COMPARISON OF SYSTEM PERFORMANCE DURING DDOS ATTACKS IN MODERN OPERATING SYSTEMS

Bachelor Degree Project in Computer Science
G2E, 22,5 ECTS
Spring Term 2017
2017-06-10

Erik Pettersson (d14eripe@student.his.se)

Supervisor: Joakim Kävrestad
Examiner: Jianguo Ding
Comparison of system performance during DDoS attacks in modern operating systems

Table of Contents

1 Introduction ................................................................................................................................. 1

2 Background .................................................................................................................................... 2
  2.1 What is DoS and DDoS? .............................................................................................................. 2
  2.2 Impact of a DDoS attack ........................................................................................................... 2
  2.3 Defence mechanisms ............................................................................................................... 2
  2.4 Related Work .......................................................................................................................... 3

3 Problem Description .................................................................................................................. 4
  3.1 Research Question .................................................................................................................. 4
  3.2 Motivation .................................................................................................................................. 4
  3.3 Objectives .................................................................................................................................. 4

4 Methodology .................................................................................................................................. 6
  4.1 Scoping ..................................................................................................................................... 6
  4.2 Planning .................................................................................................................................... 7
    4.2.1 Hypothesis Formulation .................................................................................................... 7
    4.2.2 Variable Selection .............................................................................................................. 7
    4.2.3 Experiment Design ........................................................................................................... 8
    4.2.4 Instrumentation ............................................................................................................... 8
    4.2.5 Validity Evaluation ......................................................................................................... 8
  4.3 Operation .................................................................................................................................. 9
  4.4 Analysis & Interpretation ......................................................................................................... 9
  4.5 Presentation & Package ......................................................................................................... 9

5 Designing the Experiment ........................................................................................................ 10
  5.1 Operating systems ................................................................................................................ 10
  5.2 Webserver ............................................................................................................................ 10
  5.3 Defence Methods .................................................................................................................. 11
  5.4 Simulated Attack Method ...................................................................................................... 12
  5.5 Measurement Methods ......................................................................................................... 13
  5.6 System Reliability Factors .................................................................................................. 13

6 Data Gathering .......................................................................................................................... 14
  6.1 Capturing Parameters ............................................................................................................. 14
  6.2 Filtering Data ........................................................................................................................ 14

7 Data Analysis .................................................................................................................................. 16
  7.1 CPU and RAM Usage .......................................................................................................... 16
  7.2 Round Trip Time and Packet Loss ..................................................................................... 16
  7.3 Baseline Analysis ................................................................................................................ 16
  7.4 Load Balancing Analysis .................................................................................................... 18
  7.5 Thresholding Analysis ........................................................................................................ 19
  7.6 Longer Test .......................................................................................................................... 20
8 Conclusions........................................................................................................................................24
  8.1 Conclusions Based on Analysis...................................................................................................24
  8.2 Answers to Research Questions................................................................................................24

9 Discussion.........................................................................................................................................25

10 Future work.....................................................................................................................................25

Appendix A – Scripts

Appendix B – Average CPU/RAM Data Values
Abstract

Distributed Denial of Service attacks are an ever prevalent challenge for system administrators today to overcome. The attack, which is all about restricting legitimate users access to a service, such as a web-page. Can cost companies and governments millions of dollars if not properly managed.

This study aims to explore if there is any difference in performance between some of the most modern iterations of popular server operating systems today. Those server operating systems are: Windows Server 2016, Ubuntu 16 and FreeBSD 11. And submitting them to one of the most popular DDoS attacks at the time of writing, a so called HTTP-Get request. The webservers used are some of the most widely used today, Apache and Microsoft IIS.

Each server will be submitted to attacks, and compared between one another. Different defence methods will also be tested and examined. Tests include shorter tests that is repeated multiple times for data validity, and one longer test for every condition in order to control if the results are similar. During these tests, the operating systems will measure CPU/RAM utilization, and a control computer will measure Round Trip Time.

Windows Server 2016 using IIS and FreeBSD 11 perform similarly resource wise, but Windows Server 2016 with IIS had a better Round Trip Time performance. Windows Server 2016 with Apache performs worst in all measurements, while Ubuntu 16 performs in the middle, but has the most stable performance.

**Keywords:** DDoS, DDoS Prevention, Windows Server 2016, Ubuntu 16, FreeBSD 11, HTTP-Get, IP-Thresholding, Load-Balancing.
1 Introduction

DDoS is an ever prevalent challenge for system administrators today, this study aims to contribute to the field of network security science against DDoS attacks by performing a comparison of system performance during such attacks on at the time of writing, the most modern iterations of three specific operating systems.

The report is structured in such a way that chapter 2 will give some general background information on DDoS that is necessary in order to understand the subject, further, chapter 3 will cover the aim of the study in greater detail, as well as what objectives must be achieved to produce a result. Chapter 4 will cover the methodology used to conduct this study. The design of the experiment and what tools are used to gather data is discussed in chapter 5. Chapter 6 will discuss how data is gathered and how it should be processed to be of use for interpretation. The data will then be analysed in chapter 7, using the data collected and presenting them in visual diagrams. Chapter 8 will present the conclusions that can be drawn from the data presented by this study. Chapter 9 and 10 will contain a discussion of the report and discuss possible future work in this field.
2 Background

This section will contain an overview of scientific work that has been made in the area related to this project, and will define the concepts that are necessary in order to understand the problem.

2.1 What is DoS and DDoS?

As defined by (Silberschatz, Galvin, & Gagne, 2010, p. 638), a "Denial of Service" (DoS) attack has the goal of disrupting legitimate access to a source, and is divided in two categories. The first category involves getting a device to use so much resources that no useful work can be done, an example of this would be if a user would download some sort of application from the internet that when run, uses all available processing power and available memory by opening a never-ending stream of pop-up windows on the computer, this would render the computer unable to perform any other work. The second category is where an attacker disrupts the network infrastructure of the target, which can be done in a multitude of different ways. Commonly known as “Flooding Attacks”, these attacks have in common that they send huge amounts legitimate traffic to a network device such as a router or a server in order to exhaust network/system resources or bandwidth to prevent actual legitimate users from accessing the service.

Building on from this, “Distributed Denial of Service” (DDoS) use the same mechanisms as DoS attacks, but are launched from multiple devices at once towards the same target, being remotely controlled by an attacker from another point. These multiple devices are most commonly part of a so called “Botnet”, which can consist of many million computer devices from all over the world. A computer can unwittingly become part of a botnet by being the victim of a worm, trojan horse or backdoor. This also makes it harder to recognize where the attack originate from because of the spoofed IP addresses of the bots in the botnet.

Most DoS attacks that occur nowadays are of the distributed variety, since the sheer computational power of modern company networks are often times to powerful to bring down with so few resources at the attackers disposal.

2.2 Impact of a DDoS attack

Attacks on businesses will always have a monetary cost, a report from Arbor Networks (Whalen, 2017) shows that out of the companies surveyed, 25% saw the cost of major DDoS attacks rise above $100,000 and 5% over $1,000,000.

Aside from a company suffering from monetary losses, attacks can also cause great complications for normal people. It is not unknown that online services of banks have been the targets of DDoS attacks, leaving customers unable to access the banks resources, and thus unable to access their money. Other examples than banks could be important social services that are necessary for modern society as a whole to function.

2.3 Defence mechanisms

Defence strategies against DDoS attacks comes in a number of different ways, (Zargar, Joshi, & Tipper, 2013) classifies and examines defence mechanisms according to two criteria, the
first criteria is the deployment location of the defence. Against network/transport level attacks these locations are divided in four categories: source-based, destination-based, network-based and hybrid (a.k.a. distributed). It also classifies application based attacks in two categories, these categories are destination-based and hybrid (a.k.a. distributed), application based attacks have no network level defence since that traffic is not accessible by layer 2 switches and layer 3 routers.

The second criteria is at what point in time the defence mechanism acts, these are the same for both network/transport level and application level attacks. These are divided in three categories. These categories are: "before attack", "during attack" and "after attack", dealing with attack prevention, detection and source identification/response respectively.

2.4 Related Work

On a number of occasions, research has been done in comparing the effectiveness of different defence mechanisms for DDoS attacks. Usually, this is done when testing a new model method.

This said, not much research has been done about comparing performance on different server operating systems. (Treseangrat, Kolahi, & Sarrafpour, 2015) made a similar study where Windows Server 2012 and Ubuntu 13 system performance were compared during UDP flood attacks and came to the conclusion that Ubuntu 13 outperformed Windows Server 2012 in all regards, that article inspired this study to evaluate how a BSD system performs in comparison.
3 Problem Description

This section details the problem the project focus on in more detail, as well as explaining why this research is important, and what it can add to the scientific community.

