the SYN-ACK packet the client confirms the request by sending an ACK packet. When the server receives the ACK packet it checks in the memory stack to see whether this packet corresponds to previously received SYN. If it is, then the connection is established between the client and the server and data transfer can be started. In TCP-SYN Flood attack, the attacker sends a barrage of SYN packets with spoofed IP address to the server and the server stores that information in the memory stack, sends the SYN-ACK and waits for the final ACK, which never arrives. If large amounts of SYN attack packets are sent then a Denial of Service attack can happen on the victim computer due to resource exhaustion.

There are many methods suggested to fight against this TCP-SYN attack [76, 79-81] and also service packs that can mitigate the DDoS [82].

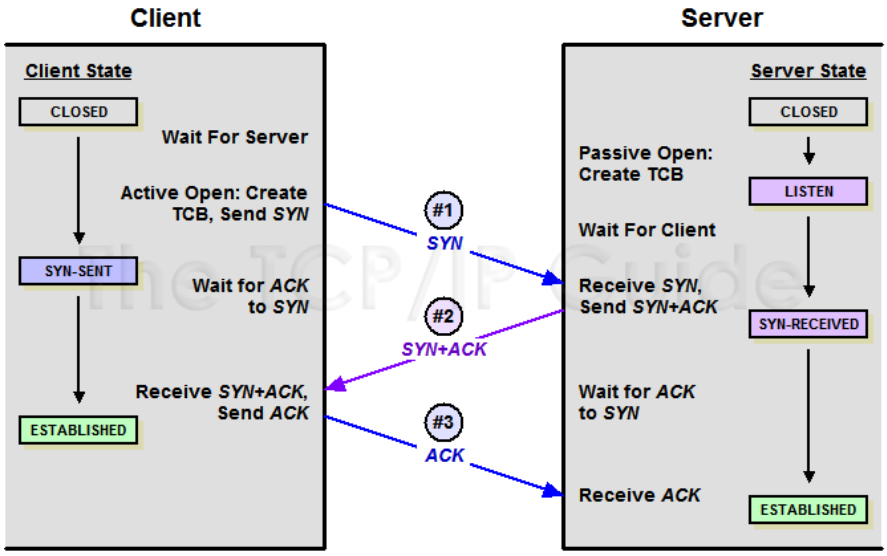


Figure 2.5 - TCP "Three-Way Handshake" Connection Establishment Procedure [83]

2.1.4 UDP Flood Attack

UDP Flood attack is a Layer-4 attack. UDP Flood vulnerabilities have been discovered during the year 1998-2000 in many systems including Microsoft products. In UDP Flood attack a barrage of UDP packets are sent to the victim computer either on specified ports or on random ports. The victim computer processes the incoming data to determine which application it has requested on that port and in case of absence of requested application on that port, the victim sends an "ICMP Destination Unreachable" message to the sender which is usually a spoofed IP. If such a barrage of requests were sent then it results in Denial of Service on the victim computer as the victim will become busy in processing those packets and sending ICMP Destination Unreachable messages. UDP flood attacks may also deplete the bandwidth of network around the victim's system. For example, by sending UDP packets with spoofed return addresses, a hacker

links one system's UDP character-generating (chargen-Port 19) service to another system's UDP echo service. As the chargen service keeps generating and sending characters to the other system, whose echo service keeps responding, UDP traffic bounces back and forth, preventing the systems from providing services [84].

2.2 Chapter Summary

In this chapter, we have reviewed the concept and background of Distributed Denial of Service Attacks. We have given a small overview of the most common types of attacks and how they take advantage of the TCP-IP protocol stack that has been previously defined within the Network Communication Protocols.

We have also reviewed the harmful capabilities found in these types of attacks, that even with very sophisticated defense mechanisms; they can bring down the performance of big server platforms around the world.

The attacks methods defined in this chapter will be used in further chapters to measure the actual impact on specific test platforms.

CHAPTER III

PHOTOVOLTAIC DATA COLLECTION SYSTEM

In this chapter, we described the database system created to store the information and generate customized reports regarding power production, and efficiency evaluation of the photovoltaic arrays on campus. This software has been developed for the purpose of analyzing and comparing the amount of power received and generated throughout a historical record which is updated on regular basis.

The database we have created can be used as an Online Analytical Processing or OLAP Relational Database developed using MySQL, and also as an Online Transaction Processing or OLTP Relational Database, depending on the activity that is being executed. A graphic user interface or GUI has also been created using C++. It is used to enable connectivity and query into the stored information.

In order to understand how this software works, and how the Solar Data is gathered, we need to describe the Photovoltaic Arrays and Solar devices located on Campus.

3.1 Solar System Description

We are able to gather solar energy from two Photovoltaic Arrays: The ENGR PV Array and TXU Array. The ENGR PV Array is a fixed array containing 24 solar panels that sum 5.184 kW power with a configuration of 216 W x 24. It has a Collector Azimuth angle of 11^{o} from the

south, and a Tilt Angle of 10^{o}. The TXU Array has a Solar Tracking System to increase the power efficiency production; it contains 2 solar trackers with 2 degrees of freedom each. Each tracker contains 12 solar panels, and their maximum rated power is 2.75 kW, summing up a total of 5.50 kW [85, 86].

The ENGR PV Array shown on Figure 3.1 generates averaged measurements every five minutes, whereas the TXU Array on Figure 3.2 is recording readings every fifteen minutes that are averaged every hour. This information is later used to populate our database records enabling the generation of customized reports, which include energy (kWh) and power (kW) measurements, efficiency calculation, and duration of the day. Both for a user selected period of time input containing start and finish date.

Figure 3.1 - ENGR PV Array at UTPA Engineering Building rooftop

A Solar Radiation Lab displayed on Figure 3.3 was recently installed on the Engineering building rooftop; this device gives us the capability to calculate the amount of power delivered by the sun throughout the day per square meter. The Solar Radiation Lab contains a Solar Tracker System able to collect measurements of direct normal irradiance (DNI), diffuse

horizontal irradiance (DHI) and global horizontal irradiance (GHI); all of these measurements are in units of watts per square meter [86].

Figure 3.2 - North Half of the TXU Sun Tracking Arrays at UTPA

Figure 3.3 – Solar Radiation Lab. Two pyranometers, one pyrheliometer, and the tracker

3.1.1 ENGR PV Array System Description

A more detailed review of the ENGR PV Array is provided in this section. The diagram that describes the ENGR PV Array System is shown in Figure 3.4. Sunlight is collected through a 5.184 KW peak power Photo Voltaic Cells Array consisting of 24 panels. Then in the form of electricity, it is taken to the Sunny Boy Inverter in a DC Voltage stream that oscillates between 200 to 400 Volts, depending on the hour of the day and the solar intensity that is being received. DC power is processed and directed into the array of batteries where it can be stored for backup power when sunlight is not available or not enough to satisfy the demand.

The Sunny WebBox is a communication device that gathers Power Data through a built-in function via RS-485 communication, which is connected to the Sunny Island and to the Sunny Boy Inverters.

The Sunny WebBox has a static IP Address, and it is connected to the local network via Ethernet communication, this way, power information is forwarded through Internet to the website www.sunnyportal.com, as shown in Figure 3.4 [87].

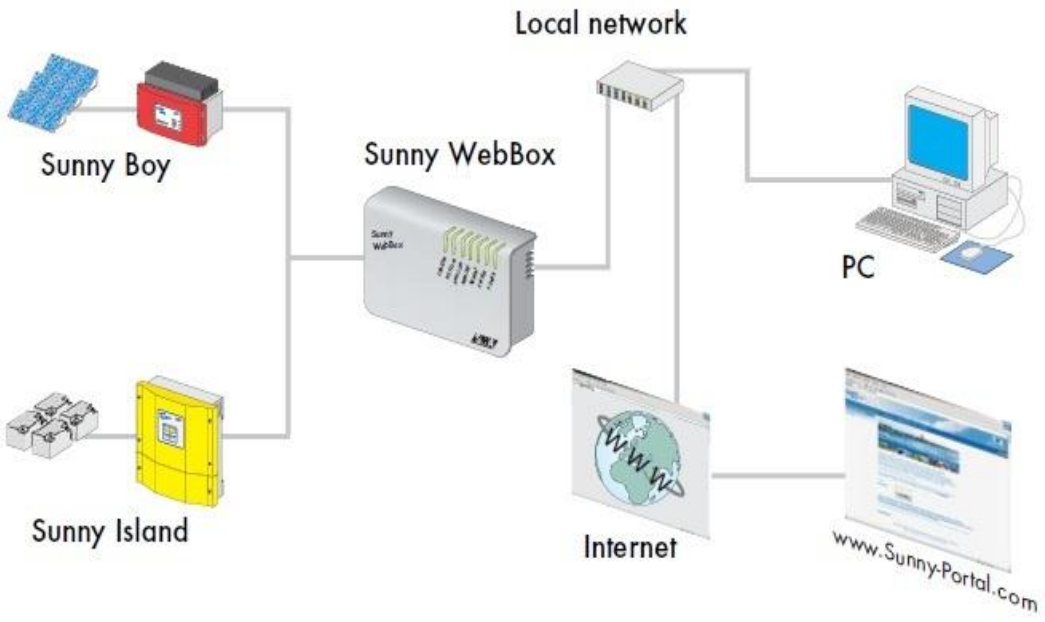


Figure 3.4 - ENGR PV Array System Logic Diagram

An alternative source of Power Data can be also collected through a computer connected to the local network using any web browser by establishing an http connection directly with Sunny WebBox.

3.1.2 TXU Sun Tracking Arrays System Description

The TXU Sun Tracking Arrays System is displayed in figure 3.5. This System is composed of two Solar Arrays with two axis trackers each. Every array has a maximum rated capacity of 2.760 kW with a configuration of 230 W x 12 summing up a total of 5.520 kW for both arrays. Each array of 12 solar panels is connected to a DC to AC Inverter and redirected to a 208 VAC Panel. Here there is a power meter connected measuring the power production on real time. This

information is later forwarded to the internet using the Local Area Network, and stored in a private database from the solar panels vendor. The power production information can be found on the following link: http://siteapp.fatspaniel.net/siteapp/simpleView.jsf?eid=553237 [88].

From the panel, the AC voltage is connected to a transformer and elevated to 480 VAC. This energy is later directed to the International Trade and Technology building in the University of Texas-Pan American, where it is used by the loads contained in this location.

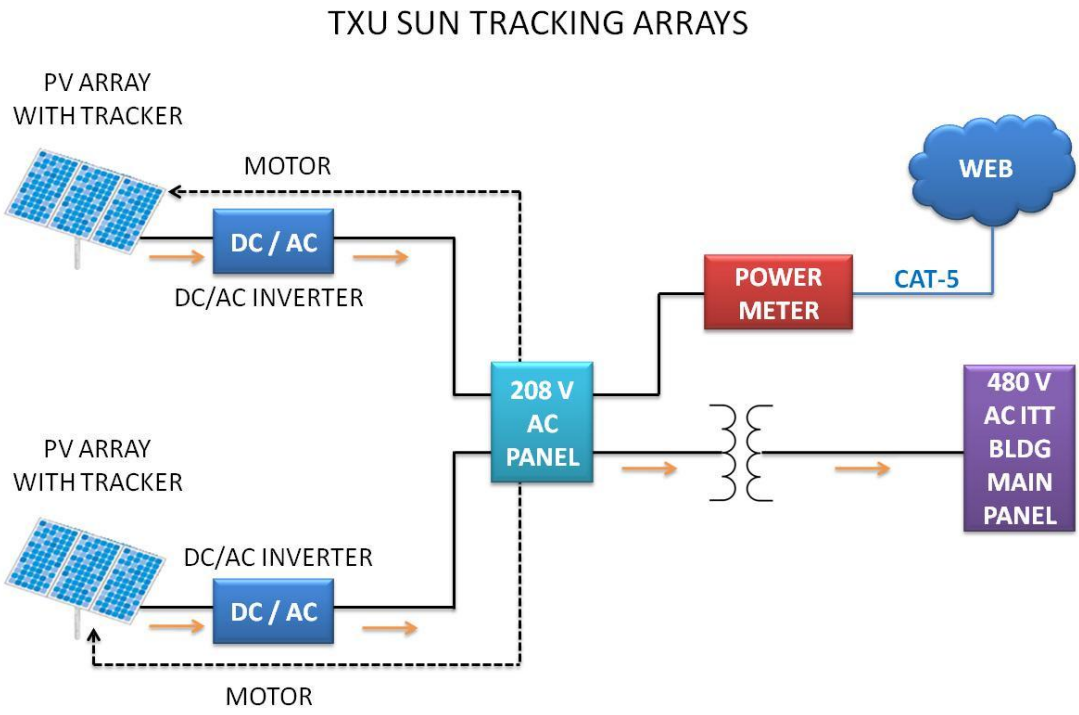


Figure 3.5 – TXU Sun Tracking Arrays System Logic Diagram

3.1.3 Solar Radiation Lab System Description

Now we will describe the Solar Radiation Lab or SRL located in the Engineering building rooftop of the University of Texas-Pan American. Figure 3.6 displays the SRL Logic Diagram. The SRL is composed of a 2-Axis GPS Solar Tracker named Solys2, which has the function to follow the sun in a very precise way. This Tracker has three sensors mounted on it that are

described as S1, S2 and S3 in fig 3.6. They are intended to measure Direct Normal Irradiance, Global Horizontal Irradiance and Diffuse Horizontal Irradiance respectively. S1 is a pyrheliometer while S2 and S3 are pyranometers. The measurements taken by these three sensors are later forwarded to the Campbell CR-1000, a data logger that forwards the information to the internet and can be accessed in the following site: http://www.nrel.gov/midc/utpa_srl/ [89]. This information is also forwarded to a PC that is connected to the same network to capture these measurements and forward them to our database, which will be described in the following sections of this chapter.

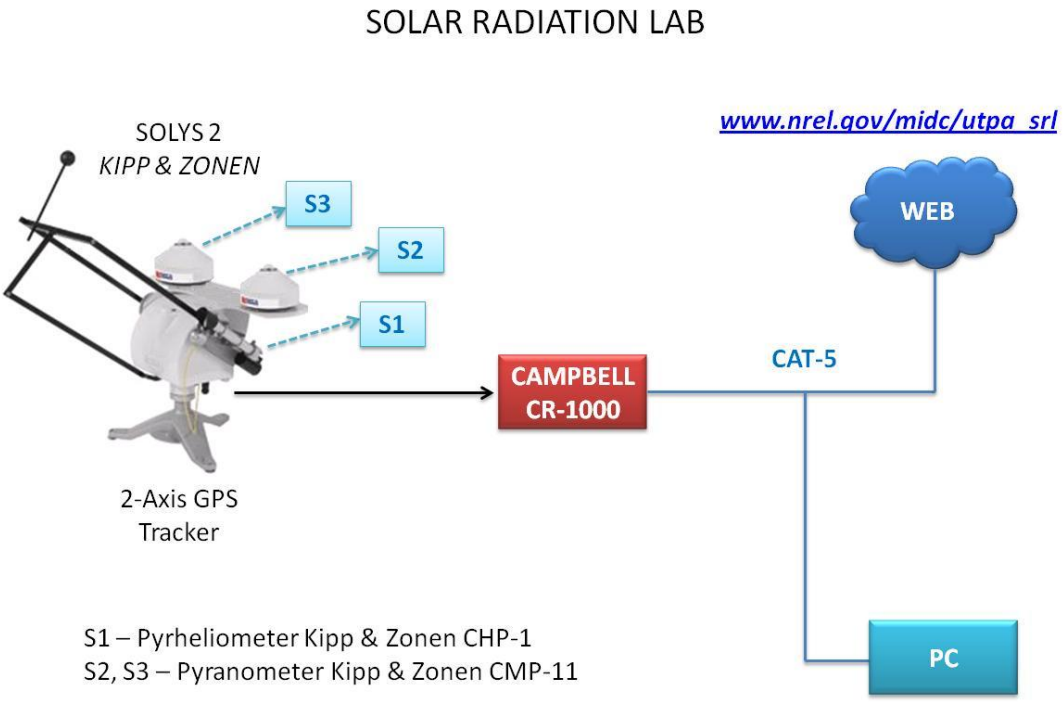


Figure 3.6 – Solar Radiation Lab System Logic Diagram [89]

Now that the Solar Systems have been described, we will review in the next section, how the database interacts with these devices and stores the generated Solar Information.

3.2 Solar Data Collection System

One of the main concerns when this Software was developed is to provide with a system capable of creating historical records of the power generated through solar energy in the University of Texas-Pan American, which can later be accessed for research and educational purposes. A second objective for the creation of this platform is to simulate a Data Collection System used in the smart grid systems, in order to test its vulnerability when a cyber-attack such as a DDoS is launched towards it.

A typical smart grid communication system, as illustrated in Figure 3.7, is a horizontal integration of one or more regional control centers, with each center supervising the operation of multiple power plants and substations. A smart grid communication system has a layered structure and performs data collection and control of electricity delivery. A regional control center typically support metering system, operation data management, power market operations, power system operation and data acquisition control. Substations contain Remote Terminal Units (RTUs), circuit breaker. Human Machine Interfaces (HMIs), communication devices (switches, hubs, and routers), log servers, data concentrators, and a protocol gateway. Intelligent Electronic Device (IEDs) are field devices, including an array of instrument transducers, tap changers, circuit re-closers, phase measuring units (PMUs), and protection relays [90].

According to IEEE 2030-2011 standard [13], the End-to-End Smart Grid communications model is making use of the Internet to establish a path for communications for the different domains of the power systems, which go from bulk generation to transmission and distribution, getting to the end customers through smart meters. Diverse kind of Area Networks are interconnected and communicated through the Internet and require this resource for proper operation of the Power Grid.

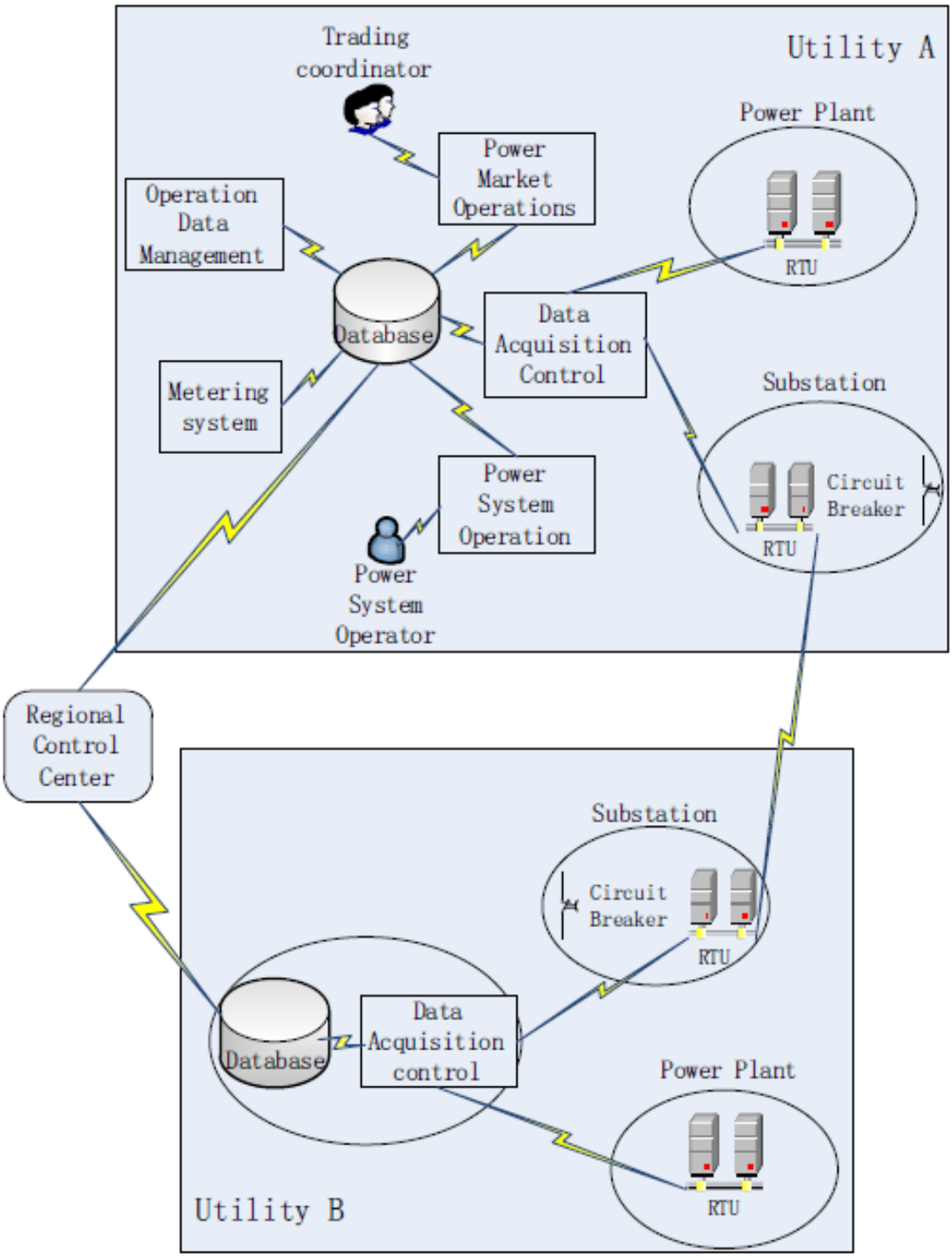


Figure 3.7 - A Smart Grid Communication System [90]

As we can observe, the database element is present in every subsystem of the Smart Grid. Since the communications standards are executed using diverse protocols including the TCP/IP suite in the IEC 61580 standard [91], and with the benefits of using the internet and the networks already established, the Smart Grid is also inheriting its vulnerabilities. From our point of view, MySQL Servers and Web Servers play an important role on SG, and they can be subject to cyber-attacks. These concerns will be further reviewed this thesis.

The "UTPA - PV Database System", as this software is named, offers an authentication scheme to regulate which users are able to access it, shown in Figure 3.8.

Once the user has correctly logged in into the system, the Main window is displayed, illustrated on Figure 3.8. It shows four main function buttons that are Import, Export, SRL, and Efficiency.

The Import button allows adding new records into the database; this is a task that needs to be executed on regular basis in order to keep consistency of the stored information with the actual gathered measurements from the solar devices.

The Export button displays a new interface that allows the user to review stored solar information from the ENGR and TXU arrays, with the option to export it to an excel file or to a bitmap in the format of a plot. This window also calculates the duration of the day using data that was generated from the ENGR Array.

With the SRL button, the user gets access to the irradiance information generated by the Solar Radiation Lab (SRL) for a specific day selected.

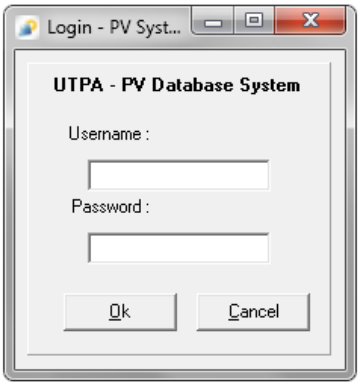


Figure 3.8 - Login Window of “UTPA - PV Database System” Software

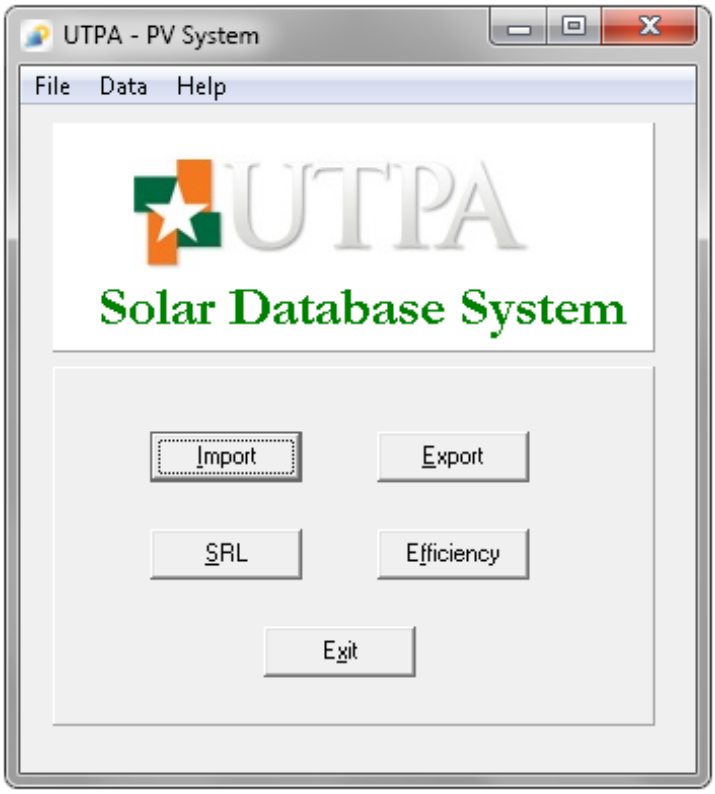


Figure 3.9 - Main Window of “UTPA - PV Database System” Software

3.2.1 Description of Import Data Form Window

In this section, we will review the functionality in depth of the Import Function briefly described previously. On Figure 3.10 we can observe the illustration of the Import Data Form window.

The information that will be imported has been previously generated by the web services of the Photovoltaic Systems described in previous section in the format of comma separated text files. The user is able to select the destination table where the information will be stored. The four options that can be chosen are: ENGR PV Array, TXU Array, Campbell Minute, and Campbell Hourly.

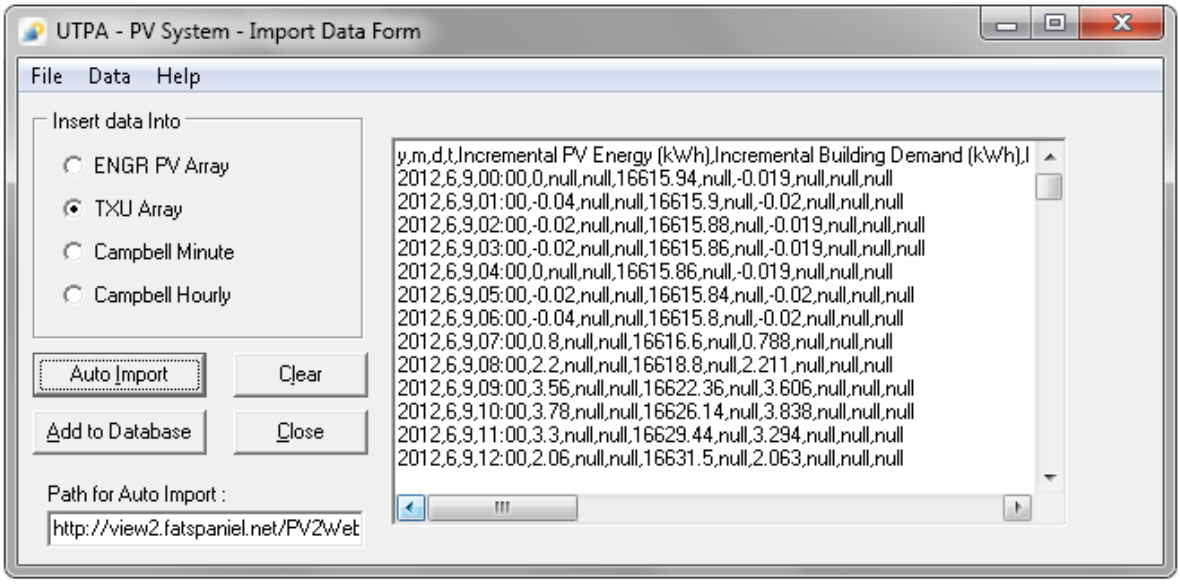


Figure 3.10 - Import Data Form Window

The ENGR PV Array table receives the files generated from the Sunny WebBox. These files have to be manually downloaded from the WebBox using a built-in web service of this device through an Ethernet connection on the same network. Once these files are gathered they are inserted using the option Open File or Open Multiple Files under the Data menu, depending on

the number of files to be imported. Then the Add to Database button is clicked and the files will be added to the corresponding table from the database. The Auto Import button allows the user to import the latest generated files automatically, but this option is not available for the ENGR PV Array.

The TXU Array option can be chosen, and the files can be added using the Auto Import button or manually using the same method described for the ENGR PV Array. If the automatic option is chosen, a real time file containing records for the last month will be downloaded from the TXU Website [88] and imported into the corresponding table from the database.

For the data gathered from the Solar Radiation Lab, we have two options; Campbell Minute, and Campbell Hourly. This is because the data provided by the Campbell logging interface generates measurements every minute and every hour, which are recorded on separate files. This gives a specific resolution of the data depending on the application that will be using it. Files can be imported using both manually and automatic options for these two tables. Since the files required for the Auto Import execution are located on the localhost hard drive, it is required that the Campbell software has been installed on it, and the records updated previously to performing this operation in order to get the most recent data.

The Import Data Form also contains a text display that shows the contents of the file when a single file is selected, or the name and location if multiple files are selected. In the illustration 3.10 an example is displayed for the case when the TXU Array is selected and the Auto Import option is executed.

3.2.2 Description of the Export User Interface Window

The Export user interface offers the capability to create graphical reports containing Max Power (kW) and Energy (kWh) information from the TXU and ENGR Arrays. It enables the user to compare in a quick way the daily photovoltaic power production.

The users first needs to select a period of time defined by a start date, and a finish date. As the following step, it needs to be selected what measurements are wished to be displayed on the graph, the options are Energy and Max Power. The user has the flexibility to choose either one of them or both at the same time.Finally the array containing the information needs to be selected. The same capability to choose one or two arrays is enabled in this section.

Once the user has selected the measurements that will be displayed, the Execute button has to be clicked and the information is displayed in two bar charts, one for the Energy and Power to be displayed, and the second one for the duration of the day throughout the selected time period. A text panel displays all this information allowing the user to view exact values.

The user has the capability to export the generated report information to an excel file, or to generate a bitmap image of the chart containing power and energy data. To perform this operation the Export menu should be selected and it will display two options: “Excel File (*.xls)”, and “Graph (*.bmp)”.

In the case the user selects an interval of time that is not available in the database, those records will not be displayed in the graphs or text window. The user has the possibility to import the missing records in order to get complete data from this form.

Figure 3.11 - Export Data Form Window

3.2.3 Description of the Solar Radiation Lab Window

In this form displayed in Figure 3.12, the user can access and export the information that has been stored regarding the Solar Irradiance received at UTPA that we have obtained on a previous selected day.

The Solar Radiation Lab (SRL) displayed on Figure 3.3, measures three different components of solar irradiance in watts per square meter. For the first it uses a pyrheliometer and measures the Direct Normal Irradiance. This measurement is abbreviated as DNI and is the

amount of solar radiation from the direction of the sun. The second is measured used a pyranometer and is called Diffuse Horizontal Irradiance or DNI. It is the radiation component that strikes a point from the sky, excluding circumsolar radiation. In the absence of atmosphere, there should be almost no diffuse sky radiation. High values are produced by an unclear atmosphere or reflections from clouds. The third measurement is done also by a second pyranometer and is called Global Horizontal Irradiance or GHI. The global radiation is said to be the sum of direct and diffuse radiation [92]. An illustration showing these components from the sun is displayed in Figure 3.13.

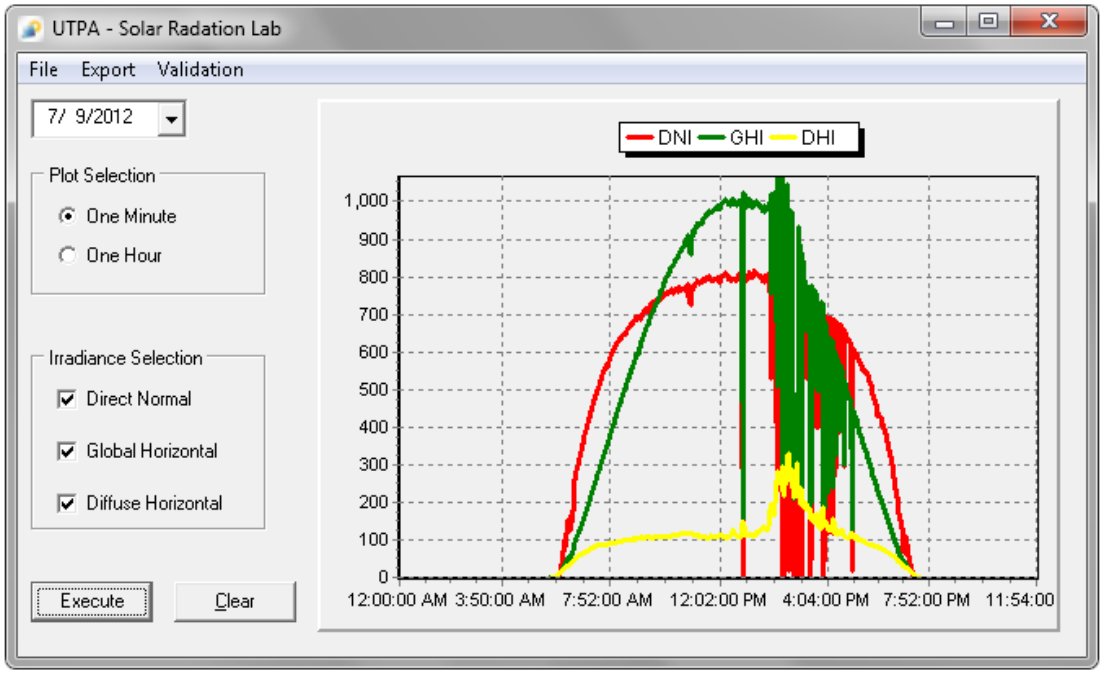


Figure 3.12 - Solar Radiation Lab Window

The user is able to select which measurement desires to be displayed, or any combination of the three. The plot selection option allows the user to choose if the data that will be displayed was sampled every minute or every hour. In the hourly mode selection the information shown represents the averaged samples for every 60 minutes.

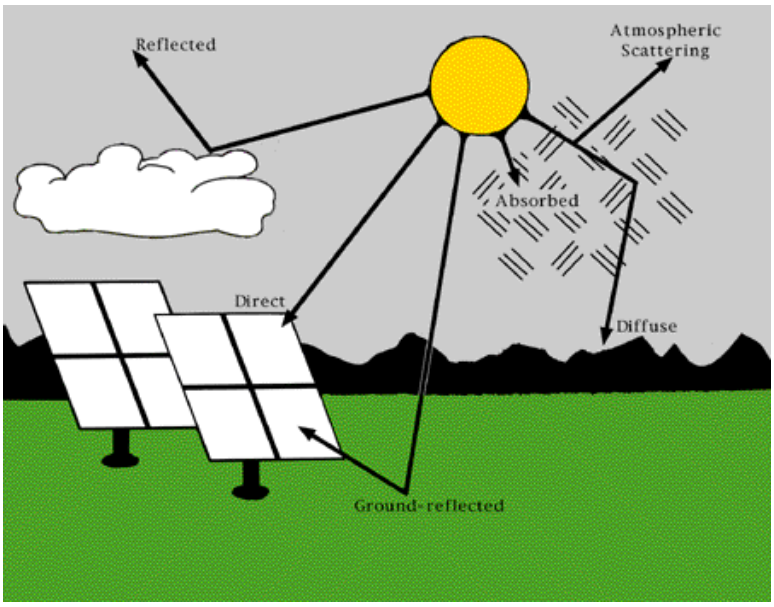


Figure 3.13 - Components of Solar Radiation [92]

The information gathered on this form can be exported to an excel file selecting the Export menu then the option Export to Excel. A file will be created containing the sampling rate selected. The chart can also be exported as bitmap file using the Save Chart option under the Export menu. This graph contains the possibility to use the left click of the mouse to zoom into the selected area, in order to get a closer look at the desired information.

3.2.4 Description of the Efficiency Calculator Window

The UTPA PV System software was designed with the capability to calculate the efficiency of the two Photovoltaic Systems. This form uses the power data collected from ENGR and TXU Arrays and compares the production with the actual irradiance received from the sun at a specific point of time. For this purpose we decided to perform these calculations at the Solar Noon, which is varying every day throughout the year depending on the earth translation movement.

In figure 3.14 we can see an example of the calculation for a selected time period. The basic calculation will require a very simple user input in order to run. The user requires selecting start and finishing dates, and the Time Zone Settings that can be Daylight Saving Time (DST) or Local Standard Time (LST) [93].

In order to create a custom calculation, the user can manually enter the data for the different parameters shown in Figure 3.14. This form offers a graphical way to visualize the efficiency results for the ENGR and TXU Arrays using a bar chart, and they can also be imported to excel or as a bitmap file. The data is presented in a text window allowing the user to quickly review the results for these calculations.

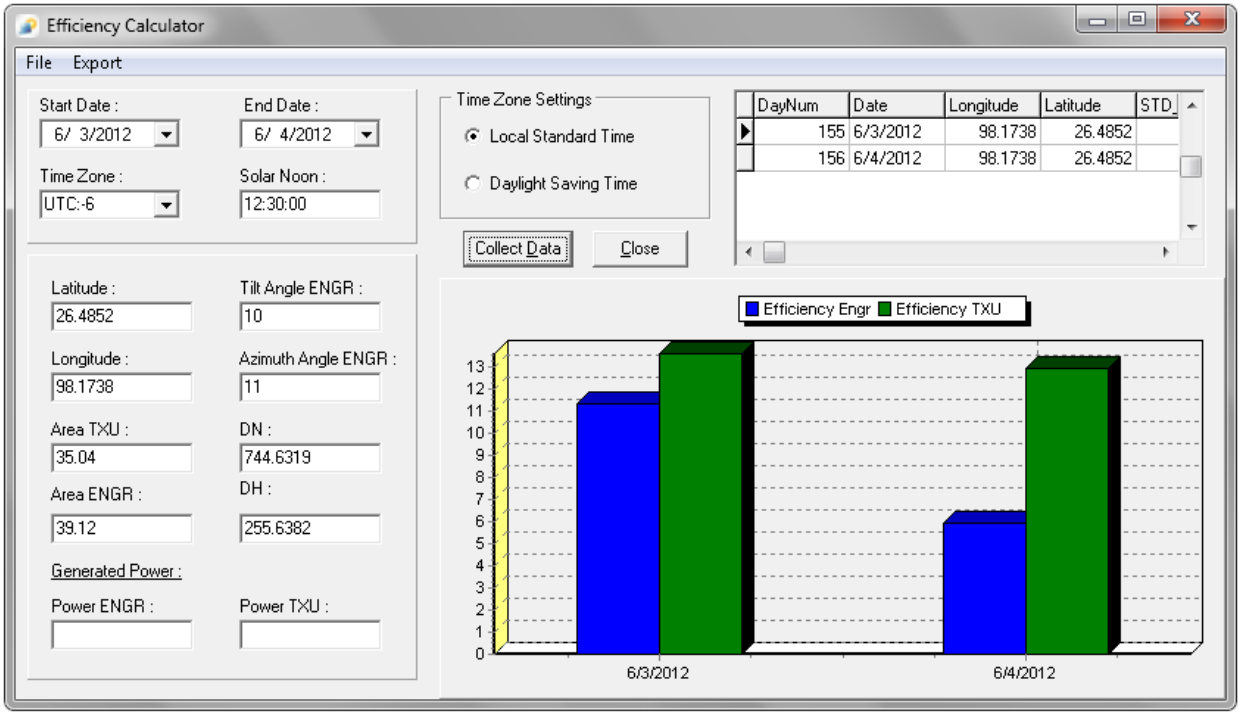


Figure 3.14 - Efficiency Calculator Window

In order to understand how the efficiency of the arrays is evaluated we have developed a system of equations presented in the following section.

3.2.5 Efficiency Calculation

With the purpose of calculating efficiency of the Photovoltaic Systems located at the University of Texas-Pan American, we need to analyze the behavior of the sun and how it influences on the amount of radiation that will be received by the Solar Panels.

As an easy way to calculate the relative angles between solar position(β, ϕ), and PV Arrays(Σ, ϕ_p), we will assign the direction of south to the x axis, east to the y, and up to z.

The angles are defined as:

β : Solar altitude or solar elevation.

ϕ : Solar azimuth, south is zero degrees going positive to the east.

Σ : Collector tilt angle.

ϕ_p: Collector azimuth, zero when collector faces the equator.

These angles are illustrated in Figure 3.15[94].