3.1 Research Question

The research question that this experiment study is looking to answer is:

“How does web-server performance differ during DDoS attacks in Windows Server 2016, Ubuntu 16 and FreeBSD server operating systems?”

The aim of this experiment is to set up an environment in which this question can be answered. In order to answer this question, and to provide some validity as to how accurate the answer this study produces for the research question, as according to (Berndtsson, Hansson, Olsson, & Lundell, 2008), an underlying questions will also have to be answered. The first of these questions are:

“What effects do different defence mechanisms have on the operating systems?”

This question could tell us whether or not the defence methods discussed in chapter 5 that are tested in this study has any varying impact on the different operating systems that will be examined in this experiment.

To answer these questions, the environment is to be set up utilizing the resources in the Network and System Administration (NSA) lab at the University of Skövde.

3.2 Motivation

With DDoS being such a prevalent threat facing system administrators, research is constantly being done on the matter, and as mentioned in chapter 2.4, very little research has been done on the subject of comparing performance on different operating systems. And none focusing on the most modern operating systems like the ones focused on in this study. This and the need to provide tools to developers of solutions against DDoS attacks for these newer operating systems is what motivates this study.

3.3 Objectives

In order to perform this study and answer the research question, four objectives has to be fulfilled to produce a result. The experiment process is generally being followed as per (Wohlin et al., 2012, pp. 73–174), and will be discussed more in chapter 4. The steps are:

Gathering of DDoS data – The first step involves gathering data regarding DDoS attacks. This would include what different manners of attacks there are and what defensive options are available. Sources for this would include other scientific reports as well as statistical data about occurrence and seriousness of DDoS attacks. Chapter 2 covers this part

Designing the experiment – The second step is to, with the data collected from the first step, design the experiment environment and parameters. This would include choosing what attack method to use as well as how to implement it. In addition, defence methods to test has to be chosen according to what is most relevant, but also most feasible to set up in
a limited lab-environment. Further this steps would include deciding just what to measure for performance and how the measuring should happen. Lastly it would include testing of the environment, in order to minimize unexpected issues that might occur during the actual performance part of the experiment.

**Performing Experiment** – In the third step, the experiment will be performed and data collected to answer the research question as according to what was determined in the second step.

**Analysis of Data** – In this step, the data collected from the third step will be analysed according to applicable statistical methods. From these results a number of hypotheses will be constructed that can provide an answer to the research question, and might also uncover findings that might be of interest for future or similar works on the subject.
4 Methodology

This chapter will further explain the details of the objectives first stated in chapter 3, and will discuss the methodology used in this study.

The method that will be used in this study is an experiment. Using an experiment was chosen because it is the most fitting method of answering the research question. Where other methods such as interviews with experts in each operating systems usage would yield skewed results. As has been presented by (Wohlin et al., 2012, pp. 16–18), performing an experiment comes in five process steps, which is not necessarily created in a waterfall order. Those steps are:

1. Scoping
2. Planning
3. Operation
4. Analysis & Interpretation
5. Presentation & Package

The study aims to follow this framework, and this chapter will give an overview of the steps and explain which objectives, as was stated in chapter 3, are fulfilled in each step. Some steps have been omitted due to this study not performing studies on people, which would have involved more complex considerations.

4.1 Scoping

The scoping process is where the groundwork for the experiment is laid, if not done correctly, rework on the experiment may have to be done or the experiment might not be able to be used to study the subject that was intended.

This process overlaps with the objective “Gathering of DDoS data”, in the way that the objective is used to do research on the subject and thus creating the scope.

The method details a template that should be used in order to create the scope, the template is:

\[
\text{Analyse} \quad \text{<Object(s) of study>}
\text{for the purpose of} \quad \text{<Purpose>}
\text{with respect to their} \quad \text{<Quality Focus>}
\text{from the point of view of the} \quad \text{<Perspective>}
\text{in the context of} \quad \text{<Context>}
\]

Where the object of study is what is being studied in the experiment, and can be products, processes, models etc. Purpose defines the intention of the experiment. Quality Focus is the primary effect that is being studied. Perspective tells the viewpoint from where the results of the experiment are interpreted. Context outlines who and what is involved in the experiment.

Filling out the template with consideration for this study would yield the following result:
Analyse **Selected operating systems** for the purpose of **Comparing system performance** with respect to their **Endurance against DDoS attacks** from the point of view of the **Researcher** in the context of **The NSA lab environment**

### 4.2 Planning

The planning process overlaps partly with the “Gathering of DDoS Data” objective and greatly overlaps with the “Designing the experiment” objective. The Planning process is continually iterated until the complete experiment design is ready.

The process can be divided into seven steps, these steps are: context selection, hypothesis formulation, variable selection, selection of subjects, experiment design, instrumentation and validity evaluation. These following sections will explain the steps and how they apply to the current study, minus the omitted ones. The actual design will be discussed in more detail in chapter 6.

#### 4.2.1 Hypothesis Formulation

In this step, the hypothesis of the study has to be formally stated. The data collected during the experiment will be used in order to, hopefully, reject the null-hypothesis so that conclusions can be drawn. The null hypothesis and the alternate hypothesis for this study is:

\[ H_0: \text{There is no difference in system performance during DDoS attacks in the specified operating systems.} \]

\[ H_1: \text{There is a difference in system performance during DDoS attacks in the specified operating systems.} \]

#### 4.2.2 Variable Selection

In this step, the dependent and independent variables has to be chosen. Independent variables are variables that can be controlled and changed in the experiment. A dependent variable is what we can measure the effect of changes in the independent variables from.

In this study, there are two independent variables. These are: Defence Method 1 and Defence Method 2. These two defence methods are explained in detail in chapter 5.3. There are three dependant variables, namely the three operating systems. These Are: Windows Server 2016, Ubuntu 16, FreeBSD 11. This means that the two independent variables will be applied to the three dependant variables.

Part of selecting independent variables is also to select measurement scales. In this study, what will be measured is largely inspired from (Treseangrat et al., 2015), and will thus measure CPU-Utilization, memory utilization, Round-Trip-Time, and packet loss. This will be discussed further in chapter 5.
4.2.3 Experiment Design

This step deals with choosing the correct design to be able to statistically analyse the results correctly. There are several design types that can be used depending on factors and treatments that are to be studied in an experiment.

This study will specify the operating systems of choice as factors, and the treatments as being the defence methods that are going to be tested are discussed in chapter 5. In addition, a baseline will be created with no defences to see how big of a difference the defences can make.

4.2.4 Instrumentation

In this step, instrumentation means three types, the objects, guidelines and measurement tools that are used for the experiment.

For this experiment, the measurement tools that will be used are the servers themselves, which will be measuring CPU-Utilization during the simulated attacks, and an external client PC, which will measure Round Trip Time (RTT). The details of how this is set up can be read in chapter five.

Guidelines for the experiment is that every attack will run for one minute, repeating at least 20 times, this data collected is then what will be used in the analysis.

4.2.5 Validity Evaluation

There are four validity threat categories that need to be considered, these are: conclusion, internal, construct and external validity. This part lists the validity threats applicable to this study, and what has been done to negate them.

These are the applicable threats to conclusion validity in this study: Low statistical power, which deals with the ability to reveal a pattern through the data, in essence, how it can be seen if there is any actual difference between the operating systems. This risk reduction is attempted by using different defence methods and repeat every test twenty times. Reliability of measures deals with how similar the results will be after every test. To reduce this risk, test sessions that are grossly out of the norm for the majority of tests, will be discarded, making sure that at least twenty successful tests have been done. The reliability of treatment implementation, deals with that the tests have to be under the same conditions every time, this concern is dealt with by making sure that all parameters are under the same conditions every time the test is run.

The applicable threats to internal validity are non-existent. Since these threats are only applicable in situations where people would be part of the experiment.

The applicable threats to construct validity are: Inadequate preoperational explication of constructs. This threat simply deals with constructs not being sufficiently defined. This is dealt with by consulting the author's supervisor between steps, to make sure that the experiment is on the right path. Mono-method bias deals with a bias being created if only one measurement is taken. This is why the experiment measures multiple performance values.
Applicable to external validity is: Interaction of setting and treatment, this threat deals with the experiment not being representative of the norm, for example an industrial practice. While many systems at the time of this reports writing might not run the latest operating systems that this study examines, since there are many older, stable, systems available, this study aims to focus on the latest ones, and using as modern tools as is available.

4.3 Operation

The operation step overlaps with the “Performing experiment” objective, in this phase the treatments are applied and data is collected.

It is important to in this step make sure everything is prepared so the operation can proceed smoothly. And that the operation follows the plan that was outlined in the planning step.

4.4 Analysis & Interpretation

This step overlaps with the “Analysing data” objective. Here, the data collected is to be analysed using appropriate statistics methods. First descriptive statistics to visually present the data, and performing hypothesis testing.