From figure 3.16, we can deduct that the solar distance to the panel can be defined as:

$$\overrightarrow{r_s} = \hat{z} r_s \sin\beta + \hat{x} r_s \cos\beta \cos\phi + \hat{y} r_s \cos\beta \sin\phi \quad (1)$$

And the unit vector in the direction to the sun is represented by:

$$\hat{r_s} = \hat{z} \sin\beta + \hat{x} \cos\beta \cos\phi + \hat{y} \cos\beta \sin\phi \quad (2)$$

The second important equation is the panel orientation (Σ : Collector tilt angle, and ϕ_p: Collector azimuth, if the panel is facing the equator, then $\phi_p = 0$). We must find an expression for the line perpendicular to the panel. This unit vector is denoted by $\hat{r_p}$, it will have three different situations that will define it. These three cases are explained next:

a) The panel is laying horizontally, then the unit vector will be:

$$\hat{r_p} = \hat{z} \cos\Sigma \quad (3)$$

The collector tilt angle is $\Sigma = 0$.

b) The panel is standing vertically, and looking to the equator, therefore the tilt and azimuth angles will be $\Sigma = 90^{o}$; $\phi_p = 0$, and the unit vector:

$$\hat{r}_p = \hat{z}\cos\Sigma + \hat{x}\sin\Sigma\cos\phi_p \quad (4)$$

c) The panel is standing vertically where the tilt angle is $\Sigma = 90^{o}$, and looking to the east the azimuth will be $\phi_p = 90^{o}$, we obtain the unit vector as:

$$\hat{r}_p = \hat{y} \quad (5)$$

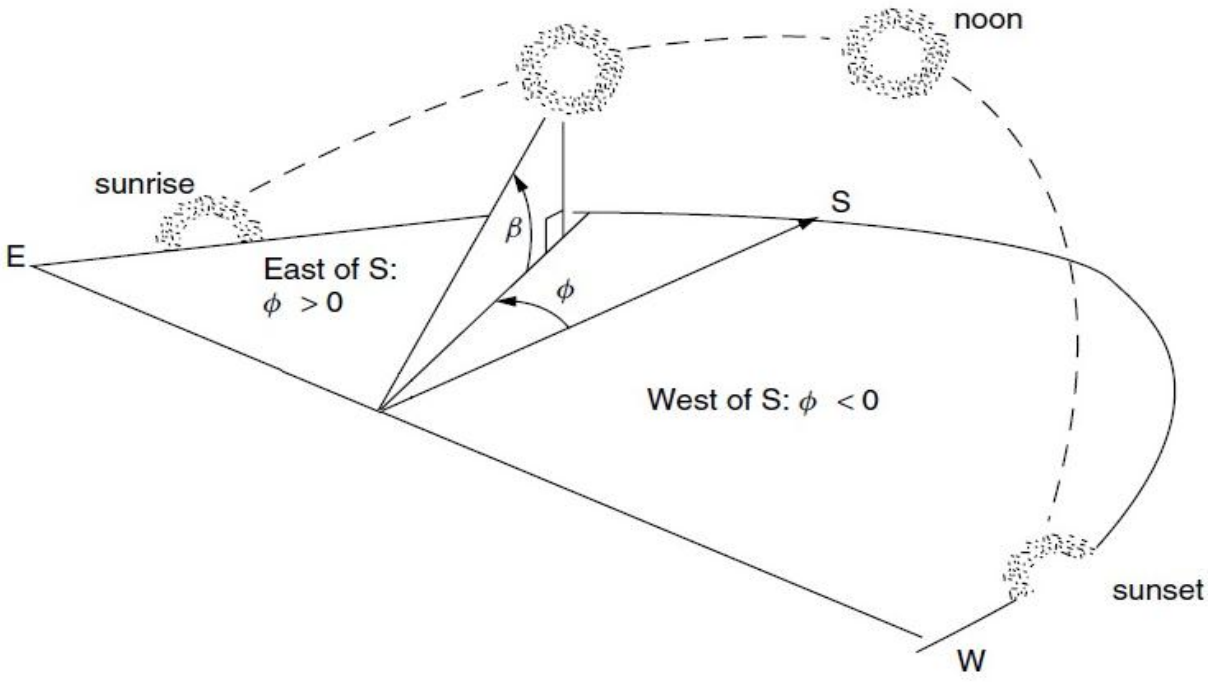


Figure 3.15 - Solar Position and Angles [94]

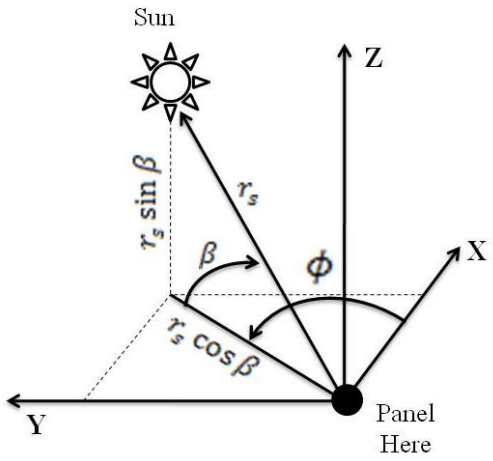


Figure 3.16 - Vector from the Panel to the Sun

Therefore all three cases are satisfied by the following equation:

$$\hat{r}_p = \hat{z}\cos\Sigma + \hat{x}\sin\Sigma\cos\phi_p + \hat{y}\sin\Sigma\sin\phi_p \qquad (6)$$

The angle between a line normal to the panel and a line pointing to the sun is obtained with the dot product from (2) & (6)

$$\cos\theta = \hat{r}_s \cdot \hat{r}_p = \sin\beta\cos\Sigma + \cos\beta\cos\phi\sin\Sigma\cos\phi_p + \cos\beta\sin\phi\sin\Sigma\sin\phi_p \qquad (7)$$

$$\cos\theta = \sin\beta\cos\Sigma + \cos\beta\sin\Sigma\cos(\phi - \phi_p) \qquad (8)$$

Equation (8) is the desired result. The sun motion angles which are altitude and azimuth (β, ϕ) are determined by:

$$\sin\beta = \cos L\cos\delta\cos\phi + \sin L\sin\phi \qquad (9)$$

$$\sin\phi = \cos\delta\frac{\sin H}{\cos\beta} \qquad (10)$$

Where: L is the site latitude, δ is the day declination, and H is the day hour (H = 0 for Solar Noon).

According to [93], the declination angle in degrees is:

$$\delta = (0.006918 - 0.399912\cos\Gamma + 0.070257\sin\Gamma - 0.006758\cos 2\Gamma +$$

$$0.000907 \sin 2\Gamma - 0.002697 \cos 3\Gamma + 0.00148 \sin 3\Gamma) \left(\frac{180}{\pi}\right) \quad (11)$$

Equation 11 delivers a more precise result for the declination angles than those calculated with the equation given in [94]:

$$\delta = 23.45 \sin\left[\frac{360}{365}(n - 81)\right] \quad (12)$$

Which considers *n* to be the day number of the year.

The day angle will be determined by:

$$\Gamma = 2\pi \frac{(d_n - 1)}{365} \quad (13)$$

This formula uses d_n as the day number, starting the count at 1 for January 1st.

3.2.5.1 UTPA's Solar Radiation Lab. As we mentioned before, this instrument is capable of measuring the following solar irradiances in W/m^2.

a) DN – Direct Normal, P_{DN}

b) DH – Diffuse Horizontal, P_{DH}

These radiation values will be needed in order to calculate efficiency. The data acquisition from the SRL system makes measurements every second; it performs 1-minute averages, writing the results in the database. These 1-minute averaged reading are also averaged every 60 minutes and recorded in another file as an hourly record.

The University of Texas-Pan American operates two solar arrays:

a) A fixed ENGR PV Array, tilt angle of $\Sigma = 11^o$, and facing the equator $\phi_p = 10^o$

b) A two axis TXU Solar Tracking Array.

Both arrays have 24 panels with capacity of 5.0kW and 5.5 kW as described previously.

3.2.5.2 Calculation of energy conversion efficiencies. The power intercepted by the ENGR

PV array & the TXU Sun Tracking Array, must be calculated. These powers are given by the following equations [94, 95]:

$$P_{ENGR} = DN \cdot \cos\theta + DH \frac{1+\cos\Sigma}{2} \quad (14)$$

$$P_{TXU} = DN \cdot 1 + DH \frac{1+\cos\Sigma}{2} \quad (15)$$

3.2.5.3 ENGR PV Array. The most convenient way to operate this array for the calculation of its operating efficiency is at solar noon and with its battery in a low state of change if it is the case for the Array to be running disconnected to the power grid. If it is grid tied, then the state of the battery will not affect this calculation.

At solar noon the azimuth angle is $\phi = 0$. From equation (11) the altitude can be shown as $\sin\beta = \cos(L - \delta)$, therefore it can be defined as $\beta = 90 - (L - \delta)$, and the incidence angle on the array from (8) if $\phi_p = 0$, for the array facing the equator, then:

$$\cos\theta = \sin\beta\cos\Sigma + \cos\beta\sin\Sigma = \sin(\beta + \Sigma) \quad (16)$$

Substituting the new equation for the altitude angle $\beta = 90 - (L - \delta)$, we obtain:

$$\cos\theta = \sin(90 - (L - \delta) + \Sigma) = \cos(L - \delta - \Sigma) \quad (17)$$

Therefore, a simplified equation for the incidence angle will be:

$$\theta = L - \delta - \Sigma \quad (18)$$

That is defined as a function of the declination.

3.2.5.4 TXU Sun Tracking Arrays. The solar power intercepted by these moving arrays is given by the equation (15); now the tilt angle Σ is a function of time. Equation (6) must be calculated from the figure 3.17:

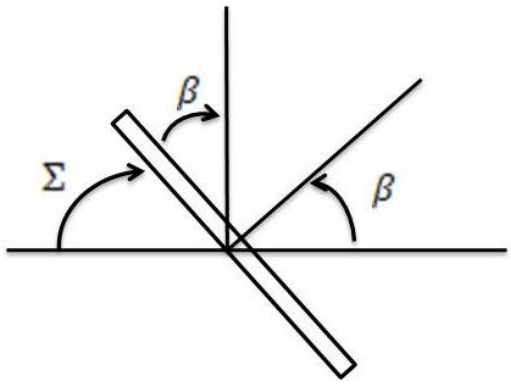


Figure 3.17 - TXU Sun Tracking Array, Panel Orientation

$$\Sigma + \beta = 90 \tag{19}$$

Now also the sun azimuth angle will be $\phi = \phi_p$, therefore $\cos\Sigma = \sin\beta$, and the power received at the TXU arrays can be expressed as:

$$P_{TXU} = DN \cdot 1 + DH \frac{1+\sin\beta}{2} \tag{20}$$

$$P_{TXU} = DN \cdot 1 + DH \left(\frac{1}{2}\right)(1 + \cos L \cos\delta \cos\phi + \sin L \sin\delta) \tag{21}$$

Again, if we perform the experiment of solar noon, the azimuth angle $\phi = 0$, then we obtain a simplified expression:

$$P_{TXU} = DN \cdot 1 + DH \left(\frac{1}{2}\right)(1 + \cos(L - \delta)) \tag{22}$$

For the cases described above the sun azimuth angle $\phi = 0$.

3.2.6 Additional Features

As an additional feature, a user can add new users to the database in order to allow them access to the system. On the main window named UTPA – PV System, the user can select the Add User option under the File menu. A new window will be shown as displayed in Figure 3.18. In order to add a new user, some information needs to be entered, such as a user name, a password, and a user level. Finally the Add User button should be clicked. This form also allows

seeing the already registered users by clicking on the Show Users button. If a user needs to be deleted, his/her user name should be entered, and the Delete User button clicked.

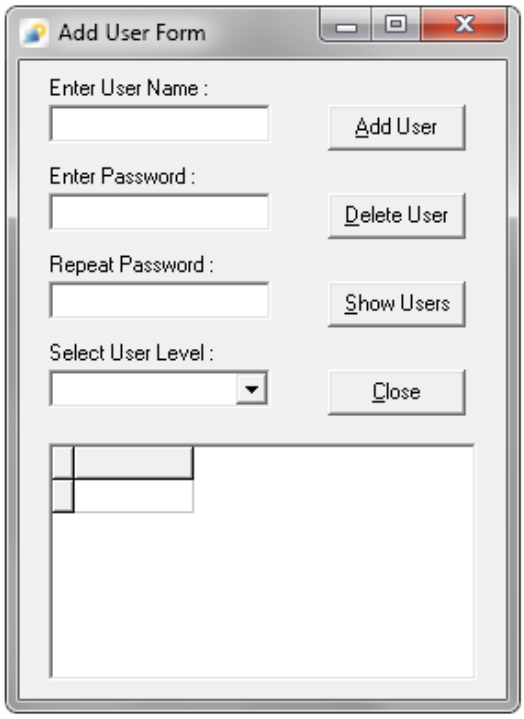


Figure 3.18 - Add User Form Window

3.3 Chapter Summary

Within chapter III, we have been able to introduce the Photovoltaic System we have in the Electrical Engineering building of the University of Texas-Pan American. Besides of the two photovoltaic arrays (ENGR PV Array and TXU Sun Tracker Arrays) and the Solar Radiation Lab, We have also introduced the Software interface developed for the purpose of gathering solar data, customized report generation, and efficiency calculation.

We have learned about the impact of using a database to create an organized scheme of our Photovoltaic System, and allow access to multiple users to this information.

The basic principles of how solar radiation is being measured, the correlation between the solar angles and the position of the panels impacts the performance, and how the efficiency calculation has been developed for our site have also been introduced in this chapter.

In the following chapters, we will advance to the experimental phase of this thesis. We will discover the impact of Distributed Denial of Service Attacks on systems like the one we have described in this chapter.

CHAPTER IV

COMPARATIVE EVALUATION OF WINDOWS 2008 SERVER WITH RED HAT LINUX 5 SERVER UNDER DDoS ATTACK

One of the main goals of a web service is to be available all the time for the users who wish to establish a connection in order to exchange needed information via Internet. Nevertheless, an attacker can cripple most of the available websites using a large enough team of zombies; a Botnet of 20,000 machines can bring down almost 90% of the Internet Websites as we can find in [50]. According to [46], the two most deployed web servers on the internet are Apache with 65.4% popularity and Microsoft IIS with 18%.

In this chapter, we present our experimental results regarding the impact of Distributed Denial of Service Attack (DDoS). We compare the performance of Windows 2008 Server with Red Hat Enterprise Linux 5 Server installed on the same hardware. DDoS attacks are considered one of the powerful attacks where the victim computer is overwhelmed with the large amount of requests of incoming traffic, thereby making the victim computer busy in replying to those requests thus consuming the system resources and causing a Denial of Service to legitimate users. To study the impact of these attacks, we created the attack traffic in the controlled lab environment of Networking Research Lab (NRL) at The University of Texas-Pan American.

We compared the behavior of Windows 2008 Server operating system with Red Hat Linux 5 Server operating system under some popular attacks such as ARP Flood, Ping Flood, ICMP

Land, Smurf Flood, TCP-SYN and UDP Flood attacks corresponding to Layer-2, Layer-3 and Layer-4 in the TCP/IP suite for the experimentation. Choosing the attacks per layer basis allows us to evaluate the protection of these operating systems against the Denial of Service attacks.

4.1 Test Plan and Experimental Setup

We have two different configurations for the experimental setup in the Network Research Lab that simulate real a world environment. First example of experimental setup is shown in Figure 4.1. In this case we consider the attack as an external attack, where the MAC Addresses contained in the datagrams are fixed, since they representing a MAC Address for the gateway router. The Internet traffic is addressed to the Victim Computer, coming from both sources legitimate users and zombie computers.

The second representation is found in Figure 4.2. This is considered an internal attack, where the hacker is creating a Botnet of zombie computers located in the same Local Area Network as that of the Victim Server. For internal attack, the packets are simulated to have different source MAC addresses, since they are generated by the computers located in the internal network. Random MAC Addresses are assigned to the packets forwarded to the server as attack traffic.

DDoS attacks were simulated in a controlled lab environment of the NRL in the Electrical Engineering Department at the University of Texas-Pan American. We simulate real attack traffic as it would be experienced by a web server deployed across the World Wide Web. As a first step, the system under test is fully used for legitimate traffic requesting to establish a connection with the web server, as this would happen in the real world, this will represent a baseline of optimal performance for the sever working under normal conditions. After a stable number of connections have been established, the server is flooded with attack traffic that has

been created by simulating multiple computers sending a barrage of corresponding attack traffic to the victim computer under test.

The attack traffic load was applied in the range of 10 Mbps to 1000 Mbps (or 1Gbps) using a Gigabit Ethernet link. The victim server is now being stressed with requests coming from legitimate clients and attack traffic. We are able to analyze its behavior same as in a real DDoS Attack scenario would happen, with the exception that in the controlled environment the attack traffic flow is handled to the system under test according to our test plan specifications.

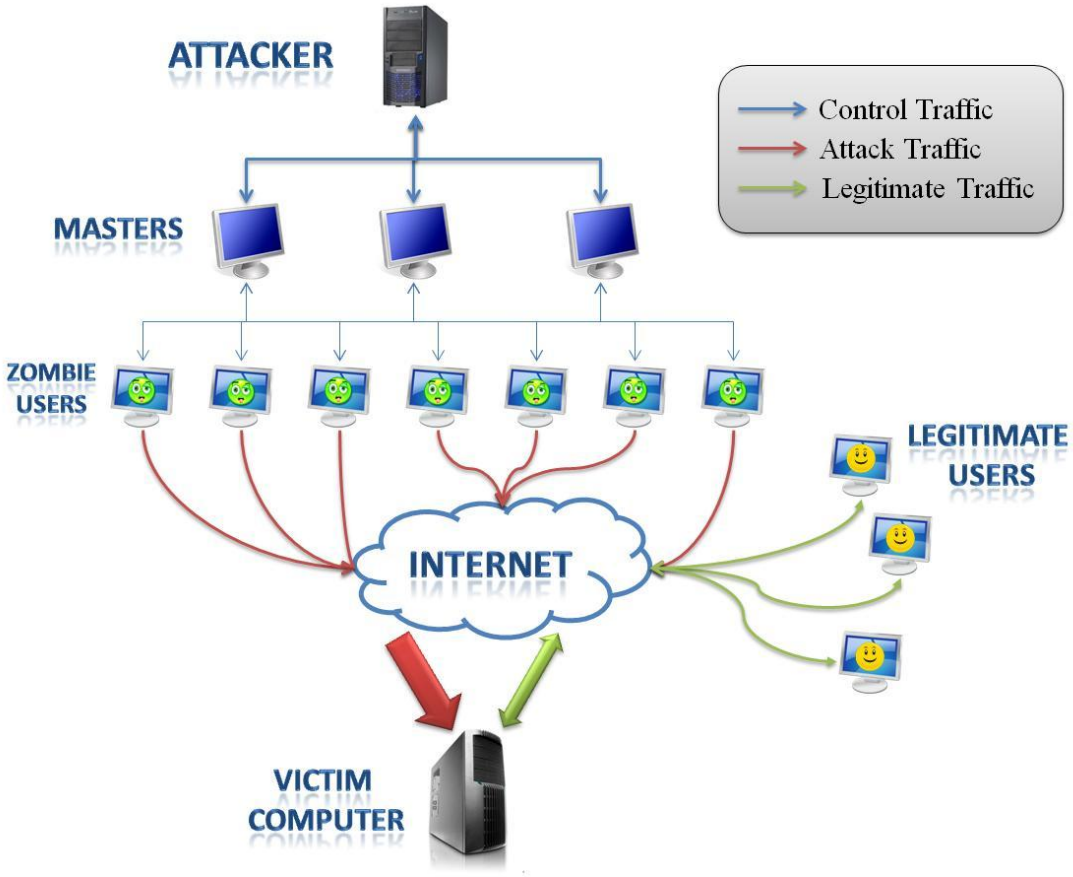


Figure 4.1 – Example of External Attack on Victim Computer

We tested Windows Server 2008 and Red Hat Linux 5 Server Operating Systems under same attack loads and on the same hardware platform. Once the client http connections were

established on a stable fashion (baseline), attack traffic was applied to the system starting from 10 Mbps, then moving to 100 Mbps up to 1Gbps in steps of 100 Mbps, for an interval of 5 minutes per load. The performance data sample rate was measured and collected every second, leading to collection of 300 measurements per load for each set of attack. For a given load, same tests were repeated three times to obtain average value for the measurements considered (total of 900 data samples for each load). Three separate runs of the same tests for each given load were repeated to ensure the same result was observed in all of the iterations. A gap of 4 seconds was introduced in between when changing from one attack load to the next one; no traffic was sent to the device under test throughout this gap. During this time, processor resources were relieved from the attack traffic allowing the legitimate users to re-establish any lost connections and go back to normal state. When this gap of time is finished, the attack is continued with the next incremental load according to the previously described procedure, until 1 Gbps traffic limit is reached.

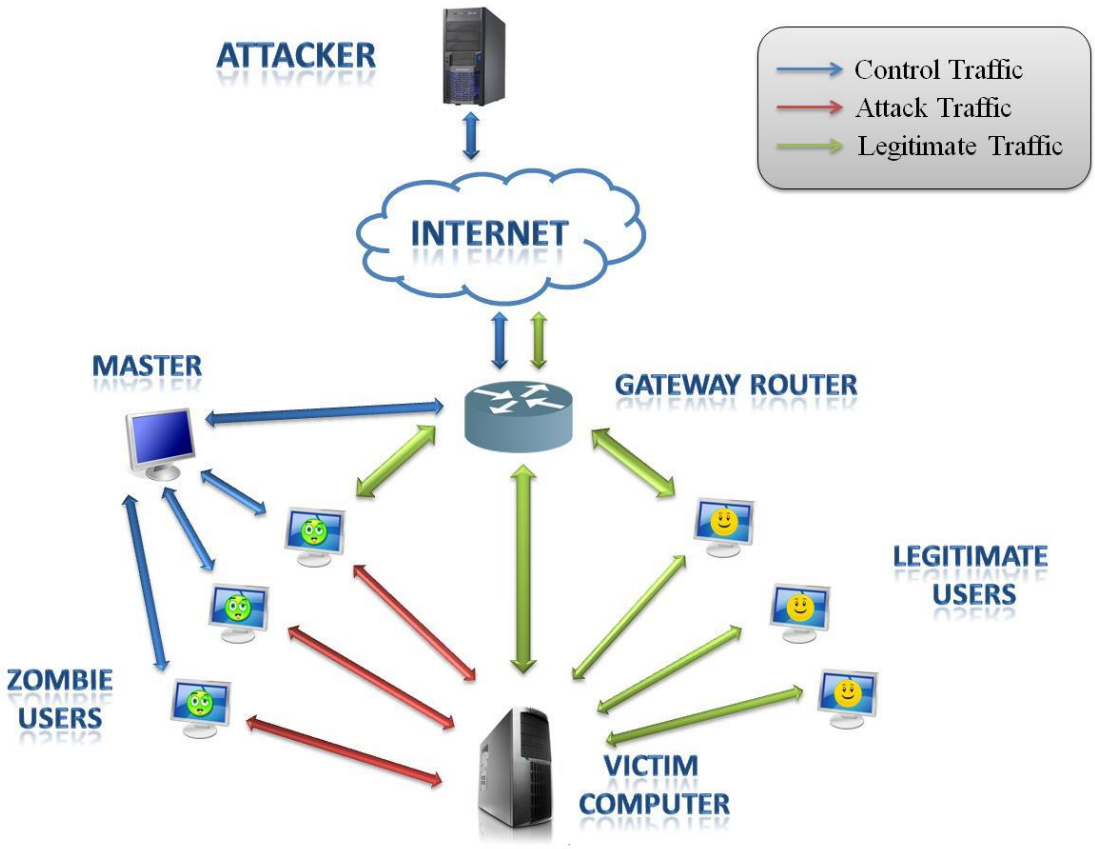


Figure 4.2 – Example of Internal Attack on Victim Computer

These tests were performed on a server computer with the following description:

- Dell PowerEdge 1800
- Processor Intel® Xeon ® CPU ES345 @ 2.33 GHz Dual Core
- RAM capacity of 4.00 GB

The two Operating Systems that were used are:

a) Windows Server 2008 R2 Enterprise. Service Pack 1, Build 7601, 64-bit Operating System.

    a. Using Internet Information Services or IIS 7 as web server.

b) Red Hat Enterprise Linux Server 5.3 (Tikanga). 64-bit Operating System.

a. Using Apache HTTP Server 2.2 as web server.

We tested both operating systems under the same conditions using the firewall configuration enabled and disabled to compare the performance on each OS for both configurations when they were under DDoS Attacks. In order to establish legitimate http connections in both servers when the Firewall was enabled, we had to modify the default settings, and added a new rule to the TCP port 80 to allow incoming connections.

4.2 Performance Evaluation Methods and Parameters

The parameters of performance evaluation considered for this experiment were the Processor utilization and the number of successfully established http connections. These resources are measured while we subject the platform under test with different DDoS attacks. Processor utilization is an important factor to be measured in Denial of Service attacks because, if the processor is fully consumed by the attack traffic, then, the system will not be able to provide any other services or execute any other processes, thereby denying the services to its legitimate user. The other factor that was not considered important is the Memory availability, since we did not notice a significant impact on this resource while the DDoS attack was being executed. If all the available main memory or RAM was consumed by the attack packets or was occupied with the processes corresponding to the attack-packets, then system can become unstable and non-responsive. So we measured memory percentage allocation to observe how it is affected during the length of the disturbance, finding a very small impact on this resource. A detailed description of the tools we used to measure the performance on the devices under test is provided in Appendix A.

4.3 Results and Discussions

In this section, we will be reviewing the results obtained from the experimental tests performed. In order to make these results valid for both operating systems, we need to ensure that the conditions used are the same on both Operating Systems.

We first created a number of stable http connections which is different on both Server Operating Systems. The http connections are representing in this experiment, the amount of traffic that would be allowed when the system under test is operating in normal conditions.

For Microsoft Windows 2008 Server, we are able to obtain the same number of stable connections with both FW ON and FW OFF configurations. The number of stable http connections on Microsoft Windows Server 2008 is displayed on Figure 4.3 and it is of 2700 for both configurations, whereas in Red Hat Enterprise Linux 5 Server, the number of stable established http connections is of 4800 when the firewall configuration is OFF and 3900 when it is ON, this is shown in Figure 4.5.

Once the stable number of http connections representing legitimate users is established, the attack traffic is introduced in the device under for both FW ON and OFF configuration under DDoS network attacks like ARP Flood, Ping Flood, ICMP Land, Smurf Flood, TCP-SYN, and UDP Flood, for both Windows 2008 Server for Red Hat Enterprise 5 Server.

One very important aspect that was found on Red Hat Enterprise 5 Server is the fact that by default it cannot receive a large number of http connections when the Firewall service is enabled, thus, creating an impact on the number of connections very similar to a denial of service even when only legitimate traffic is being received. A log file is created by the SELinux service when the limit of connections is reached that is shown in Appendix B. The command required to show this parameter needs to be executed as a super user and is:

```
[root@localhost]# sysctl net.ipv4.netfilter.ip_conntrack_max
net.ipv4.netfilter.ip_conntrack_max = 65536
```

As we can see, the number of connections that the Apache web server is limited to host is 65536 http connections. With the following command we can know the number of current connections:

```
[root@localhost]#  wc -l /proc/net/ip_conntrack
 65747 /proc/net/ip_conntrack
```

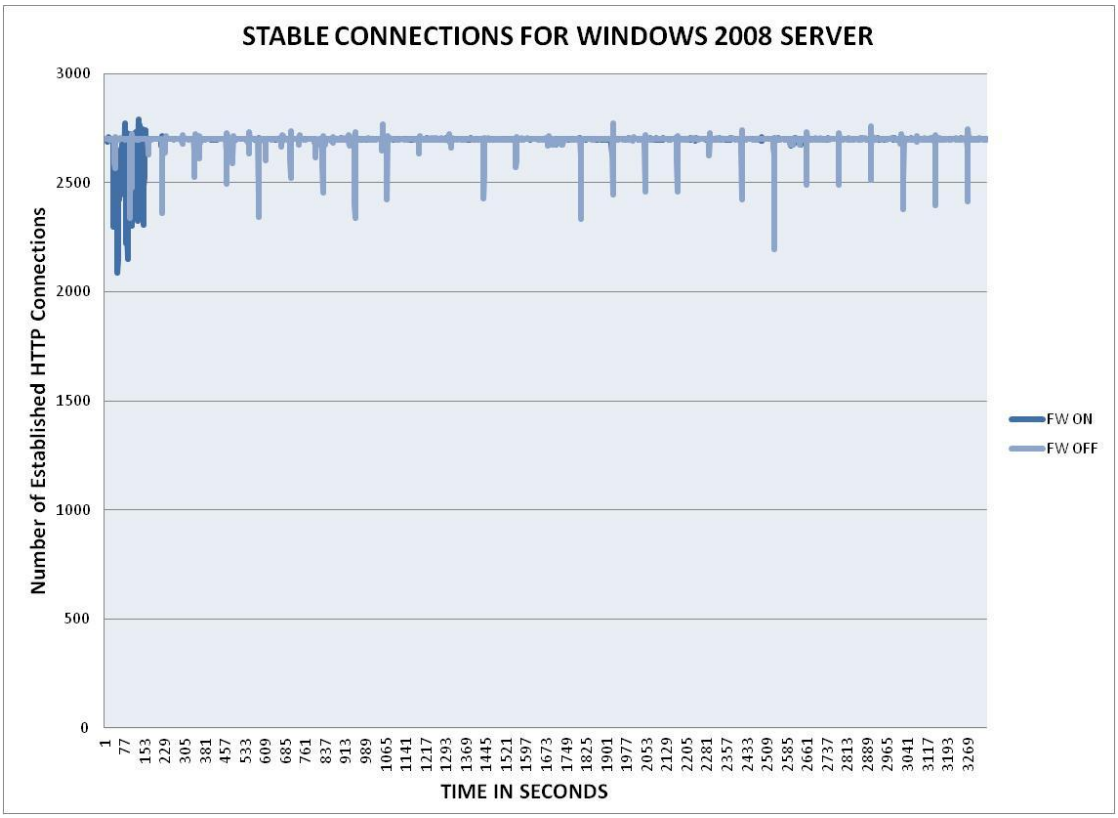


Figure 4.3 - Stable number of http connections for Microsoft Windows Server 2008 R2

Apache web server is now showing a number of 65747 http connections, which is exceeding the limit set by default. Once this limit is reached, the Apache web server will start to drop any other incoming connections, thus, creating the effect of a denial of service attack that has been captured and shown in Figure 4.4, where legitimate connections are dropped even though there is

no attack traffic in the network. New connections will be accepted once the http connections limit is released.

As we can observe, the number of connections is not stable and we cannot evaluate the performance of the Apache web server under such conditions, therefore, we modified the limit of connections to get a stable number of connections which will allow us for evaluating the attack at different loads by observing how the number of connections is affected when the disturbance is entered in the system. As the attack load is incremented the number of legitimate users that can access the web server will be diminished, thus creating a Denial of Service when the number is relatively low compared when it is stable and free of attack.

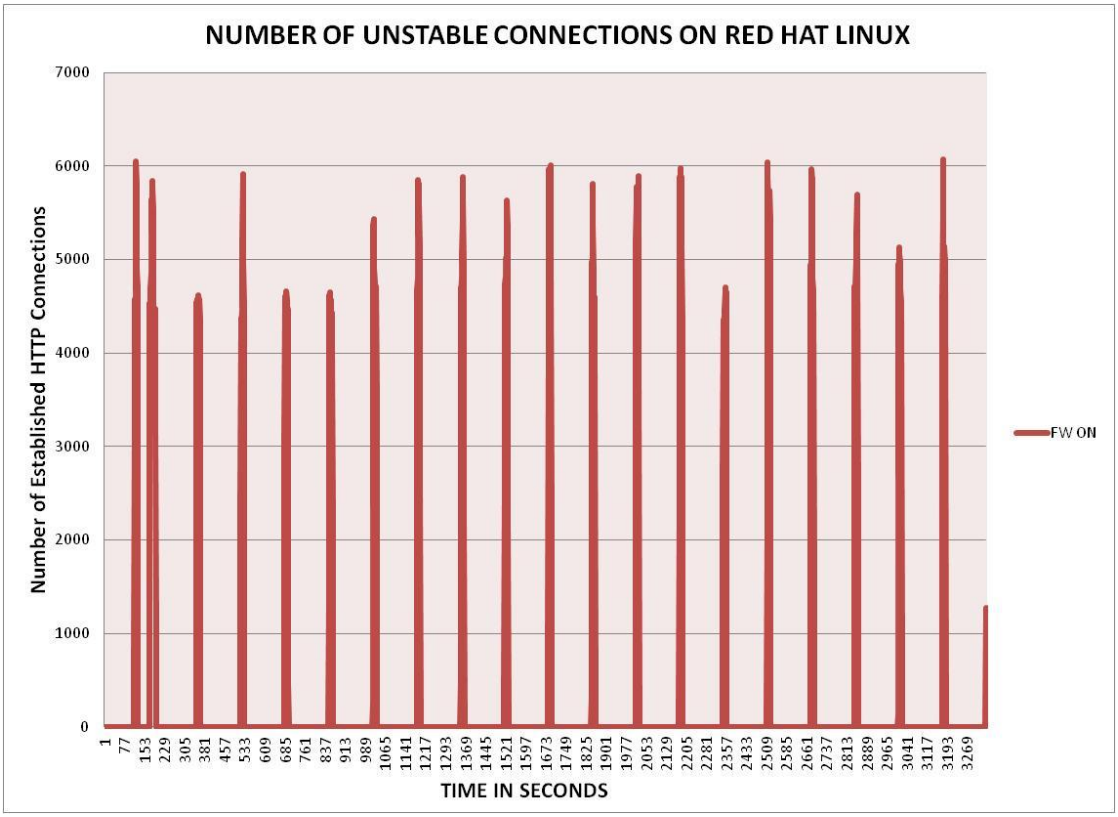


Figure 4.4 - Unstable connections on Red Hat Linux for default Firewall configuration

This behavior has been corrected by manually modifying this parameter changing the limit of http connections that the Apache Web Server is configured to accept. The command has to be executed in the following directory as a super user:

```
[root@localhost ipv4]# pwd
/proc/sys/net/ipv4
```

For testing purposes, we will increase this number to 1000000, in order to accept the maximum number of connections that the Apache web server is able to host using the following command:

```
[root@localhost ipv4]# echo 1000000 > ip_conntrack_max
```

Now that we have changed this parameter, a stable number of connections can be achieved for Firewall On configuration, as shown in figure 4.5.

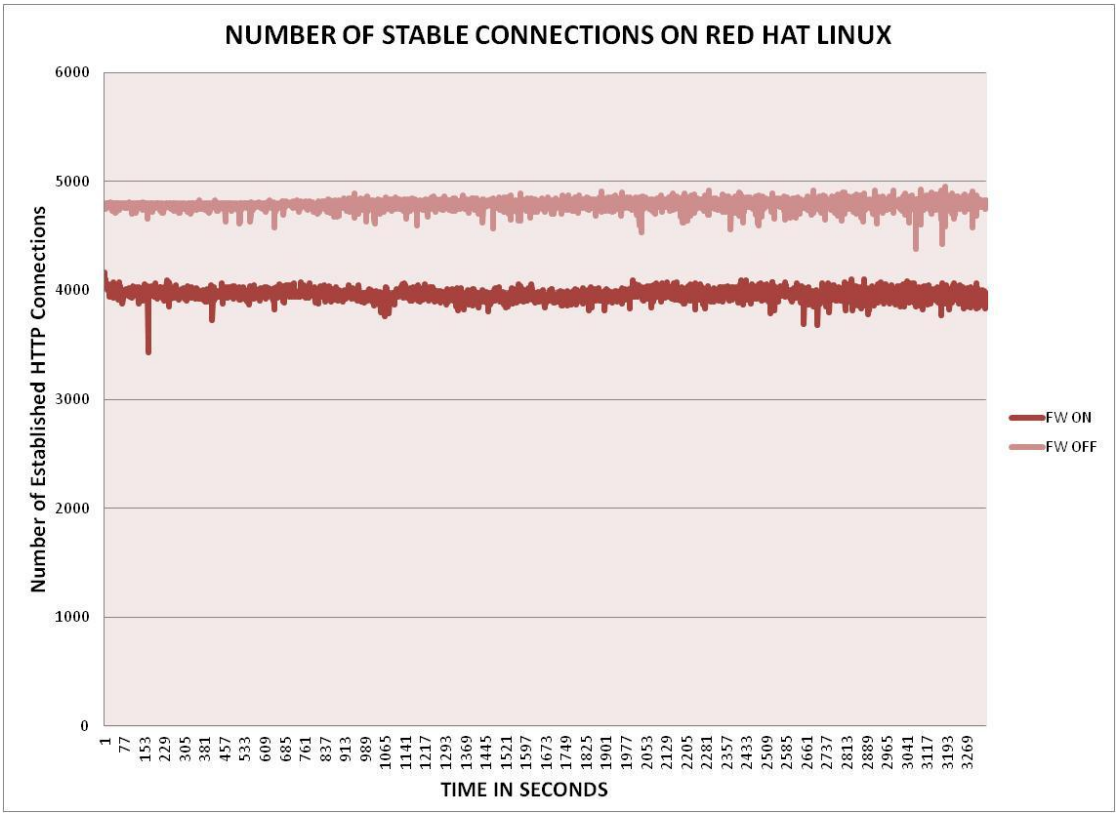


Figure 4.5 - Stable number of http connections for Red Hat Enterprise Linux 5

Now we will compare the test results we obtained for every configuration in the following sections for both operating systems. We will use the following methodology to describe the tests; first we will show the results for the External attack on Windows followed by Linux, then for Internal Attack Windows in the first term, then for Linux. This will be done on each one of the DDoS attack types that we are performing.

4.3.1 ARP Flood Attack

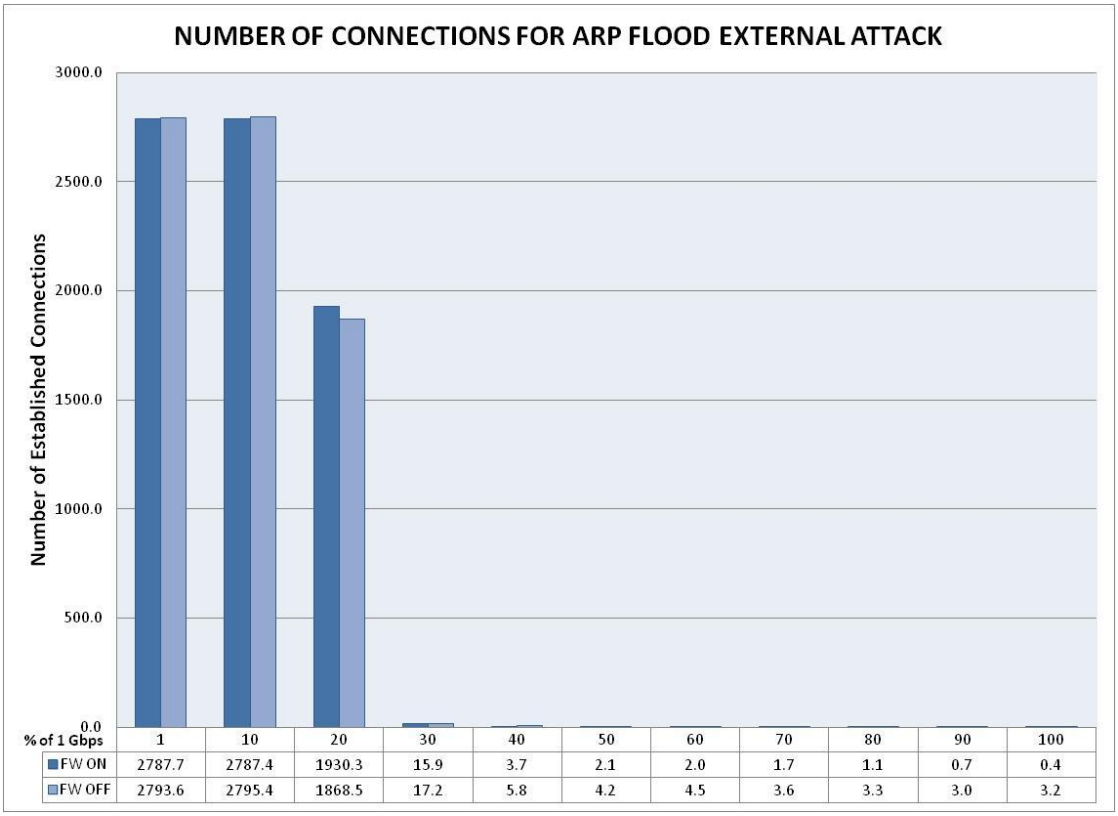
The attack loads that we are using are based on the percentage of a 1 Gbps Ethernet link. We noticed that the low intensity attacks did not have a big impact on the number of connections, but when it was increased to a high intensity rate, the number of connections was dramatically affected differently for every attack type, still following a similar fashion. Once that we obtained a stable number of http connections, the attack load was introduced in the following rates 10, 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 Mbps. In the figures displayed we show these measurements and also the average number of connections that each operating system was able to keep throughout every load for the duration of the attack.

As a color convention, we have used blue for all the figures representing measurements from the Microsoft Windows 2008 Operating System, and red for the figures with Red Hat Linux 5 Operating System information.

4.3.1.1 ARP Flood External Attack. On figure 4.6 we can observe how the Windows Server was going through a connection denial when the attack rate was 20%, which brought the number of connections close to 1900. There was not a noticeable change between the FW ON and OFF configurations. At 30%, the number of connections was decreased in a considerable way, and after that from 40 to 100%, the connections are very close to zero. For example on the

measurement for 40% attack load, the number of connections obtained for FW ON configuration was 3.7, which represents average value over a measurement data set of 900 samples.

For the Red Hat Server, displayed in figure 4.7, we notice a greater number of connections for the attack rate of 10% and lower compared to windows, but at 20%, we found that the number is reduced to 1500 connections, which represents a greater loss of connections compared to windows. On the Red Hat Server we also notice that the number of connections is greater when the Firewall is disabled for low intensity attack rates. After 30% of attack load, the same behavior is observed in Red Hat Linux as in the Windows Server.