4.5 Presentation & Package

This step involves presenting the data found during the experiment in some form. In this case, this report. And has been adjusted during the process.
Designing the Experiment

This chapter will detail how the experiment environment is set up. The network topology of the experiment can be seen in Figure 5-1. It consists of two subnets, one subnet contains the attacking machines, and a deployment server that also supplies DHCP to attackers, see chapter 5.4. It also contains the computer which will act as the data gathering monitor, this is also the host used to remotely start the testing process. The second subnet contains the server to be attacked.

Both subnets have their own Cisco 2960 switch that all hosts in their respective subnet is connected to. The subnets are separated by a Cisco 2800 router running with IOS version 12.4. As these models were available for use in the NSA lab environment, they were chosen for this experiment.

Figure 5-1: Network Topology (Authors Own)

5.1 Operating systems

The server operating systems examined in this study are Microsoft Windows Server 2016, Ubuntu 16, and FreeBSD 11. These choices were made because of the choices in a similar study (Treseangrat et al., 2015) to use Microsoft Windows Server 2012 and Ubuntu 13. To use newer versions of these operating systems was seen as interesting to see if the results would be different. FreeBSD was then added, since sources as for example ('Most popular BSD distributions', 2014) claim that it is the most widely used BSD system at the time.

5.2 Webserver

According to ('February 2017 Web Server Survey', 2017), Apache is the webserver software which is historically most widely used on active sites on the internet, and has been for many years, with Nginx and Microsoft IIS (Internet Information Services) coming in at second and third place. Although Apache and Nginx are most commonly used on UNIX based systems such as Linux and BSD, it also has a version for Windows. Thus the choice to use Apache2 as
the webserver software would be the most prudent. On Windows Server 2016 there are several compiled versions of Apache2. This study will use the “ApacheHaus” compiled version (‘Welcome to the Apache Haus’, n.d.). However, certain sources (UpGuard, 2014) mention that Apache2 may not perform optimally on Windows systems, thus, the Windows Server 2016 operating system will also be tested with Microsoft’s own IIS 10.0 webserver software, which is more closely tied to.

All servers used in this experiment are created as virtual machines hosted on a VMWare vSphere 6.0 server (‘Server Virtualization with VMware vSphere’, n.d.). All machines use the same underlying hardware. Which is an Intel Xeon W3550 3.07 GHz processor and 8GB of DDR3 RAM.

### 5.3 Defence Methods

The defence methods that will be used in this experiment are the following two: IP Thresholding (Subrahami, 2011, p. 20) and Load Balancing. (‘Network Load Balancing Technical Overview’, 2000). These defence methods were chosen out of the defence methods used in (Treseangrat et al., 2015) because of their simplicity to apply.

IP Thresholding is the practice of limiting the amount of traffic that is allowed to pass on an interface. In this experiment, Thresholding is implemented on the Cisco 2800 router, limiting the amount of http traffic allowed into the interface the attacker subnet is connected to 80 000 bits per second.

Load Balancing entails using a "Proxy & Load Balancing Server" to be the machine that services requests. In this case http requests. The server then proceeds to forward requests to a number of defined backend webservers. To do this, a clone of each webserver was created and deployed on a secondary vSphere server. As the actual load balancer to forward requests, “Pound” (Pound, n.d.) is used. Pound is a Load Balancer application that can be run on most Linux systems. In this case it was run on an Ubuntu 16 server. It was chosen because of its ease of configuration and functions it could perform which were necessary for this experiment. It was configured to split the requests between the servers 50/50. This creates a simple change in the network topology which can be seen in Figure 5-2.
5.4 Simulated Attack Method

The simulated attack on the webserver will be performed by using two computers in the NSA-lab environment.

A pre-configured distribution of the Ubuntu operating system will be deployed to the computers through a network PXE boot. This PXE boot was created in order to be able to easily control the number of attack computers, should it be needed. This is also done so as to minimize hard-disk drives needed, and thus the setup time required to perform the experiment. The PXE boot will also put the operating systems on the hosts RAM. This is so that they will not need network connection to the deployment server since that could have an unwanted impact on test results.

This pre-configured operating system comes installed with the “Siege” tool (Fulmer, n.d.). Which will be used to create TCP traffic to simulate a “HTTP GET” Flood attack. All the hosts will run and stop at the same time by being remotely controlled by a remote workstation.

The simulated attack is started by the monitoring host in the 10.0.1.0 subnet using a number of scripts that can be examined in appendix A. Ubuntu and Siege was chosen for the attacking hosts due to the ability to run Siege remotely through the command line, which is what the script does.

During a DDoS attack, the goal is to flood the system with so much traffic that the system will be overwhelmed. However, since the goal of this study is to examine how well the systems perform during strong loads. Too much attack traffic can’t be used as that would put all server systems at 100% load, which is a result where no difference between the systems can be seen. Thus, this study uses a lighter traffic load so as not to completely overwhelm the servers.
5.5 Measurement Methods

A number of tools will be used in this experiment to measure the variables first mentioned in chapter 4.2.2. To reiterate, the variables that will be measured is CPU-Utilization, RAM Memory usage, Round-Trip-Time and packet loss.

CPU-Utilization and memory usage will be measured on Windows Server 2016 with the built in “Performance Monitor” service built into the operating system. On Ubuntu 16 and FreeBSD, the “Top” command will be used to do the same.

All the above CPU-Monitoring tools will be running slightly before, during and after the tests. All will create outputs that can be saved and checked after the test has run.

Round-Trip-Time will be measured with the “TCPing” tool (Fulkerson, 2016), which is a tool that can measure the Round-Trip-Time of a connection. This will also show if any packet loss occurs.

Measurements are started by the monitoring host in the 10.0.1.0 subnet using a number of scripts. One script were created for each operating system. These scripts can be examined in appendix A. The tools “Top” and “Performance Monitor” were chosen as measurement tools due to the operating systems having them installed by default. For measuring Round-Trip-Time, “TCPing” were chosen as it was being used in (Treseangrat et al., 2015).

5.6 System Reliability Factors

This experiment is set up as a completely isolated network utilizing the resources in the NSA-lab at the University of Skövde. As such, there is no physical network connection to the university network. This allows the experiment to run without risking harm to the school network. It also allows the experiment to be free from outside influences that could affect network performance.

All devices, that is, computers, routers and switches are all password protected to ensure that no unauthorized editing of settings to the experiment environment can occur.

The experiment is run in different phases. First all baseline tests are done after one another for all three systems. Only one operating system is ever powered on and running at any point. This process is then repeated for the two defence methods detailed in chapter 5.3.

A freshly rebooted system might perform differently than one that has been running for some time. However, in this test, it was decided to test performance without restarting. As detailed in chapter 6.1, there is a 40 second period between all 20 tests opportunities where the server is not under any load. This allows the servers to revert back to a stadium where they are idle and resources are not under load.
6 Data Gathering

This chapter deals with the gathering and processing of data. It will also give examples of how gathered data looks, and how it was cleaned out in order to be more easily processed.

6.1 Capturing Parameters

Every operating system testing phase run for two minutes. Where twenty seconds go into collecting data without an ongoing attack, then data is collected during an attack that lasts one minute, data is collected for twenty seconds after the attack, and then a period of twenty seconds passes before the test begin again. This is repeated twenty times.

During the test, each operating systems respective tool takes a measure of CPU and RAM usage every two seconds. As the “Top” command has minor differences between Ubuntu and FreeBSD, different arguments had to be used. These parameters can be examined more closely in the test scripts presented in appendix A.

Also during the test, the monitor computer checks Round-Trip-Time and packet loss at the same intervals that the hardware monitors on the servers run, that is, take a measure every two seconds.

6.2 Filtering Data

Since data is not completely the same in the output for CPU and memory utilization. Scripts will be used to convert all data to a more uniform format. This format will be in .csv table format, so that the data can easily be used to create diagrams and perform calculations. Examples of the three different types of raw performance data can be seen in Figure 6-1. See Table 6-1 for an example of how the cleaned data files will look.

<table>
<thead>
<tr>
<th>Measure Number</th>
<th>CPU % Used</th>
<th>Memory % Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>8.3</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>8.2</td>
</tr>
<tr>
<td>3</td>
<td>13.6</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Table 6-1: Example Cleaned CPU/RAM Data
Windows CPU measurement will be rounded to the nearest 1 decimal value. The CPU value of the top command will be calculated by simply taking the idle percentage value, and subtract it from 100, leaving the total percentage of processor usage.

The same will be done for the memory value, getting a percentage value by calculating how much memory is free from the total memory available.

Measurement for Round-Trip-Time will also be converted into a more easily managed format. Since this data looks the same for all tests, only one script is needed. See Figure 6-2 for an example of how raw Round-Trip-Time data looks.