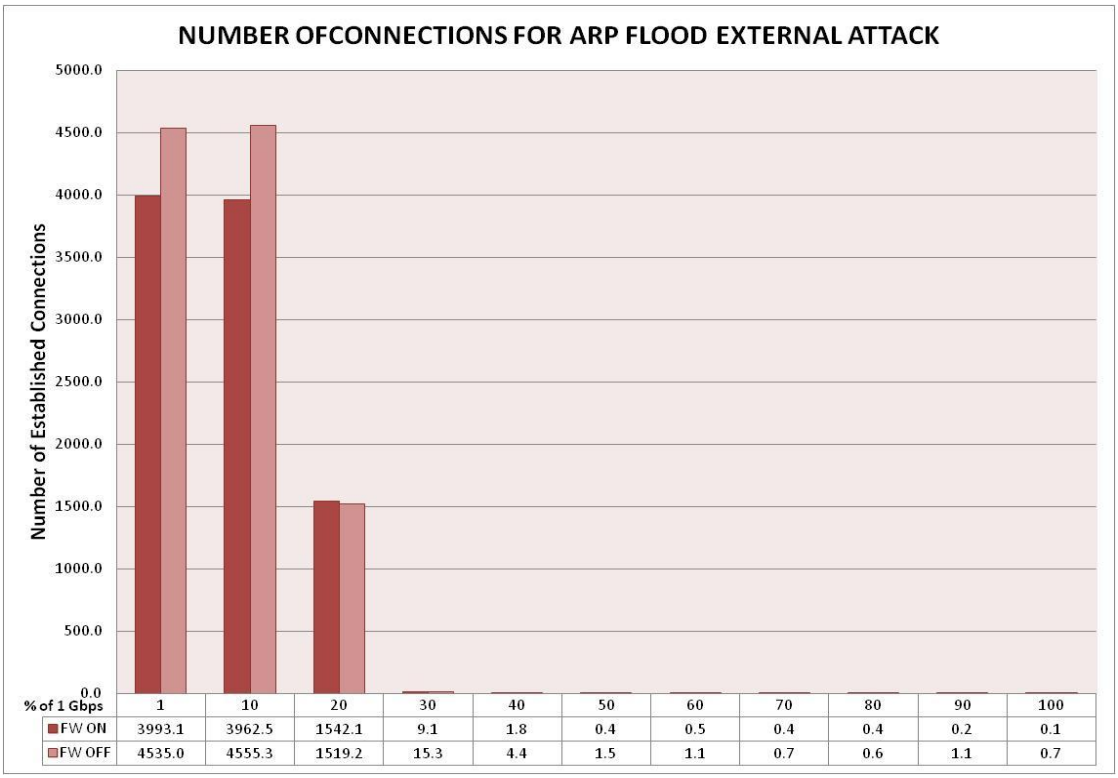


% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2787.7	2787.4	1930.3	15.9	3.7	2.1	2.0	1.7	1.1	0.7	0.4
FW OFF	2793.6	2795.4	1868.5	17.2	5.8	4.2	4.5	3.6	3.3	3.0	3.2

Figure 4.6 - Connections for ARP Flood External Attack on Microsoft Windows Server 2008 R2

We can conclude so far, that the Windows Firewall is defending better the number of connections that the one from Linux at 20% of attack rate, although Linux is able to host greater number of connections for the lower intensity attack rate.

In figures 4.8 and 4.9, we display the processor utilization for both Windows and Linux operating systems, respectively. As we can observe, the processor performance of Linux is better than Windows since it is using at most 78% of processor consumption, whereas Windows is consuming up to 97%, and at the same time, it is hosting less number of connections. In both operating systems we can see that the processor consumption is greater when the firewall is enabled, the difference is not very big, and the impact on performance can be considered null.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	3993.1	3962.5	1542.1	9.1	1.8	0.4	0.5	0.4	0.4	0.2	0.1
FW OFF	4535.0	4555.3	1519.2	15.3	4.4	1.5	1.1	0.7	0.6	1.1	0.7

Figure 4.7 - Connections for ARP Flood External Attack on Red Hat Linux 5 Server

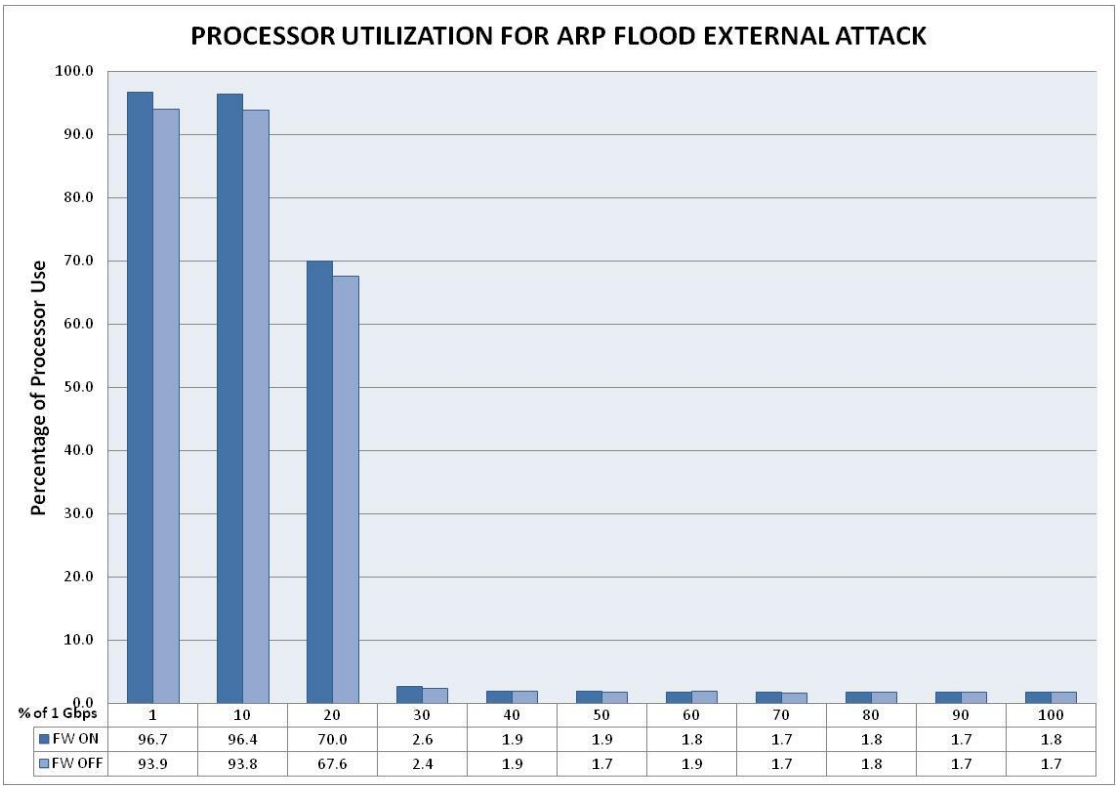


Figure 4.8 - Processor Utilization for ARP Flood External Attack on Microsoft Windows Server 2008 R2

It is important to notice that for low attack rates, the number of connections that Linux is able to host, is greater when the firewall is disabled, and the processor resources consumed is less compared to the firewall ON configuration. This behavior is observed to continue during the rest of attack types that we will be analyzing, whereas in windows, even the processor behavior is very similar, the number of connections for both firewall configurations remains almost equal.

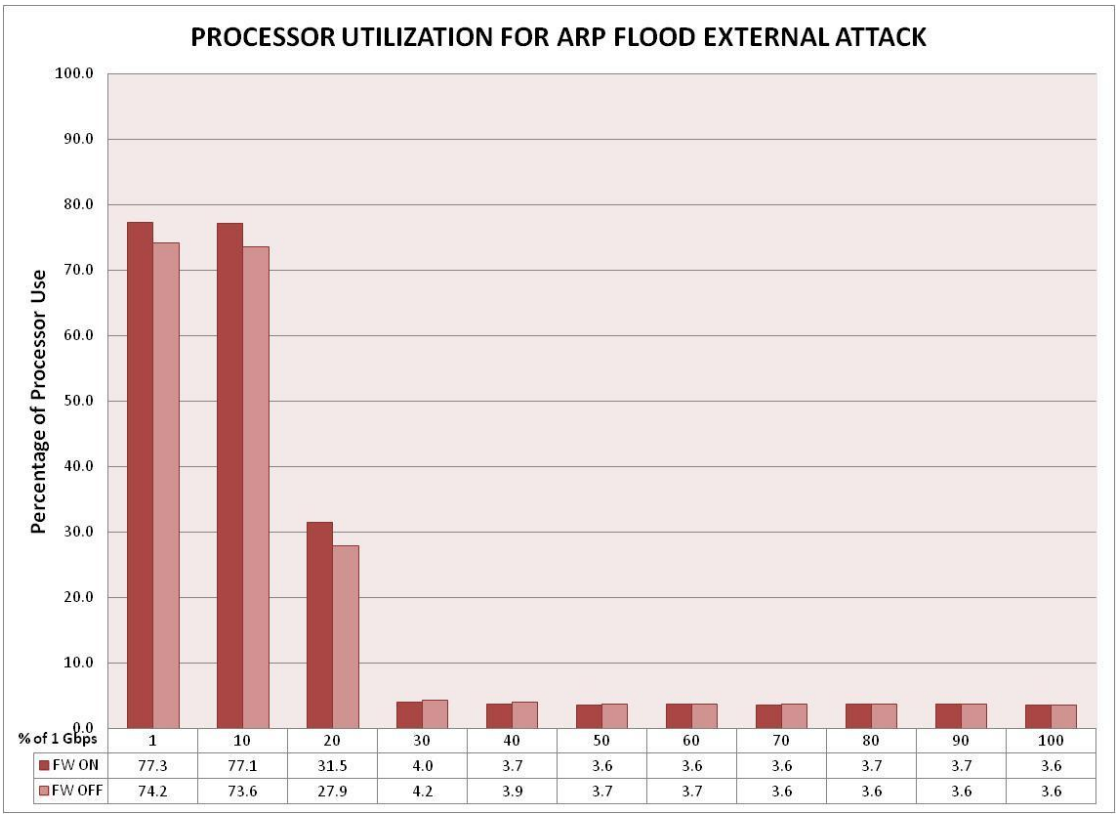


Figure 4.9 - Processor Utilization for ARP Flood External Attack on Red Hat Linux 5 Server

4.3.1.2 ARP Flood Internal Attack. In figure 4.10 the ARP flood internal attack we can still observe a small number of connections on Windows for 30% attack rate when firewall is enabled. It shows to be less harmful than the external attack configurations. Processor is consumed very similarly for both attack configurations internal and external on Windows shown in figure 4.12.

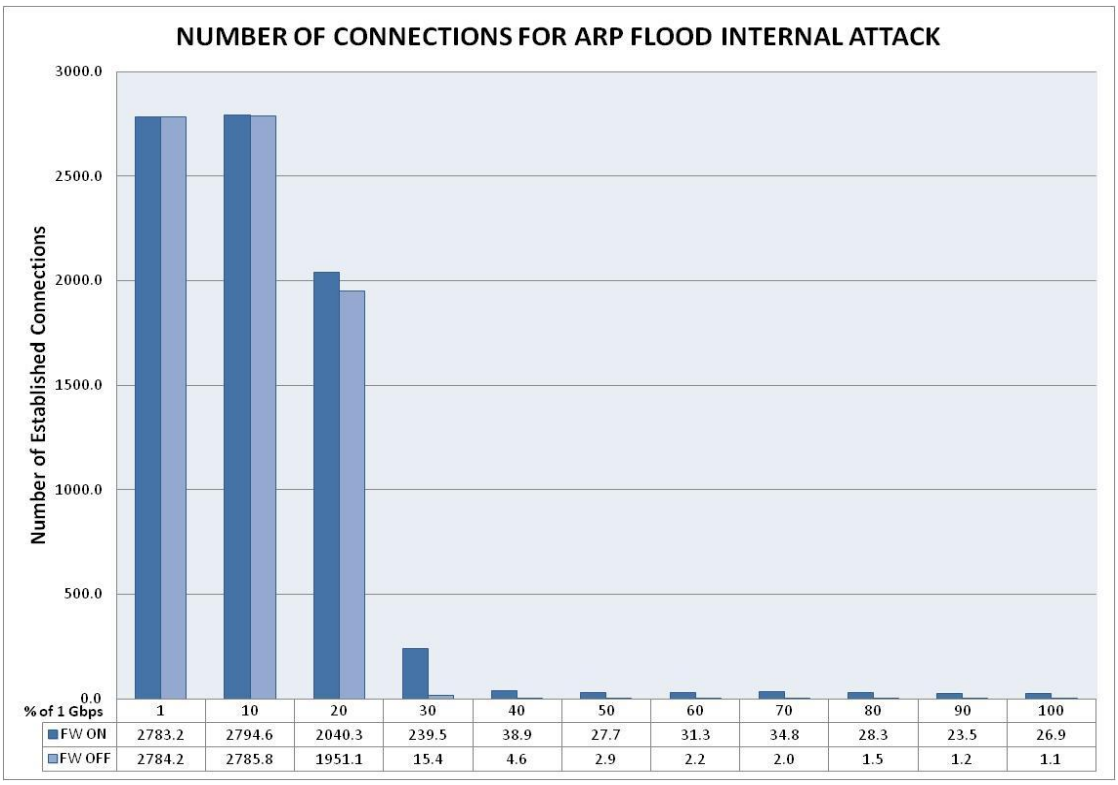


Figure 4.10 - Connections for ARP Flood Internal Attack on Microsoft Windows Server 2008 R2

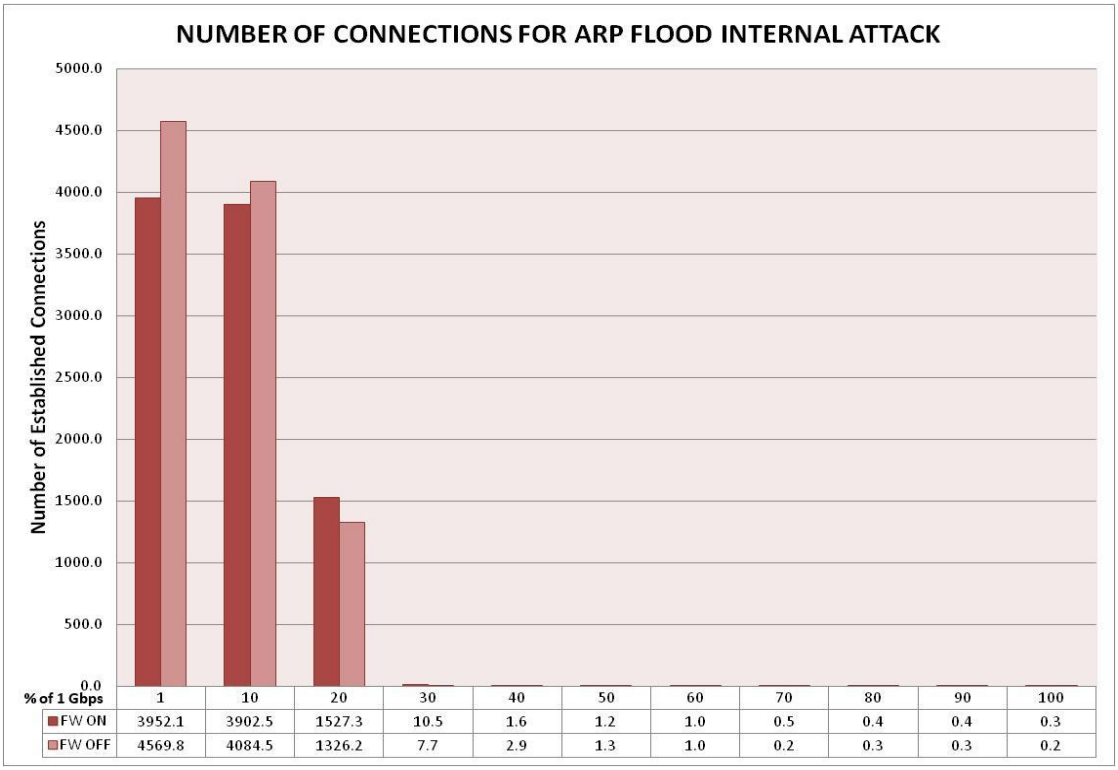
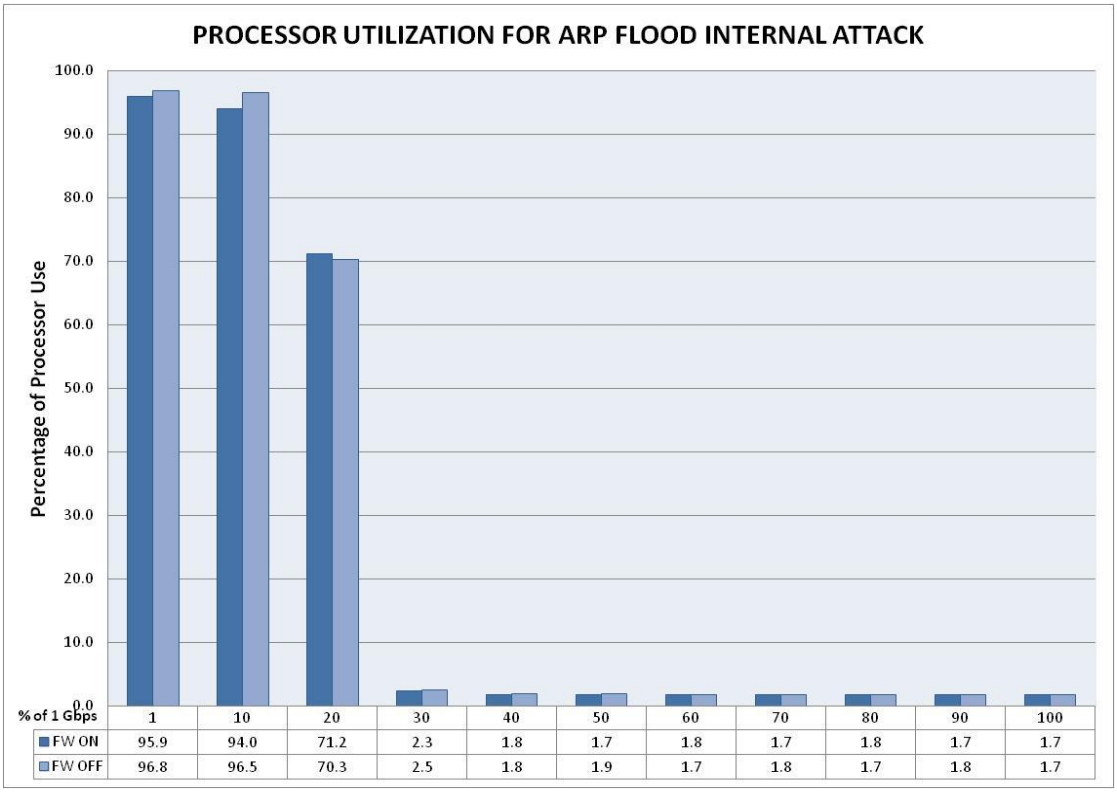


Figure 4.11 - Connections for ARP Flood Internal Attack on Red Hat Linux 5 Server

For the case of Linux from figure 4.11, the number of connections remains almost the same in the internal attack as for the external attack, but the processor utilization when the firewall is disabled was observed to be greater, displayed in figure 4.13. The number of connections was almost null for the 30% attack rate and higher, the attack traffic created a big impact on processor resources, compared to the firewall enabled, where it made a good job keeping the processor resources low.

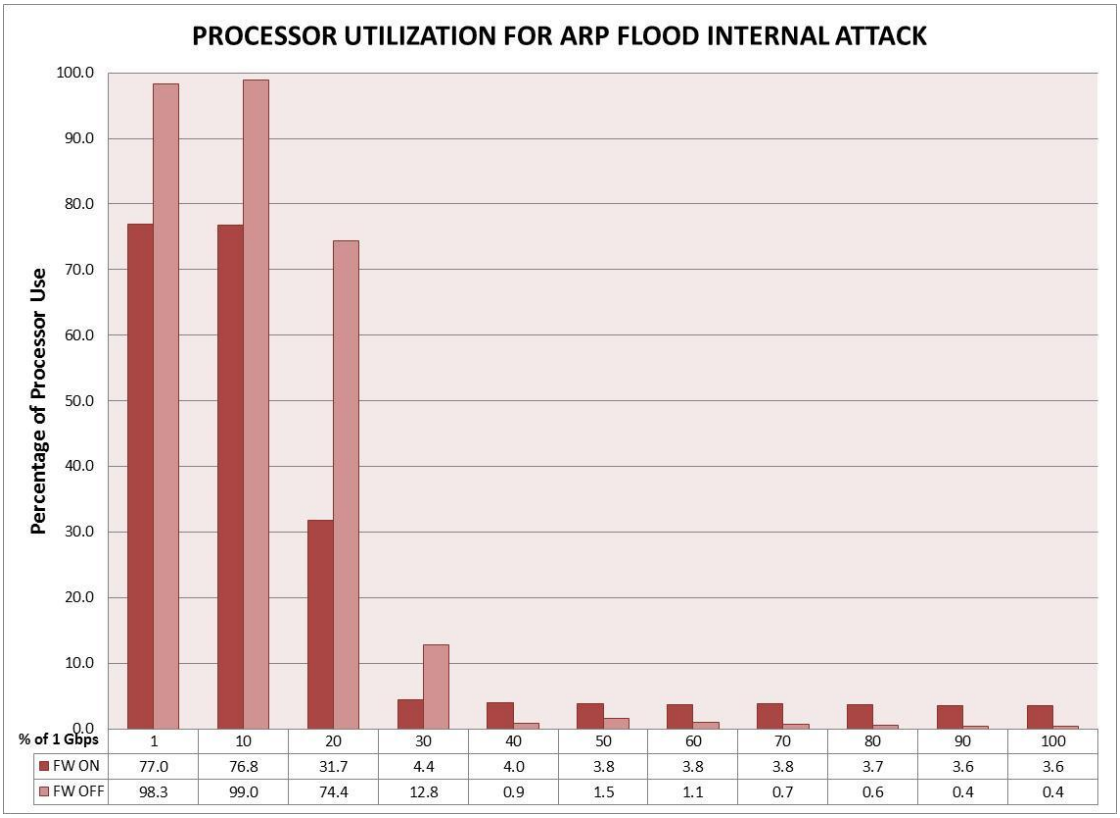
In general, we can conclude for ARP flood attack, that for Windows it has more impact the external attack for both firewall configurations.

For Linux the internal attack proved to be more harmful than the external attack.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	95.9	94.0	71.2	2.3	1.8	1.7	1.8	1.7	1.8	1.7	1.7
FW OFF	96.8	96.5	70.3	2.5	1.8	1.9	1.7	1.8	1.7	1.8	1.7

Figure 4.12 - Processor Utilization for ARP Flood Internal Attack on Microsoft Windows Server 2008 R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	77.0	76.8	31.7	4.4	4.0	3.8	3.8	3.8	3.7	3.6	3.6
FW OFF	98.3	99.0	74.4	12.8	0.9	1.5	1.1	0.7	0.6	0.4	0.4

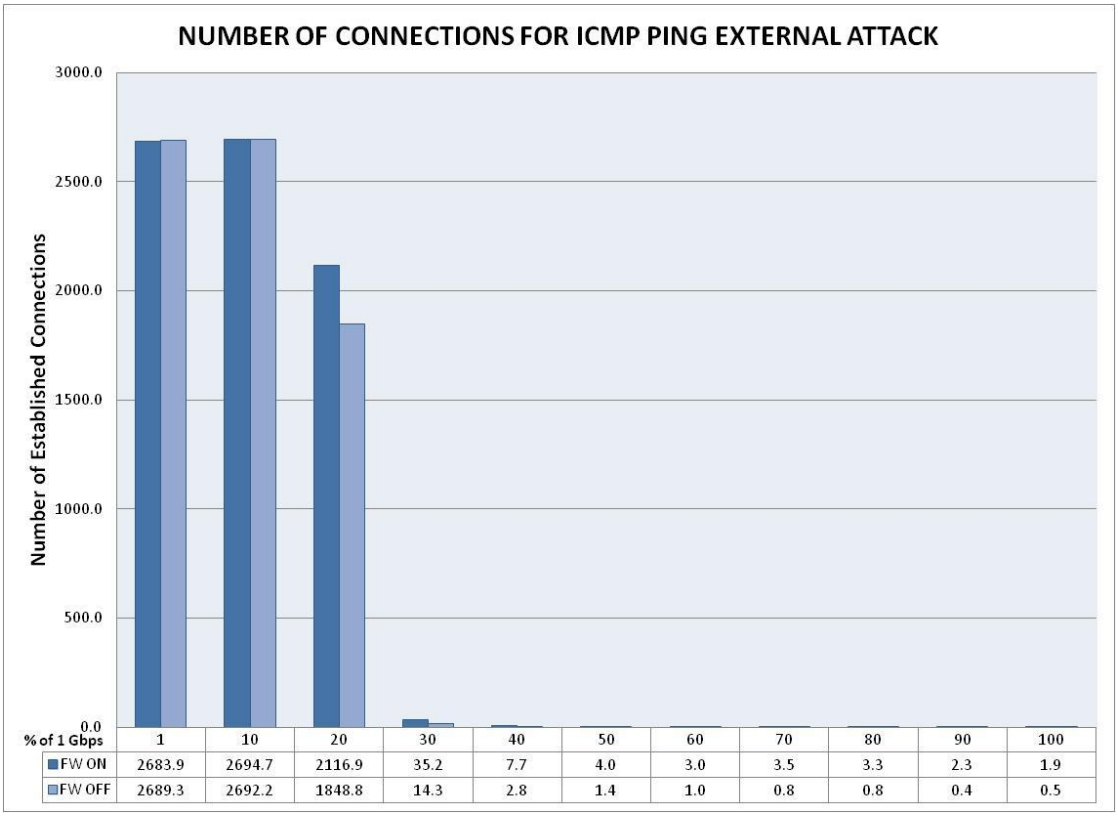
Figure 4.13 - Processor Utilization for ARP Flood Internal Attack on Red Hat Linux 5 Server

4.3.2 Ping Flood Attack

In the following subsections, we will analyze what we consider one of the most common attack types, the Ping Flood Attack. We will review both configurations External and Internal Attacks.

4.3.2.1 Ping Flood External Attack. In figures 4.14 and 4.15 we can observe the number of connections for Windows and Linux operating systems when the server is under Ping flood external attack. For low intensity attack rates from 1% to 10%, the attack had no significant impact on any platform. Linux proved to be able to host greater number of connections than Windows, and the connections hosted by Linux were greater when the firewall was disabled.

When the attack rate was increased to 20%, we noticed that Windows is defending better keeping more connections compared to Linux. From 30% to 100%, the number of connections was almost zero on both operating systems.

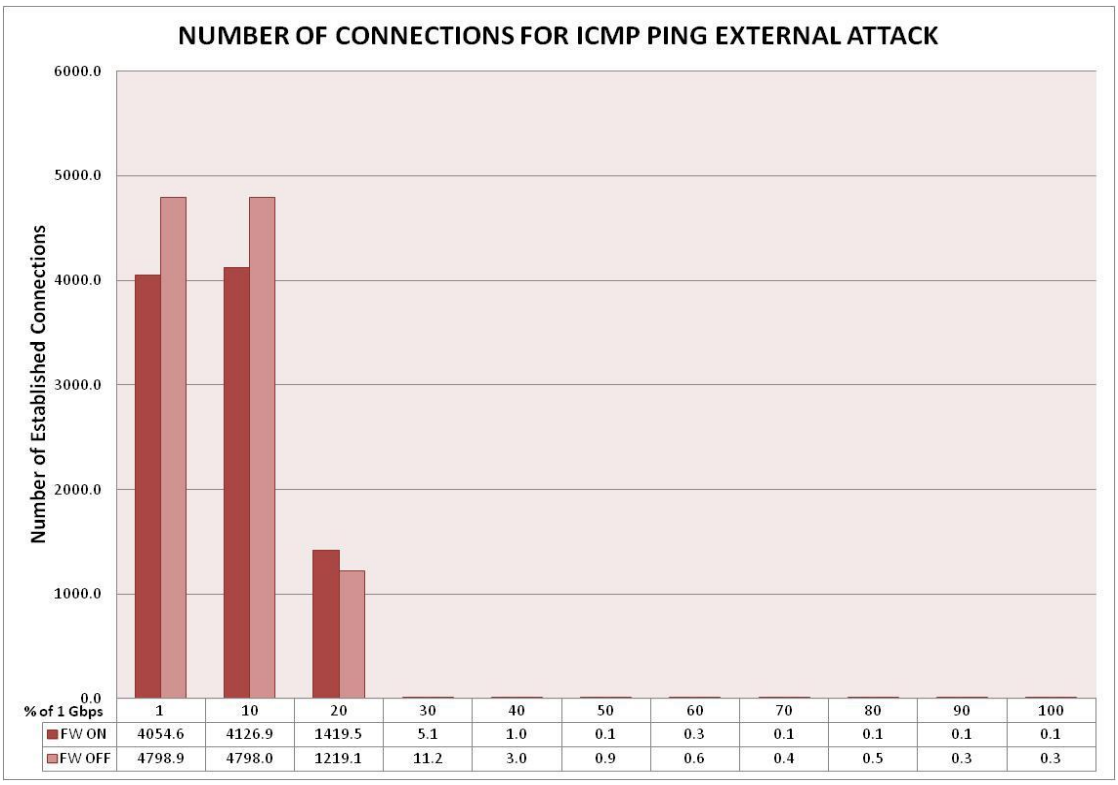


% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2683.9	2694.7	2116.9	35.2	7.7	4.0	3.0	3.5	3.3	2.3	1.9
FW OFF	2689.3	2692.2	1848.8	14.3	2.8	1.4	1.0	0.8	0.8	0.4	0.5

Figure 4.14 - Connections for Ping Flood External Attack on Microsoft Windows Server 2008 R2

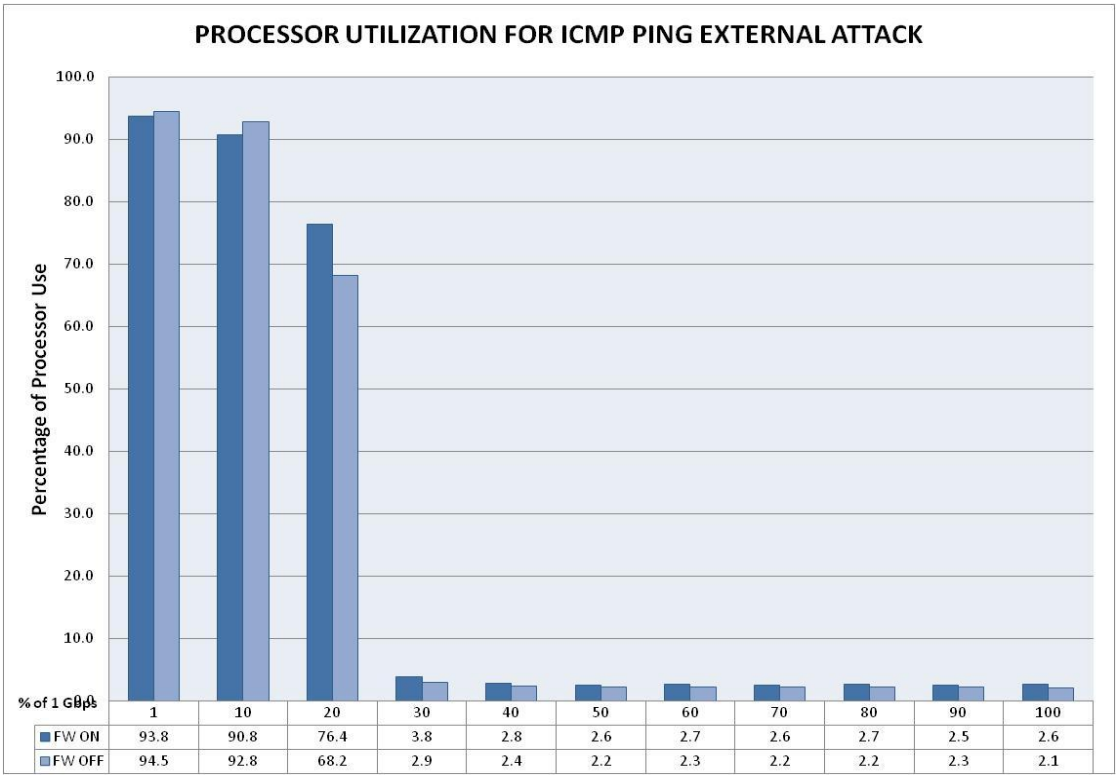
For the processor utilization, we show the figure 4.16 and 4.17 for Windows and Linux, respectively. We can see that for the attack rates from 1% to 20%, Windows is using more processor resources compared to Linux.

We observe that for both operating systems, the processor consumption for this attack is directly related to the number of connections being hosted by the system. Once the connections are lost due to the attack traffic, the processor resources are released.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	4054.6	4126.9	1419.5	5.1	1.0	0.1	0.3	0.1	0.1	0.1	0.1
FW OFF	4798.9	4798.0	1219.1	11.2	3.0	0.9	0.6	0.4	0.5	0.3	0.3

Figure 4.15 - Connections for Ping Flood External Attack on Red Hat Linux 5 Server



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	93.8	90.8	76.4	3.8	2.8	2.6	2.7	2.6	2.7	2.5	2.6
FW OFF	94.5	92.8	68.2	2.9	2.4	2.2	2.3	2.2	2.2	2.3	2.1

Figure 4.16 - Processor Utilization for Ping Flood External Attack on MS Windows Server 2008 R2

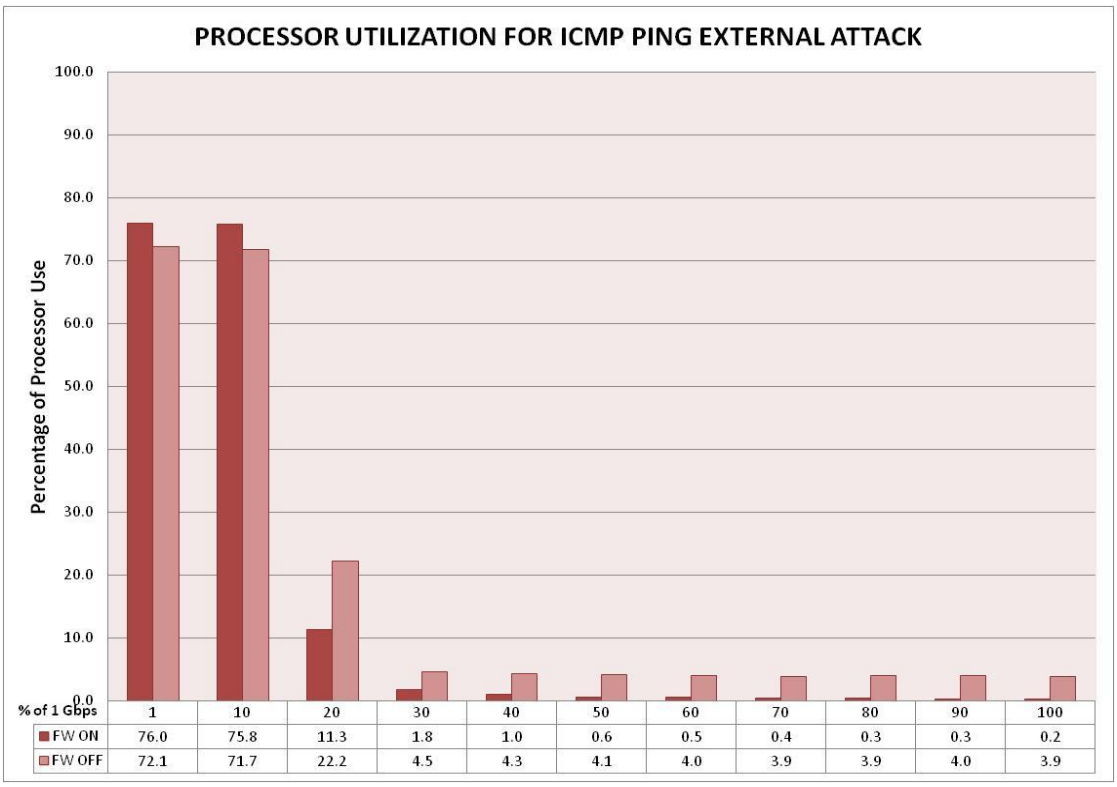


Figure 4.17 - Processor Utilization for Ping Flood External Attack on Red Hat Linux 5 Server

4.3.2.2 Ping Flood Internal Attack. For the Ping flood internal attack we present figures 4.18 and 4.19 to represent the number of connections for Windows and Linux platforms.

For this attack configuration, both servers are hosting completely the number of http connections up to 30% of the attack rate. We can observe that Linux is hosting more connections that Windows for this attack range, but at 40% Linux and Windows lose connections but Linux is performing poorly compared to Windows in order to keep connections.

At 50% of attack rate, Linux has already lost most of the http connections, while Windows is still keeping alive a few of them. For further attack rates, both operating systems are not able to sustain a big number of connections.

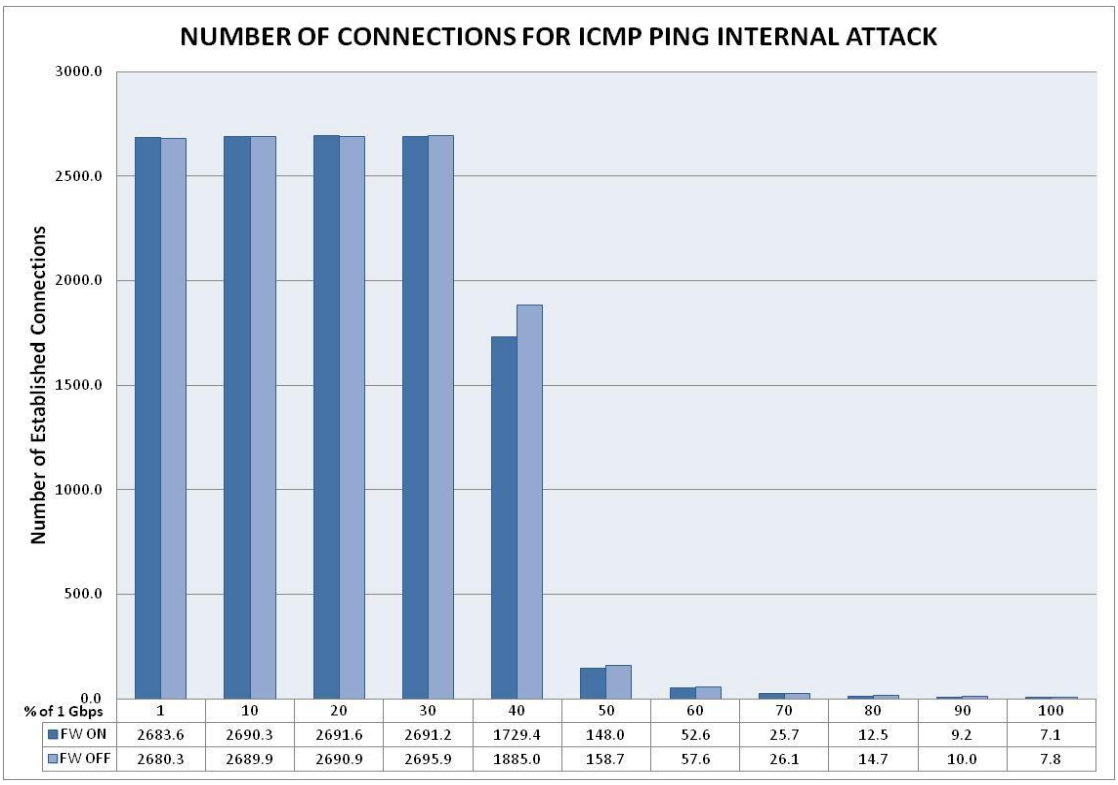


Figure 4.18 - Connections for Ping Flood Internal Attack on Microsoft Windows Server 2008 R2

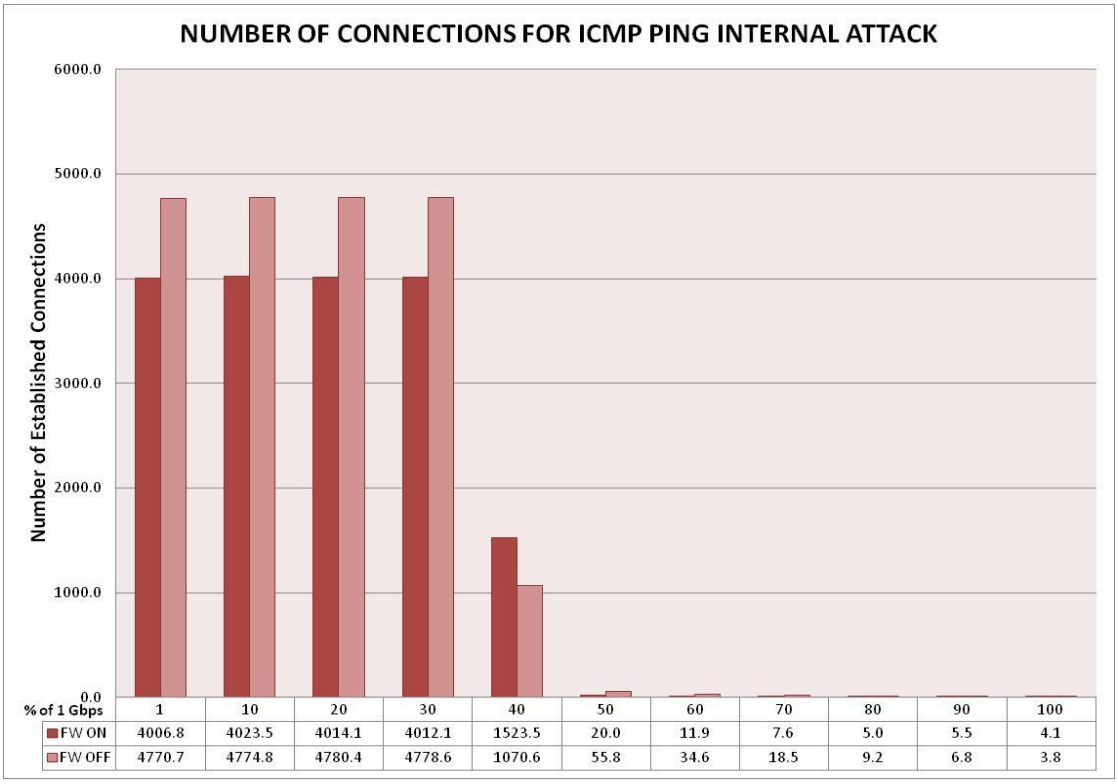
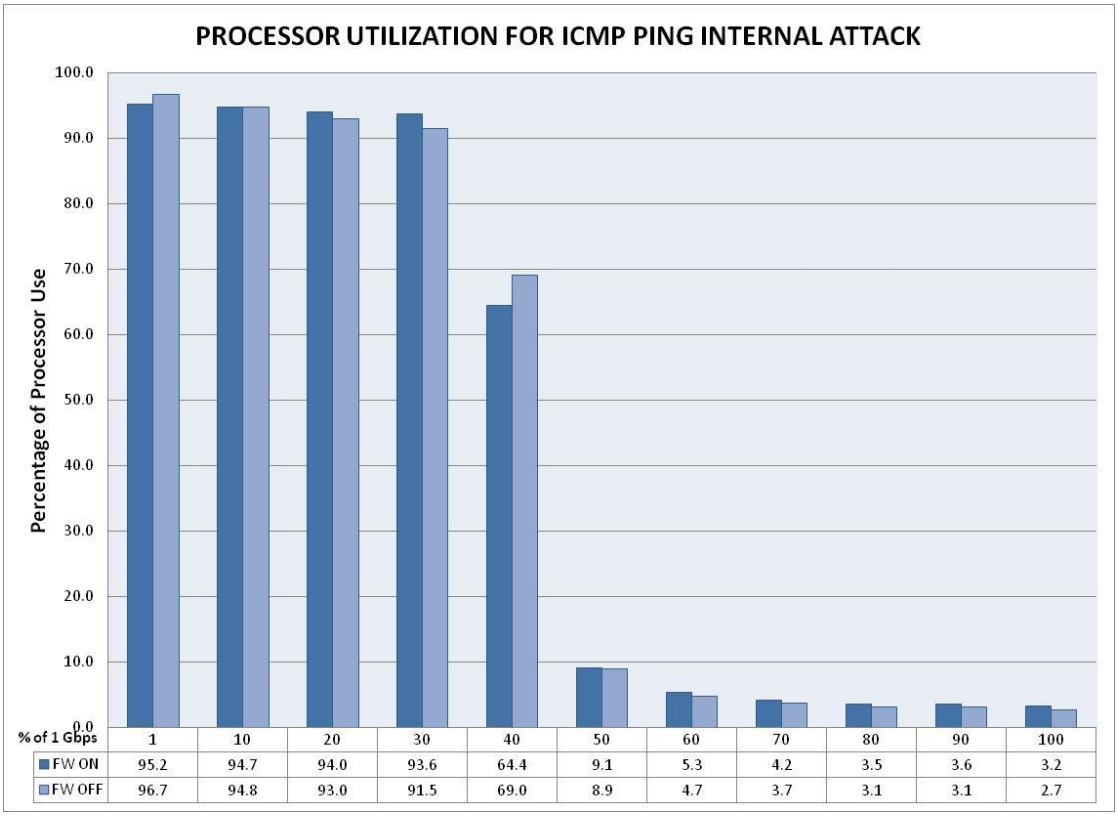


Figure 4.19 - Connections for Ping Flood Internal Attack on Red Hat Linux 5 Server

Regarding the processor utilization, Linux shows to perform better for attack rates lower than 500 Mbps consuming up to 78% of processor resources, compared to windows, which is using up to 97% of processor, leaving only small processor resources available for other operations.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	95.2	94.7	94.0	93.6	64.4	9.1	5.3	4.2	3.5	3.6	3.2
FW OFF	96.7	94.8	93.0	91.5	69.0	8.9	4.7	3.7	3.1	3.1	2.7

Figure 4.20 - Processor Utilization for Ping Flood Internal Attack on Microsoft Windows 2008 Server R2

For the Ping Flood Attack configurations, we can conclude that the external type of attack is much more harmful to both operating systems than the internal attack. Bothe operating systems experience loss of http connections at lower rates of attack when they are under external attack.

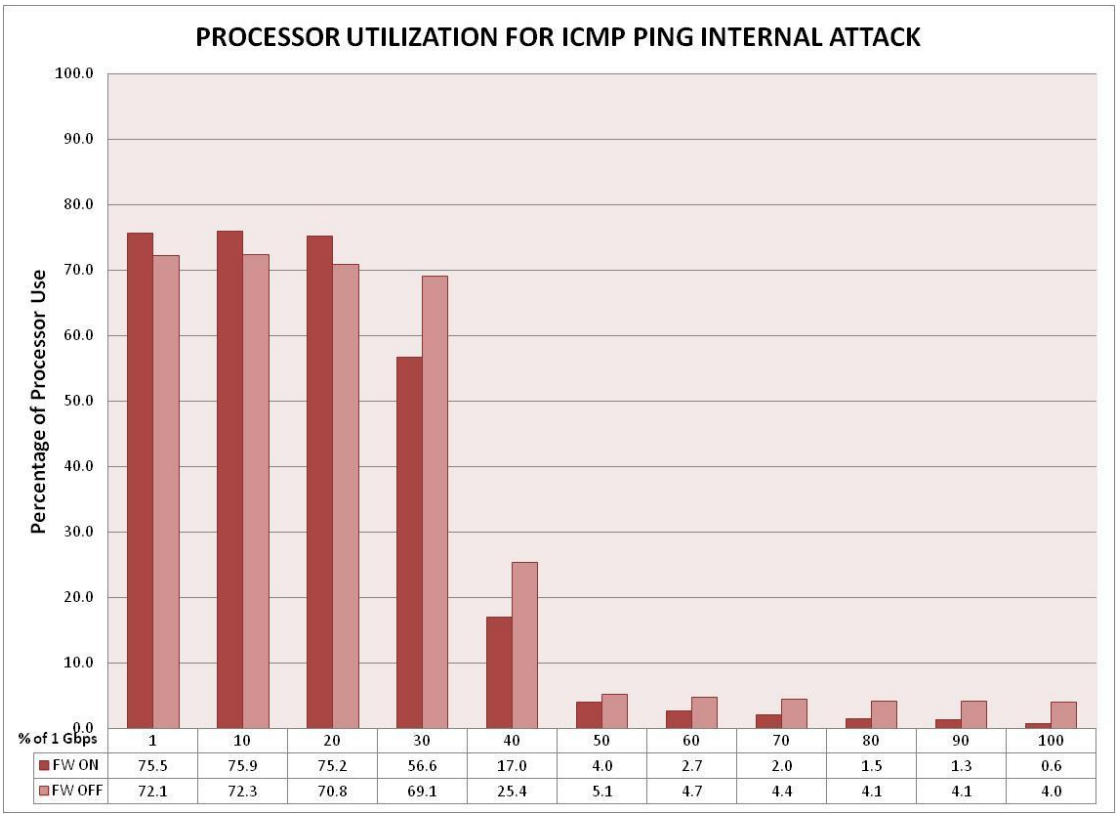


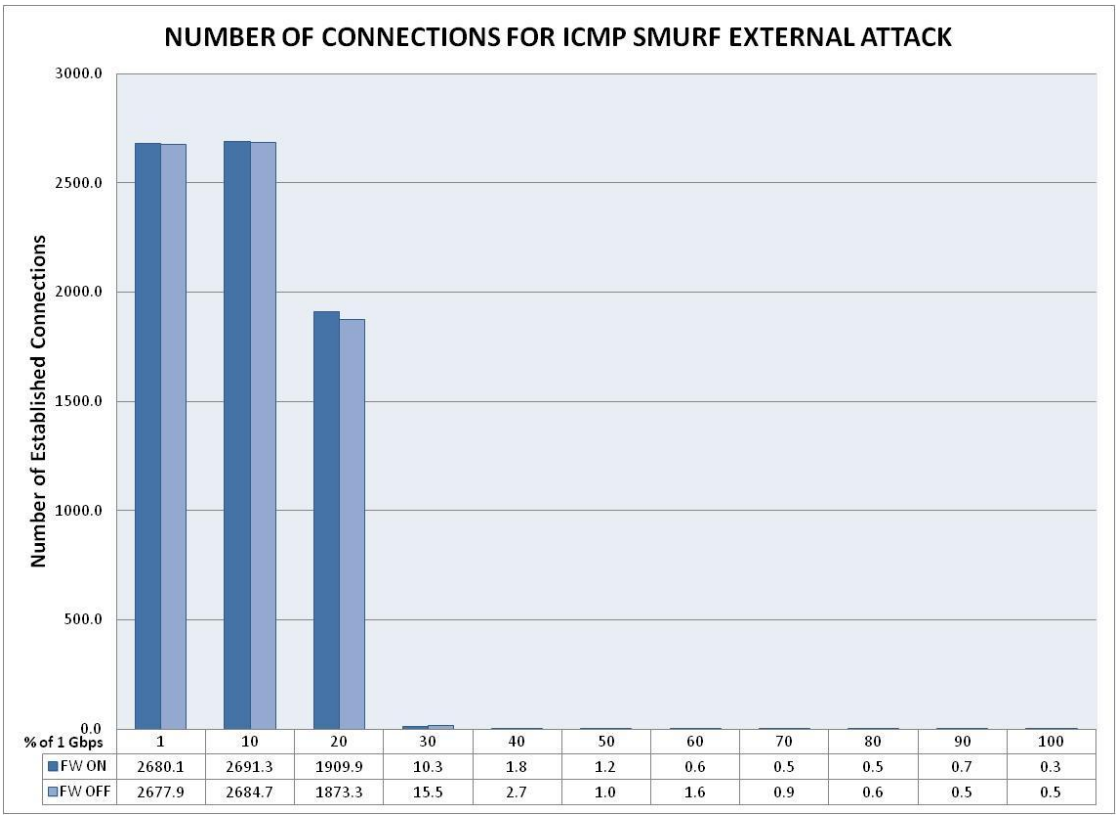
Figure 4.21 - Processor Utilization for Ping Flood Internal Attack on Red Hat Linux 5 Server

4.3.3 Smurf Attack

4.3.3.1 Smurf Flood External Attack. In this section we present the findings for Smurf External Attack. In Figures 4.22 and 4.23, we find the number of connections throughout the attack for both Windows and Linux platforms. In figures 4.24 and 4.25, the processor utilization report is presented for these operating systems.

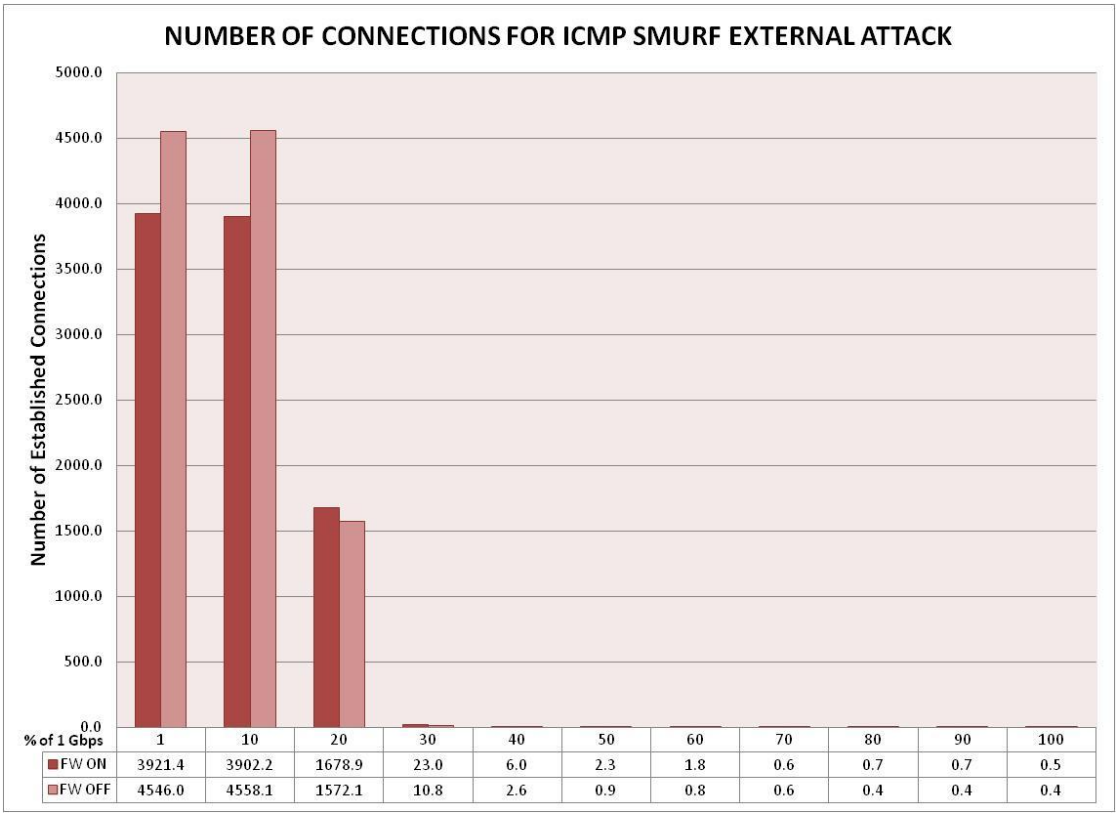
We observe that for low intensity rates of external attack below 10%, there is no impact on any of the two platforms, for this range of attack; Red Hat is performing better hosting more connections than Windows. On the next attack load that is 20%, Windows hosts around 1900 http connections compared to Linux that has 1678 and 1572 connections for both firewall ON and OFF configurations respectively.

For the attack load range of 30% to 100%, both operating systems are not able to sustain enough connections, thus creating a denial of service for legitimate users.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2680.1	2691.3	1909.9	10.3	1.8	1.2	0.6	0.5	0.5	0.7	0.3
FW OFF	2677.9	2684.7	1873.3	15.5	2.7	1.0	1.6	0.9	0.6	0.5	0.5

Figure 4.22 - Connections for Smurf Flood External Attack on Microsoft Windows 2008 Server R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	3921.4	3902.2	1678.9	23.0	6.0	2.3	1.8	0.6	0.7	0.7	0.5
FW OFF	4546.0	4558.1	1572.1	10.8	2.6	0.9	0.8	0.6	0.4	0.4	0.4

Figure 4.23 - Connections for Smurf Flood External Attack on Red Hat Linux 5 Server

In the following two figures 4.24 and 4.25, we analyze the performance of the processor resources for Windows and Linux operating systems.

Linux is taking the lead position for processor utilization performance under this type of attack. It is consuming at most 77% of processor while hosting a greater number of connections than Windows, which consumes up to 96% of processor.

We can appreciate that the Smurf external attack, has not a big impact on the processor performance for any of the two platforms, instead the processor is consumed by the number of http connections that each server is hosting, as this number is decreased due to the increase on the attack load, the processor resources are released.

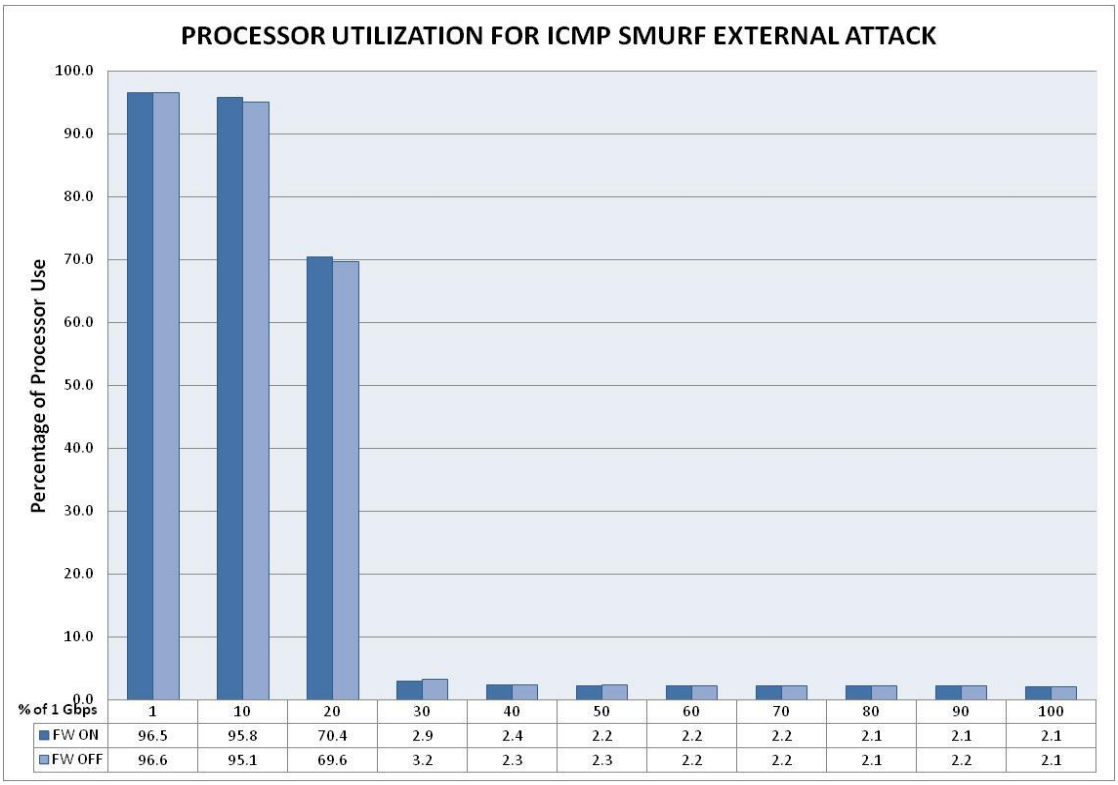


Figure 4.24 - Processor Utilization for Smurf Flood External Attack on MS Windows 2008 Server R2

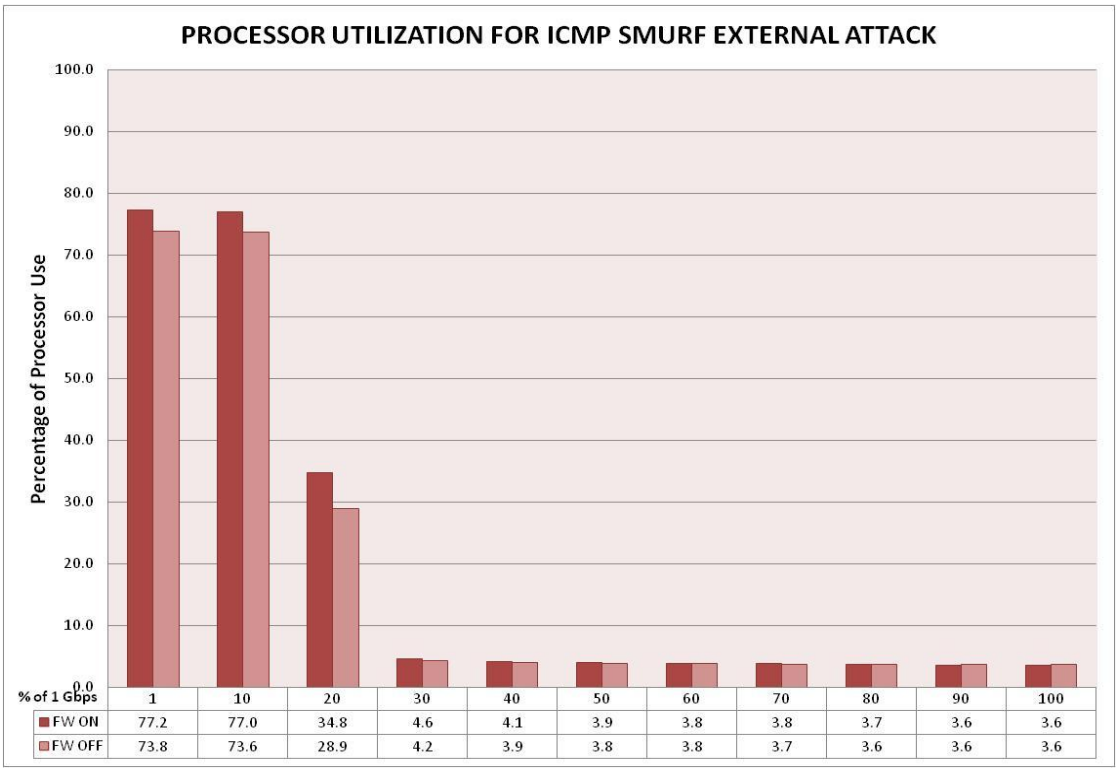
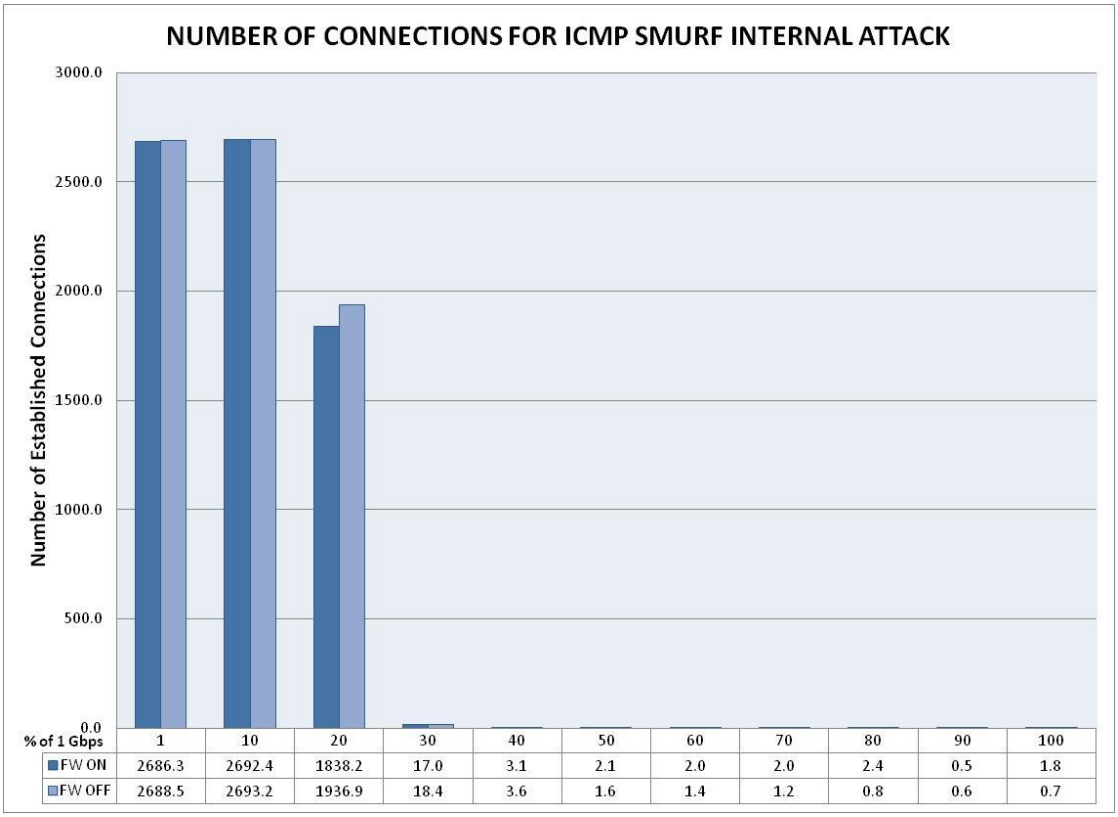


Figure 4.25 - Processor Utilization for Smurf Flood External Attack on Red Hat Linux 5 Server

4.3.3.2 Smurf Flood Internal Attack. In this section we will review the Smurf Flood Internal Attack. Figures 4.26 and 4.27 represent the number of connections hosted by Windows and Linux, while 4.28 and 4.29 display the processor utilization for these platforms, respectively.

We noticed that the impact on the server under this attack, is very similar to the one found for external attack. No significant change was discovered between internal and external Smurf flood attack.

We can conclude that for Smurf flood attack, the impact it has represents the same hazard when it is executed in an internal or an external network, opposite to Ping flood attack, which showed to have a greater impact when the attack was coming from external sources.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2686.3	2692.4	1838.2	17.0	3.1	2.1	2.0	2.0	2.4	0.5	1.8
FW OFF	2688.5	2693.2	1936.9	18.4	3.6	1.6	1.4	1.2	0.8	0.6	0.7

Figure 4.26 - Connections for Smurf Flood Internal Attack on Microsoft Windows 2008 Server R2

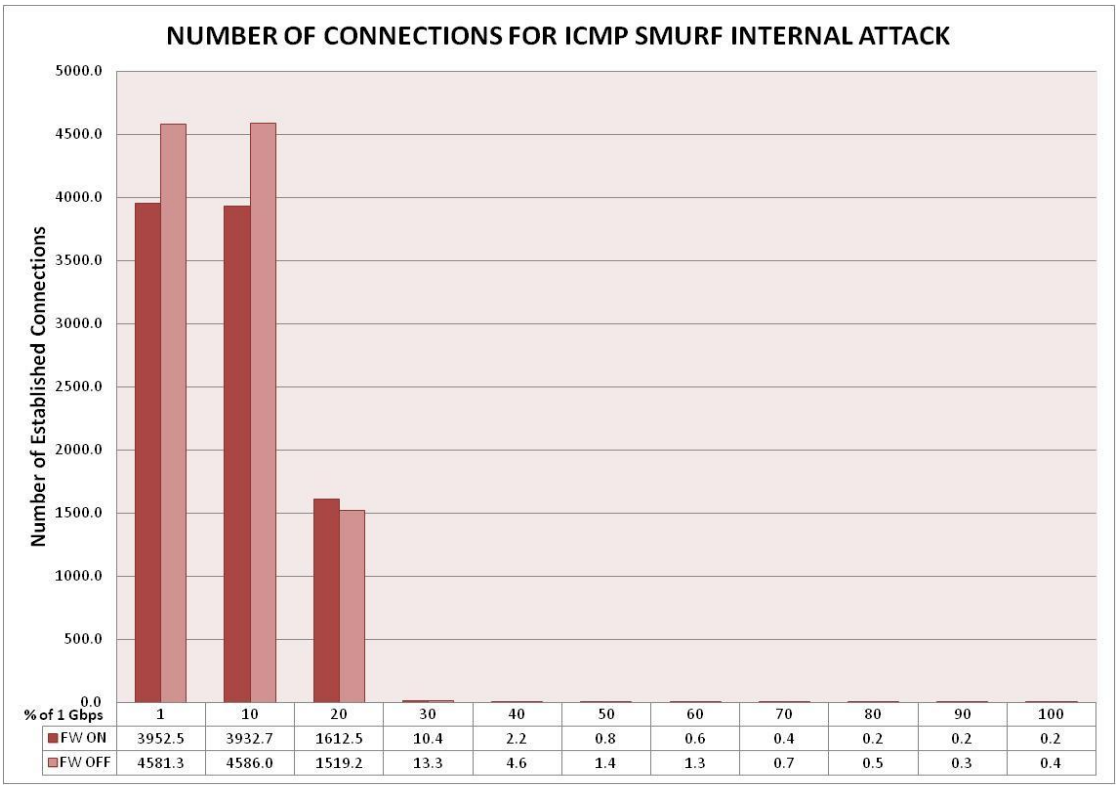


Figure 4.27 - Connections for Smurf Flood Internal Attack on Red Hat Linux 5 Server

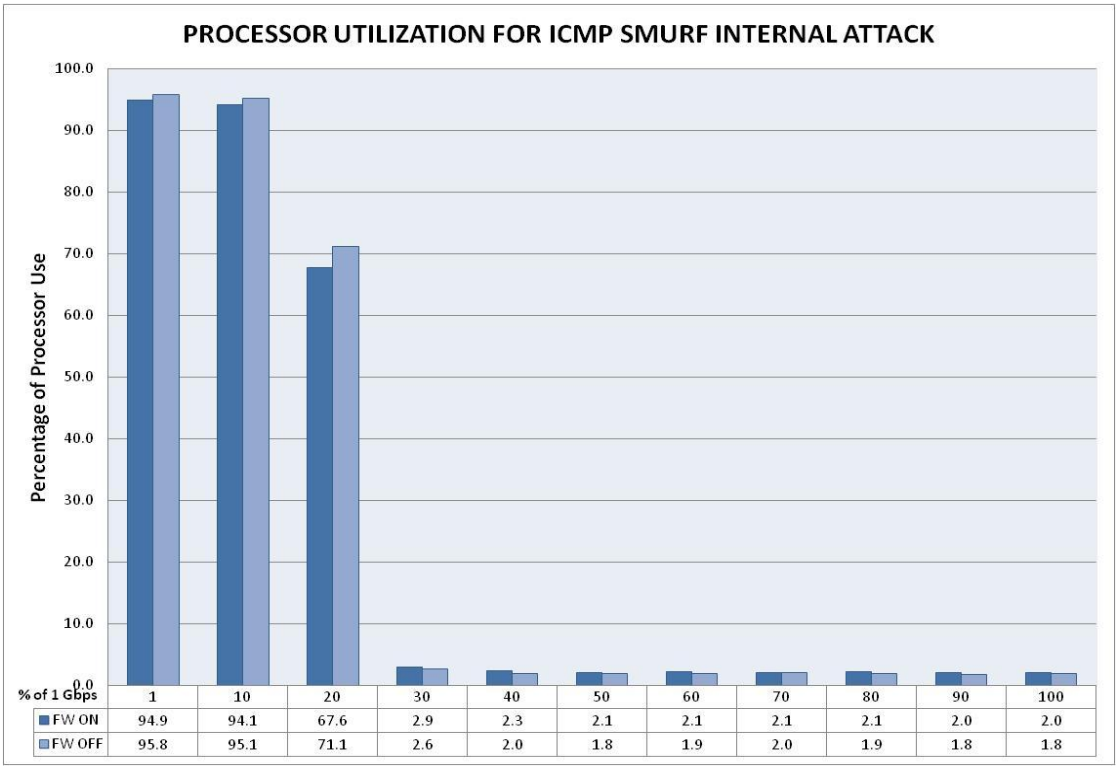
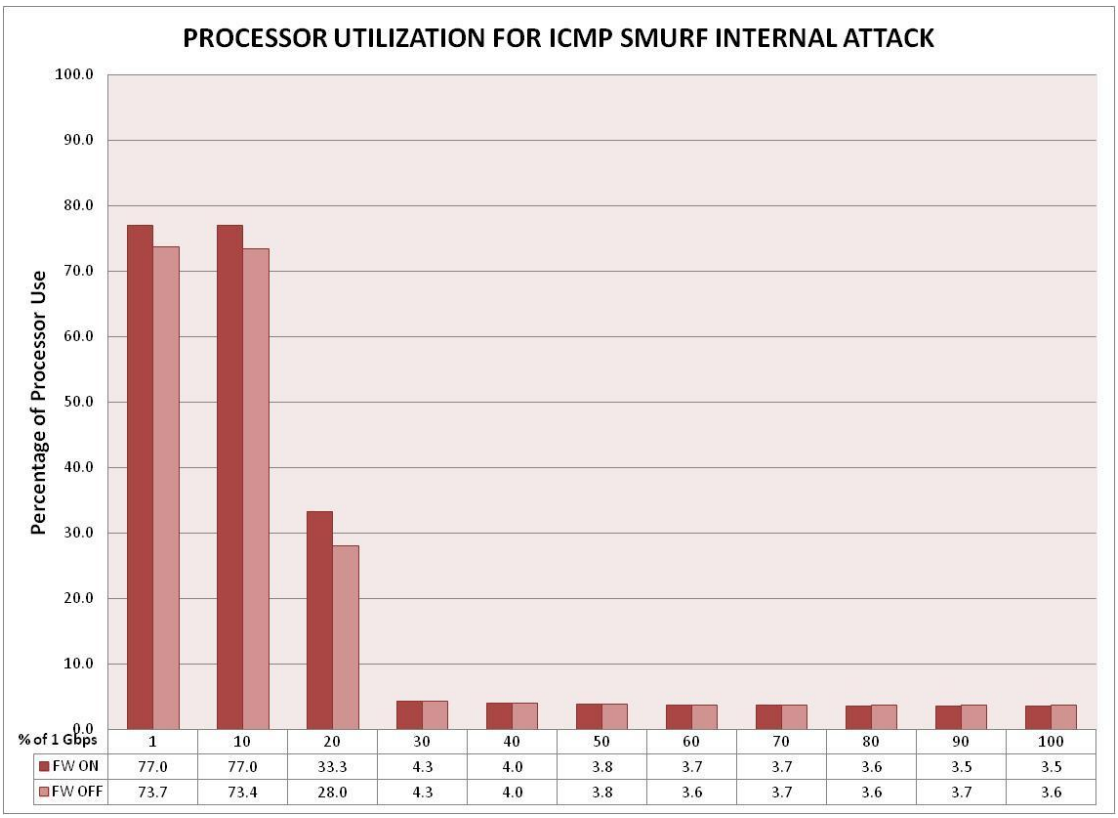


Figure 4.28 - Processor Utilization for Smurf Flood Internal Attack on MS Windows 2008 Server R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	77.0	77.0	33.3	4.3	4.0	3.8	3.7	3.7	3.6	3.5	3.5
FW OFF	73.7	73.4	28.0	4.3	4.0	3.8	3.6	3.7	3.6	3.7	3.6

Figure 4.29 - Processor Utilization for Smurf Flood Internal Attack on Red Hat Linux 5 Server

Even though Smurf Attack did not consume significant processor resources for high attack rates, the impact on the number of connections that were being hosted was lethal, bringing that number close to zero with only 30% of attack rate. This attack proved to be very successful on both platforms for both internal and external configurations.

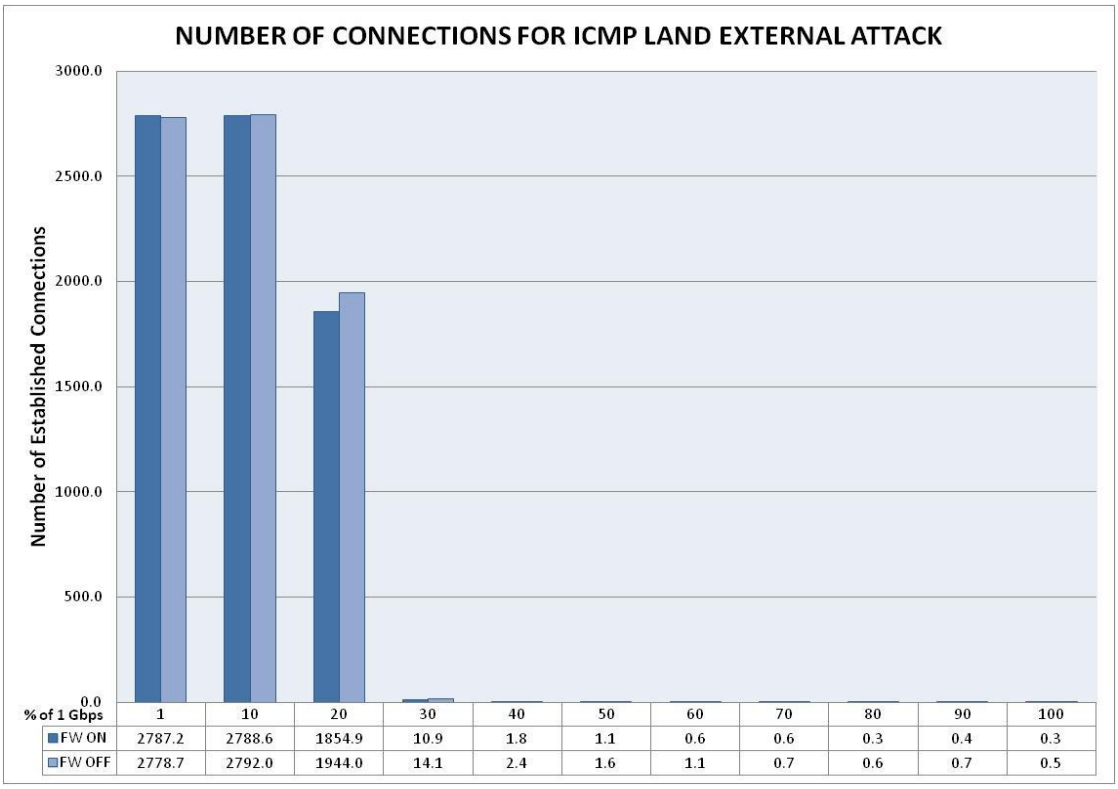
4.3.4 ICMP LAND Attack

4.3.4.1 ICMP LAND External Attack. In this section we can observe the results obtained for the Land external attack. The number of connections that Windows is able to host is represented in Figure 4.30. For both ON and OFF firewall configurations, we can observe a very similar behavior when the server is under disturbance. Windows Server is defending very well for attack loads of 10% and lowers since it is not losing any connections, when it is receiving

20% of attack load there is a small impact in the number of connections decreasing to around 1900 for firewall ON and OFF dispositions, for attack rates of 30% and higher, the operating system is losing the majority of connections being requested by legitimate users, thus, causing a denial of service attack to them.

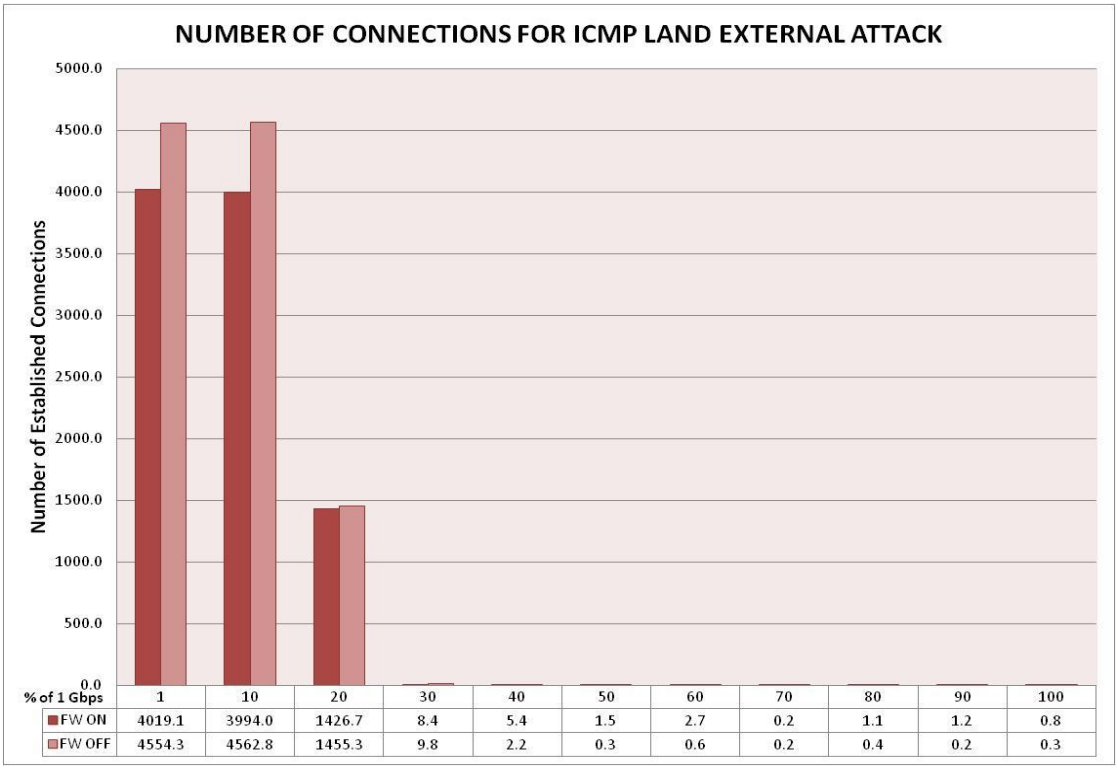
Red Hat proves to host more connections for firewall OFF configuration (Figure 4.31). The number of http connections observed for low attack intensity rates of 10% and lower, is higher than the one found in Windows, but for the attack load of 20%, it is losing a lot of connections staying behind on comparison with Windows. On attack rates of 30% or higher the same behavior from Windows was observed.

Processor utilization performance is shown in Figures 4.32 and 4.33 for Windows and Linux respectively. We can observe that Linux is able to handle better the processor resources compared to Windows, since it is using at most 77% while hosting a greater number of http users. Windows on the other hand, is able to withstand the attack better for the load of 20%. For the rest of the attack both operating systems are handling their processor utilization in a reasonable manner considering also that they are losing the number of connections due to the high intensity traffic of the incoming attack, which drives them to cause denial of service to the legitimate users.