**Figure 6-2: Example Round-Trip-Time Data**

<table>
<thead>
<tr>
<th>Time</th>
<th>Receive Time</th>
<th>Status</th>
<th>Bytes</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.158ms</td>
<td>1.864ms</td>
<td>200</td>
<td>6065</td>
<td>~26031.441</td>
</tr>
<tr>
<td>1.208ms</td>
<td>1.818ms</td>
<td>200</td>
<td>6065</td>
<td>~26686.410</td>
</tr>
<tr>
<td>1.201ms</td>
<td>1.830ms</td>
<td>200</td>
<td>6065</td>
<td>~26520.807</td>
</tr>
</tbody>
</table>
7 Data Analysis

In this chapter, the method for analysing data will be explained. That is, how the raw data collected has been processed in order to be able to use the data for comparisons. In addition the data will be analysed for the baseline and defence methods applied.

7.1 CPU and RAM Usage

As was mentioned in Chapter 6.2, raw data files are filtered out into more uniform and manageable forms using PowerShell scripts. These scripts parse through all twenty files for each test, and removes extraneous information, leaving only the data that is interesting. These scripts can be examined in Appendix A.

These sorted data files are then combined into a Microsoft Excel file where the average value is calculated for every test moment, of which there are 50. That is, since there are 50 measurements during each test, the average value for every 2 seconds across all tests are calculated. The result of these calculations can be examined in Appendix B.

The data of averages that have been created is then used to create diagrams using Microsoft Excel 2013, this is done so as to give a visual demonstration of the performance difference. These averages is what have been examined in this chapter when mentioning highest and lowest values taken.

Due to a slight error in timing for test done on the FreeBSD systems, there were 4 less measurement moments done during the test for those systems, thus these lines start 8 seconds into the every testing session to bring the diagram results more in line with the other systems, this choice was made as there are still 10 measurements made before the attack which was deemed adequate for checking before-attack results.

Lastly this chapter will also present the findings of a longer, one hour test session. This test session was made in order to see if the results during extended attacks are the same as during shorter tests.

7.2 Round Trip Time and Packet Loss

The nature of how “TCPing” measures Round-Trip-Time (RTT) provides a challenge in how to present the results. As the tool started one measurement, and then did not perform another one until the first measurement was either successful or failed. This makes a comparison of every test interval as for CPU and RAM as detailed in chapter 7.1 unfitting for this comparison.

As such, RTT and packet loss will be examined in a case by case basis. Where the highest and lowest values recorded for every defence methods will be examined.

7.3 Baseline Analysis

As can be seen in Figure 7-1. All systems perform at about an equal rate before the tests at about 0-2%. Some systems utilize a little bit more at the first measurement, this spike in CPU activity is because of the measurement program starting. The worst performer during
the baseline tests during attacks were the Windows Apache system, which utilized on average between 40.98% and 46.96% between every measurement during tests. Interestingly, Windows with IIS webserver performed best with averaging between 21.665% and 25.42% during each measurement. Close to this was FreeBSD that utilized between 24.345% and 28.64%. Lastly in the middle is Ubuntu which utilized between 29.025% and 35.25%.

We can also see in Figure 7-2, that RAM usage remains steady throughout the entire test session, and thus is not impacted at all during these shorter tests, which we will see is the norm during all tests.

As for RTT, the pre-attack results for Windows IIS was between 0.8 to 1.2 ms, during the attacks results become more varied with between 1 to 1.6 ms, spiking rarely at 7 ms. For Windows Apache the pre-attack results stay between 1 and 1.25 ms, during the attack it goes up to between 1.2 to 3 ms, rarely spiking as high as 17ms. Ubuntu results at pre-attack generally at between 0.8 to 1.2 ms, sometimes going as low as 0.6ms. During the attack, response is very similar, only rarely going above 1.2ms. Lastly, the FreeBSD results are pre-attack between 1 and 1.25 ms, and during attack goes up to values ranging between 1 and 5 ms.

![Baseline CPU Comparison Diagram](image)

**Figure 7-1: Baseline CPU Comparison Diagram**
7.4 Load Balancing Analysis

As can be seen in Figure 7-3, the placement of best performer is again Windows IIS at first place with values between 10.71% and 13.7%, followed closely by FreeBSD with results between 15.08% and 17.395%. Ubuntu with 18.625% and 20.31%, and lastly Windows Apache with the greatest gap of results between 17.715% and 27.88%.

Again it can be seen in Figure 7-4 that RAM is remaining steady throughout the entire test, and is thus under negligible impact.

RTT for Windows IIS during these tests are between 0.9 and 1.1 during both pre- and during attacks, occasionally it spikes as high as 1.5 but these instances are rare. For Windows Apache pre-attack are between 0.9 and 1.1 ms. During attacks it increases to 1.1 to 3 ms. With occasional spikes at +1000 ms. With Ubuntu the general RTT during and before attack is between 0.8 and 1.1 ms. During two testing sessions, the first RTT as the attack started was at around 3000 ms. Lastly, for FreeBSD the RTT is between 0.9 and 1.1 ms pre-attack, and 1.1 to 3 ms during the attack, spiking rarely at +100 ms.
7.5 Thresholding Analysis

As can be seen in Figure 7-5, this defence method puts all operating systems at very similar levels. With Windows IIS using only between 0.87% and 2.45%, Windows Apache between
1.21% and 2.49%. Ubuntu uses between 0.895% and 1.7% and FreeBSD between 0.795% and 2.15%.

Like the other tests, RAM is essentially unaffected, with only KB being added and removed during the test.

RTT for this defence method is for all systems generally short and similar, interspersed with spikes of two to nine thousands of milliseconds. It is also the only test in which there are actual packet losses. This is because of the nature of the defence that packets are dropped as soon as a set amount of specific bits have passed the interface.

When the spikes do not happen, Windows IIS has a span of 0.8 to 1.3 ms, Windows Apache between 0.9 to 1.5 ms. The span for Ubuntu is between 0.8 to 1.2 ms. Lastly, FreeBSD has a span of 0.8 to 1.5 ms.

Figure 7-5: Thresholding Comparison Diagram

### 7.6 Longer Test

In these longer tests, only the Baseline and Load Balancing will be examined, as the results for the thresholding test are so similar that it is hard to notice any actual difference between the systems, similar to what can be seen in the shorter thresholding test analysis in chapter 7.5.

These tests are made using the same defensive methods, but using two additional attack computers, this is done so as to see if the ratio between the most successful performance from the longer tests are any different from the shorter, lighter test attacks. In addition, the tests collected measurements at an equal rate as the shorter tests, resulting in a total of
1850 measurements during the full hour test. In order to be able to visualize this in a chart, only every 10th value was used. Therefore, the charts contain data from 185 measurements.

Figure 7-6, which depicts the long test baseline, shows that once again, the worst performer during the baseline is Windows 2016 with Apache installed, generally ranging between 50-90%, followed by Ubuntu at a more stable 60-70% span. Windows IIS and FreeBSD are performing at a very similar rate in the 34-56% range, although Windows IIS spiked measured above this at five occurrences.

In Figure 7-8, which shows the results from the longer load balancing test, it can be seen that the relationship is the same, with Windows Apache performing worst at 22-48.8%, going to 50% and above at 8 opportunities. Followed by Ubuntu with the most stable results at between 30-41%. Once again FreeBSD and Windows IIS performing very similarly, with Windows IIS measuring generally between 11-36.4%, it did measure above this at three opportunities. And FreeBSD measuring between 13-31%, measuring above this at one opportunity.

![Figure 7-6: Long Test Baseline CPU Comparison](image)
Comparison of system performance during DDoS attacks in modern operating systems

Figure 7-7: Long Test Baseline RAM Comparison

Figure 7-8: Long Test Load-Balancing CPU Comparison
As can be seen in Figure 7-7 and Figure 7-9 figures, RAM usage does not fluctuate but have steady increases and decreases. FreeBSD and Ubuntu has a higher increase than the Windows systems. This could be attributed to measurement data taken, taking up more space on those systems.

The latency during the longer baseline tests for Windows IIS was in the range of 0.8-1.68 ms, it did however measure between 2-7.5 ms at five occurrences, and between 11-15.7 ms at five occurrences. Ubuntu performed at 0.6-1.7 ms, and only went above 2 ms at two occurrences. Windows Apache and FreeBSD performed similarly here at 10-43 ms, although Windows went above 60ms at four occurrences, while FreeBSD never went above 43 ms.

During the Load Balancing tests, a similar ratio could be observed. With Windows IIS having a general RTT of 0.6-1.67 ms during attacks, going between 30-61 ms four times, and above 3000 ms three times. Coming close is Ubuntu with an RTT of 0.6-1.7 ms during attacks, rising to between 8-14 ms six times, and spiking at over 40 ms four times. FreeBSD and Windows Apache were close to each other once again with Windows Apache generally performing between 11-29.8 ms, and spiking above 3000 ms three times. Whereas FreeBSD performed generally within 13-29 ms, rising above 30 ms two times, and above 3000 ms two times as well.
8 Conclusions

This chapter will draw conclusions from the analysis of this report, as well as discuss whether the research questions posed in chapter 3.1 have been answered.