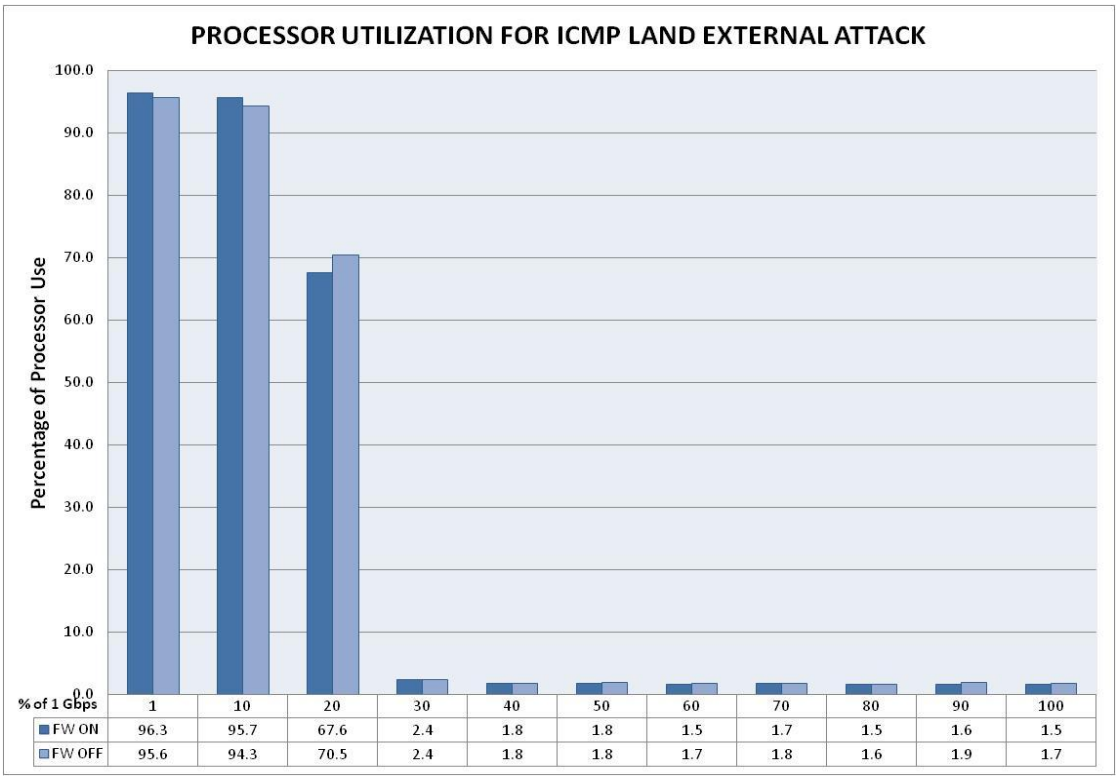
% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2787.2	2788.6	1854.9	10.9	1.8	1.1	0.6	0.6	0.3	0.4	0.3
FW OFF	2778.7	2792.0	1944.0	14.1	2.4	1.6	1.1	0.7	0.6	0.7	0.5

Figure 4.30 - Connections for ICMP Land External Attack on Microsoft Windows 2008 Server R2



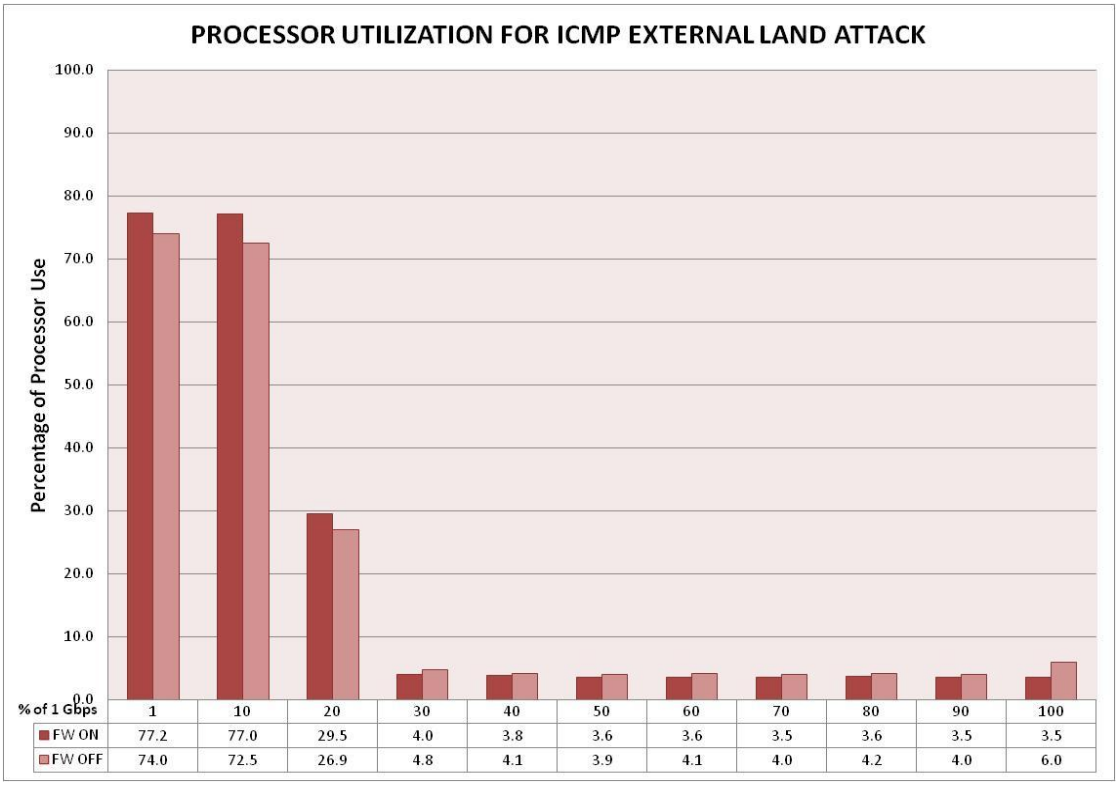
% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	4019.1	3994.0	1426.7	8.4	5.4	1.5	2.7	0.2	1.1	1.2	0.8
FW OFF	4554.3	4562.8	1455.3	9.8	2.2	0.3	0.6	0.2	0.4	0.2	0.3

Figure 4.31 - Connections for ICMP Land External Attack on Red Hat Linux 5 Server



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	96.3	95.7	67.6	2.4	1.8	1.8	1.5	1.7	1.5	1.6	1.5
FW OFF	95.6	94.3	70.5	2.4	1.8	1.8	1.7	1.8	1.6	1.9	1.7

Figure 4.32 - Processor Utilization for ICMP Land External Attack on MS Windows 2008 Server R2



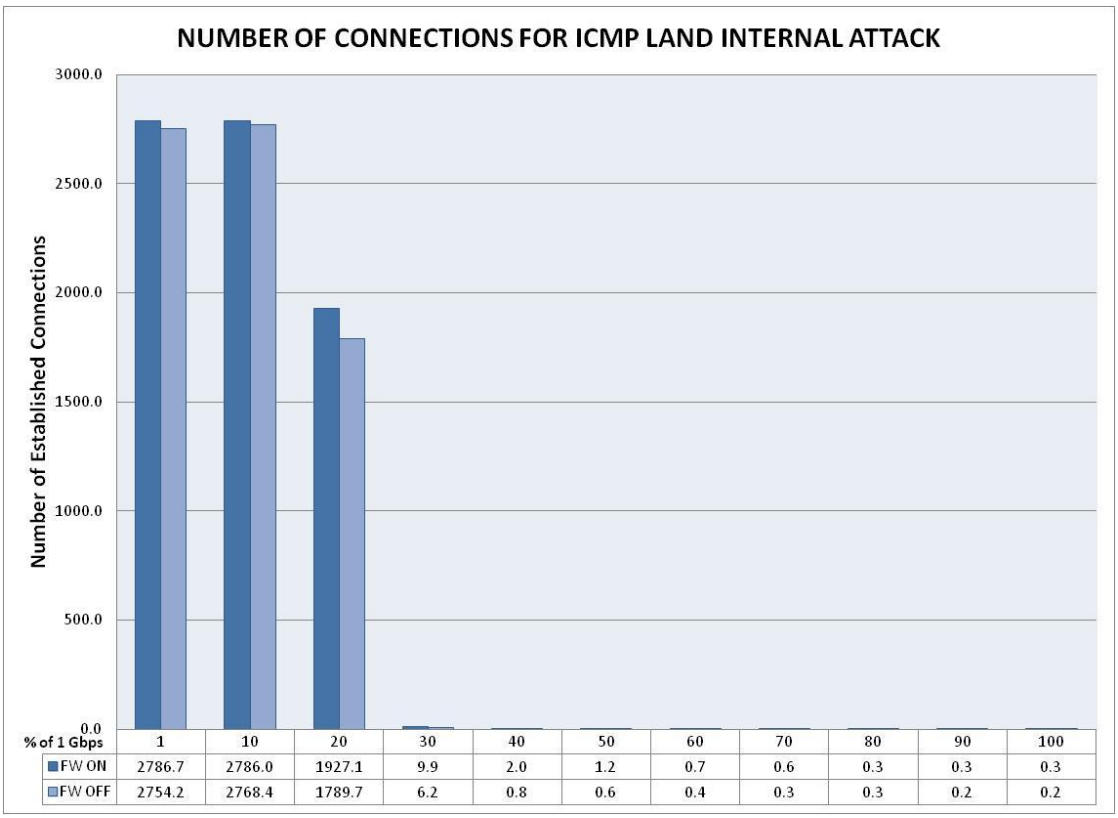
% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	77.2	77.0	29.5	4.0	3.8	3.6	3.6	3.5	3.6	3.5	3.5
FW OFF	74.0	72.5	26.9	4.8	4.1	3.9	4.1	4.0	4.2	4.0	6.0

Figure 4.33 - Processor Utilization for ICMP Land External Attack on Red Hat Linux 5 Server

4.3.4.2 ICMP LAND Internal Attack. For the case of ICMP Land Internal Attack, we observe that both operating systems are behaving similarly to the external attack. In Figures 4.34 thru 4.37 we present the results on http connections for Windows and Linux under this type of attack, and for the processor utilization on the same order.

A stable number of connections were observed for both operating systems when the attack intensity rate is 10% or lower. When the attack load is incremented to 20%, the number of connections that both operating systems are able to host is decrease, for this case, Windows takes the lead position being able to sustain about 1800 http users.

About the processor utilization performance, we have noticed in previous attacks that Windows is performing poorly compared to Linux, since Linux is always reserving some room for other operations using the processor resources at a maximum level of 80%, whereas Windows is consuming up to 98% in the most of the cases to process the incoming connections.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2786.7	2786.0	1927.1	9.9	2.0	1.2	0.7	0.6	0.3	0.3	0.3
FW OFF	2754.2	2768.4	1789.7	6.2	0.8	0.6	0.4	0.3	0.3	0.2	0.2

Figure 4.34 - Connections for ICMP Land Internal Attack on Microsoft Windows 2008 Server R2

We can conclude this section saying that both operating systems have proven to have the same behavior when they are subjected to the ICMP Land attacks on both internal and external configurations.

We observe as well that in overall, Linux performs better than Windows when it is under this type of attacks by being able to host a greater number of connections consuming less processor resources. The only case that Windows proved to perform better is when it was under 20% of attack load since it was able to sustain more connections at this rate compared to Linux.

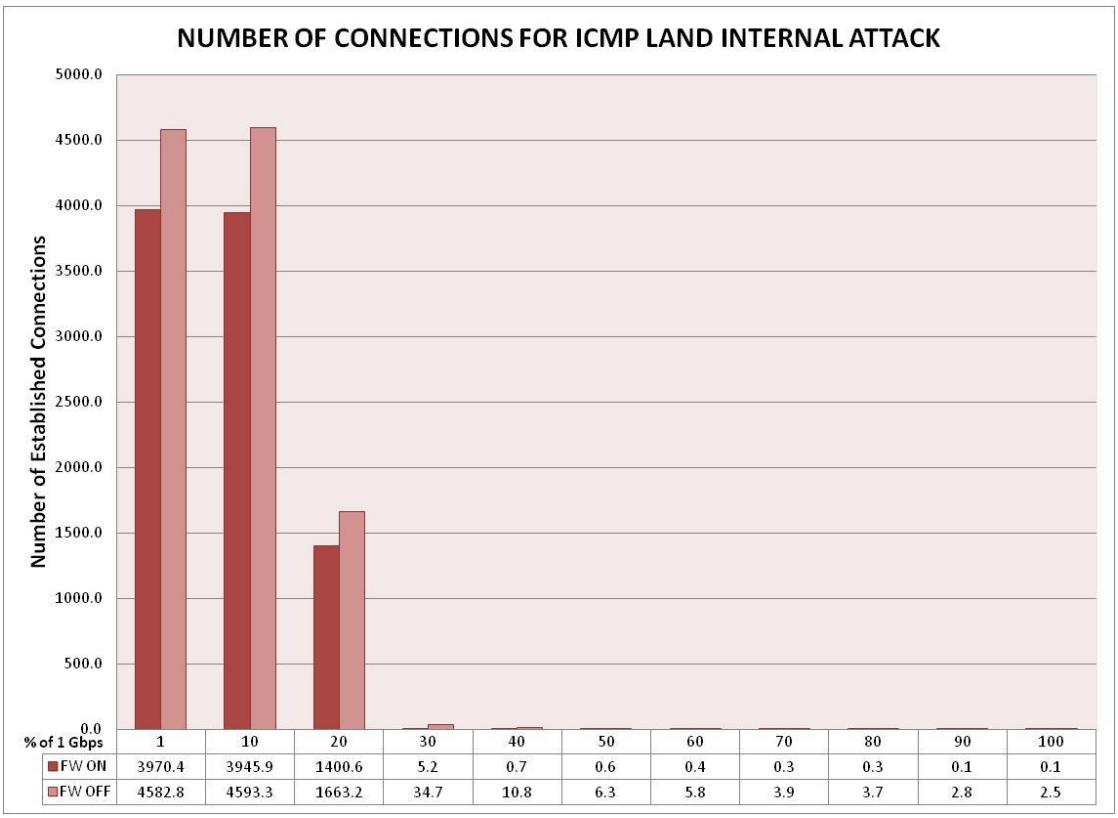
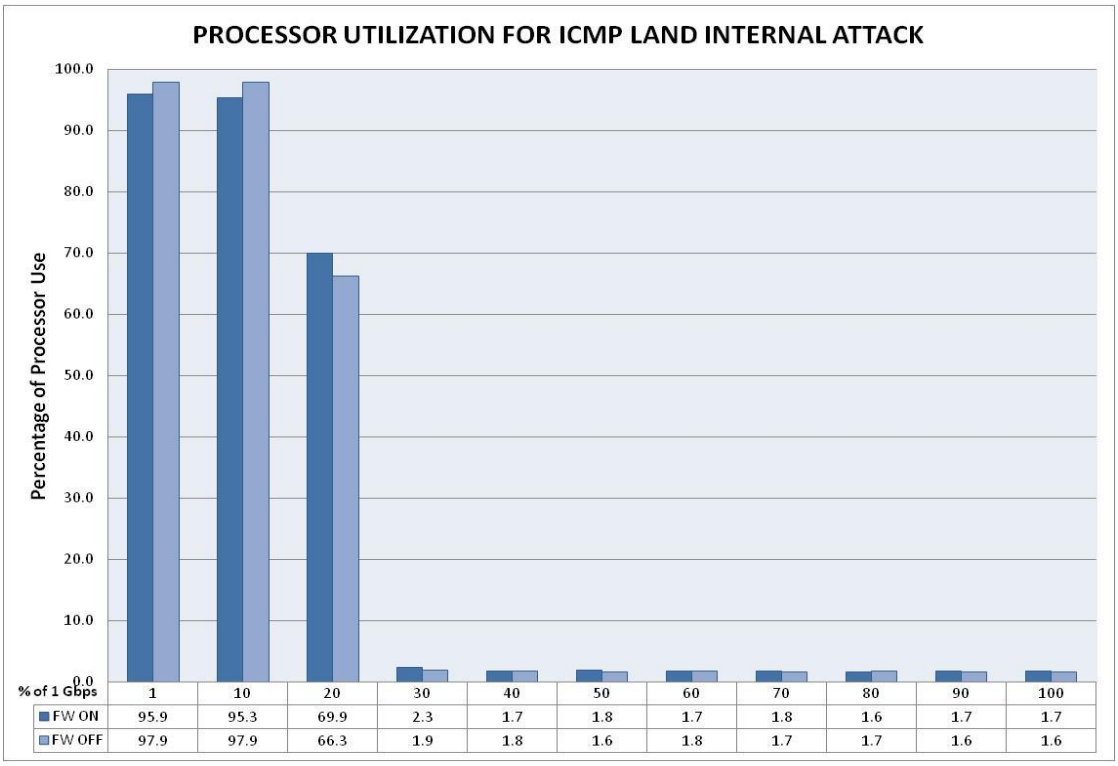
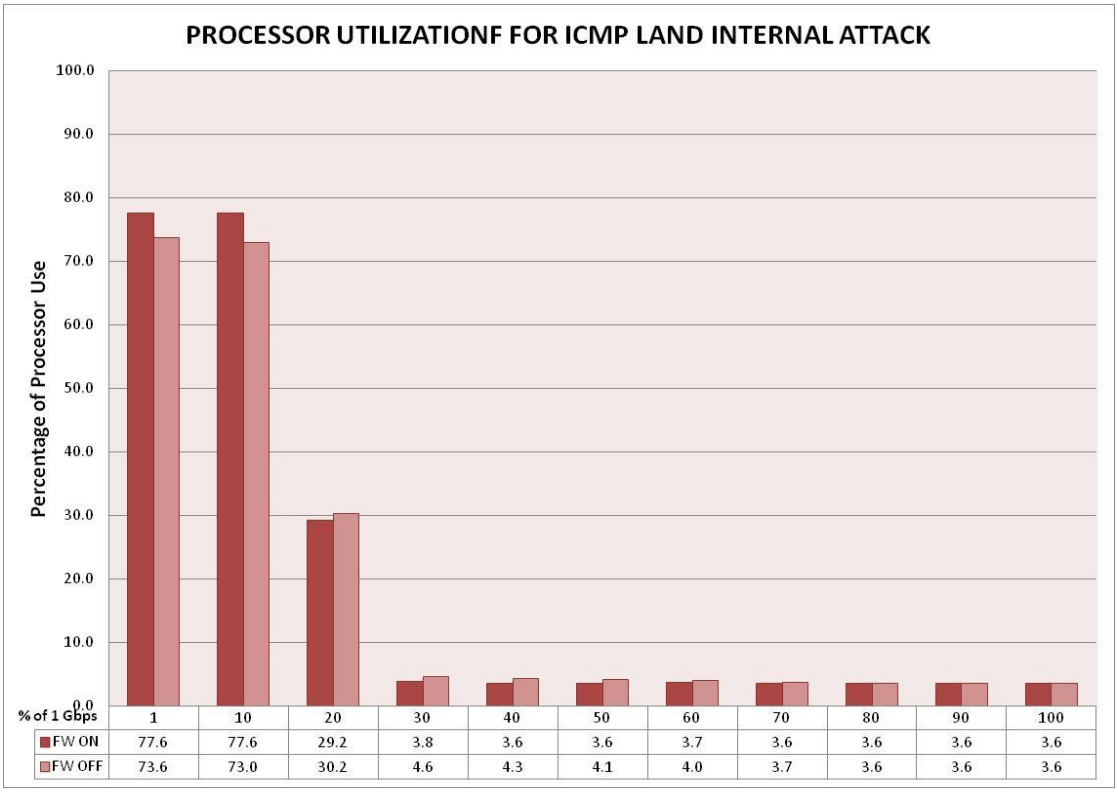


Figure 4.35 - Connections for ICMP Land Internal Attack on Red Hat Linux 5 Server



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	95.9	95.3	69.9	2.3	1.7	1.8	1.7	1.8	1.6	1.7	1.7
FW OFF	97.9	97.9	66.3	1.9	1.8	1.6	1.8	1.7	1.7	1.6	1.6

Figure 4.36 - Processor Utilization for ICMP Land Internal Attack on MS Windows 2008 Server R2



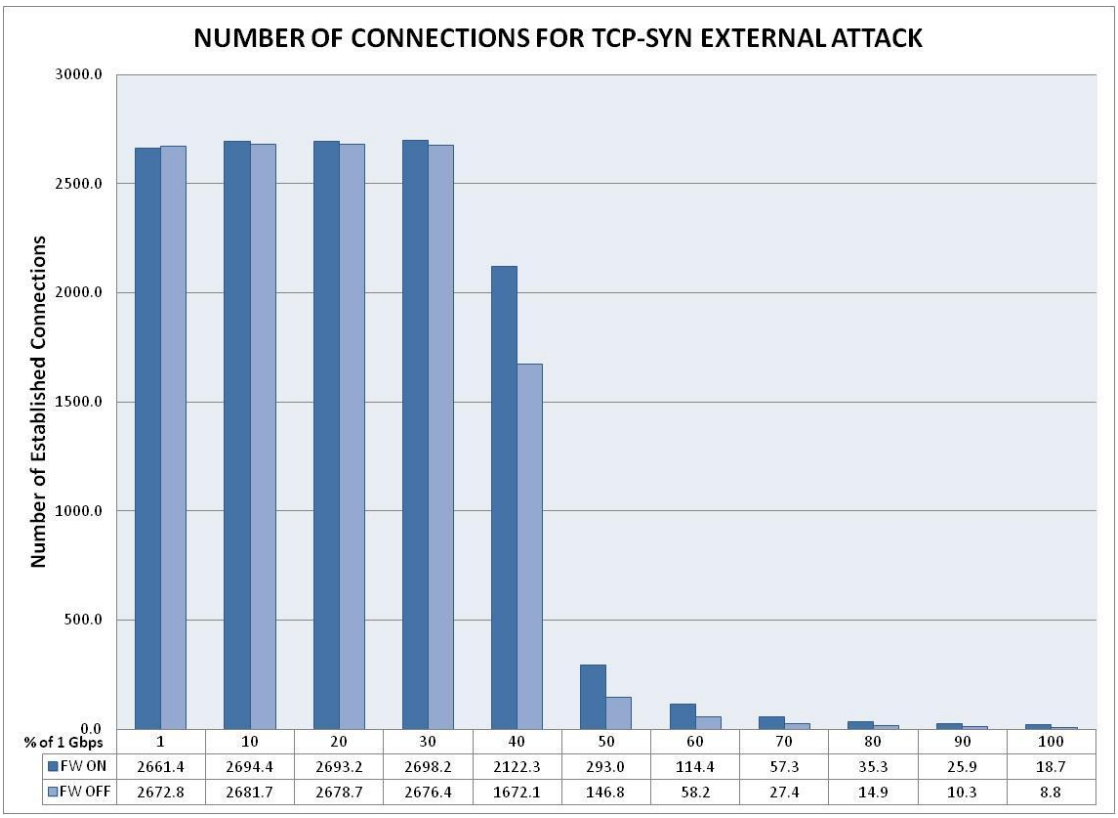
% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	77.6	77.6	29.2	3.8	3.6	3.6	3.7	3.6	3.6	3.6	3.6
FW OFF	73.6	73.0	30.2	4.6	4.3	4.1	4.0	3.7	3.6	3.6	3.6

Figure 4.37 - Processor Utilization for ICMP Land Internal Attack on Red Hat Linux 5 Sever

4.3.5 TCP-SYN Flood Attack

4.3.5.1 TCP-SYN Flood External Attack. In this section we will describe the findings from the experiments done on Windows and Linux server under TCP-SYN Flood External Attack.

In Figure 4.38 we can observe how Windows is performing under this type of attack. It shows to defend very well the connections up to a 30% of the attack rate, compared to previous attacks where it was starting to lose connections at only 20% of the attack rate. At 40%, the number of connections is decreased in a greater way for firewall OFF configuration, but it is still withstanding the attack on a very good way. When the attack load is 50%, Windows has lost most of the connections still keeping 293 for firewall ON and 146 when firewall is disabled. After 60% of attack load, we can still get some connections but the number is very low, although the firewall ON configurations proves to withstand greater number of connections compared to firewall OFF.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2661.4	2694.4	2693.2	2698.2	2122.3	293.0	114.4	57.3	35.3	25.9	18.7
FW OFF	2672.8	2681.7	2678.7	2676.4	1672.1	146.8	58.2	27.4	14.9	10.3	8.8

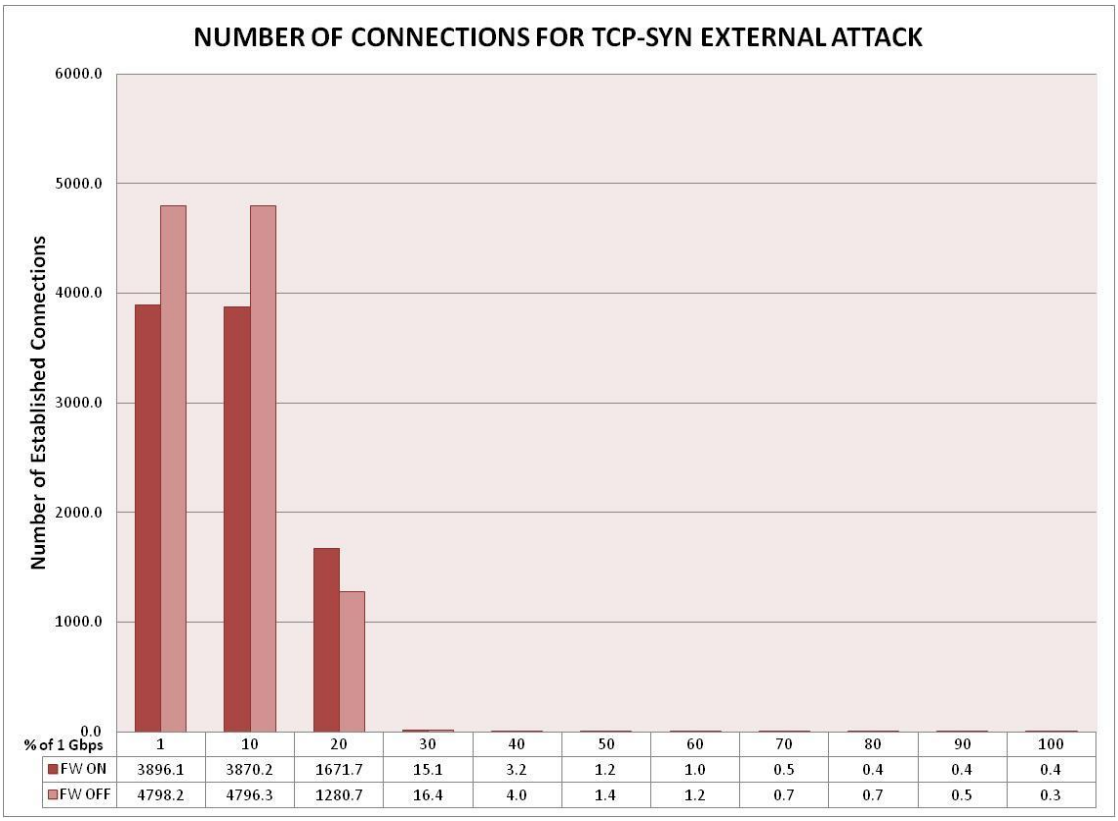
Figure 4.38 - Connections for TCP-SYN External Attack on Microsoft Windows 2008 Server R2

On the case of the Red Hat Server, we can observe its connection performance in Figure 4.39. Linux is performing better that Windows only for attack rates of 10% and lower, since it is able to host a greater number of connections, but in general for the rest of the attack, it is performing poorly because for this case, Linux is not able to withstand higher attack rates thus, losing most of the connections from 30% and higher loads.

This is a result that was not very obvious at the beginning since the TCP-SYN attack is known as being a very dangerous kind of attack which can bring down the number of connections very easily.

We can say Windows is performing very well in defending against this attack since it is able to keep connections throughout the whole duration of the disturbance. Linux on the other hand is

not showing any protections against TCP-SYN, and it is not able to withstand the attack at high intensity rates.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	3896.1	3870.2	1671.7	15.1	3.2	1.2	1.0	0.5	0.4	0.4	0.4
FW OFF	4798.2	4796.3	1280.7	16.4	4.0	1.4	1.2	0.7	0.7	0.5	0.3

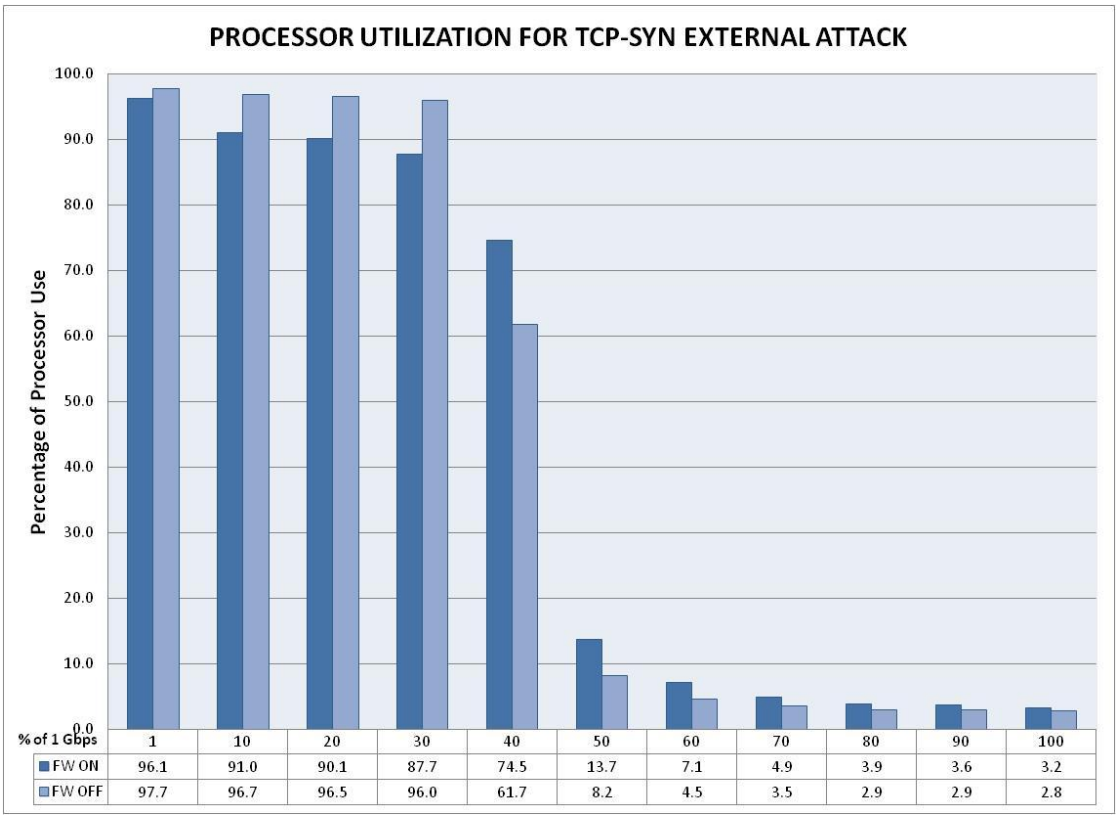
Figure 4.39 - Connections for TCP-SYN External Attack on Red Hat Linux 5 Server

The processor utilization graphics are presented in Figures 4.40 and 4.41 for Windows and Linux servers respectively when they are being tested for TCP-SYN external attack.

Windows operating system is using more processor resources when the firewall is deactivated for attack loads of 30% and smaller. After that, it is using more resources with firewall ON configuration for the fact that it is hosting a greater number of connections on this firewall setup.

Linux on the other hand, has a better processor performance for the whole attack, but it is performing very poor defending against the attack traffic. That is why the processor resources

are released since it is losing most of the connections thus, causing a denial of service to legitimate traffic.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	96.1	91.0	90.1	87.7	74.5	13.7	7.1	4.9	3.9	3.6	3.2
FW OFF	97.7	96.7	96.5	96.0	61.7	8.2	4.5	3.5	2.9	2.9	2.8

Figure 4.40 - Processor Utilization for TCP-SYN External Attack on Microsoft Windows 2008 Server R2

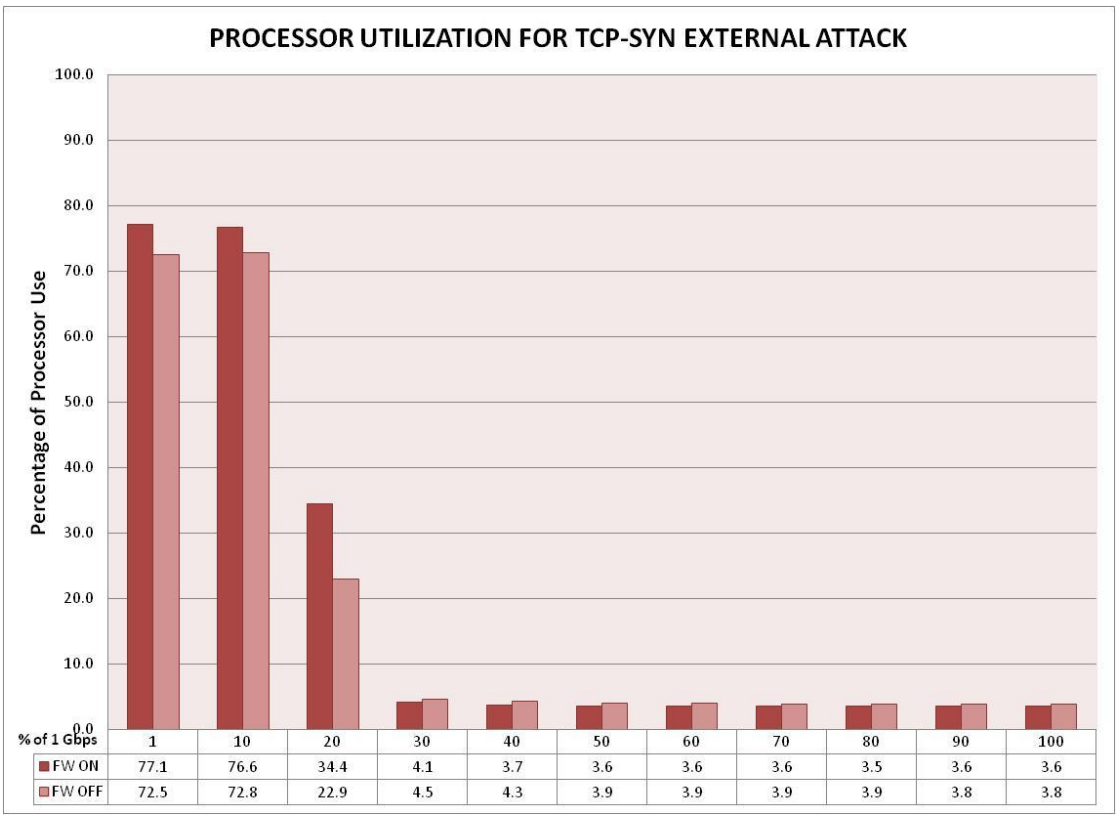


Figure 4.41 - Processor Utilization for TCP-SYN External Attack on Red Hat Linux 5 Server

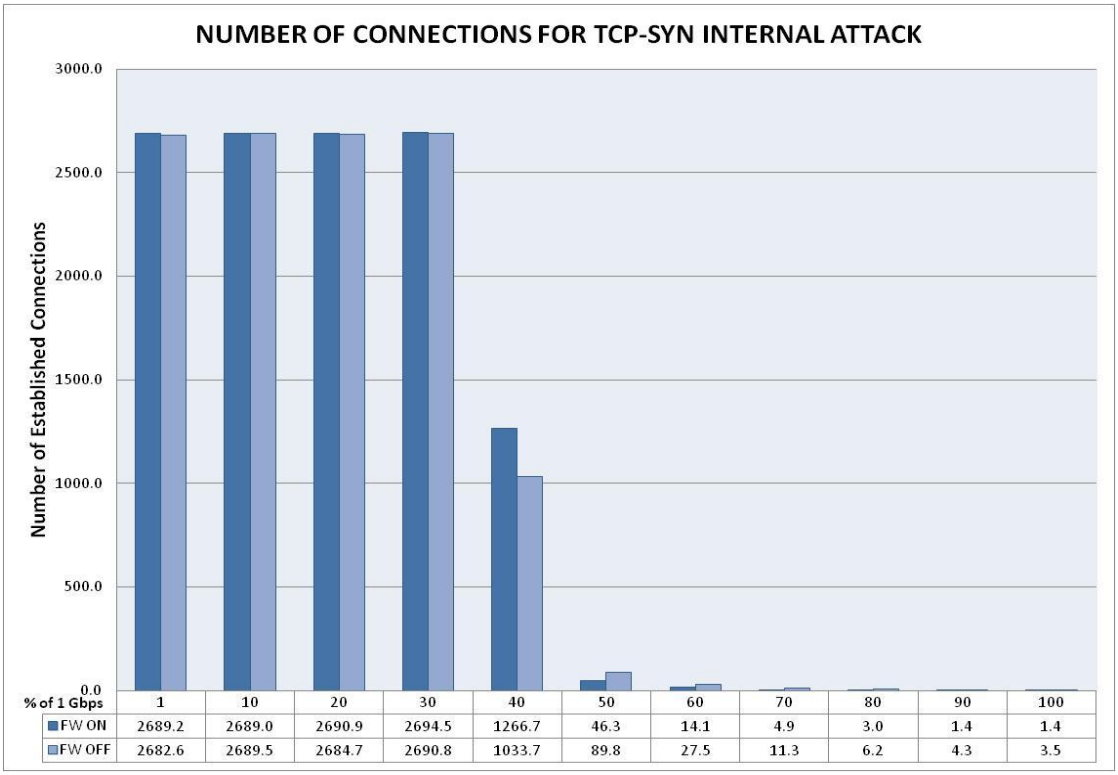
4.3.5.2 TCP-SYN Flood Internal Attack. In this section we will review the results obtained for Windows and Linux when they are being subject of the TCP-SYN flood internal attack configuration.

For this attack configuration, we observe a very stable Windows platform keeping all the incoming http connections for attack loads of 30% and lower. When the attack load is increased to 40% we are able to notice a significant number of connections for both; firewall ON and OFF configurations. From 50% attack load and higher, Windows is losing most of the connections due to the intensity of the attack traffic. If we compare Windows performance under internal and external attacks, it shows to be performing very similarly, still being able to host more connections for the external attack, thus, we can conclude that the internal attack for TCP-SYN

flood represents a higher threat for this operating system. The test results for Windows are presented graphically in Figure 4.42.

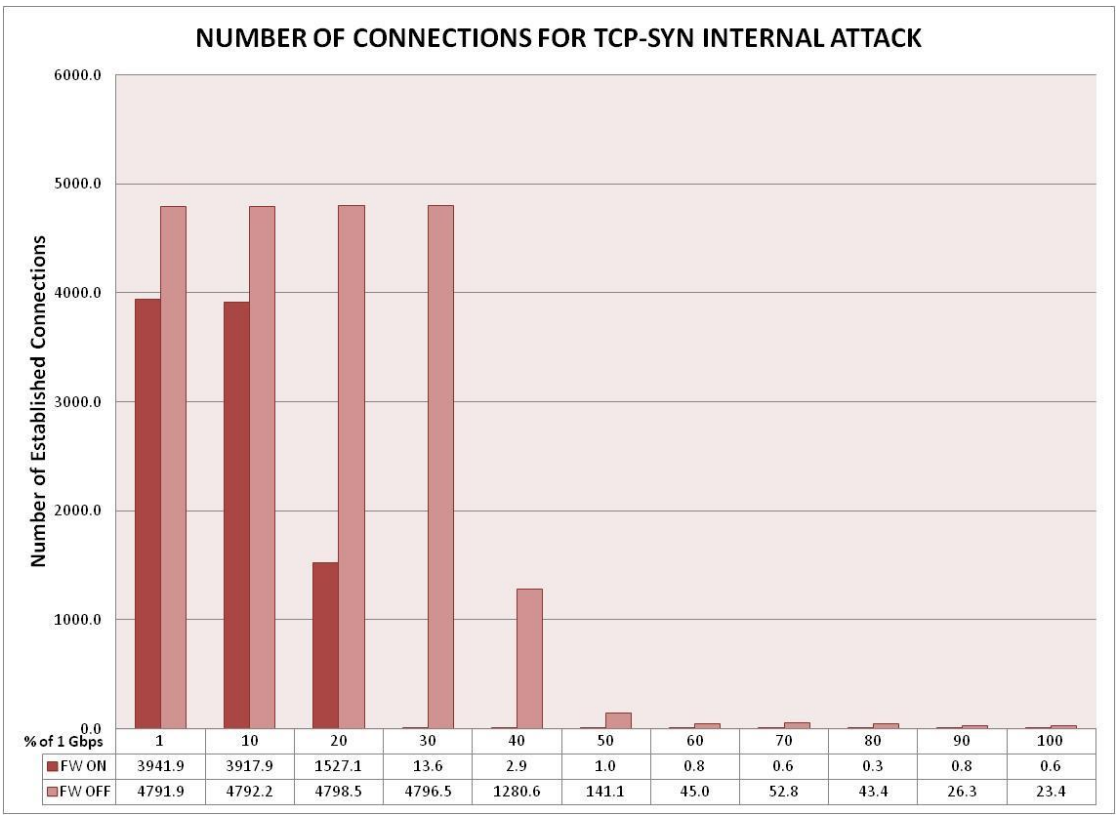
In Figure 4.43 we observe the behavior of Linux under this attack type. Usually we have observed that Linux is able to host a greater number of connections when its firewall is disabled compared to when it is enabled. This type of internal attack is not the exception, but the results prove that Linux is performing much better with the firewall OFF. It is able to sustain almost 4800 connections up to a 30% attack load, compared to the firewall ON at this rate, which has already lost most of the connections. At the next attack load that is 40%, the firewall OFF setup still is able to host 1280 connections and for further attack loads, it is incurring into a denial of service losing the majority number of connections.

For the internal attack using TCP-SYN flood, Windows has proved to perform better than Linux being able to survive for greater attack loads for both Firewall configurations.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2689.2	2689.0	2690.9	2694.5	1266.7	46.3	14.1	4.9	3.0	1.4	1.4
FW OFF	2682.6	2689.5	2684.7	2690.8	1033.7	89.8	27.5	11.3	6.2	4.3	3.5

Figure 4.42 - Connections for TCP-SYN Internal Attack on Microsoft Windows 2008 Server R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	3941.9	3917.9	1527.1	13.6	2.9	1.0	0.8	0.6	0.3	0.8	0.6
FW OFF	4791.9	4792.2	4798.5	4796.5	1280.6	141.1	45.0	52.8	43.4	26.3	23.4

Figure 4.43 - Connections for TCP-SYN Internal Attack on Red Hat Linux 5 Server

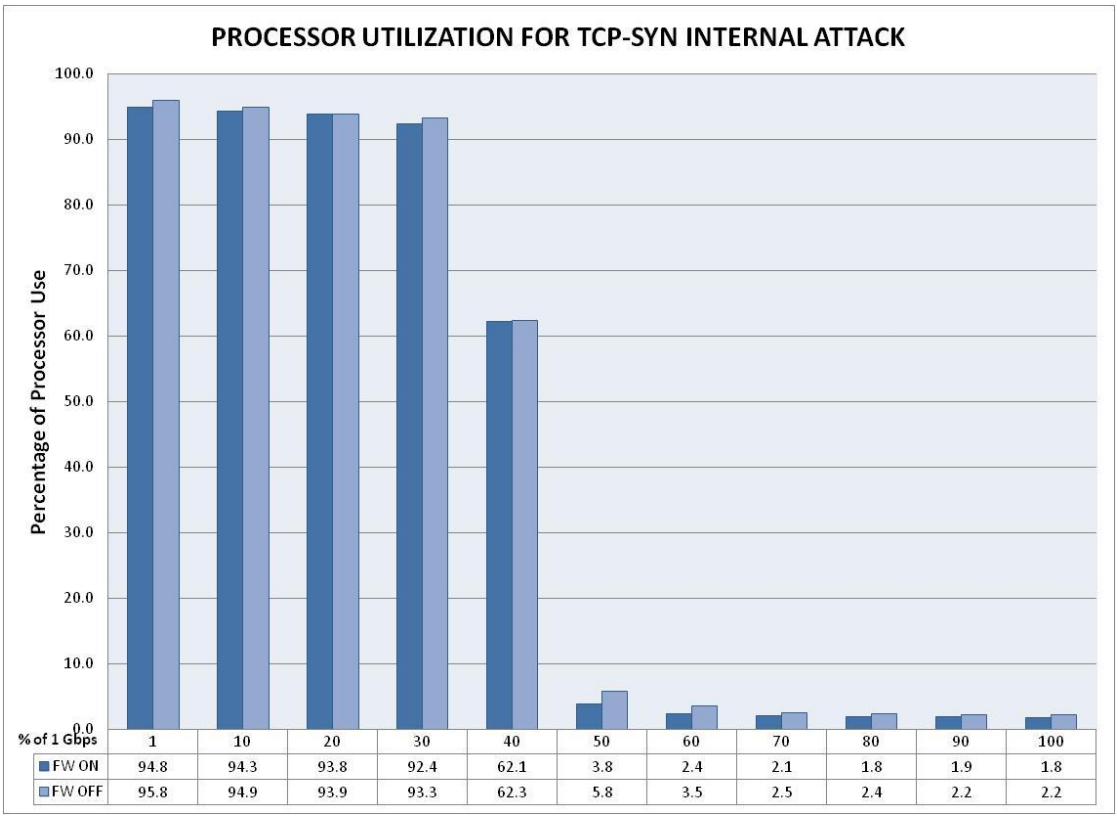
For processor utilization analysis under TCP-SYN internal attack, we present the following Figures 4.44 and 4.45 for Windows and Linux, respectively.

We can observe that the processor utilization for both operating systems is related directly to the number of connections they are able to hold.

For Windows the behavior is very similar for firewall ON and OFF configurations, whereas in Linux the firewall OFF represents a greater consumption of resources compared to firewall ON. This is due to the fact that it is not able to sustain enough connections for the second mode.

If we compare processor resources of Windows with Linux, we find that Linux is keeping a greater amount of processor resources available. Windows has shown to perform better in an overall rating for TCP-SYN attack internal and external configurations when compared to Linux,

since it is able to host connections at higher level of attack loads. Linux is performing poorly incurring into denial of service for even small loads.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	94.8	94.3	93.8	92.4	62.1	3.8	2.4	2.1	1.8	1.9	1.8
FW OFF	95.8	94.9	93.9	93.3	62.3	5.8	3.5	2.5	2.4	2.2	2.2

Figure 4.44 - Processor Utilization for TCP-SYN Internal Attack on Microsoft Windows 2008 Server R2

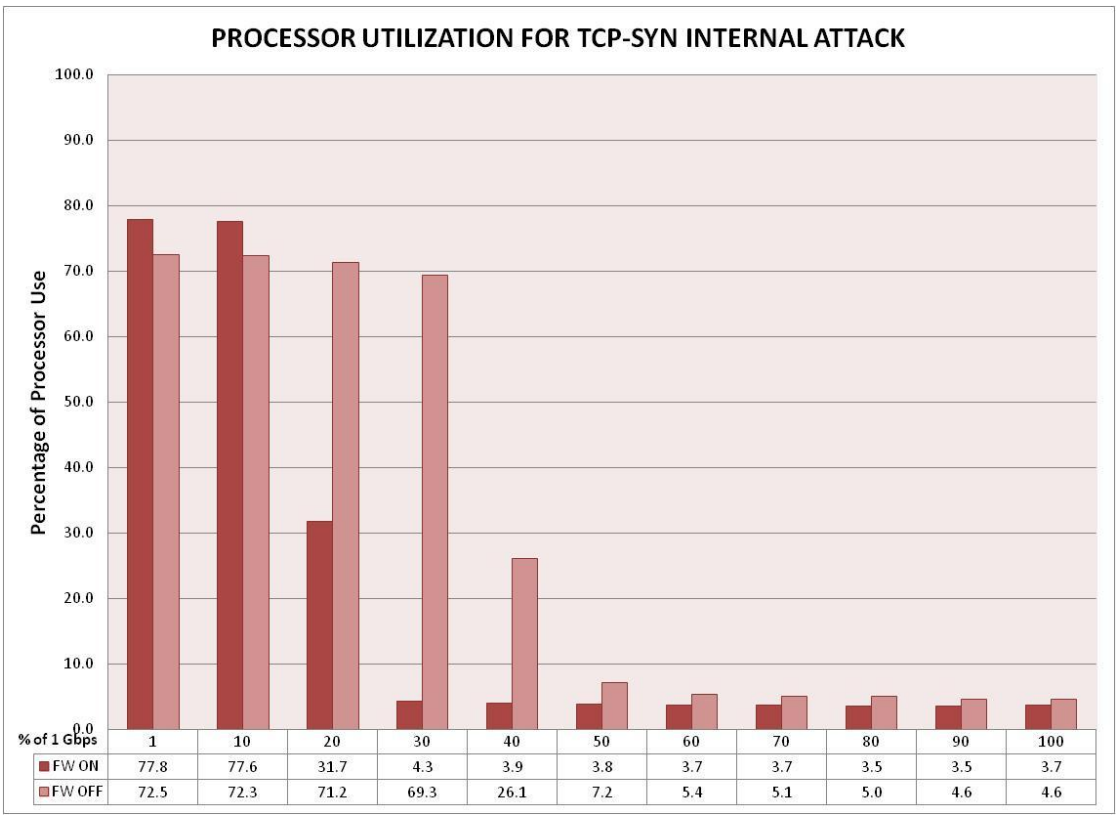


Figure 4.45 - Processor Utilization for TCP-SYN Internal Attack on Red Hat Linux 5 Server

4.3.6 UDP Flood Attack

In this section we will be reviewing the results obtained for the Windows and Linux server operating systems when they are under UDP Flood External and Internal Attacks.

4.3.6.1 UDP Flood External Attack. We can observe the graphics showing this type of attack for number of connections on Windows in Figure 4.46, and for Linux in Figure 4.47.

Even though UDP is located on the layer 4 of the TCP/IP protocol suite, similarly to TCP, the performance observe for this attack is very different from the one of the TCP-SYN attack.

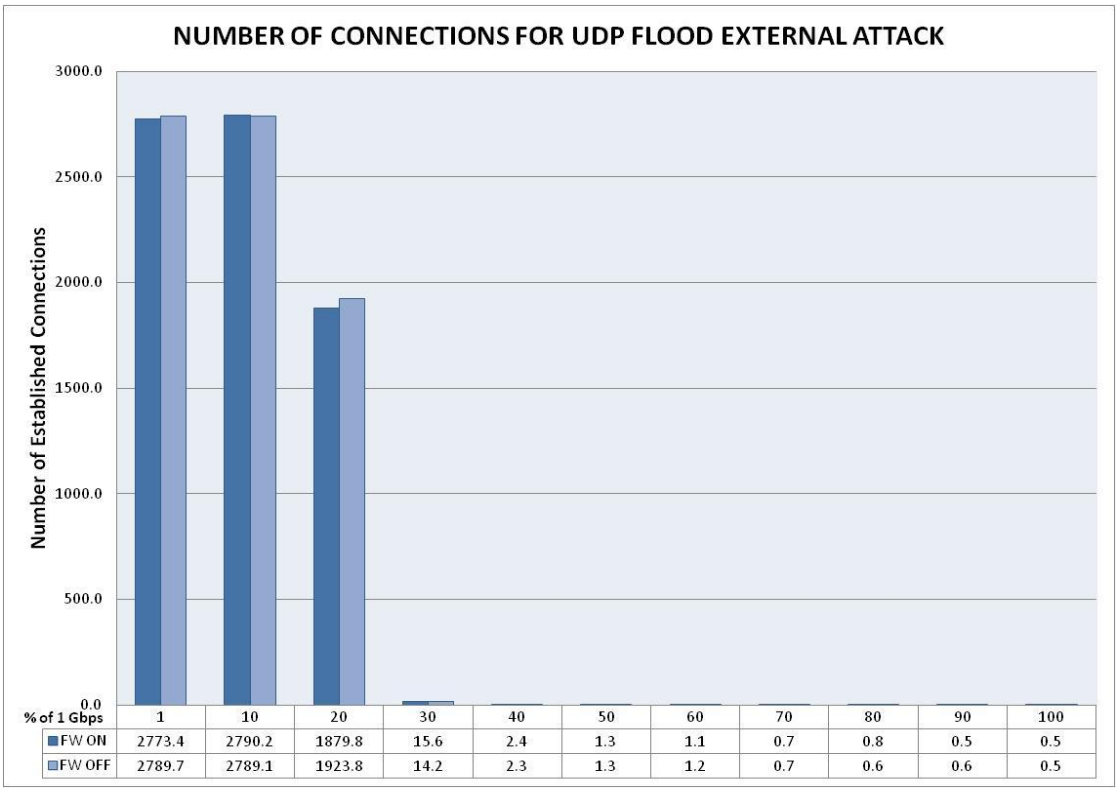
Windows is hosting a complete number of connections only for low intensity attack rates of 10% and lower. When the attack rate is 20%, connections are diminished but they still survive to

the attack. From 30% to 100% the attack is defeating the Windows operating system creating a denial of service to legitimate users, and the connections are dropped to almost zero.

Linux is behaving very similarly to Windows on this attack, only that the number of connections it is able to host for low intensity attack rates of 10% and lower is larger than Windows. When the attack load is increased to 20%, Linux still can host some connections but from 30% to 100% the connection rate is lost and Linux incurs into a denial of service at these rates.

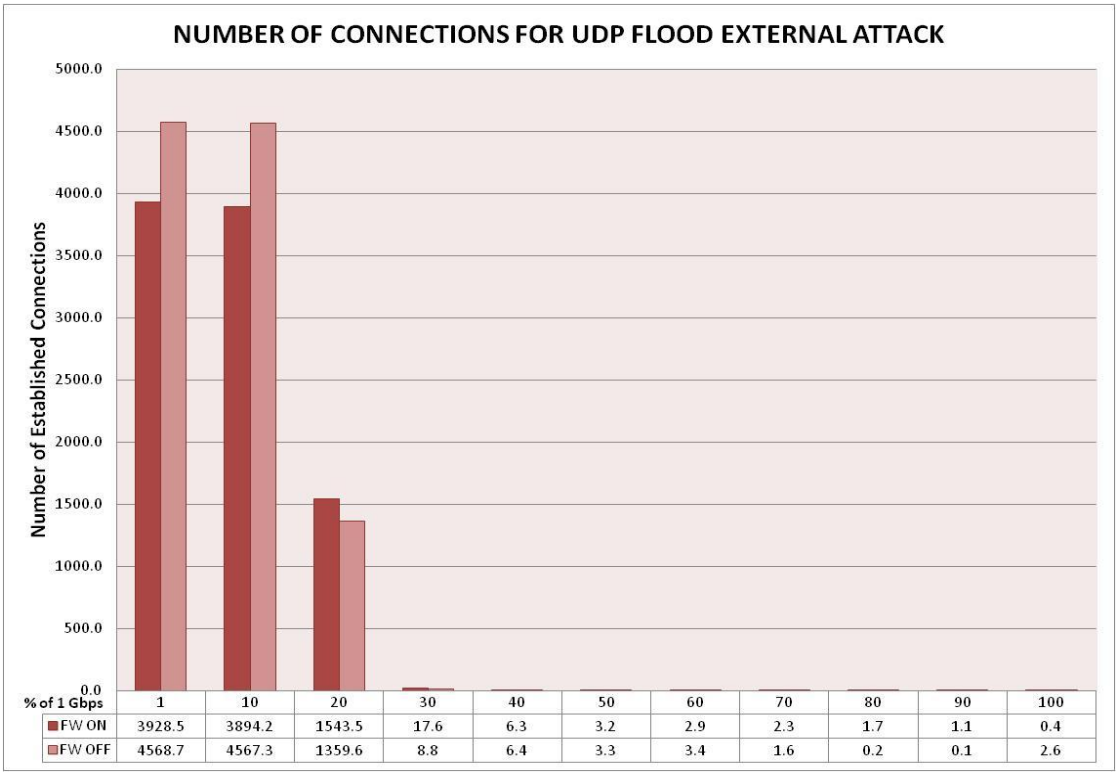
Regarding processor performance, Figures 4.48 and 4.49 present the measurements obtained for Windows and Linux operating systems. Linux is handling its resources better than Windows since at most it is consuming up to 77% of processor compared to 96% of processor consumption when both operating systems are hosting the greater number of connections.

For high attack loads of 30% or more, both operating systems lose most of their connections, and the processor resources are released and remain low for the rest of the attack.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2773.4	2790.2	1879.8	15.6	2.4	1.3	1.1	0.7	0.8	0.5	0.5
FW OFF	2789.7	2789.1	1923.8	14.2	2.3	1.3	1.2	0.7	0.6	0.6	0.5

Figure 4.46 - Connections for UDP Flood External Attack on Microsoft Windows 2008 Server R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	3928.5	3894.2	1543.5	17.6	6.3	3.2	2.9	2.3	1.7	1.1	0.4
FW OFF	4568.7	4567.3	1359.6	8.8	6.4	3.3	3.4	1.6	0.2	0.1	2.6

Figure 4.47 - Connections for UDP Flood External Attack on Red Hat Linux 5 Server

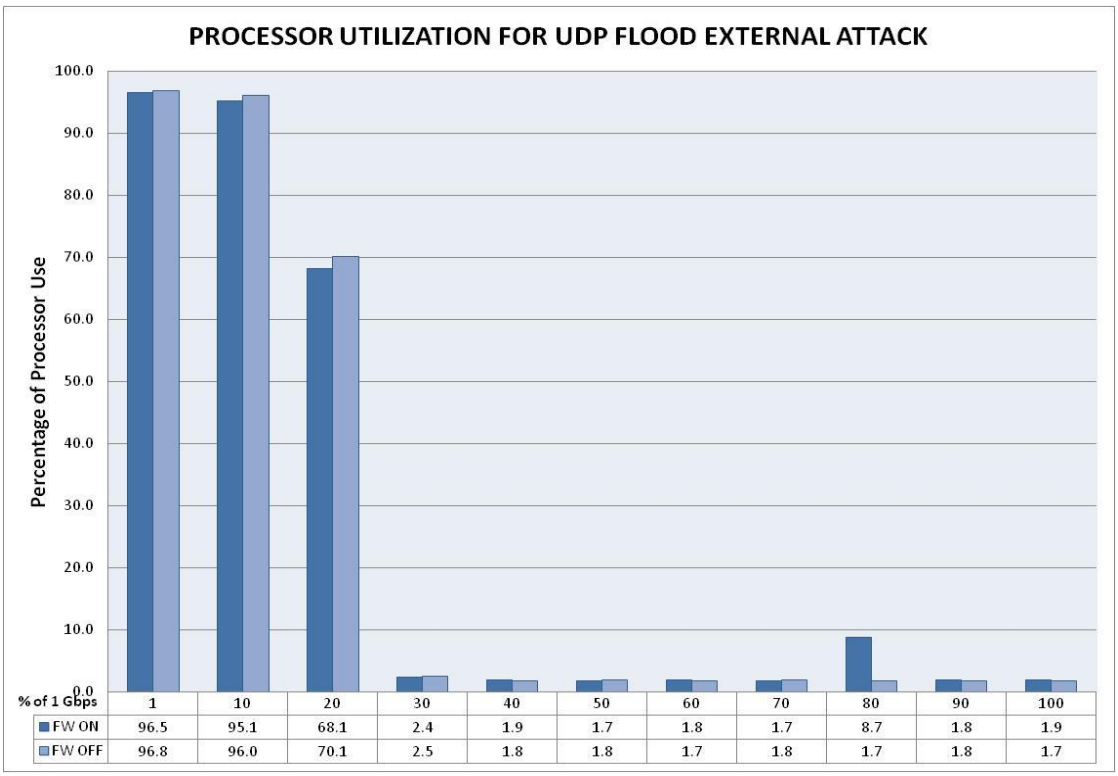


Figure 4.48 - Processor Utilization for UDP Flood External Attack on MS Windows 2008 Server R2

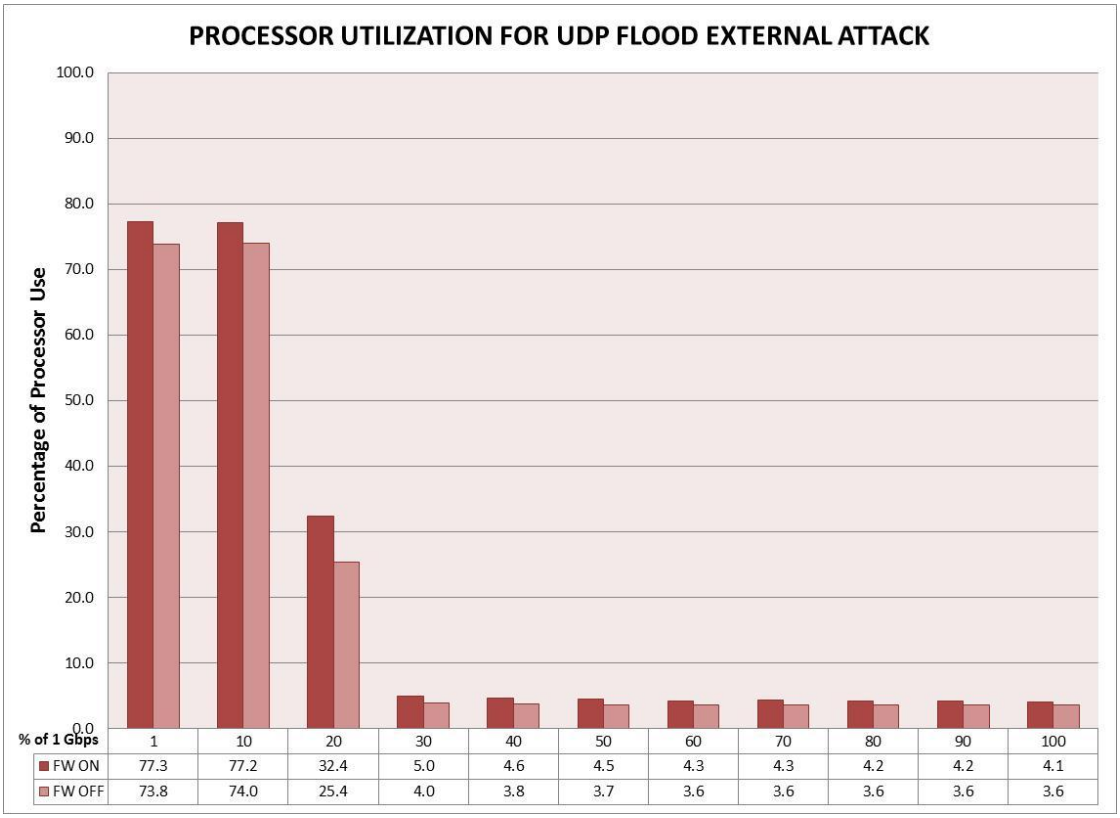


Figure 4.49 - Processor Utilization for UDP Flood External Attack on Red Hat Linux 5 Server

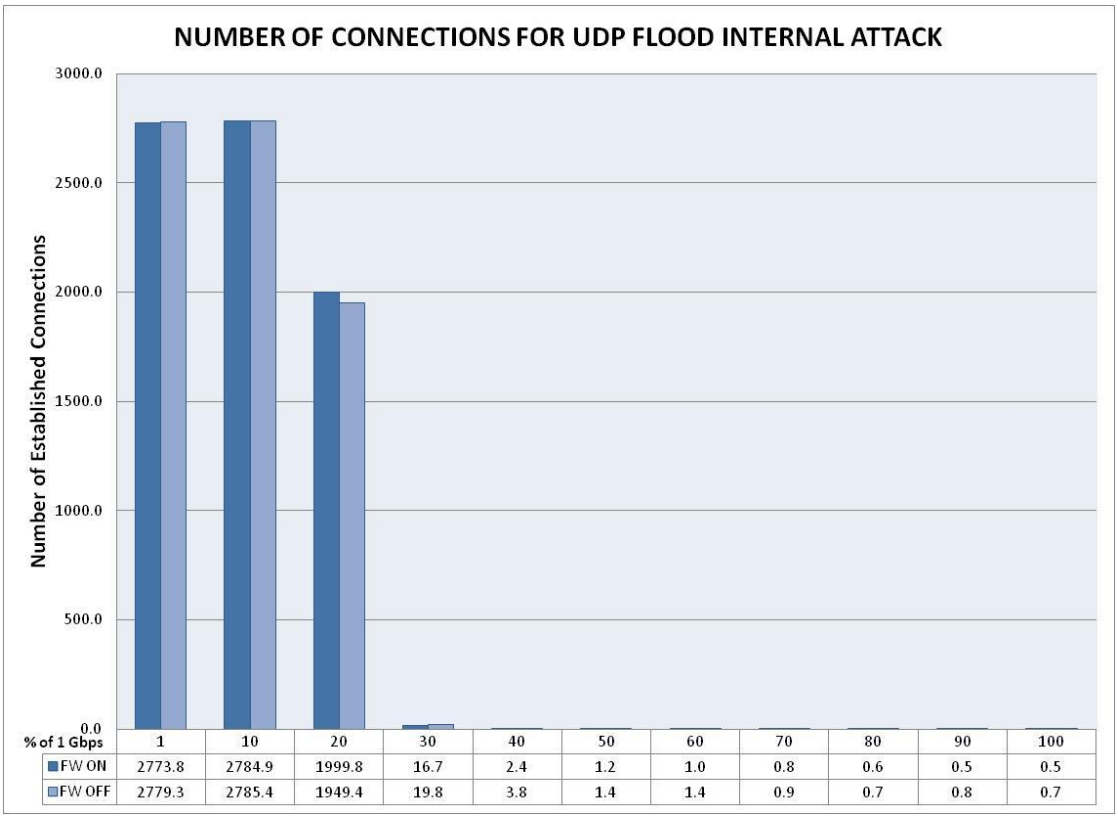
4.3.6.2 UDP Flood Internal Attack. In this section we will analyze the UDP Flood internal attack configuration and its impact on both operating systems.

Windows is not showing a big difference from the external attack configuration, whereas Linux has shown an improved defense against this internal attack for the firewall ON configuration.

Windows is keeping a stable number of connections for attack loads of 10% and lower. It starts losing connections at 20%, and from 30% and higher rates the number of connections is close to zero for both; firewall ON and OFF setups.

Linux has been able to host stable number of connections up to 30% of attack load, when the firewall is activated. At 40% of attack load the connections are decreased to 1454, and for larger attack load the number of connections is close to zero. When the firewall is disabled, Linux is

able to host connections only before 20% of attack load, but for further rates, the denial of service is presented to legitimate users.



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	2773.8	2784.9	1999.8	16.7	2.4	1.2	1.0	0.8	0.6	0.5	0.5
FW OFF	2779.3	2785.4	1949.4	19.8	3.8	1.4	1.4	0.9	0.7	0.8	0.7

Figure 4.50 - Connections for UDP Flood Internal Attack on Microsoft Windows 2008 Server R2

The processor resources analysis for this attack type is shown in figures 4.52 and 4.53. Both Windows and Linux operating systems are proving to use processor resources directly to host http connections.

Windows shows to have a lower performance on handling processor resources compared to Linux since it consumes up to 97% of its resources while it hosts less http connections than Linux. Linux consumes at most 77% of processor resources for the same purpose.

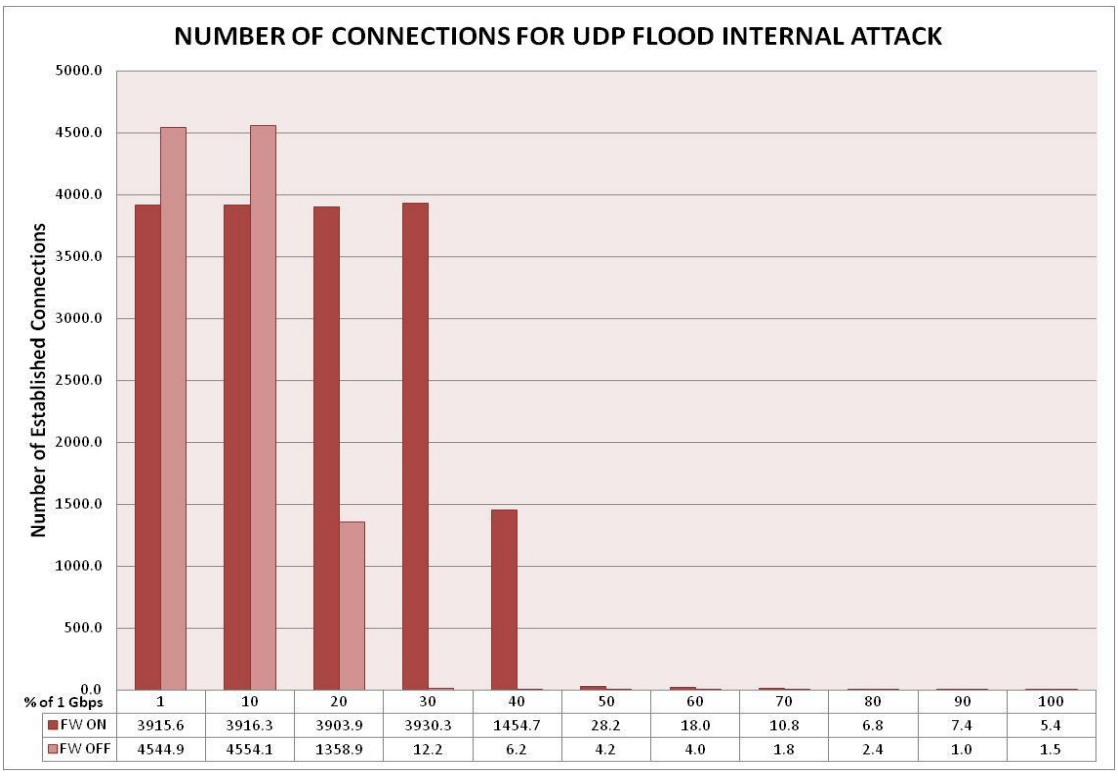


Figure 4.51 - Connections for UDP Flood Internal Attack on Red Hat Linux 5 Server

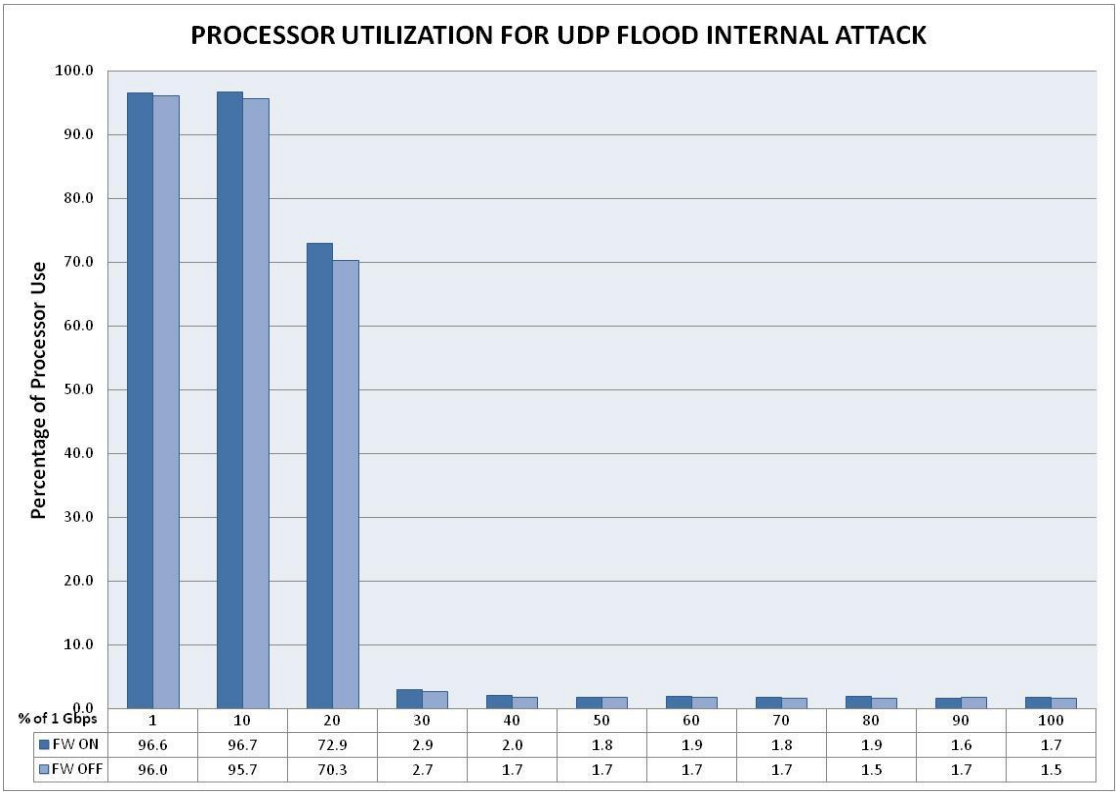
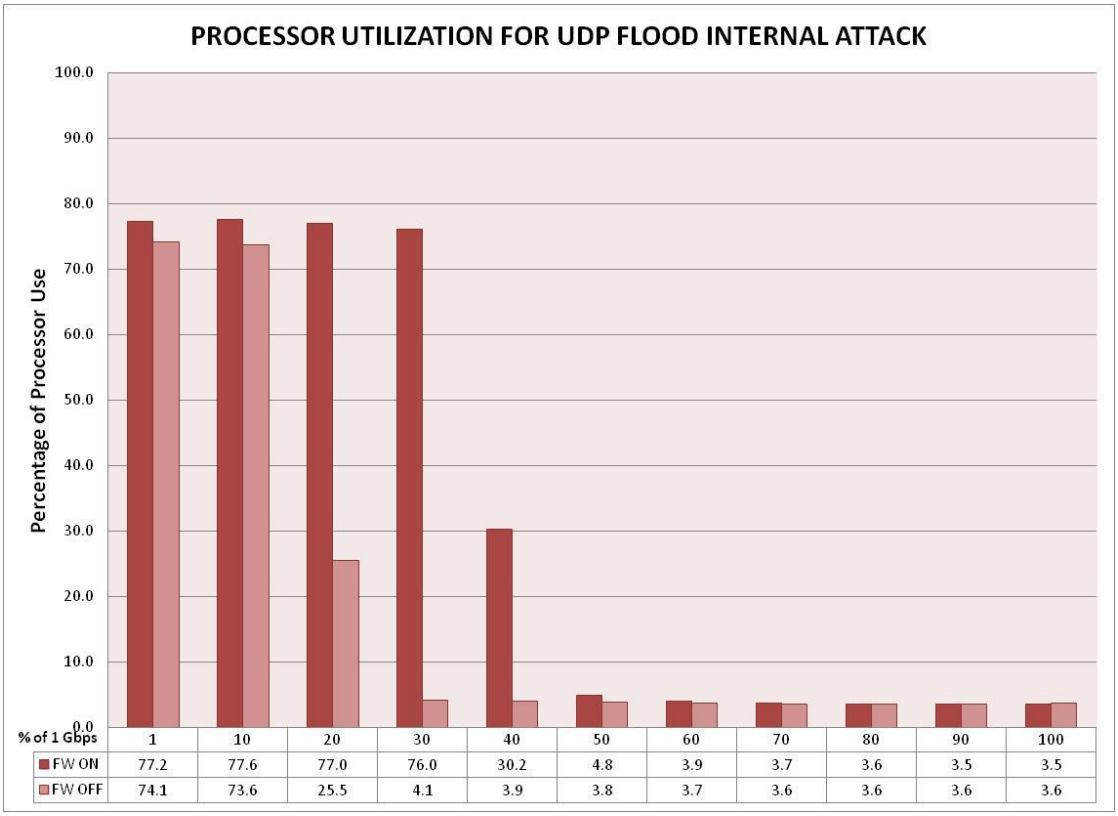


Figure 4.52 - Processor Utilization for UDP Flood Internal Attack on Microsoft Windows 2008 Server R2



% of 1 Gbps	1	10	20	30	40	50	60	70	80	90	100
FW ON	77.2	77.6	77.0	76.0	30.2	4.8	3.9	3.7	3.6	3.5	3.5
FW OFF	74.1	73.6	25.5	4.1	3.9	3.8	3.7	3.6	3.6	3.6	3.6

Figure 4.53 - Processor Utilization for UDP Flood Internal Attack on Red Hat Linux 5 Server

4.4 Chapter Summary

We can conclude this chapter by summarizing the most important findings on the both platforms that were tested. We have performed an exhaustive analysis of different types of denial of service attacks using a hierarchical point of view of the information layers 2, 3 and 4 from the TCP/IP protocol suite. When we tested both server platforms, they showed to have both strengths and weaknesses depending on the scenarios presented and on the kind of attack performed on each operating system.

In order to summarize the test results obtained in pervious chapters for Windows 2008 and Red Hat Linux5 servers under different types of DDoS attacks, we present comparison in a tabular form for two parameters: (i) Number of HTTP Connections supported, (ii) Processor

Utilization and (iii) Performance Comparison of Internal vs. External Attacks. The different combinations of analysis evaluated include variations like Internal and External Attack, and Firewall enabled and disabled for each type of attack we have tested.

(i) Number of HTTP Connections. In the table below, this parameter represents a total number of HTTP connections that were supported by the servers under different attack types. A server was considered a winner if it supported a higher number of HTTP connections for a given attack type. In the table 4.1, the word 'WIN' and 'LOSE' was used to indicate their performance in supporting higher number of connections for a given attack type. 'EVEN' is used in the table 4.1 to indicate that their performance is comparable for a given attack.

(ii) Processor Utilization. This parameter represents which operating system was able to make a better use of the Processor resources while it was under attack and free of attack. These results are shown in table 4.2.

(iii) Performance Comparison of Internal vs. External Attacks. In table 4.3, we present an evaluation of the performance results compared for Internal and External Attacks. We are able to analyze which attack has a greater impact on each operating system. The word "Better" is used to represent a superior performance obtained for that specific attack configuration. If the word "Worst" is used, it means this configuration resulted in causing more damage to the equipment under test.

Table 4.1 shows that the Red Hat Linux 5 operating system is able to host a greater number of connections for 5 out of 6 attack types, viz. ARP attack, Ping attack, Smurf attack, Land attack and UDP attack. However, for the TCP-SYN attack, it was Windows 2008 that performed better

than that of the Red Hat Linux system. We can deduce from table 4.1 that for most of the attack types, Linux is able to host a larger number of connections compared to Windows.

Table 4.1 – Comparison of Number of HTTP Connections

NUMBER OF HTTP CONNECTIONS									
ATTACK TYPE	INTERNAL ATTACK				EXTERNAL ATTACK				CATEGORY WINNER
	FW ON		FW OFF		FW ON		FW OFF		
	WINDOWS	LINUX	WINDOWS	LINUX	WINDOWS	LINUX	WINDOWS	LINUX	
ARP Flood	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
Ping Flood	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
Smurf Attack	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
ICMP Land	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
TCP - SYN	WIN	LOSE	EVEN	EVEN	WIN	LOSE	WIN	LOSE	WINDOWS
UDP FLOOD	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX

The following table 4.2 compares the processor utilization of the two servers under different attack types. The one using least amount of processor resource was considered a winner for the given attack type, indicated by 'WIN' in the table; and the one consuming higher processor resource, is indicated by 'LOSE' in the table 4.2.

Table 4.2 – Summarized Study of Processor Utilization for Web Server

PROCESSOR UTILIZATION									
ATTACK TYPE	INTERNAL ATTACK				EXTERNAL ATTACK				CATEGORY WINNER
	FW ON		FW OFF		FW ON		FW OFF		
	WINDOWS	LINUX	WINDOWS	LINUX	WINDOWS	LINUX	WINDOWS	LINUX	
ARP Flood	LOSE	WIN	EVEN	EVEN	LOSE	WIN	LOSE	WIN	LINUX
Ping Flood	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
Smurf Attack	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
ICMP Land	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
TCP - SYN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX
UDP FLOOD	LOSE	WIN	LOSE	WIN	LOSE	WIN	LOSE	WIN	LINUX

We present in table 4.3 a performance comparison of Internal and External attacks. We noticed that for most configurations and attack types, the impact produced for both Internal and

External attacks was very similar. We noticed that for the ICMP Ping Flood Attack, the External Attack was always worst for all the configurations on both Windows and Linux operating systems. Same behavior was also observed for Linux under TCP-SYN and UDP Flood Attacks when the firewall was disabled. For all other configurations the general performance was very similar on both attack configurations.

Table 4.3 – Performance Comparison of Internal vs. External Attacks for Web Servers

PERFORMANCE COMPARISON OF INTERNAL VS EXTERNAL ATTACKS								
ATTACK TYPE	WINDOWS				LINUX			
	FW ON		FW OFF		FW ON		FW OFF	
	INTERNAL	EXTERNAL	INTERNAL	EXTERNAL	INTERNAL	EXTERNAL	INTERNAL	EXTERNAL
ARP Flood	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME
Ping Flood	BETTER	WORST	BETTER	WORST	BETTER	WORST	BETTER	WORST
Smurf Attack	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME
ICMP Land	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME
TCP - SYN	SAME	SAME	SAME	SAME	SAME	SAME	BETTER	WORST
UDP FLOOD	SAME	SAME	SAME	SAME	SAME	SAME	BETTER	WORST

For the majority of the tests, Linux has been able to withstand the greater number of connections at low intensity rates of incoming attack traffic, besides, it has been able to prove that it can handle the processor resources in a more efficient manner compared to Windows.

In the case of Internal versus External attacks, we have observed that for most cases the difference in the performance impact on both platforms is not significant, but for some specific attacks the external attack has represented a greater threat than internal attack.

It is also important to mention that Linux Operating System by default was incurring into denial of service due to the limited number of maximum connections allowed. This setting had to be modified in order to allow more http connections.

We have considered Windows to be more user-friendly and easier to configure even by users with just an entry level of expertise, whereas Linux required a more advanced knowledge to be configured in order to obtain the best performance when it was subject of a Distributed Denial of Service Attack.

CHAPTER V

EVALUATION OF RENEWABLE DATA COLLECTION SYSTEMS UNDER DDoS ATTACK

Data collection is a very important aspect of every system and subsystem on the Power Generation and Distribution Industry. By the deployment of an organized data collection scheme, Companies may know how much energy and power has been produced and distributed throughout the network, how it was consumed by users and how they can improve the system and make it more efficient in order to get most benefits of the utilities.

With help of past data analysis, the Power Generation Industry can generate forecasts and create expectations on how much demand they can expect for a specific period of time.

This is one of the main reasons why there should always exist good reliability on the information stored in the databases. It should always be available to desired users, so they can access the servers on demand basis to store, modify or pull specific data in order to generate customized reports.

In this chapter, we will review the impact produced on a Data Collection System when the Database Server gathering information is exposed to a Distribute Denial of Service Attack.

5.1 Test Plan and Experimental Setup

Databases are nowadays widely deployed and used throughout the world employing Relational Database Management Systems (RDBMS). One of the most common ways to managing this data is through the use of a special-purpose programming language called Structured Query Language (abbreviated like SQL). Power Generation and Distribution Systems use this language to store data coming from different stages or subsystems.

We have presented a Photovoltaic Data Collection System on chapter III that uses the advantages of MySQL Server to store and access the solar production data gathered at our site. This System is using MySQL Server, the Open Source software platform which helps perform these tasks.

In order to measure the impact on the Photovoltaic Data Collection System we have chosen to test it using an Open Source MySQL Benchmark software called Hammerora [96]. Hammerora uses the TPC-C Standard [97] to perform a MySQL benchmark evaluation of the system. It counts the number of Transactions per Minute and the New Orders per Minute that the System under Test can perform. It is tested on a predefined database architecture that simulates an Online Transaction Processing (OLTP) application. We are interested in testing the OLTP aspect of the database due to its higher requirement for fast processing of transactions, where an Online Analytical Processing (OLAP) database usually is used for long queries which take longer to be executed and does not requires a high transaction throughput.

A more detailed description of how Hammerora performs the Benchmark will be presented in section 5.2 from this chapter.

The platforms used to perform these tests are the same presented on chapter IV of this thesis, but this time, we will be performing the evaluation over the MySQL Database System. The Device under Test description is detailed below:

Dell PowerEdge 1800

- Processor Intel® Xeon ® CPU ES345 @ 2.33 GHz Dual Core
- RAM capacity of 4.00 GB

The two Operating Systems that were used are:

a) Windows Server 2008 R2 Enterprise. Service Pack 1, Build 7601, 64-bit Operating System.
    a. Using MySQL 5.5.25a Community Server.
b) Red Hat Enterprise Linux Server 5.3 (Tikanga). 64-bit Operating System.
    a. Using MySQL 5.5.28 Community Server.

The testing scenario is displayed in Figure 5.1. The legitimate user's traffic is diminished due to the impact on processor and bandwidth consumption coming from the zombie users, which are being controlled by the attacker. As a consequence, attack traffic is generated and directed towards their target, the MySQL Database Server, as shown in this experimental setup.

To ensure consistency in the tests we have performed, all the tests are repeated 3 times and then averaged. In some of the graphs presented in further sections we will see that there is some variation for specific configuration. This variation has been reduced to the minimum by the number of iterations we have evaluated.

It is also important to mention that for each benchmark test, Hammerora is configured to do a 2 minute ramp-up test, prior to the actual benchmark. So it can reach a stable number of transactions throughout the duration of the test which is 5 minutes after the ramp-up.