8.1 Conclusions Based on Analysis

The data presented in chapter 7 has shown that there is a clear difference with how each operating system, with specific webserver software installed, performs during the same amount of attack load during the shorter, more data powerful tests. Whereas during the longer, more intense test, the operating systems blend together more in their performance. Where FreeBSD and Windows IIS perform similarly, and Windows Apache and Ubuntu performs closely together.

Since the analysis part is dealing with averaging values across twenty separate tests, it is difficult to put an exact number on the effect the different defence methods have on the operating systems. It does seem however that the effectiveness of the load balancing defence is about 50% on the CPU/RAM performance, which is not very surprising considering all attack traffic is split in half. The thresholding defence method brings all operating systems down to a very similar level. With only CPU usage of Windows Apache performing at 0.4% more load than the other three.

As for RTT, a great difference in performance could not be observed pre- and during attacks in the shorter tests. Only differing between a half and two milliseconds for each operating system. In general though, Windows IIS and Ubuntu performed RTT about 0.2 milliseconds faster than Windows Apache and FreeBSD. This did not change notably for the different defence methods, except during the thresholding defence, where the nature of it created packet loss and RTT in the thousands due to the router not letting more traffic pass after a certain point.

During the longer tests, more spikes in Round-Trip-Time could be observed, as well as an actual difference. During these tests, it could be seen that Windows IIS and Ubuntu performed at an on average similar level at below 2 milliseconds. Whereas Windows Apache and FreeBSD generally had an RTT of above 10 milliseconds.

It should be noted, that the longer test suffer from the validity threat of low statistical power, since it only consist of one extended test.

An interesting thing to note, is that across both the long and shorter tests, Ubuntu has performed the most stable CPU-wise. This is especially obvious during the longer test, where Figure 7-6 and Figure 7-8 show us that it has a much smoother curve than the other participants.

8.2 Answers to Research Questions

Judging from the results produced by this study, the answer to the sub-question posed in chapter 3.1, is that the defence methods have a similar effect across all operating systems. And neither system performs better with them than another.
Comparison of system performance during DDoS attacks in modern operating systems

The results produced by this study shows that there is a big difference in how much CPU power is utilized during attacks. The ratio of best performer remaining steady through all the shorter tests. Windows IIS and FreeBSD are very closely tied CPU performance wise during the shorter tests, and overlapping greatly in the longer test. However, the longer tests showed that Windows IIS in general has a faster RTT. Ubuntu is shown to not be the best performer, but it is the most stable CPU-wise. Windows Apache has been shown to be have the worst performance, both in CPU and RTT.

These results differ from (Treseangrat et al., 2015), which tested UDP flood attacks against Windows Server 2012 and Ubuntu 13, where for example Ubuntu 13 performed better with 1% over Windows Server 2012 during that experiment. Where in this study, the difference is more noticeable. This difference may be because of the nature of the attack or configuration of web service.

9 Discussion
This study has produced results in a lab setup. And while it was shown that in this study, Windows IIS perform best, it is important to remember that all systems have been used with their original configurations intact. And a live webservers performance is highly dependent on each situational need for that specific network and estimated workload. This study has shown that there is a difference how these particular machines performed.

The results of this study could be used by system administrators as one of the steps of selecting a server operating system for webservers they administrate. As this study have shown that there is a difference between how well the different operating systems handles the same amount of traffic. This means that the operating systems potentially have a greater capacity for dealing with high traffic outside of a DDoS attack. Other things system administrators should take into consideration for selecting operating system is the security of the systems. This is however outside of the scope of this study and not it's focus.

During the study's duration, some changes were made from the initial plans. Initial plans were to test against different attack methods, in particular, it was planned to test against UDP Flood attacks. This was however omitted due to time restraints. Another change that was made were the testing times. Initially, every test in of the twenty repeated tests this study performs were planned to perform for five minutes and were in this shape during initial testings. However, this time was shortened due as it was noticed that there did not seem to be a noticeable difference in performance between five and one minute tests.

10 Future work
This study has tested specific operating systems against a specific attack with specific defence mechanisms. The defence mechanisms used in this study is very basic and simple in their usage, therefore it might be interesting to in the future test with more in depth and complex defences. This study has also used default configurations of each webserver software on each operating system. A future study could be to test with webserver settings optimized for a specific system, so as to get a more real-life accurate simulation.
References


Appendix A – Test Scripts
The following is a series of scripts that are used in order to perform the experiment. Each script have the same functions. These functions are:

1. Start the CPU/RAM data gathering process. This entails connecting to the server computer and starting the CPU/RAM gathering tool. These are “Windows Performance Monitor” for Windows Server 2016, and “Top” for FreeBSD 11 and Ubuntu 16. The commands vary for this in each script.

2. Start the RTT data gathering process. This is done the same in all scripts by using the TCPing tool.

3. Wait for 20 seconds, then start the attack. This is done by connecting to the attack computers by ssh and starting the siege tool. A while loop is used in order to connect to several attack computers. The script was constructed as such to be able to increase attack computers easily if the need arrived. This function is the same in all scripts.

4. Wait for twenty seconds before restarting the script, repeating it X amount of times. In this case, it was set to repeat 20 times.

Ubuntu Test Script

$IPField = "10.0.1."
$TestCounter = 12

#Go to folder that contains putty and TCPing.exe
cd "C:\Program Files\PuTTY\"

#Loop that repeats test a number of times
while ($TestCounter -le 20){
    if ($TestCounter -lt 10){
        $TestNumber = "0$TestCounter"
    } else {
        $TestNumber = $TestCounter
    }
    #These filepaths are changed depending on which test is being done
    $OutputPath = "c:\measure\Ubuntu\UbuntuTest_XXX$TestNumber.csv"
    $LatencyPath = "c:\measure\Ubuntu\UbuntuLatencyTest_XXX$TestNumber.csv"
    $StartIP = 30
    start powershell -argument "plink student@10.0.2.100 top -d 2 -n 50 -p 1 -b > $OutputPath"
    start powershell -argument "tcping -n 20 -i 2 -h 10.0.2.100 > $LatencyPath"
    write-output "Measuring Test $TestCounter Activated on Ubuntu 16"
    Start-Sleep -s 20
}

#Loop that starts a number of attack machines
while ($StartIP -le 31){
    $StartIP = $StartIP
    $Added = $IPField + $StartIP
    start powershell -argument "plink custom@$Added /usr/local/bin/siege -t 1m -c255 10.0.2.100"
Comparison of system performance during DDoS attacks in modern operating systems

```powershell
# start powershell -argument "plink custom@$Added ping -c 5 10.0.1.5"
Write-output "$Added is started"
$StartIP = $StartIP+1
Start-Sleep -s 1
}
#Initiate grace period
Start-Sleep -s 100
write-output "Test $TestCounter Has ran."
$TestCounter = $TestCounter+1
}```
FreeBSD Test Script

$IPField = "10.0.1."
$TestCounter = 1

# Go to folder that contains putty and TCPing.exe
cd "C:\Program Files\PuTTY"

# Loop that repeats test a number of times
while ($TestCounter -le 20){
    if ($TestCounter -lt 10){
        $TestNumber = "0$TestCounter"
    } else{
        $TestNumber = $Testcounter
    }
    # These filepaths are changed depending on which test is being done
    $OutputPath = "c:\measure\FreeBSD\BSDTest_XXX$TestNumber.csv"
    $LatencyPath = "c:\measure\FreeBSD\BSDLatencyTest_XXX$TestNumber.csv"
    [int]$StartIP = 30
    start powershell -argument "plink -pw XXX root@10.0.2.100 top -d 50 -s 2 %CPU > $OutputPath"
    start powershell -argument "tcping -n 20 -i 2 -h 10.0.2.100 > $LatencyPath"
    write-output "Measuring Test $TestCounter Activated FreeBSD 11"
    Start-Sleep -s 20

    # Loop that starts a number of attack machines
    while ($StartIP -le 31){
        $StartIP = $StartIP
        $Added = $IPField + $StartIP
        start powershell -argument "plink custom@$Added siege -t 1m -c255 10.0.2.100"
        Write-output "$Added is started"
        $StartIP = $StartIP+1
        Start-Sleep -s 1
    }
    # Initiate grace period
    Start-Sleep -s 100
    write-output "Test $TestCounter Has ran."
    $TestCounter = $TestCounter+1
}
Windows Test Script

```powershell
$IPField = "10.0.1."
$TestCounter = 1
#Go to folder that contains putty and TCPing.exe
cd "C:\Program Files\PUTTY\"
#Create Password so as no not need inputting when measurement starts
$Username = "Administrator"
$Password = "XXX"
$Secstr = New-Object -TypeName System.Security.SecureString
$Password.ToCharArray() | ForEach-Object {$secstr.AppendChar($_)}
$Cred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $Username, $Secstr

#Loop that repeats test a number of times
while ($TestCounter -le 20){
    if ($TestCounter -lt 10){
        $TestNumber = "0$TestCounter"
    } else{
        $TestNumber = $Testcounter
    }
    #These filepaths are changed depending on which test is being done
    $OutputPath = "c:/measure/Windows/WindowsIISTest_XXX$TestNumber"+".csv"
    $LatencyPath = "c:\measure\Windows\WindowsIISTest_XXX$TestNumber"+"Latency.csv"
    [int]$StartIP = 30
    $CPUCounter = "\Processor(_Total)\% Processor Time"
    $MemCounter = "\Memory\% Committed Bytes In Use"
    $Array = @()
    $Array += $CPUCounter
    $Array += $MemCounter
    $Array += $OutputPath
    Invoke-Command -ComputerName 10.0.2.100 -ArgumentList $Array -ScriptBlock { Get-Counter $args[0,1] -SampleInterval 2 -MaxSamples 50 > $args[2]} -credential $Cred -AsJob
    start powershell -argument "tcping -n 20 -i 2 -h 10.0.2.100 > $LatencyPath" #55
    write-output "Measuring Test $TestCounter Activated on Windows Server 2016"
    Start-Sleep -s 20
    ##Loop that starts a number of attack machines
    while ($StartIP -le 31){
        $Added = $IPField + $StartIP
        start powershell -argument "plink custom@$Added siege -t 1m -c255 10.0.2.100"
    }
    write-output "$Added is started"
    $StartIP = $StartIP+1
    Start-Sleep -s 1
}
#Initiate grace period
Start-Sleep -s 100
write-output "Test $TestCounter Has ran."
$TestCounter = $TestCounter+1
```
Appendix B – Average Data Values

The following tables shows the results of the data gathering process. The data presented in the table are averaged values from 20 separate testing sessions. To clarify: Count 1 is the average value of Count 1 from all 20 testing sessions. Count 2 is the average from Count 2 of all 20 testing session, and so on. As is explained in chapter 7.1.

The percentages are percentages of all CPU/RAM resources used. For example. As according to the system resources specified in chapter 5.2, if the “Windows IIS CPU(%)” column shows 23.27, it is 23.27% of the total CPU resources of the system being used at that point.

### Baseline Data

<table>
<thead>
<tr>
<th>Count</th>
<th>Windows IIS CPU(%)</th>
<th>Windows IIS RAM(%)</th>
<th>Windows Apache CPU(%)</th>
<th>Windows Apache RAM(%)</th>
<th>Ubuntu CPU(%)</th>
<th>Ubuntu RAM(%)</th>
<th>FreeBSD CPU(%)</th>
<th>FreeBSD RAM(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.94</td>
<td>8,675</td>
<td>1,755</td>
<td>8,835</td>
<td>9.2</td>
<td>5.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.96</td>
<td>8,685</td>
<td>1.03</td>
<td>8.84</td>
<td>0.3</td>
<td>5.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.265</td>
<td>8,695</td>
<td>0.24</td>
<td>8.83</td>
<td>0.175</td>
<td>5.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.705</td>
<td>8,645</td>
<td>0.12</td>
<td>8.83</td>
<td>0.25</td>
<td>5.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>8,645</td>
<td>0.08</td>
<td>8.83</td>
<td>0.25</td>
<td>5.6</td>
<td>0.3</td>
<td>5.965</td>
</tr>
<tr>
<td>6</td>
<td>0.775</td>
<td>8,65</td>
<td>0</td>
<td>8.83</td>
<td>0.175</td>
<td>5.6</td>
<td>0.24</td>
<td>5.965</td>
</tr>
<tr>
<td>7</td>
<td>0.08</td>
<td>8,66</td>
<td>0.24</td>
<td>8.83</td>
<td>0.225</td>
<td>5.585</td>
<td>0.28</td>
<td>5.965</td>
</tr>
<tr>
<td>8</td>
<td>1.545</td>
<td>8,67</td>
<td>0.08</td>
<td>8.835</td>
<td>0.225</td>
<td>5.585</td>
<td>0.34</td>
<td>5.965</td>
</tr>
<tr>
<td>9</td>
<td>0.155</td>
<td>8,675</td>
<td>0.12</td>
<td>8.83</td>
<td>0.275</td>
<td>5.55</td>
<td>0.38</td>
<td>5.965</td>
</tr>
<tr>
<td>10</td>
<td>0.04</td>
<td>8,69</td>
<td>0.635</td>
<td>8.83</td>
<td>0.275</td>
<td>5.55</td>
<td>0.22</td>
<td>5.965</td>
</tr>
<tr>
<td>11</td>
<td>0.24</td>
<td>8,715</td>
<td>0.04</td>
<td>8.83</td>
<td>0.275</td>
<td>5.52</td>
<td>0.16</td>
<td>5.965</td>
</tr>
<tr>
<td>12</td>
<td>0.04</td>
<td>8,68</td>
<td>0</td>
<td>8.83</td>
<td>0.275</td>
<td>5.52</td>
<td>0.2</td>
<td>5.965</td>
</tr>
<tr>
<td>13</td>
<td>0.31</td>
<td>8,685</td>
<td>0.16</td>
<td>8.83</td>
<td>0.15</td>
<td>5.52</td>
<td>0.22</td>
<td>5.965</td>
</tr>
<tr>
<td>14</td>
<td>0.04</td>
<td>8,68</td>
<td>0.04</td>
<td>8.83</td>
<td>0.3</td>
<td>5.52</td>
<td>0.08</td>
<td>5.965</td>
</tr>
<tr>
<td>15</td>
<td>0.115</td>
<td>8,685</td>
<td>0</td>
<td>8.83</td>
<td>0.225</td>
<td>5.52</td>
<td>0.18</td>
<td>5.965</td>
</tr>
<tr>
<td>16</td>
<td>0.82</td>
<td>8,725</td>
<td>22.35</td>
<td>8.83</td>
<td>4.525</td>
<td>5.52</td>
<td>13.795</td>
<td>5.965</td>
</tr>
<tr>
<td>17</td>
<td>12.35</td>
<td>8,685</td>
<td>42.125</td>
<td>8.83</td>
<td>29.025</td>
<td>5.53</td>
<td>24.375</td>
<td>5.99</td>
</tr>
<tr>
<td>18</td>
<td>23.44</td>
<td>8,665</td>
<td>44.63</td>
<td>8.83</td>
<td>33.68</td>
<td>5.53</td>
<td>24.615</td>
<td>6.015</td>
</tr>
<tr>
<td>19</td>
<td>23.21</td>
<td>8,675</td>
<td>43.415</td>
<td>8.83</td>
<td>33.765</td>
<td>5.585</td>
<td>25.61</td>
<td>6.02</td>
</tr>
<tr>
<td>20</td>
<td>24.38</td>
<td>8,765</td>
<td>43.655</td>
<td>8.83</td>
<td>33.695</td>
<td>5.585</td>
<td>25.99</td>
<td>6.015</td>
</tr>
<tr>
<td>21</td>
<td>23.27</td>
<td>8,67</td>
<td>43.795</td>
<td>8.83</td>
<td>33.955</td>
<td>5.6</td>
<td>25.635</td>
<td>6.02</td>
</tr>
<tr>
<td>22</td>
<td>23.46</td>
<td>8,67</td>
<td>42.58</td>
<td>8.83</td>
<td>34.02</td>
<td>5.6</td>
<td>25.08</td>