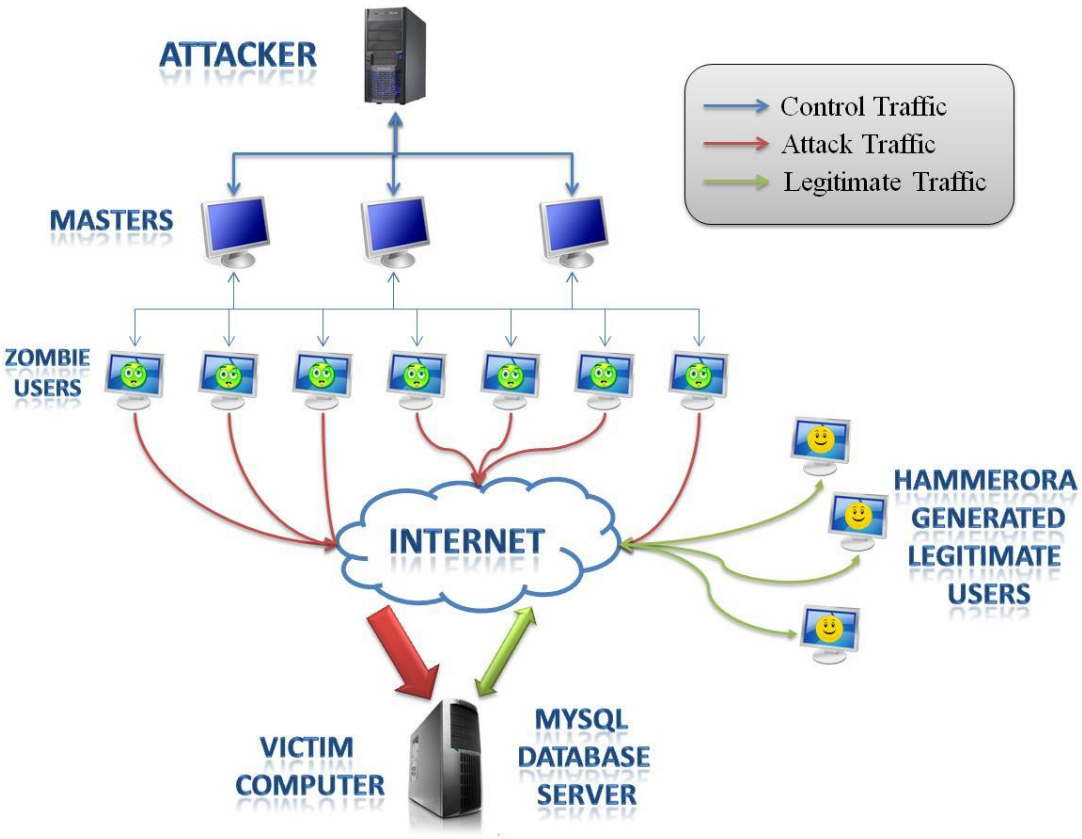


Figure 5.1 - Experimental Setup for MySQL Server attack in the Network Research Laboratory

5.2 Performance Evaluation Methods and Parameters

To get a better understanding of how the evaluation tests were performed a brief description of the TPC-C Standard and the Hammerora software will be provided next.

Hammerora implements a Benchmark test on MySQL Platforms using the TPC-C Standard, which is recognized by major database vendors as the standard for Online Transaction Processing [98]. The TPC-C Standard has been in included into Hammerora, which allows testing for any MySQL environment. Some of the significant advantages of such implementation are that it is reliable, scalable and tested to produce consistent results [99].

TPC-C Standard implements a computer system from a company that sells 100,000 items to different users who complete orders and get fulfilled. The items are kept in warehouses. Each warehouse has a 10 sales districts and each district serves 3000 customers.

The size of the company is not fixed and it can be changed to fit the specifications of the System that we will be testing. The workload of the system will consist of a mix of five transactions as follows:

- New Order: receive a new order from a customer: 45%.
- Payment: update the customers balance to record a payment: 43%.
- Delivery: deliver orders asynchronously: 4%.
- Order-Status: retrieve the status of customer's most recent order: 4%.
- Stock-Level: return the status of the warehouse's inventory: 4%.

The main measurements TPC-C will generate are Transactions per Minute (TPM) and New Orders per Minute (NOPM), which will be the parameters of performance metric used to analyze the DUT behavior when there is some disturbance on the Network due to DDoS Attacks.

For the purpose of testing our System to the Maximum capacity, we have reduced the thinking and keying time parameter to zero, allowing by simulation of a single user to generate thousands of transactions that represent a very large environment. This simulates an environment that hosts a large number of MySQL Transactions.

The first step to measure the severity of the attack is to know how the system performs under normal load conditions and zero disturbance. For such reason, a Baseline Benchmark is created, and illustrates the optimal behavior of the System.

After the Baseline has been created for a different number of users, we introduce attack traffic at different levels going from 10% to 100% of a 1000 Mbps Ethernet Network, increasing

in steps of 100 Mbps. The results presented in this chapter are the average of three runs at each load for every type of attack evaluated.

We have chosen to use two types of Distributed Denial of Service Attacks. TCP-SYN and ICMP Ping Attacks were selected based on observations of technology news because they have been very popular and are selected by many attackers.

The Firewall configuration used for both Microsoft Windows Server 2008 and Red hat Linux 5 Server Operating Systems is enabled. An exception was added to the firewall to allow incoming traffic addressed to port TCP 3307, used specifically for MySQL Server to process transactions into the database and queries from the users.

We assume the attacks to be directed from external networks. To achieve that purpose, random IP addresses are used on the incoming attack traffic and a fixed MAC addresses on such packets, since they must go through the gateway router prior to arrive to the Database Server. This is represented by the configuration displayed in Fig. 5.1.

On the following sections we will present the results obtained from these tests for both Operating Systems chosen. We will be able to compare the performance of MySQL under DDoS Attacks on both platforms.

5.3 Results and Discussions

In this section we present the results obtained from the experimental testing for MySQL Server when it is being subject of a Distributed Denial of Service Attack. This Research has been evaluated on the Network Research Lab from the Engineering Department of the University of Texas-Pan American. We are simulating a mass deployment of data components in smart grid environments.

In Figure 5.2 we can observe the Baseline of MySQL Benchmark Report for Microsoft Windows 2008 Server. The number of Transactions per Minute is presented on the left Y-Axis, and the units are in multiples of 1000, abbreviated as TPM and plotted with a red line. On the Right side of the Y-Axis, we can see the number of New Orders per Minute or NOPM represented by the green line.

The X-Axis represents the number of Virtual Users that are accessing the database simultaneously starting from 1 user to 24 users. The maximum number of TPM and NOPM is reached when the number of users is 2; it gets close to 250,000 TPM and 3,700 NOPM. Then a significant decrease in these number is recorded when the number of users is incremented getting close to 35,000 TPM and 300 NOPM.

The number of TPM and NOPM are started by the Hammerora benchmark software in a ramp up period of two minutes, after that they have reached a stable number and they are tested during 5 minutes. The total number of TPM and NOPM is calculated by Hammerora during the 5 minutes of testing and represents the total number of transactions and new orders succeeded by all users in test per minute.

Figure 5.3 represents the results of the same test run on Red Hat Linux Server 5. It follows the same pattern of fig. 5.2, but the maximum number of TPM is close to 325,000 and NOPM almost 5,000 when the number of virtual users is 2. As the number of virtual users is increased, the Transactions and New Orders per minute are decreased getting close to 80,000 TPM and 1200 NOPM when the number of 24 virtual users is configured.

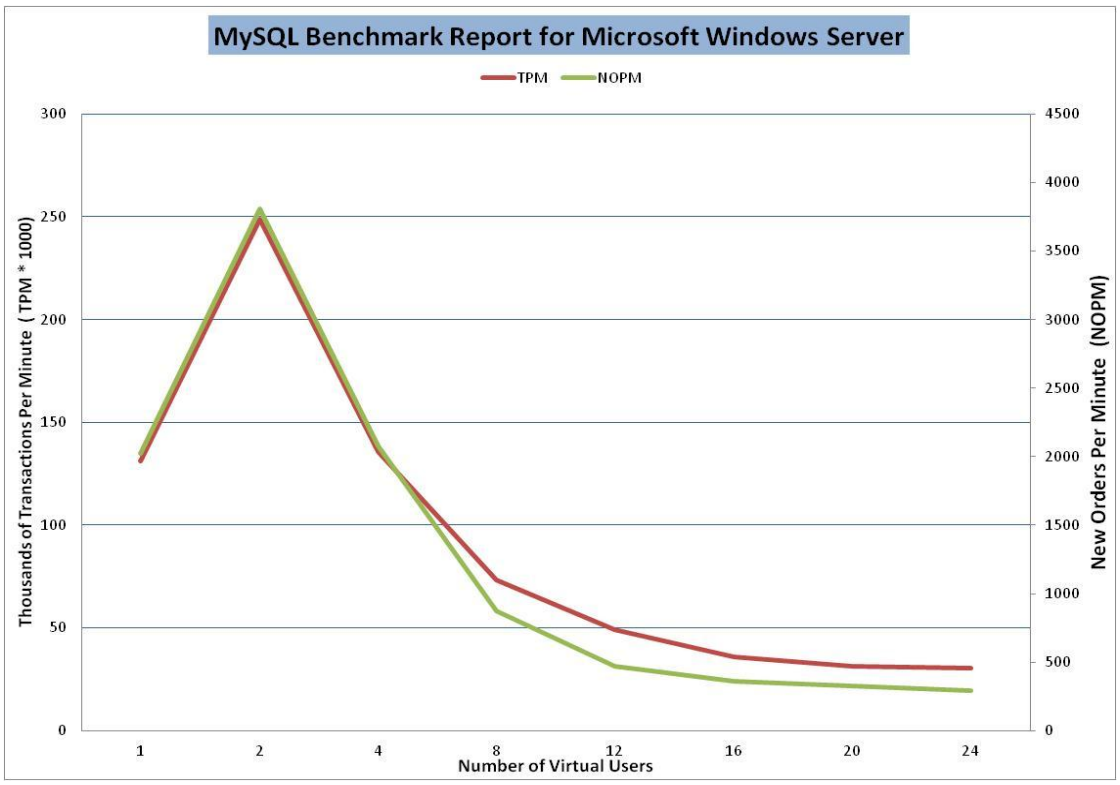


Figure 5.2 - MySQL Benchmark Report for MS Windows Server 2008 R2

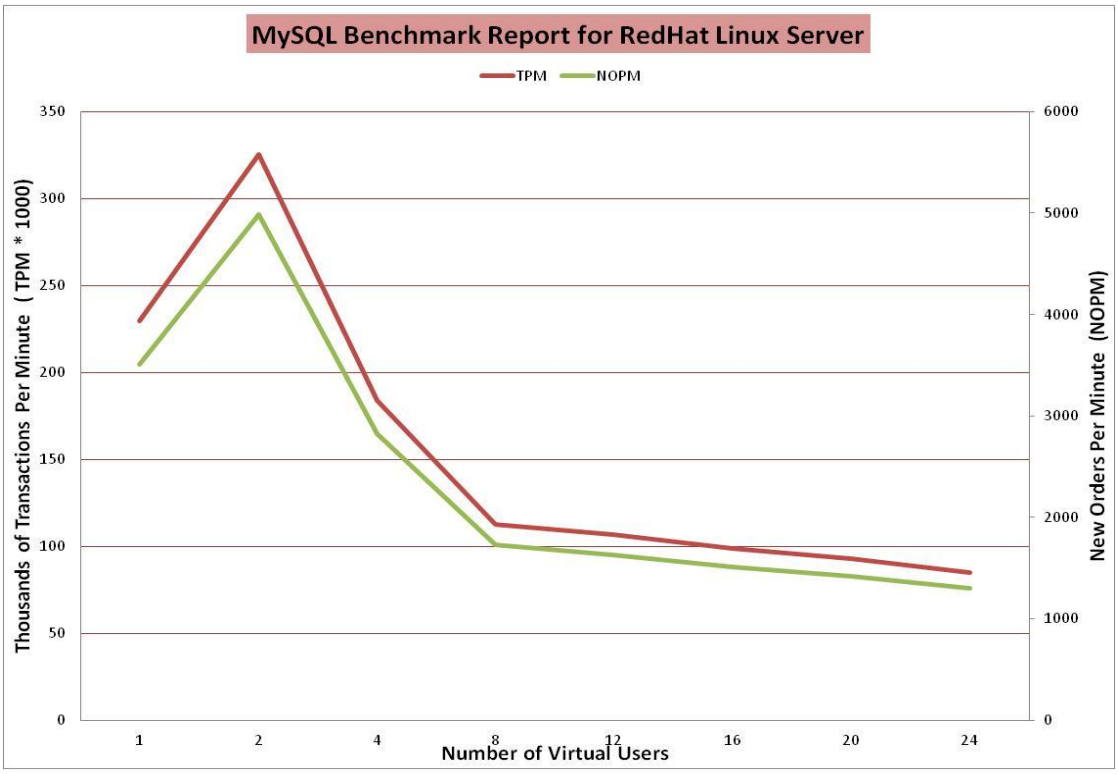


Figure 5.3 – MySQL Benchmark Report for Red Hat Linux Server 5

For the Benchmark option, we can see that Linux has proved to be able to process a greater number of MySQL transactions throughout the duration of the test compared to Windows. We will analyze the behavior of both operating systems performance when they are under disturbance on the following sections of this chapter.

5.3.1 ICMP Ping Flood Attack

In this section we will discuss the results obtained from testing both Operating Systems under ICMP Ping Flood Attack. The performance graphs will show three main results. The green line represents the System Performance when there is No Attack Load present. The orange line represents the performance under a50% Attack Load, and the red line show the 100% Attack Load results. These measurements are evaluated on a 1 Gbps Ethernet Line, thus 50% is the equivalent for 500 Mbps, and 100% is the same as 1 Gbps of Attack Traffic.

TPM and NOPM measurements will be now separated into two different graphs and a third plot will be showing the percentage of Central Processor Unit (CPU) utilization consumed while the test was performed.

Figures 5.4, 5.5 and 5.6 represent the results obtained from Microsoft Windows Server 2008 for TPM, NOPM and CPU utilization respectively.

We can observe that for the Transactions per Minute and New Orders per Minute graphs, the Ping Attack impact when the traffic is 500 Mbps is not noticeable for 1 virtual user. Actually, the number of TPM and NOPM is almost the same at this configuration. However, when the number of virtual users is increased starting from 2 up to 24, there is an impact close to 20% in the number of transactions. When the Attack traffic is increased to 100% a higher impact on the number of transactions is observed for configurations of low number of Virtual Users below 16,

starting at 16 virtual users, the number of TPM and NOPM is very similar for the 50% and 100% attack loads, but slightly different from the no attack configuration.

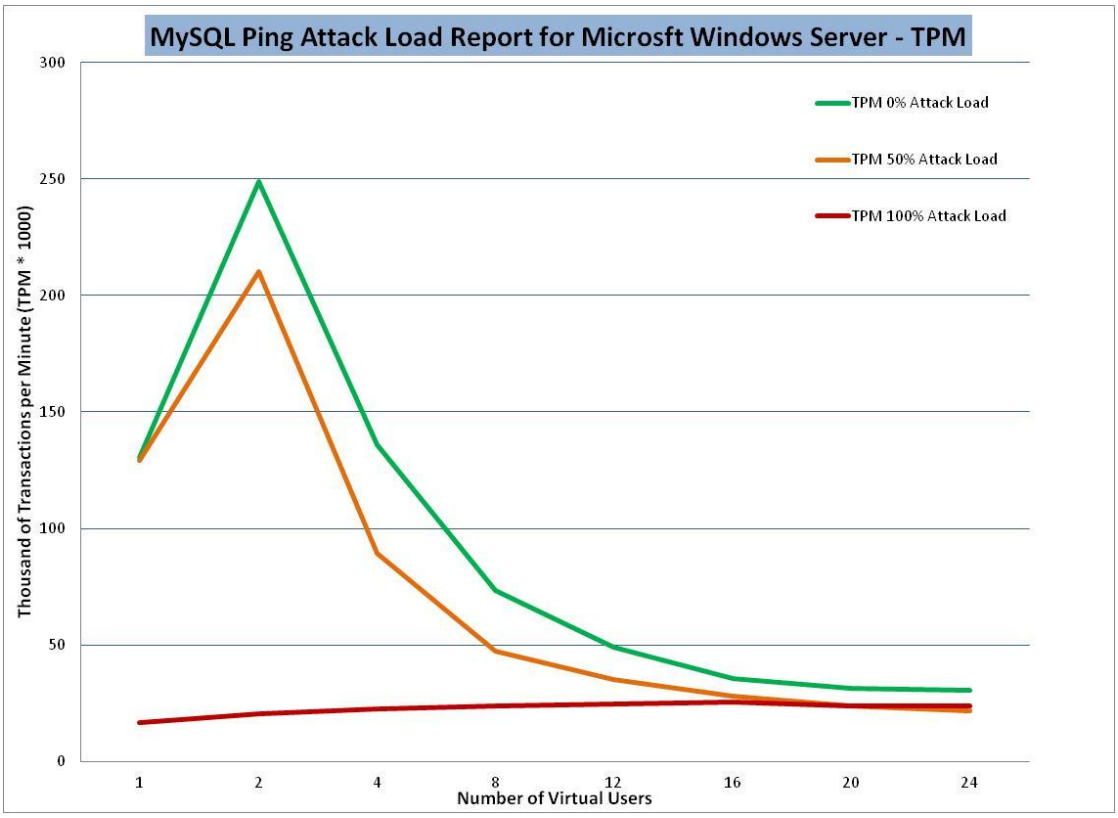


Figure 5.4 – TPM obtained by MySQL under Ping Attack for MS Windows Server 2008 R2

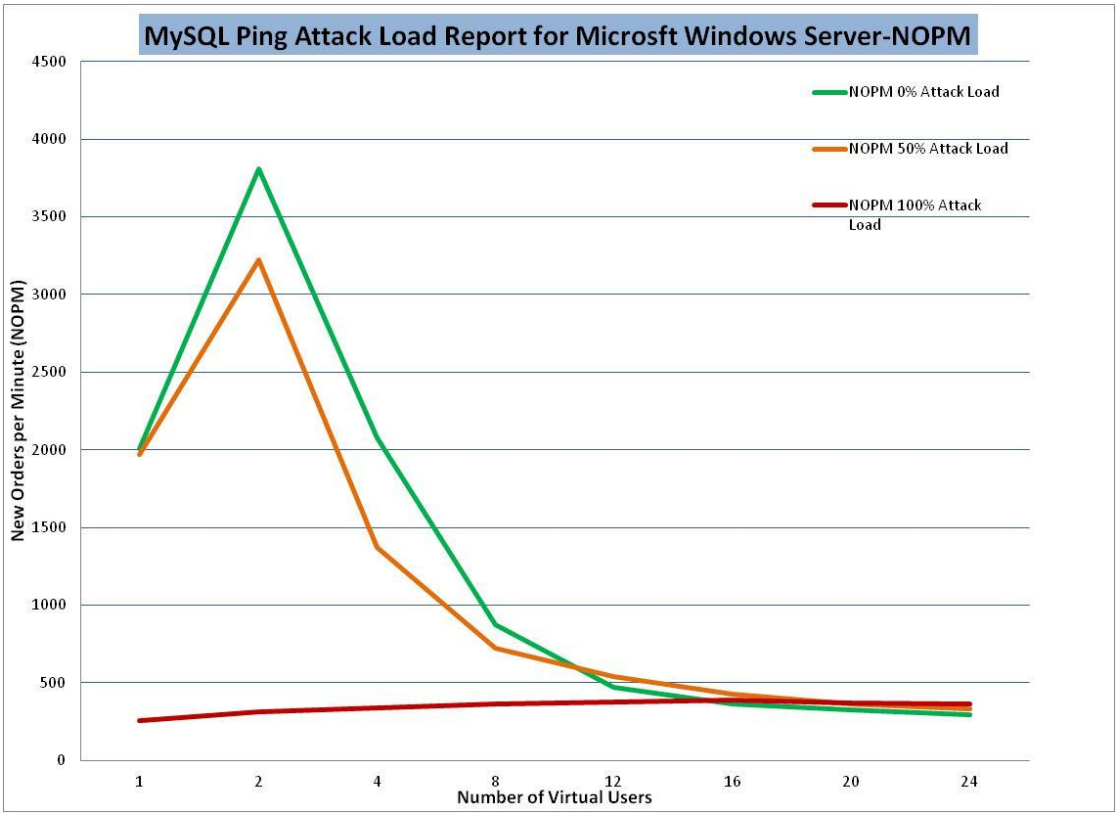


Figure 5.5 – NOPM obtained by MySQL under Ping Attack for MS Windows Server 2008 R2

In Figure 5.6, we can see the Processor utilization graph. The CPU is most consumed when the attack load is not present, and it shows to be proportional to the number of transactions that the MySQL Database Server is dealing with.

As the attack load is increased, the CPU utilization starts being released due to a loss in transactions caused by the network overload created by the Ping Attack. The processor resources are more consumed by the Database Management System than by stopping the attack. The main impacts of the ICMP Ping Attack have been discovered on the higher attack rates like 1 Gbps, and when the number of virtual users that are accessing the Database Server is lower than 16. For configurations of 16 virtual users and higher, the performance under different attack load remains the same as the obtained by the no attack curve.

We can also see that for a high number of virtual users the database server itself performs much lower, even when the attack load is not present in the system being the same as if there was an attack being done, and therefore, limiting the number of transactions that the virtual users are able to process. The behavior MySQL has is similar to a Denial of Service Attack for a high number of virtual users.

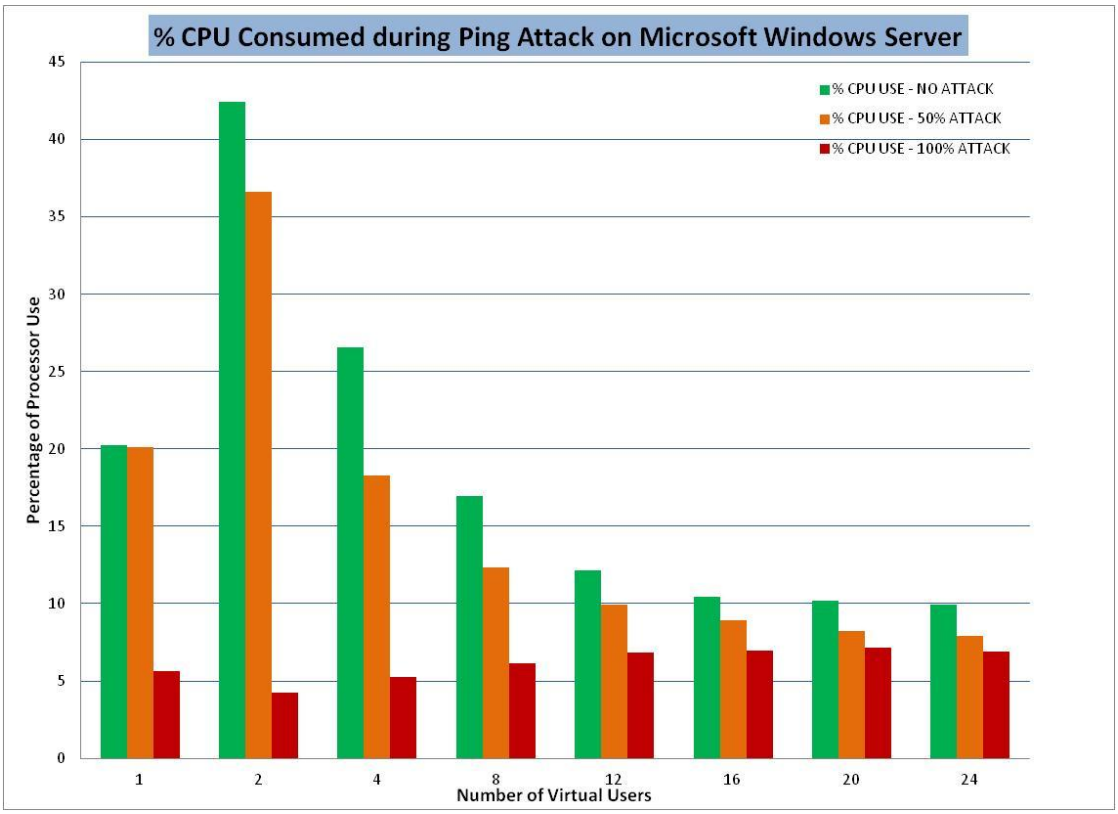


Figure 5.6 – Processor Utilization under Ping Attack for MS Windows Server 2008 R2

In the following three figures 5.7, 5.8 and 5.9, we present the performance of MySQL Server under ICMP Ping Attack evaluated on Red Hat Linux Server Platform.

The number of Transactions per Minute is presented in Fig. 5.7. We can observe a significant increase in the number of transactions compared to Windows Server.

The largest amount of transactions is reached when the number of virtual users is 2 being close to 325,000, and then it decreases as the number of virtual users is increased getting close to

80,000 when the number of virtual users is 24. These results were obtained for the No attack configuration.

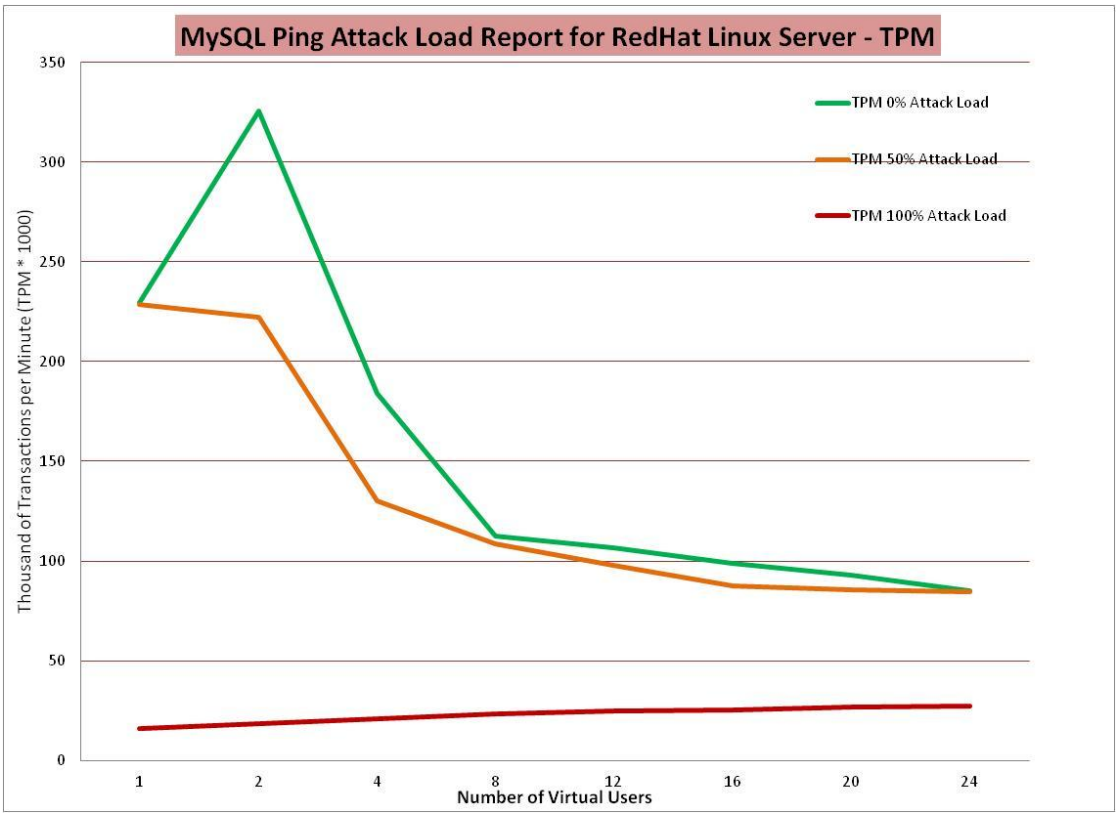


Figure 5.7 - TPM obtained by MySQL under Ping Attack for Red Hat Linux Server 5

When the Attack load is increased to 50%, we can observe a significant impact when the number of users is 2, the TPM are reduced to 220,000. This particular attack load has the most impact on the 2 and 4 virtual user's configurations, for the other number of users, we can observe that the impact on the number of transactions is very small on the 50% attack load when compared to the Baseline, and when there is only 1 virtual user, the performance under this attack load is the same as the one obtained when there is no attack present.

NOPM shows a very similar behavior proportional to the TPM when it is subject to 50% attack load. The major impact has been produced when the number of virtual users was 2 and 4. For the other number of virtual users, the impact was noticed to be much lighter.

When the attack load is increased to 100%, the impact on the number of TPM and NOPM is very noticeable. The attack traffic is driving the Database Server to refuse MYSQL connections and the legitimate traffic is affected in a great way.

The impact is almost the same for all the different number of virtual users. TPM are decreased close to 20,000 and NOPM to around 300. These records prove that the attack load is creating a denial of service to many users and/or transactions.

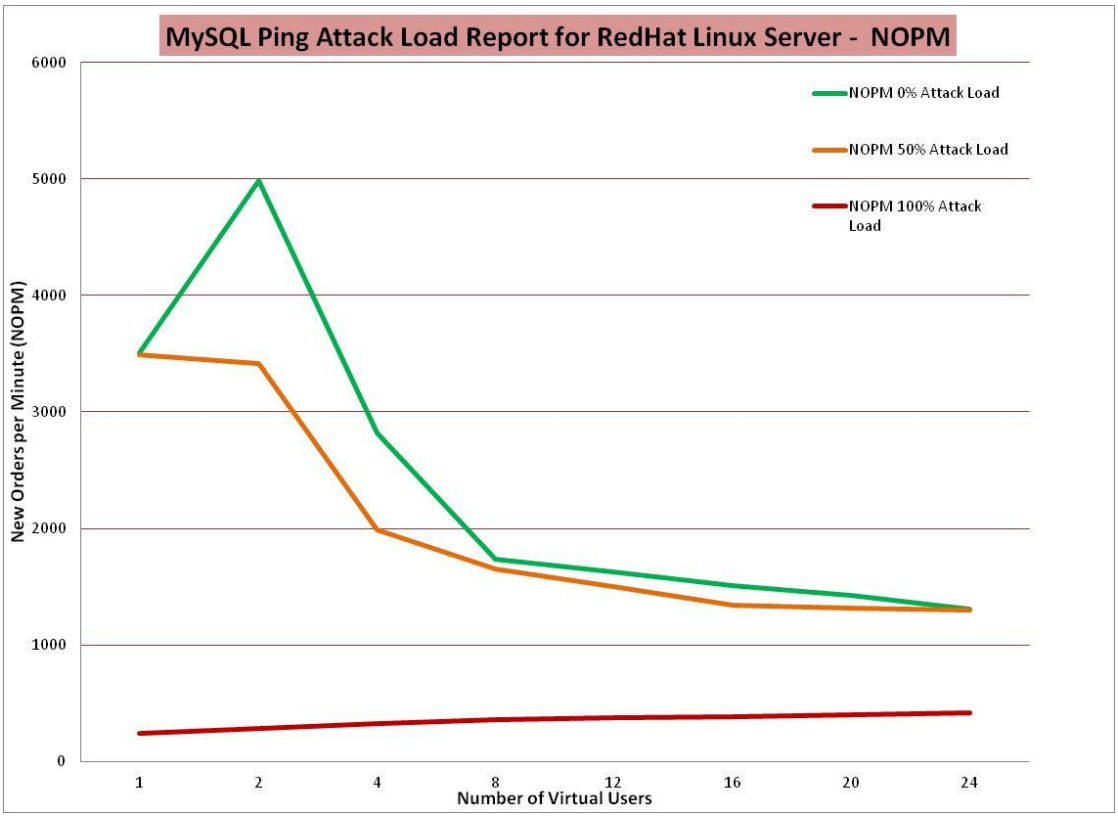


Figure 5.8 - NOPM obtained by MySQL under Ping Attack for Red Hat Linux Server 5

The CPU resources consumed during this test are presented in figure 5.9, and this plot helps us to understand better how the operating system is reacting to the attack. We observe that the processor is most consumed when the attack load is not present on the system. This shows proportionality between the number of transactions and the processor consumption, being this behavior very similar to the one observed in Windows. One of the main differences that we can

notice between Windows and Linux is that windows is handling much better the processor resources for all different number of virtual users, although, the number of transactions was a little smaller. The maximum CPU used by Windows was 42%, logged when the number of virtual users was 2 and the attack load 0%. For all other number of virtual users the processor resources were released and as a consequence, the number of TPM and NOPM were also diminished. In Linux the minimum amount of CPU consumed for the No Attack configuration is 47%, and it goes up to 67% for the majority of the baseline benchmark.

We can deduce that Windows has a better CPU resources management for MySQL, and Linux can host a larger number of Transactions. Linux is consuming most of its CPU resources when the attack load is 0% and 50%, but it is still keeping a high number of transactions and new orders per minute. The CPU consumption continues to be high for most of the user configurations, except for the 100% attack load, where it is showing to release the CPU resources due to a significant decrease in the TPM and NOPM. Whereas Windows has shown to release CPU resources for both 0% and 50% attack configurations at a low number of virtual users, but the number of transaction is less than compared to Linux, and for the 100% attack load, the CPU stays close to 6% for all the configuration of virtual users.

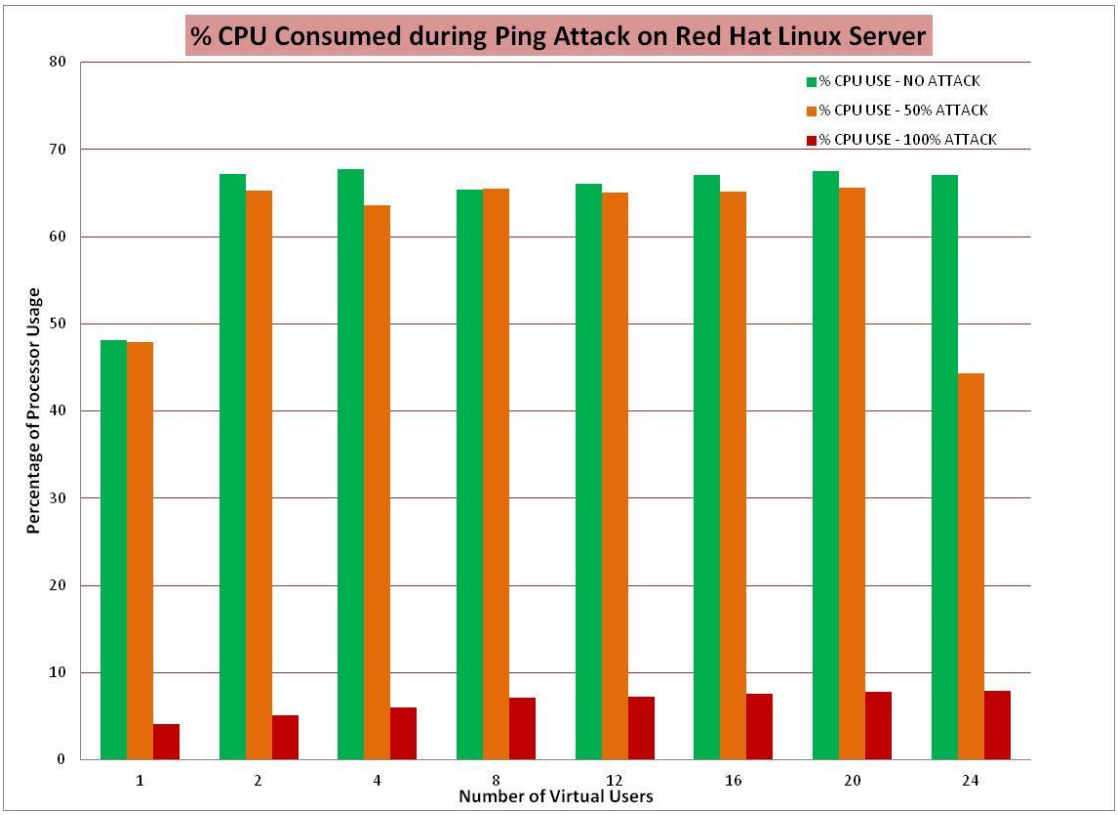


Figure 5.9 – Processor Utilization under Ping Attack for Red Hat Linux Server 5

When Linux is under 100% attack load, the processor resources are released getting close to 8%, but the number of TPM and NOPM are lost and the system is now under Denial of Service to Legitimate Users.

5.3.2 TCP-SYN Flood Attack

In this section, we will review the results obtained from the tests performed on the DUT when it was subject to a TCP-SYN Flood Attack.

In order to keep consistency, the format we use in the graphs presented is the same used in the previous section.

The TPM performance is presented in figure 5.10. We observe how the number of transactions when the attack load is 50% is smaller than the TPM of 0% attack load, but the

difference is 20% or less. Even though the attack is affecting the performance delivered to Legitimate Users, they are able to still establish a connection to the MYSQL Database Server, and execute transactions. The major impact of the attack is noticed when the number of virtual users is between 2 and 16. For 16 virtual users or more, the performance is affected on a less severe way when the attack is present.

When the attack load is increased to 100% the number of TPM recorded is very low compared to the no attack curve when the number of virtual users is 12 or lower. We can see how the attack traffic is creating a denial of service to the legitimate users of the system limiting the number of transactions the users can achieve. We can also observe that for higher number of virtual users the impact of the attack is not as harmful. Actually the number of transactions achieved at these configurations is almost the same to 50% attack load and the difference with the 0% attack load is very small. Microsoft Windows 2008 Server performs almost the same when it is under attack and free of attack for configurations of 16 and more virtual users.

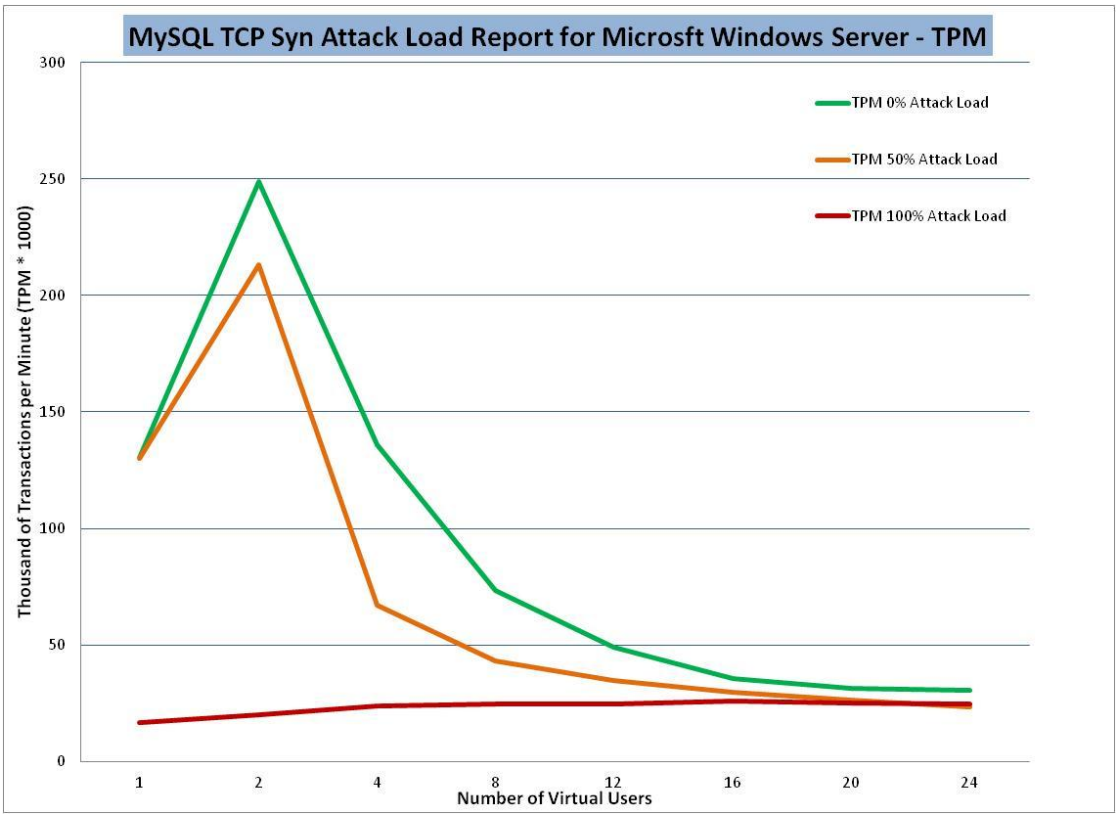


Figure 5.10 - TPM obtained by MySQL under TCP-SYN Attack for MS Windows Server 2008 R2

The behavior for NOPM is displayed in figure 5.11, and it is very similar to the one observed for the TPM. There is a lot of proportionality between these two benchmarks.

We can observe how the major impact of the attack is also noted in when the attack traffic is 100%, and the number of virtual users is less than 16.

The CPU consumption chart is presented in figure 5.12. We can notice how the system resources are consumed proportionally to the number of transactions executed. The greatest CPU consumption is logged when the configuration of virtual users is 2 and the attack load is 0%, reaching at most 43% of processor resources. For all other configurations, the processor resources are almost always 20% consumption or less.