<td>6.015</td>
</tr>
<tr>
<td>23</td>
<td>22.395</td>
<td>8,67</td>
<td>42.465</td>
<td>8.83</td>
<td>33.79</td>
<td>5.6</td>
<td>25.715</td>
<td>6.02</td>
</tr>
<tr>
<td>24</td>
<td>23.215</td>
<td>8,67</td>
<td>44.015</td>
<td>8.83</td>
<td>34.08</td>
<td>5.6</td>
<td>25.09</td>
<td>6.02</td>
</tr>
<tr>
<td>25</td>
<td>23.335</td>
<td>8,66</td>
<td>44.4</td>
<td>8.83</td>
<td>34.01</td>
<td>5.6</td>
<td>25.695</td>
<td>6.025</td>
</tr>
<tr>
<td>26</td>
<td>23.78</td>
<td>8,665</td>
<td>43.18</td>
<td>8.83</td>
<td>34.15</td>
<td>5.6</td>
<td>24.69</td>
<td>6.025</td>
</tr>
<tr>
<td>27</td>
<td>23.915</td>
<td>8,665</td>
<td>43.915</td>
<td>8.83</td>
<td>33.735</td>
<td>5.62</td>
<td>26.245</td>
<td>6.035</td>
</tr>
<tr>
<td>28</td>
<td>23.67</td>
<td>8,665</td>
<td>43.1</td>
<td>8.83</td>
<td>34.305</td>
<td>5.62</td>
<td>26</td>
<td>6.03</td>
</tr>
<tr>
<td>29</td>
<td>23.255</td>
<td>8,665</td>
<td>44.915</td>
<td>8.83</td>
<td>34.32</td>
<td>5.675</td>
<td>25.075</td>
<td>6.04</td>
</tr>
<tr>
<td>30</td>
<td>23.365</td>
<td>8,665</td>
<td>42.85</td>
<td>8.83</td>
<td>34.71</td>
<td>5.675</td>
<td>26.705</td>
<td>6.04</td>
</tr>
<tr>
<td>Count</td>
<td>Windows IIS CPU(%)</td>
<td>Windows IIS RAM(%)</td>
<td>Windows Apache CPU(%)</td>
<td>Windows Apache RAM(%)</td>
<td>Ubuntu CPU(%)</td>
<td>Ubuntu RAM(%)</td>
<td>FreeBSD CPU(%)</td>
<td>FreeBSD RAM(%)</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------</td>
<td>--------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>---------------</td>
<td>---------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>2,74</td>
<td>7,545</td>
<td>2,605</td>
<td>9,095</td>
<td>6,91</td>
<td>3,945</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1,405</td>
<td>7,545</td>
<td>1,065</td>
<td>9,09</td>
<td>0,05</td>
<td>3,945</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0,545</td>
<td>7,525</td>
<td>1,565</td>
<td>9,085</td>
<td>0,25</td>
<td>3,945</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0,08</td>
<td>7,525</td>
<td>0,43</td>
<td>9,07</td>
<td>0,125</td>
<td>3,945</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0,2</td>
<td>7,525</td>
<td>0,08</td>
<td>9,075</td>
<td>0,075</td>
<td>3,945</td>
<td>0,22</td>
<td>5,435</td>
</tr>
<tr>
<td>6</td>
<td>0,08</td>
<td>7,525</td>
<td>0,04</td>
<td>9,075</td>
<td>0,175</td>
<td>3,945</td>
<td>0,2</td>
<td>5,44</td>
</tr>
<tr>
<td>7</td>
<td>1,205</td>
<td>7,545</td>
<td>0,12</td>
<td>9,085</td>
<td>0,1</td>
<td>3,945</td>
<td>0,18</td>
<td>5,44</td>
</tr>
<tr>
<td>8</td>
<td>0,12</td>
<td>7,545</td>
<td>0,36</td>
<td>9,095</td>
<td>0,1</td>
<td>3,945</td>
<td>0,16</td>
<td>5,44</td>
</tr>
<tr>
<td>9</td>
<td>0,545</td>
<td>7,54</td>
<td>0,16</td>
<td>9,095</td>
<td>0,075</td>
<td>3,935</td>
<td>0,2</td>
<td>5,44</td>
</tr>
<tr>
<td>10</td>
<td>0,315</td>
<td>7,54</td>
<td>0,63</td>
<td>9,095</td>
<td>0,05</td>
<td>3,935</td>
<td>0,16</td>
<td>5,44</td>
</tr>
<tr>
<td>11</td>
<td>0,235</td>
<td>7,54</td>
<td>0,08</td>
<td>9,115</td>
<td>0,275</td>
<td>3,925</td>
<td>0,12</td>
<td>5,44</td>
</tr>
<tr>
<td>12</td>
<td>0,08</td>
<td>7,54</td>
<td>0,7</td>
<td>9,125</td>
<td>0,25</td>
<td>3,925</td>
<td>0,16</td>
<td>5,44</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>7,54</td>
<td>0,115</td>
<td>9,12</td>
<td>0,2</td>
<td>3,93</td>
<td>0,16</td>
<td>5,44</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>7,54</td>
<td>0,24</td>
<td>9,105</td>
<td>0,375</td>
<td>3,93</td>
<td>0,08</td>
<td>5,44</td>
</tr>
<tr>
<td>15</td>
<td>0,08</td>
<td>7,54</td>
<td>0,195</td>
<td>9,11</td>
<td>0,125</td>
<td>3,93</td>
<td>0,14</td>
<td>5,44</td>
</tr>
<tr>
<td>16</td>
<td>6,425</td>
<td>7,535</td>
<td>9,875</td>
<td>9,11</td>
<td>2,425</td>
<td>3,93</td>
<td>4,605</td>
<td>5,44</td>
</tr>
<tr>
<td>17</td>
<td>13,13</td>
<td>7,535</td>
<td>17,715</td>
<td>9,115</td>
<td>15,41</td>
<td>3,93</td>
<td>9,88</td>
<td>5,465</td>
</tr>
<tr>
<td>18</td>
<td>11,54</td>
<td>7,535</td>
<td>20,86</td>
<td>9,075</td>
<td>19,23</td>
<td>3,93</td>
<td>13,31</td>
<td>5,52</td>
</tr>
<tr>
<td>19</td>
<td>12,36</td>
<td>7,535</td>
<td>22,99</td>
<td>9,085</td>
<td>19,355</td>
<td>3,945</td>
<td>15,19</td>
<td>5,55</td>
</tr>
<tr>
<td>Count</td>
<td>Windows IIS CPU(%)</td>
<td>Windows IIS RAM(%)</td>
<td>Windows Apache CPU(%)</td>
<td>Windows Apache RAM(%)</td>
<td>Ubuntu CPU(%)</td>
<td>Ubuntu RAM(%)</td>
<td>FreeBSD CPU(%)</td>
<td>FreeBSD RAM(%)</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>--------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>1</td>
<td>2.2</td>
<td>8.645</td>
<td>2.04</td>
<td>7.675</td>
<td>2.085</td>
<td>3.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.47</td>
<td>8.625</td>
<td>1.15</td>
<td>7.945</td>
<td>0.15</td>
<td>3.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.665</td>
<td>8.6</td>
<td>0.315</td>
<td>7.88</td>
<td>0.275</td>
<td>3.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.385</td>
<td>8.61</td>
<td>0.355</td>
<td>7.815</td>
<td>0.175</td>
<td>3.86</td>
<td>0.1</td>
<td>5.8</td>
</tr>
<tr>
<td>5</td>
<td>1.635</td>
<td>8.615</td>
<td>0.28</td>
<td>7.645</td>
<td>0.05</td>
<td>3.86</td>
<td>0.12</td>
<td>5.8</td>
</tr>
<tr>
<td>6</td>
<td>1.05</td>
<td>8.6</td>
<td>0.24</td>
<td>7.64</td>
<td>0.15</td>
<td>3.86</td>
<td>0.22</td>
<td>5.8</td>
</tr>
<tr>
<td>7</td>
<td>0.715</td>
<td>8.615</td>
<td>0.355</td>
<td>7.645</td>
<td>0.1</td>
<td>3.86</td>
<td>0.22</td>
<td>5.8</td>
</tr>
<tr>
<td>8</td>
<td>0.08</td>
<td>8.615</td>
<td>0.08</td>
<td>7.645</td>
<td>0.25</td>
<td>3.86</td>
<td>0.235</td>
<td>5.805</td>
</tr>
</tbody>
</table>
Comparison of system performance during DDoS attacks in modern operating systems

<table>
<thead>
<tr>
<th></th>
<th>0.235</th>
<th>8.615</th>
<th>0.12</th>