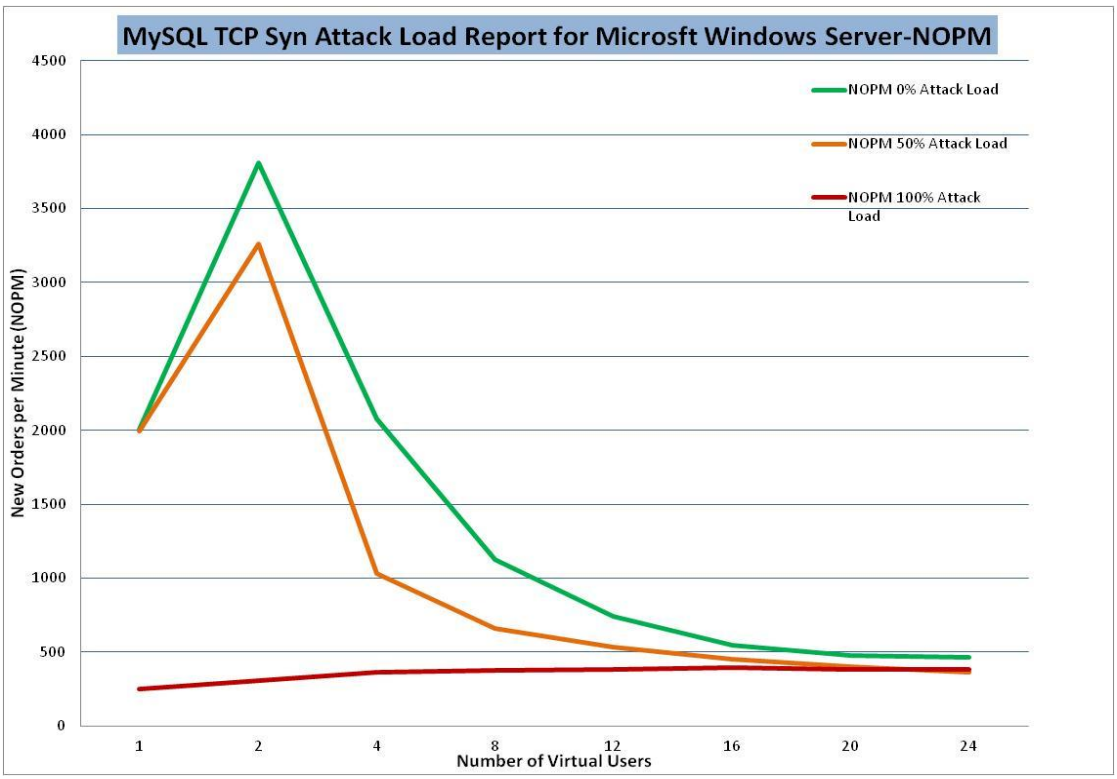


Figure 5.11 - NOPM obtained by MySQL under TCP-Syn Attack for MS Windows Server 2008 R2

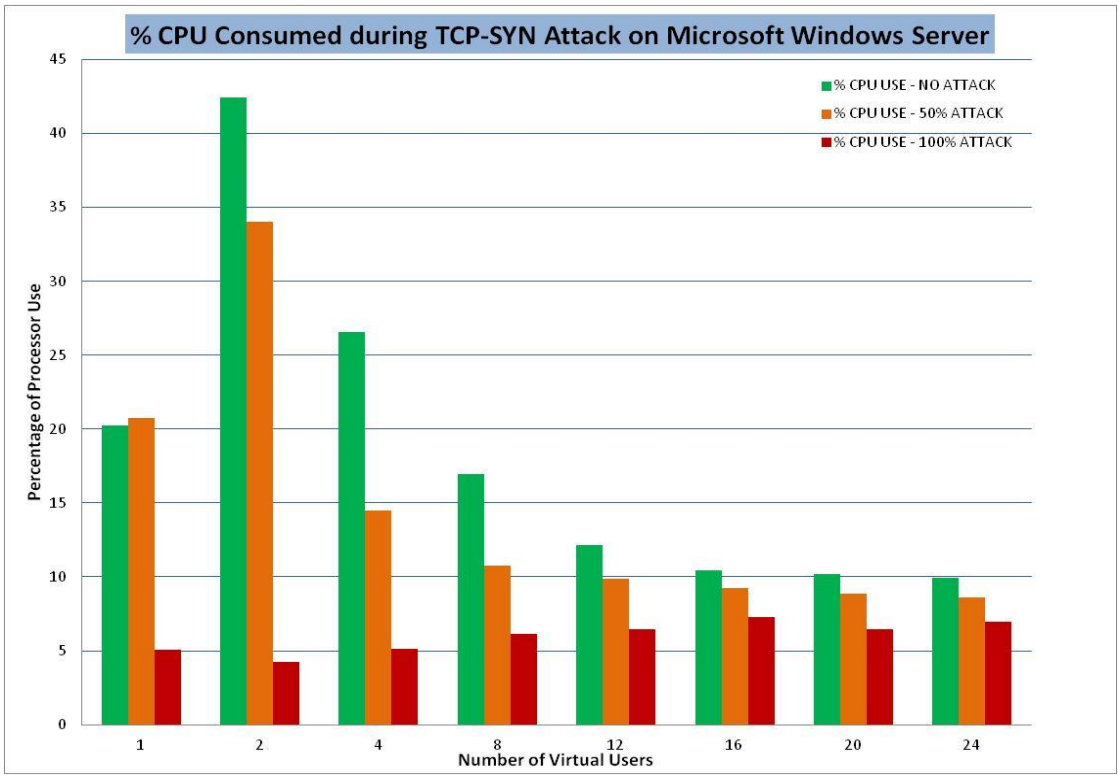


Figure 5.12 – Processor Utilization under TCP-SYN Attack for MS Windows Server 2008 R2

Now we will discuss the results obtained for the Red Hat Linux Server 5 Operating System. We can notice in figures 5.13 and 5.14 that the number of TPM and NOPM is larger than the obtained for the Windows environment. Linux gets around 320,000 TPM when the number of virtual users configured is 2 and the Attack Load is not present. For lager number of virtual users the number of TPM is decreasing getting close to 80,000 at 24 virtual users.

One of the main reasons behind we have observed that the number of TPM and NOPM is always lager for 2 virtual users compared to other configurations, is because the device under test is a Dual-Core Processor architecture, and the load gets distributed evenly for each core. If our processor was Quad-Core, then we would be able to see the most number of TPM and NOPM for 4 virtual users and so on.

When the server is under 50% attack load, the performance is affected mainly for 2 and 4 virtual users. For other configurations the decrease in number of TPM and NOPM is not as severe.

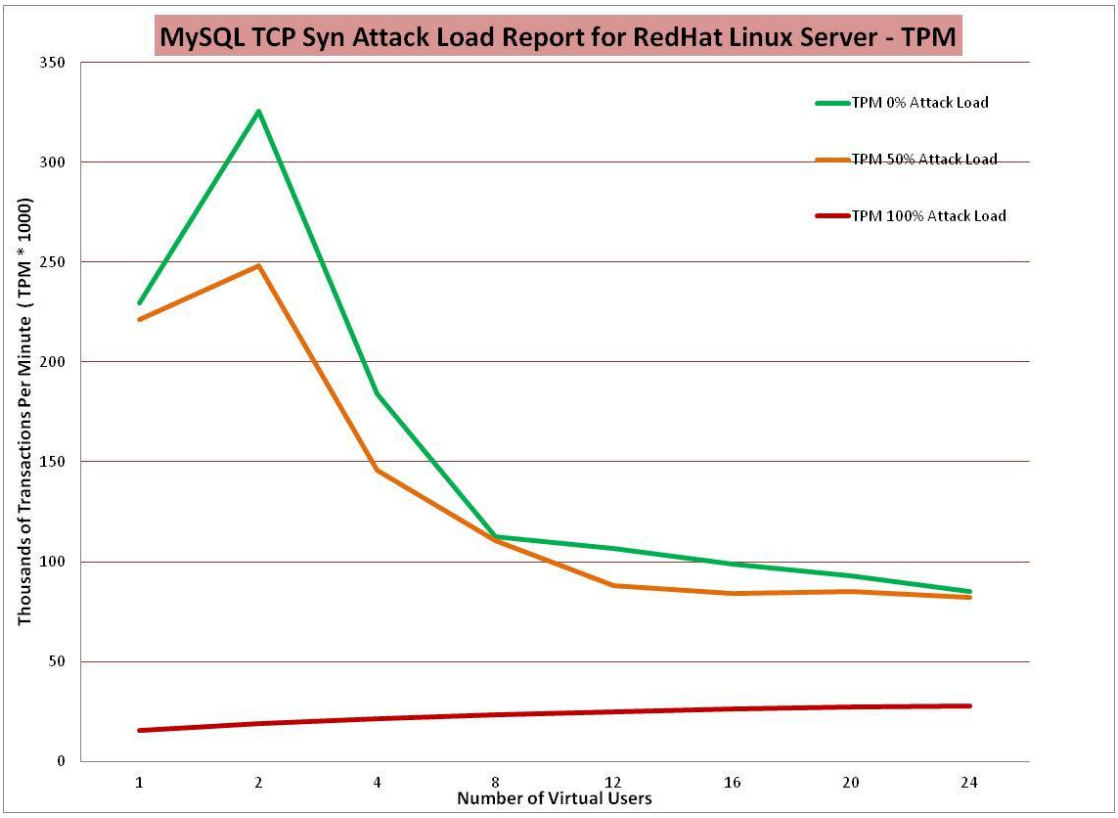


Figure 5.13 - TPM obtained by MySQL under TCP-SYN Attack for Red Hat Linux Server 5

For the 100% attack load we can see that impact on the number of TPM and NOPM is very similar, and that for this attack configuration, the transactions achieved is very small compared to the system under 0% or 50% attack load. We can conclude that the Linux Server is now being subject of a Denial of Service. Legitimate users will experience now difficulties to access the database to save, modify or retrieve information. We can also see that for attack loads different from 100%, Linux OS is being able to host more Transactions and New Orders per Minute compared to Windows Server.

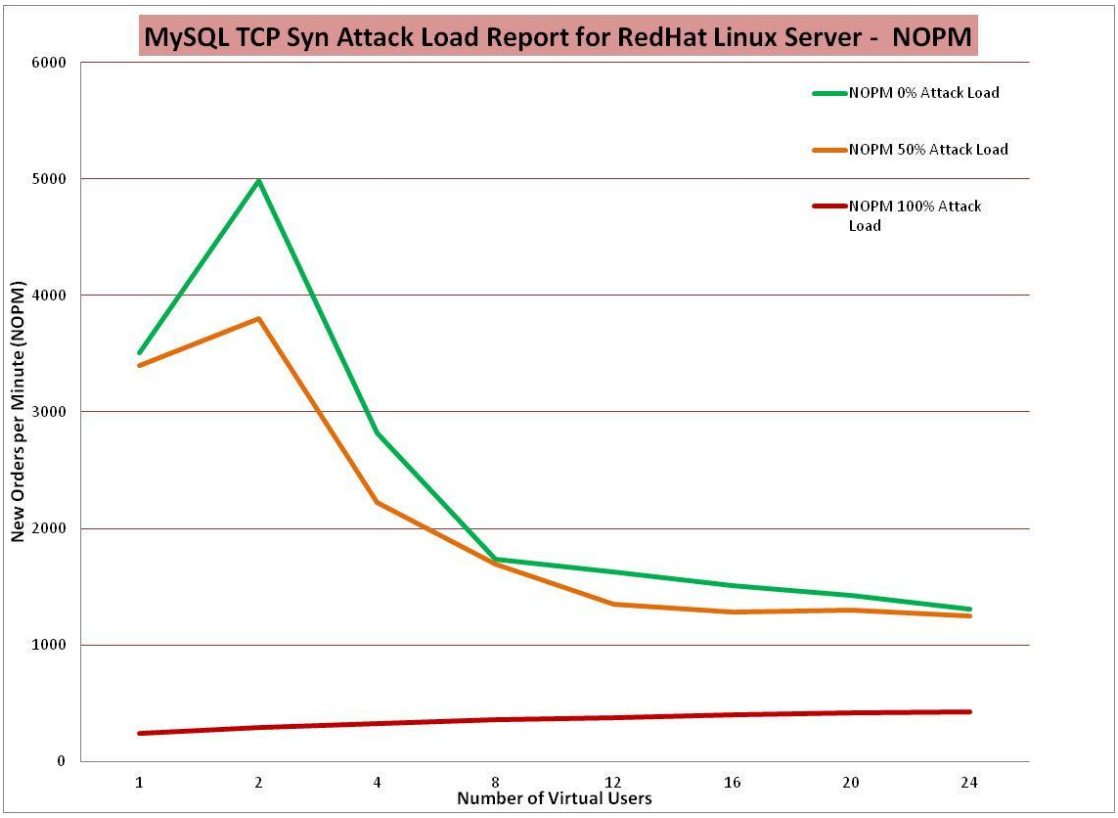


Figure 5.14 - NOPM obtained by MySQL under TCP-SYN Attack for Red Hat Linux Server 5

The processor utilization chart for Linux OS is presented in figure 5.15. The CPU consumption observed for Linux is very different from the one observed in Windows. Linux is showing to exhaust more the processor reaching up to 68% for most virtual user's configurations, where Windows is under 20% for the majority of the test, but the number of TPM and NOPM Windows can serve is also less.

We can also see that the processor is consumed directly proportional to the number of transactions. For 100% attack load, the processor resources are released and get close to 8% due to the decrease noticed on the number of TPM and NOPM.

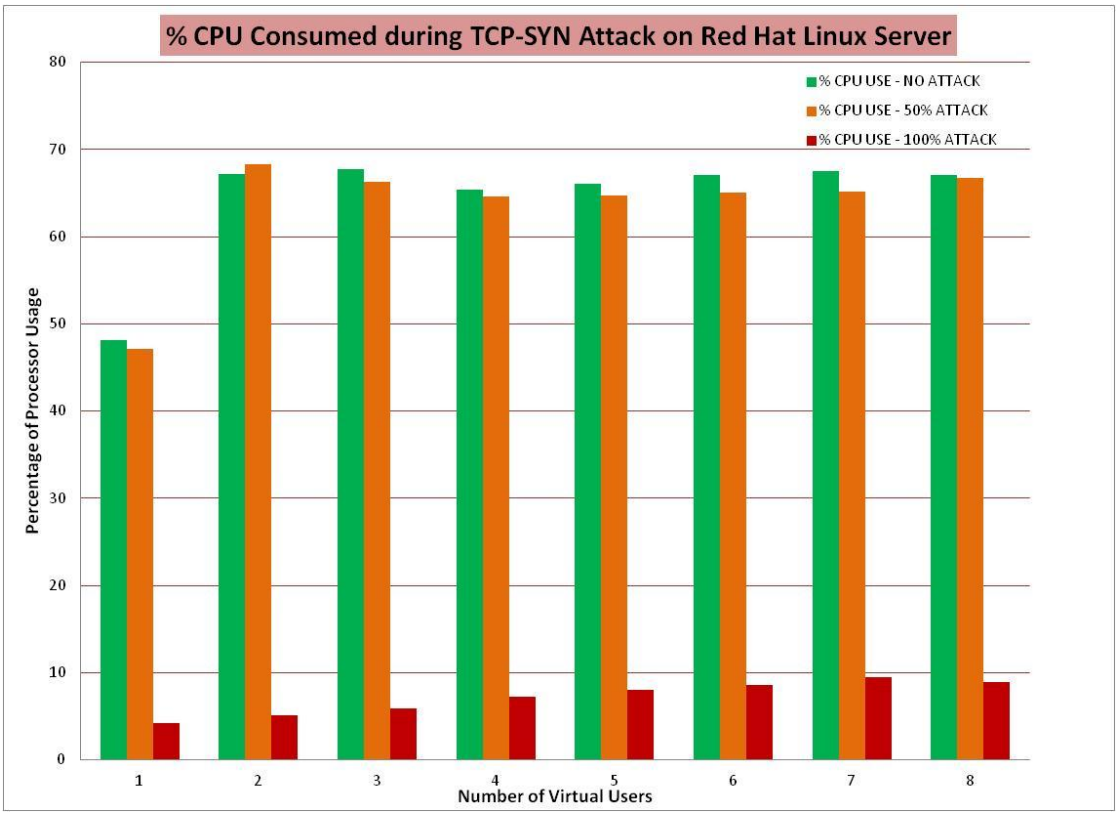


Figure 5.15 – Processor Utilization under TCP-SYN Attack for Red Hat Linux Server 5

5.4 Chapter Summary

In this chapter, we presented the results obtained for the Performance MYSQL tests run in Windows Server 2008 and Red Hat Linux 5 Server under two types of Distributed Denial of Service Attacks; ICMP Ping Flood and TCP-SYN Flood.

The results obtained proved the severity of attack traffic and how it affects the number of MYSQL Transactions a system can perform.

We present summarized results from the tests evaluated in four tables following the same format as Section 4.4 for each of the attacks we have performed on both operating systems (TCP-SYN Attack and ICMP Ping Flood Attack).

We will evaluate the operating systems based on the following parameters:

- Number of TPM and NOPM. We look at the number of transactions and new orders that each system is able to process per minute. The winner of each category will be the operating system which obtains a greater number of TPM and NOPM for all attack rates. The results obtained for this category are presented in table 5.1.
- Processor Utilization. This parameter is describing how well each operating system is utilizing its processor resources. The winner system will be the one who is consuming less processor resources, thus, leaving room for other operations that the server might perform. Table 5.2 will present the results obtained for these measurements evaluation.

In the following table 5.1, we compare the number of transactions and new orders per minute obtained. A Clear winner for this category on both attack type is Linux. This result has been obtained due to the fact that Linux has been able to process greater number of transactions for most attack rates tested.

In table 5.2 we observe the comparison on Processor resources. Windows has proven to manage its processor resources in a smarter way when it is processing MySQL transactions. Linux on the other hand, has behaved poorly consuming a lot of processor resources even when the transactions obtained were decreasing. Windows has been considered the winner for this category.

Table 5.1 – Comparison evaluation of TPM and NOPM performance

NUMBER OF TPM AND NOPM			
ATTACK TYPE	WINDOWS	LINUX	CATEGORY WINNER
Ping Flood	LOSE	WIN	LINUX
TCP - SYN	LOSE	WIN	LINUX

Table 5.2 – Summarized study of Processor Utilization for MySQL performance

PROCESSOR UTILIZATION			
ATTACK TYPE	WINDOWS	LINUX	CATEGORY WINNER
Ping Flood	WIN	LOSE	WINDOWS
TCP - SYN	WIN	LOSE	WINDOWS

We noticed that Windows was able to handle Processor resources better than Linux, but Linux was able to process a greater number of transactions. The processor resources for both Operating Systems have proven to be directly proportional to the number of MYSQL Transactions performed.

These two Operating Systems are able to handle both types of DDoS Attacks for most loads except for 100%, when the attack load has consumed the network resources and by the congestion created, the MySQL Database Servers incur into Denial of Service to Legitimate users.

We also observed that Windows incurs into a behavior similar to Denial of Service for high number of virtual users even when there was no attack present, because it reduced the number of transactions being processed, whereas Linux diminished the number of transactions for higher number of users, it was always greater than Windows when there was no attack load.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

The presence of Denial of Service Attacks is present and is increasing on daily basis. Even though a lot of security measures are implemented to stop this threat, attackers are still able to figure out how to find security breaches in sophisticated secured systems and reach their targets. Attackers are still causing big impacts on normal operation of platforms, creating service and monetary losses, and posing a huge security menace that can have immeasurable consequences.

In chapter III, we have developed a Database System to gather and store Solar Data generated from the Solar Systems. The Database System allows us to generate reports with the gathered Solar Data in an organized and normalized way, so that it can be compared between the two solar systems located in the University of Texas-Pan American.

It was observed in chapter IV the results obtained from the performance tests of two of the main web server systems deployed worldwide, Apache and Internet Information Services used in Red Hat Linux Server 5 and Microsoft Windows Server 2008 R2 respectively, when they are exposed to diverse Distributed Denial of Service Attacks. We tested different configurations for each attack classification.

Our experimental setup allowed testing for internal and external attack configurations. We also tested with Firewall enabled and disabled configurations to analyze the impact difference of the attack due to this integrated protection feature in each Operating System.

There is a noticeable weakness of the tested operating systems to protect themselves from DDoS attacks. The loss of http connections at small attack rates reveals that an attacker can easily bring down a web server and impact on the connections hosted for legitimate users. These results can be considered critical for the proper operation of the Smart Grid, which might produce huge consequences on the communication interfaces for Power Systems.

In chapter V, we studied the results obtained from MySQL Server testing when it was exposed to DDoS attacks. A noticeable impact on the performance of the MySQL service was observed for both Windows Server and Red Hat Sever platforms. With the loss of MySQL transactions, the users are affected and data cannot be stored into the database server.

One of the consequences of being victim of a DDoS attack for a MySQL server can be a noticeable reduction of transactions being processed. This would prevent legitimate users from access and retrieve data required for optimal operation of the grid, thus, causing impact on logging and monitoring systems, or even on the operations of the power resources. These results indicate that a massive blackout may occur due to miscommunication of Power Generation, Transmission and Distribution phases on the power grid, having devastating consequences on the geographical areas affected.

We can deduce that there is a continuous risk on the everyday operations of communications systems of the Smart Grid; the consequences might also stop the legitimate users from normal operations.

As a future work we can suggest the study of the security impact of threats like denial of service attacks on wireless devices, since there is a big usage of this communication scheme planned for the Smart Grid for Home Area Networks, Industrial Area Networks, and other kind of wireless networks. For the work done in this thesis, we have used the micro smart grid system

located at the University of Texas-Pan American. Another future work suggestion is to introduce the usage of Load control and monitoring systems into the communication interfaces to simulate an environment like a real Smart Grid. Then, introduce the networks attacks presented in this thesis and analyze the behavior on the loads (End Users) when the network disturbance is present.

REFERENCES

[1] “Internet Protocol”, RFC 791. http://datatracker.ietf.org/doc/rfc791/ . Last Access: 2/11/2012.

[2] “How was the Justice Department Web site attacked?” http://www.washingtonpost.com/blogs/federal-eye/post/how-was-the-justice-department-web-site-attacked/2012/01/19/gIQA6EGHDQ_blog.html. Last Access: 2/11/2012.

[3] Anonymous Launches "Largest Attack Ever on Government and Music Industry Sites". http://www.securityweek.com/anonymous-launches-largest-attack-ever-government-and-music-industry-sites. Last Access: 2/12/2012.

[4] DoS Attack Cripples Internet Root Servers. http://www.informationweek.com/news/197003903. Last Access: 2/11/2012.

[5] Events of 21-Oct-2002. http://c.root-servers.org/october21.txt Last Access: 2/11/2012.

[6] Twitter Crippled by Denial-of-Service Attack. http://news.cnet.com/8301-13577_3-10304633-36.html. Last Access: 2/11/2012.

[7] Data Finds over 1 m UK Home PCs Belonging to Botnets. http://www.ddosattacks.net/2011/12/11/httpwww-spamfighter-comnews-17155-data-finds-over-1-m-uk-home-pcs-belonging-to-botnets-htm/. Last Access: 2/11/2012.

[8] Major Denial of Service Vulnerability Affects Most Web Servers. http://www.ddosattacks.net/2012/01/04/major-denial-of-service-vulnerability-affects-most-web-servers/. Last Access: 2/11/2012.

[9] Understanding Denial of Service. http://www.informit.com/articles/article.aspx?p=386163&seqNum=5. Last Access: 2/12/2012.

[10] Statistics of DDoS. http://www.shadowserver.org/wiki/pmwiki.php/Stats/DDoSCharts#toc2. Last Access: 2/12/2012.

[11] "The Smart Grid" http://www.smartgrid.gov/the_smart_grid#smart_grid. Last Access: 07/04/2012.

[12] Hank Kenchington, et Al. "Securing Tomorrow's Grid", Public Utilities Fortnightly, July 2011.

[13] "IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads", IEEE Standards Coordinating Committee 21, September 2011.

[14] H. E. Brown, S. Suryanarayanan, "A Survey Seeking a Definition of a Smart Distribution System".

[15] Saifur Rahman, "Smart Grid Expectations what will make it a reality" IEEE power & energy magazine September/October 2009.

[16] M. Hashmi , S. Hänninen, and K. Mäki, "Survey of Smart Grid Concepts, Architectures, and Technological Demonstrations Worldwide", IEEE 2011.

[17] Yi Du, “Summary on Smart Grid and its Automatic Control”, IEEE 2011.

[18] Jeffrey Taft, PhD, Paul De Martini, Leonardo von Prellwitz, “Utility Data Management & Intelligence”, Cisco Systems, Inc. May 2012.

[19] Jianming Qiu et Al, “Use of Real-time/historical database in Smart Grid”, IEEE 2011.

[20] Yong Wang et Al, “Analysis of Smart Grid Security Standards”, IEEE 2011.

[21] F. Boroomand et Al, “Cyber Security for Smart Grid: A Human-Automation Interaction Framework”

[22] NISTIR 7628, “Guidelines for Smart Grid Cyber Security”, NIST, US Department of Commerce, August 2010.

[23] “Cisco Connected Grid Security for Field Area Network”, Cisco Systems, Inc. 2012.

[24] “Why IP Is the Right Foundation for the Smart Grid”, Cisco Systems, Inc. 2010.

[25] Maria B. Line et Al, “Cyber Security Challenges in Smart Grids”

[26] Alexandru Stefanov, Chen-Ching Liu, “Cyber–Power System Security in a Smart Grid Environment”, IEEE 2011.

[27] Y. Yang et Al, “Impact of Cyber-Security Issues on Smart Grid”

[28] Zhuo Lu et Al, “Review and Evaluation of Security Threats on the Communication Networks in the Smart Grid”, IEEE 2010.

[29] Patrick McDaniel, Sean W. Smith, “Security and Privacy Challenges in the Smart Grid”, IEEE Security and Privacy Magazine, May/June 2009.

[30] Alessandro Barenghi, Gerardo Pelosi, “Security and Privacy in Smart Grid infrastructures”, IEEE 2011.

[31] “Security in the smart grid”, ABB white paper 2009.

[32] Emiliano Pallotti, Federica Mangiatordi, “Smart Grid Cyber Security Requirements”, IEEE 2011.

[33] Anthony R. Metke, Randy L. Ekl, “Smart Grid Security Technology”, IEEE 2010.

[34] Ming Li1 and Wei Zhao2 “Detection of Variations of Local Irregularity of Traffic under DDOS Flood Attack”, Hindawi Publishing Corporation, Mathematical Problems in Engineering Volume 2008, Article ID 475878.

[35] Jiang Feng; Yuan Liu; "The Research of DDoS Attack Detecting Algorithm Based on the Feature of the Traffic," Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on, vol., no., pp.1-4, 24-26 Sept. 2009.

[36] Zhang Sheng; Zhang Qifei; Pan Xuezeng; Zhu Xuhui; , "Detection of Low-rate DDoS Attack Based on Self-Similarity," IEEE Education Technology and Computer Science (ETCS), 2010 Second International Workshop on , vol.1, no., pp.333-336, 6-7 March 2010.

[37] Dalia Nashat, Xiaohong Jiang and Susumu Horiguchi “On the Detection of DDoS Attackers for Large-Scale Networks”, Graduate School of Information Sciences, Tohoku University, Sendai, Japan, 980-8579.

[38] Zhongmin Wang; Xinsheng Wang; "DDoS attack detection algorithm based on the correlation of IP address analysis," Electrical and Control Engineering (ICECE), 2011 International Conference on, vol., no., pp.2951-2954, 16-18 Sept. 2011.

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

[40] Terence K.T. Law, John C.S. Lui, David K.Y. Yau "You Can Run, But You Can't Hide: An Effective Statistical Methodology to Trace Back DDoS Attackers", IEEE Transactions on Parallel and Distributed Systems, Vol. 16, No. 10, October 2005.

[41] Cheol-Joo Chae; Seoung-Hyeon Lee; Jae-Seung Lee; Jae-Kwang Lee; , "A Study of Defense DDoS Attacks Using IP Traceback," Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, vol., no., pp.402-408, 11-13 Oct. 2007.

[42] Philip Hunter "Distributed Denial of Service (DDOS) Mitigation Tools".

[43] Cheol-Joo Chae; Seoung-Hyeon Lee; Jae-Seung Lee; Jae-Kwang Lee; , "A Study of Defense DDoS Attacks Using IP Traceback," Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, vol., no., pp.402-408, 11-13 Oct. 2007.

[44] Jelena Mirkovic and Peter Reiher. 2004. A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Comput. Commun. Rev. 34, 2 (April 2004), 39-53.

[45] Liang Hu; Xiaoming Bi; , "Research of DDoS attack mechanism and its defense frame," Computer Research and Development (ICCRD), 2011 3rd International Conference on , vol.4, no., pp.440-442, 11-13 March 2011.

[46] Usage of web servers for websites. http://w3techs.com/technologies/overview/web_server/all. Last Access: 2/12/2012.

[47] David Duke, "What is the Difference between Denial of Service (DoS) and Distributed Denial-of-Service (DDoS)?", Cryptic Software.

[48] "Denial of Service Attacks", http://www.cert.org/tech_tips/denial_of_service.html Last Access: 07/09/2012.

[49] "Securing against Denial of Service Attacks", http://www.w3.org/Security/Faq/wwwsf6.html Last Access: 07/09/2012.

[50] A. Srivastava et Al. "A Recent Survey on DDoS Attacks and Defense Mechanisms", D. Nagamalai, E. Renault, and M. Dhanushkodi (Eds.): PDCTA 2011, CCIS 203, pp. 570–580, 2011. Springer-Verlag Berlin Heidelberg 2011.

[51] Christos Douligeris, Aikaterini Mitrokotsa, "Ddos Attacks And Defense Mechanisms: Classification and State of the Art", Christos Douligeris, Aikaterini Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art, Computer Networks, Volume 44, Issue 5, 5 April 2004.

[52] Dr. Abbass Asosheh, Naghmeh Ramezani, “A Comprehensive Taxonomy of DDoS Attacks and Defense Mechanism Applying in a Smart Classification”, WSEAS Transactions on Computers, v 7, n 4, p 281-290, April 2008.

[53] Mohan, H.S.; Reddy, A.R.; "An Effective Defense against Distributed Denial of Service in Grid," Integrated Intelligent Computing (ICIIC), 2010 First International Conference on, vol., no., pp.84-89, 5-7 Aug. 2010.

[54] Z. Morley Mao et Al. “Analyzing Large DDoS Attacks Using Multiple Data Sources”, SIGCOMM’06 Workshops September 11-15, 2006, Pisa, Italy.

[55] Jirí Schäfer, Michal Drozd, “Detecting Network Attacks Using Behavioral Models”, Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on, vol.2, no., pp.753-758, 15-17 Sept. 201.

[56] Ping Du, Akihiro Nakao, “DDoS Defense Deployment with Network Egress and Ingress Filtering”, Communications (ICC), 2010 IEEE International Conference on , vol., no., pp.1-6, 23-27 May 2010.

[57] Dennis Distler, “Performing Egress Filtering”, SANS Institute, InfoSec Reading Room. August 19, 2008.

[58] P. Ferguson & D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which Employ, IP Source Address Spoofing”, RFC 2827, Cisco Systems, Inc. May 2000.

[59] Xianjun Geng and Andrew B. Whinston, “Defeating distributed denial of service attacks”, IT Pro, August 2000.

[60] R.R. Talpade, G. Kim, and S. Khurana, “NOMAD: Traffic-based Network Monitoring Framework for Anomaly Detection”, IEEE Symposium on Computers and Communications - Proceedings, p 442-451, 1999.

[61] David C. Plummer, “An Ethernet Address Resolution Protocol”, RFC 826, November 1982.

[62] Sanjeev Kumar, “Impact of Distributed Denial of Service (DDoS) Attack Due to ARP Storm”, Lecture Notes in Computer Science, v 3421, n II, p 997-1002, 2005, Networking - ICN 2005.

[63] “ARP Message Format”, http://www.tcpipguide.com/free/t_ARPMessageFormat.htm, Last Access: 07/09/2012.

[64] J. Postel, “Internet Control Message Protocol”, RFC 792, September 1981.

[65] Sanjeev Kumar, “PING Attack - How bad is it?” Elsevier, Computers & Security Journal, Volume 25, Issue 5, July 2006, Pages 332–337.

[66] Jun Xu, Wooyong Lee, “Sustaining Availability of Web Services under Distributed Denial of Service Attacks”, IEEE Transactions on Computers, Vol. 52, No. 2, February 2003.

[67] Rocky K. C. Chang, “Defending against Flooding Based Distributed Denial of Service Attacks: A Tutorial”, Communications Magazine, IEEE, vol.40, no.10, pp. 42- 51, Oct 2002.

[68] Dan Sterne et Al. “Autonomic Response to Distributed Denial of Service Attacks”, W. Lee, L. M´e, and A. Wespi (Eds.): RAID 2001, LNCS 2212, pp. 134–149, 2001. Springer-Verlag Berlin Heidelberg 2001.

[69] Vyas Sekar, “LADS: Large-scale Automated DDoS detection System”, In Proc. of USENIX ATC, pages 171-184, 2006.

[70] Yu Chen, Kai Hwang and Wei-Shinn Ku, “Collaborative Detection of DDoS Attacks over Multiple Network Domains”, IEEE Transactions on Parallel and Distributed Systems, TPDS-0228-0806.

[71] Peter G. Neumann, “Denial of Service Attacks”, Communications of the ACM, April 2000/Vol. 43, No. 4.

[72] Zong-Lin Li Guang-Min Hu Dan Yang, “Global Abnormal Correlation Analysis for DDoS Attack Detection”, Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, vol., no., pp.310-315, 6-9 July 2008.

[73] Sanjeev Kumar, “Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet”, Second International Conference on Internet Monitoring and Protection (ICIMP 2007).

[74] S. Kumar, M. Azad, O. Gomez, R. Valdez,” Can Microsoft’s Service Pack2 (SP2) Security Software Prevent SMURF Attacks?”, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006).

[75] John McCormick, “Possible LAND Attack Vulnerability Affects Windows XP and 2003”, http://www.techrepublic.com/article/possible-land-attack-vulnerability-affects-windows-xp-and-2003/5611467, Last Access: 07/09/2012.

[76] 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, 10-16 Feb. 2010.

[77] Jordan Spencer Cunningham, “US, South Korean Websites under Attack, North Korea Suspected”, http://www.osnews.com/story/21797/US_South_Korean_Websites_Under_Attack_North_Korea_Suspected, Last Access: 07/09/2012.

[78] Robert McMillan & Martyn Williams, “US government sites bombarded by botnet”, http://news.techworld.com/security/118814/us-government-sites-bombarded-by-botnet/, Last Access: 07/09/2012.

[79] Pi-E Liu, Zhong-Hua Sheng, “Defending Against Tcp Syn Flooding with a New Kind of Syn-Agent”, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008.

[80] Vladimir V., Shakhov and Hyunseung Choo, “On Modeling Counteraction against TCP SYN Flooding”, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v 5200 LNCS, p 574-583, 2008.

[81] Wei Chen, Dit-Yan Yeung, “Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing”, Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on, vol., no., pp. 38, 23-29 April 2006.

[82] Sanjeev Kumar and Einar Petana, “Mitigation of TCP-SYN Attacks with Microsoft’s Windows XP Service Pack2 (SP2) Software”, icn, pp.238-242, Seventh International Conference on Networking (icn 2008), 2008.

[83] “TCP Connection Establishment Process: The "Three-Way”, http://www.tcpipguide.com/free/t_TCPConnectionEstablishmentProcessTheThreeWayHandsh-3.htm, Last Access: 07/09/2012.

[84] Felix Lau et Al. “Distributed Denial of Service Attacks”, In IEEE International Conference on Systems, Man, and Cybernetics, pages 2275-2280, 2000.

[85] Jaime Ramos, Leonel Aguilera, Et Al. “Commissioning A 5 kW PV Array for Electrical Engineering University Curriculum”, American Society for Engineering Education, 2011.

[86] Leonel Aguilera, Jaime Ramos & Sanjeev Kumar, “UTPA Solar System Efficiency”, American Society for Engineering Education, 2012.

[87] Leonel Aguilera, “Validation of Power Data provided by Sunny Web Box for Renewable Solar Energy”, UTPA, Award Winning Poster at “HESTEC Science Symposium”, during Hestec 2010.

[88] “TXU Sun Tracking Arrays Website”, http://view2.fatspaniel.net/PV2Web/feed?feed=Dashboard/Commercial/RGDetailedViewFeed&eid=553237&format=table&style=csv, Last Access: 07/10/2012.

[89] “Solar Research Lab (SRL) - UTPA”, http://www.nrel.gov/midc/utpa_srl/, Last Access: 10/29/2012.

[90] Ye Yan, Yi Qian, Hamid Sharif and David Tipper, “A Survey on Cyber Security for Smart Grid, Communications”, Communications Surveys & Tutorials, IEEE , vol.PP, no.99, pp.1-13, 0.

[91] Tarlochan S. Sidhu, and Yujie Yin, “Modelling and Simulation for Performance Evaluation of IEC61850-Based Substation, Communication SystemsSidhu, T.S.; Yujie Yin; , "Modelling and Simulation for Performance Evaluation of IEC61850-Based Substation Communication Systems," Power Delivery, IEEE Transactions on , vol.22, no.3, pp.1482-1489, July 2007.

[92] “Glossary of Solar Radiation Resource Terms”, http://rredc.nrel.gov/solar/glossary/, Last Access: 07/10/2012.

[93] Muhammad Iqbal “An Introduction to Solar Radiation”, Academic Press 1983.

[94] Gilbert M. Masters “Renewable and Efficient Electric Power Systems”, 2004 John Wiley &Sons.

[95] T. Stoffel, et Al, “Concentrating Solar Power. Best Practices Handbook for the Collection & Use of Solar Resource Data”. Technical Report NREL/TP-550-47465, Sept 2010.

[96] “Hammerora", http://hammerora.sourceforge.net/index.html , Last Access: 10/30/2012.

[97] “TPC-C Standard”, http://www.tpc.org/tpcc/detail.asp , Last Access: 10/30/2012.

[98] “Oracle Benchmarks”, http://www.oracle.com/us/solutions/performance-scalability/index.html , Last Access: 10/30/2012.

[99] “Hammerora for Mysql”, http://hammerora.sourceforge.net/hammerora_mysql_oltp_v1.5.pdf, Last Access: 10/30/2012.

APPENDIX A

APPENDIX A

TOOLS EMPLOYED FOR PERFORMANCE MEASUREMENT

The tools employed to measure performance for Microsoft Windows 2008 Server R2 and Red Hat Linux 5 Server, during the tests studied on chapter IV and chapter V will be presented in this section.

In order to measure the performance on Microsoft Windows 2008 Server R2, we used the Performance Log Tools provided by this Operating System. It allows creating a customized text (.csv) report of the desired parameters that we require monitoring during a specific period of time. The parameters used for logging performance during the tests evaluated in this thesis are the Processor and Memory.

We configured a log file containing Processor and Memory utilization, which was recorded with a sample rate of 1second for the duration of the tests.

To measure the performance on Red Hat Linux Server 5, we used the command line interface and executed the SAR (System Activity Reporter) command to create a log file containing Processor and Memory utilization with sampling rate of 1 second.

The command required to log the Processor behavior used was:

```
sar -u 1 3700 >> cpu.txt
```

This command is using the SAR monitoring library included in Linux. The '-u' parameter indicates that the resource to be monitored was the Processor, the '1' parameter is the number of seconds that will be waited before taking an additional measurement, and the '3700' parameter is

the number of times this command will be executed. After that the output of this tool will be appended into a text file named “cpu.txt”.

An example of this parameter output is partially displayed next:

```
Linux 2.6.18-128.el5 (localhost.localdomain)    10/09/2012

09:55:53 AM       CPU     %user     %nice   %system   %iowait    %steal
%idle
09:55:54 AM       all      0.00      0.00      0.00      0.00      0.00
100.00
09:55:55 AM       all      8.04      0.00      1.01      5.03      0.00
85.93
09:55:56 AM       all     26.00      0.00      3.00     17.50      0.00
53.50
09:55:57 AM       all     23.88      0.00      2.99     19.90      0.00
53.23
09:55:58 AM       all     24.50      0.00      2.50     19.50      0.00
53.50
```

To measure the memory resources into a log file, the same monitoring tool was used in the following form:

```
sar -r 1 3700 >> mem.txt
```

The only parameters changed are the ‘-r’, which represents the logging of Memory resources, and the output file name.

These files are later used to examine processor and memory behavior during the attack.

APPENDIX B

APPENDIX B

LOG FILE FROM SELINUX ON RED HAT LINUX

Partial log file generated by SELinux when the number of incoming http connections is greater than the limit set in the Apache Web Server.

```
Jul 16 23:09:40 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:09:52 localhost last message repeated 3 times
Jul 16 23:10:00 localhost kernel: printk: 27 messages suppressed.
Jul 16 23:10:00 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:00 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:09 localhost kernel: printk: 2 messages suppressed.
Jul 16 23:10:09 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:10 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:16 localhost kernel: printk: 3 messages suppressed.
Jul 16 23:10:16 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:21 localhost kernel: printk: 1 messages suppressed.
Jul 16 23:10:21 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:26 localhost kernel: printk: 1 messages suppressed.
Jul 16 23:10:26 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:30 localhost kernel: ip_conntrack: table full, dropping
packet.
Jul 16 23:10:45 localhost kernel: printk: 1 messages suppressed.
Jul 16 23:10:45 localhost kernel: ip_conntrack: table full, dropping
packet.
```

BIOGRAPHICAL SKETCH

Leonel Aguilera Zambrano was born on October 6, 1983. He finished his undergraduate studies in the Instituto Tecnologico de Saltillo on 2005. He obtained the degree of Electronic Engineer. He has two and a half years' experience as an Electrical Designer for the automotive industry. He finished his Masters in Science in Electrical Engineering from The University of Texas-Pan American, Edinburg, Texas, on December, 2012. His current mailing address is,

1711 W. Portales Dr. Apt 1

Edinburg TX. 78541 - 6040.

His Publications and Poster Presentations achieved during his Master Degree are:

- Leonel Aguilera, Jaime Ramos, Sanjeev Kumar, "UTPA Solar System Efficiency", American Society for Engineering Education, 2012.
- Jaime Ramos, Leonel Aguilera, Et Al. "Commissioning A 5 kW PV Array for Electrical Engineering University Curriculum", American Society for Engineering Education, 2011.
- Leonel Aguilera, Sanjeev Kumar, "Web Performance of Windows vs. Linux under DDoS Network Attack", ***Award Winning Poster*** for HESTEC 2011 competition, UTPA.
- Leonel Aguilera, Jaime Ramos, Sanjeev Kumar "UTPA Solar Arrays Efficiencies", Poster for HESTEC 2011 competition, UTPA.
- Leonel Aguilera, Jaime Ramos, Sanjeev Kumar, "Validation of Power Data provided by Sunny WebBox for Renewable Solar Energy", ***Award Winning Poster*** for HESTEC 2010 competition, UTPA.