<th>7.64</th>
<th>0.325</th>
<th>3.86</th>
<th>0.28</th>
<th>5.81</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.93</td>
<td>8.61</td>
<td>2.535</td>
<td>7.645</td>
<td>0.3</td>
<td>3.86</td>
<td>1.685</td>
<td>5.815</td>
</tr>
<tr>
<td>11</td>
<td>1.505</td>
<td>8.615</td>
<td>1.71</td>
<td>7.645</td>
<td>1.7</td>
<td>3.865</td>
<td>1.62</td>
<td>5.815</td>
</tr>
<tr>
<td>12</td>
<td>1.405</td>
<td>8.625</td>
<td>1.37</td>
<td>7.645</td>
<td>1.42</td>
<td>3.865</td>
<td>2.15</td>
<td>5.92</td>
</tr>
<tr>
<td>13</td>
<td>1.765</td>
<td>8.6</td>
<td>1.445</td>
<td>7.645</td>
<td>1.195</td>
<td>3.865</td>
<td>1.57</td>
<td>6.035</td>
</tr>
<tr>
<td>14</td>
<td>1.215</td>
<td>8.54</td>
<td>1.675</td>
<td>7.64</td>
<td>1.22</td>
<td>3.865</td>
<td>1.495</td>
<td>6.05</td>
</tr>
<tr>
<td>15</td>
<td>2.45</td>
<td>8.545</td>
<td>2.435</td>
<td>7.64</td>
<td>1.195</td>
<td>3.865</td>
<td>1.58</td>
<td>6.12</td>
</tr>
<tr>
<td>16</td>
<td>1.635</td>
<td>8.54</td>
<td>1.95</td>
<td>7.64</td>
<td>1.29</td>
<td>3.865</td>
<td>1.61</td>
<td>6.185</td>
</tr>
<tr>
<td>17</td>
<td>1.205</td>
<td>8.54</td>
<td>1.83</td>
<td>7.64</td>
<td>1.415</td>
<td>3.875</td>
<td>1.435</td>
<td>6.21</td>
</tr>
<tr>
<td>18</td>
<td>1.135</td>
<td>8.545</td>
<td>2.025</td>
<td>7.64</td>
<td>1.295</td>
<td>3.875</td>
<td>1.34</td>
<td>6.225</td>
</tr>
<tr>
<td>19</td>
<td>0.87</td>
<td>8.545</td>
<td>1.955</td>
<td>7.64</td>
<td>1.295</td>
<td>3.89</td>
<td>1.32</td>
<td>6.225</td>
</tr>
<tr>
<td>20</td>
<td>1.525</td>
<td>8.55</td>
<td>1.415</td>
<td>7.64</td>
<td>1.19</td>
<td>3.89</td>
<td>0.795</td>
<td>6.215</td>
</tr>
<tr>
<td>21</td>
<td>1.57</td>
<td>8.565</td>
<td>1.645</td>
<td>7.64</td>
<td>1.61</td>
<td>3.895</td>
<td>1.16</td>
<td>6.195</td>
</tr>
<tr>
<td>22</td>
<td>1.445</td>
<td>8.605</td>
<td>1.865</td>
<td>7.64</td>
<td>1.295</td>
<td>3.895</td>
<td>1.1</td>
<td>6.195</td>
</tr>
<tr>
<td>23</td>
<td>0.99</td>
<td>8.57</td>
<td>1.86</td>
<td>7.635</td>
<td>1.045</td>
<td>3.92</td>
<td>1.28</td>
<td>6.195</td>
</tr>
<tr>
<td>24</td>
<td>1.34</td>
<td>8.565</td>
<td>1.885</td>
<td>7.635</td>
<td>1.3</td>
<td>3.92</td>
<td>1.01</td>
<td>6.2</td>
</tr>
<tr>
<td>25</td>
<td>1.68</td>
<td>8.57</td>
<td>1.79</td>
<td>7.635</td>
<td>1.2</td>
<td>3.925</td>
<td>1.435</td>
<td>6.185</td>
</tr>
<tr>
<td>26</td>
<td>1.565</td>
<td>8.575</td>
<td>1.21</td>
<td>7.63</td>
<td>1.075</td>
<td>3.925</td>
<td>1.415</td>
<td>6.165</td>
</tr>
<tr>
<td>27</td>
<td>1.71</td>
<td>8.575</td>
<td>1.53</td>
<td>7.63</td>
<td>1.265</td>
<td>3.93</td>
<td>1.4</td>
<td>6.155</td>
</tr>
<tr>
<td>28</td>
<td>1.17</td>
<td>8.61</td>
<td>1.69</td>
<td>7.63</td>
<td>1.175</td>
<td>3.93</td>
<td>1.3</td>
<td>6.145</td>
</tr>
<tr>
<td>29</td>
<td>1.25</td>
<td>8.675</td>
<td>1.665</td>
<td>7.63</td>
<td>1.175</td>
<td>3.945</td>
<td>1.07</td>
<td>6.15</td>
</tr>
<tr>
<td>30</td>
<td>1.64</td>
<td>8.58</td>
<td>1.28</td>
<td>7.63</td>
<td>1.345</td>
<td>3.945</td>
<td>1.395</td>
<td>6.15</td>
</tr>
<tr>
<td>31</td>
<td>1.41</td>
<td>8.59</td>
<td>1.285</td>
<td>7.63</td>
<td>1.1</td>
<td>3.96</td>
<td>1.22</td>
<td>6.145</td>
</tr>
<tr>
<td>32</td>
<td>1.175</td>
<td>8.61</td>
<td>1.83</td>
<td>7.63</td>
<td>1.25</td>
<td>3.96</td>
<td>1.13</td>
<td>6.14</td>
</tr>
<tr>
<td>33</td>
<td>1.28</td>
<td>8.62</td>
<td>1.865</td>
<td>7.63</td>
<td>1.22</td>
<td>3.98</td>
<td>1.41</td>
<td>6.135</td>
</tr>
<tr>
<td>34</td>
<td>2.025</td>
<td>8.635</td>
<td>2.49</td>
<td>7.63</td>
<td>0.895</td>
<td>3.98</td>
<td>1.075</td>
<td>6.13</td>
</tr>
<tr>
<td>35</td>
<td>1.72</td>
<td>8.95</td>
<td>1.5</td>
<td>7.63</td>
<td>1.05</td>
<td>3.98</td>
<td>1.33</td>
<td>6.125</td>
</tr>
<tr>
<td>36</td>
<td>1.165</td>
<td>8.885</td>
<td>1.48</td>
<td>7.63</td>
<td>1.05</td>
<td>3.98</td>
<td>1.375</td>
<td>6.11</td>
</tr>
<tr>
<td>37</td>
<td>1.065</td>
<td>8.8</td>
<td>1.55</td>
<td>7.63</td>
<td>1.245</td>
<td>3.99</td>
<td>1.355</td>
<td>6.11</td>
</tr>
<tr>
<td>38</td>
<td>1.48</td>
<td>8.65</td>
<td>1.665</td>
<td>7.63</td>
<td>1.555</td>
<td>3.99</td>
<td>1.205</td>
<td>6.105</td>
</tr>
<tr>
<td>39</td>
<td>1.095</td>
<td>8.65</td>
<td>1.215</td>
<td>7.63</td>
<td>1.265</td>
<td>3.99</td>
<td>0.98</td>
<td>6.09</td>
</tr>
<tr>
<td>40</td>
<td>1.02</td>
<td>8.655</td>
<td>1.84</td>
<td>7.63</td>
<td>1.19</td>
<td>3.99</td>
<td>0.745</td>
<td>6.085</td>
</tr>
<tr>
<td>41</td>
<td>0.48</td>
<td>8.66</td>
<td>0.275</td>
<td>7.63</td>
<td>1.345</td>
<td>3.985</td>
<td>0.3</td>
<td>6.06</td>
</tr>
<tr>
<td>42</td>
<td>0.47</td>
<td>8.66</td>
<td>0.39</td>
<td>7.63</td>
<td>0.7</td>
<td>3.985</td>
<td>0.3</td>
<td>6.04</td>
</tr>
<tr>
<td>43</td>
<td>0.39</td>
<td>8.625</td>
<td>0.24</td>
<td>7.63</td>
<td>0.375</td>
<td>3.98</td>
<td>0.32</td>
<td>6.02</td>
</tr>
<tr>
<td>44</td>
<td>0.275</td>
<td>8.615</td>
<td>0.2</td>
<td>7.63</td>
<td>0.325</td>
<td>3.98</td>
<td>0.26</td>
<td>6.015</td>
</tr>
<tr>
<td>45</td>
<td>0.28</td>
<td>8.615</td>
<td>0.16</td>
<td>7.63</td>
<td>0.375</td>
<td>3.955</td>
<td>0.32</td>
<td>6.005</td>
</tr>
<tr>
<td>46</td>
<td>0.275</td>
<td>8.61</td>
<td>0.315</td>
<td>7.63</td>
<td>0.225</td>
<td>3.955</td>
<td>0.24</td>
<td>5.985</td>
</tr>
<tr>
<td>47</td>
<td>0.315</td>
<td>8.61</td>
<td>0.28</td>
<td>7.63</td>
<td>0.575</td>
<td>3.945</td>
<td>0.24</td>
<td>5.96</td>
</tr>
<tr>
<td>48</td>
<td>0.2</td>
<td>8.61</td>
<td>0.04</td>
<td>7.63</td>
<td>0.175</td>
<td>3.945</td>
<td>0.18</td>
<td>5.945</td>
</tr>
<tr>
<td>49</td>
<td>0.12</td>
<td>8.605</td>
<td>0.315</td>
<td>7.63</td>
<td>0.225</td>
<td>3.93</td>
<td>0.18</td>
<td>5.925</td>
</tr>
<tr>
<td>50</td>
<td>0.355</td>
<td>8.605</td>
<td>0.275</td>
<td>7.625</td>
<td>0.15</td>
<td>3.93</td>
<td>0.22</td>
<td>5.92</td>
</tr>
</tbody>
</table>