honeyCAPTCHA: AN ENHANCED INTRUSION DETECTION FRAMEWORK

By

ABDULLAHI, Mukhtar Ahmad

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES,

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MARCH, 2021

honeyCAPTCHA: AN ENHANCED INTRUSION DETECTION FRAMEWORK

By

ABDULLAHI, Mukhtar Ahmad

P16PSCS8043

A DISSERTATIONSUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES,
AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF
MASTER OF SCIENCE (M.Sc.) DEGREE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCE

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA.

MARCH, 2021

CERTIFICATION

This project dissertation entitled "honeyCAPTCHA: An Enhanced Intrusion Detection Framework" by ABDULLAHI MUKHTAR AHMAD (P16PSCS8043) meets the regulations governing the award of M.Sc. in Computer Science and is approved for its contribution to knowledge and literary presentation.

_____ _____

Dr. S. Aliyu Date
Chairman, Supervisory Committee

_____ _____

Professor S.B. Junaidu Date
Member, Supervisory committee

_____ _____

Dr. M. Abdulkarim Date
Head of Department

_____ _____

Professor S.A. Abdullahi Date
Dean, School of Postgraduate Studies

DECLARATION

I hereby declare that this dissertation titled "**honeyCAPTCHA: An Enhanced Intrusion Detection Framework**" was carried out by me under the supervision of **Dr. S. Aliyu** and **Prof S. B. Junaidu.** This research work has not been presented for the award of any degree in any institution before now and all sources of information are specifically acknowledged by means of references.

ABDULLAHI, Mukhtar Ahmad _____

Student Name **Date**

DEDICATION

This research work is dedicated to my entire family

ACKNOWLEDGEMENT

Many thanks to Almighty Allah for bestowing upon me the capacity to carry out this research work. I sincerely acknowledge the intellectual guidance, encouragement and supervision of my supervisors: Dr. S. Aliyu and Professor S.B. Junaidu who tirelessly through thick and thin, painstakingly ensured that this work is balanced and completed in good time. Sirs, I am most grateful, may the Almighty Allah guide and preserve all that relates to you and your families in your race through life.

My gratitude also goes to Head of Department, Prof. S. B. Junaidu and other lecturers of the Department. May the Lord keep you and grant all that your heart desire in this profession.

My heartfelt appreciation goes to my family, for putting in me the discipline that brought me this far.

ABSTRACT

Internet is no doubt inevitable; it has a tremendous impact in our lives. Despite its importance, internet comes with many challenges, among which is security. From the literature, several attempts have been made to develop a secure and user-friendly spam detection technique. But these attempts linger between these two fundamental issues: the robustness and the usability. The passiveness of some Intrusion Detection Systems for example, the CAPTCHA, which failed to detect some forms of attacks with low usability, is of major concern to researchers in recent years. In this work, an enhanced intrusion detection framework named honeyCAPTCHA capable of detecting web page crawlers, resilient and efficient to users is designed to solve the inefficient and weak security settings from the traditional IDS. The system is mainly considered as an option to the CAPTCHA-BASED IDS model which inherited the passiveness and inefficient problems that lingered between all the traditional IDPS. Eliminating CAPTCHA entirely on the system gateway, providing a cognitive CAPTCHA test, setting response time to the dummy page and a fallback opportunity are the mean ingredients that makes the proposal system successful. The proposed system outperforms the existing system considering its performance measure with a detection rate (DR) of 76%, which is 1.7 times the detection rate of the existing system and a false positive rate (FPR) of 10% as against that of the existing system with 36% FPR. This shows that our system is more robust compare to the existing system. The usability of proposed system measured using BDR and BNR is 1.5 times that of the existing system, this shows how efficient proposed system is to users when compared to the existing system. Both systems were also compared based on the standard IDS evaluation metrics CID, of which our system is 2.26 times the existing system. Indeed, honeyCAPTCHA can be guaranteed to be a more secure and user friendly IDP system over the use of directs CAPTCHA.

Table of Contents

List of Figures

List of Tables

Abbreviations

AIS	Artificial Immune System
AUC	Area Under a Curve
BDR	Bayesian Detection Rate
CAPTCHA	Completely Automated Turing Test to Tell Computer and Human Apart
Caret	**C**lassification **A**nd **Re**gression **T**raining
CID	Intrusion Detection Capability
CR	Classification Rate
DR	Detection Rate
e1071	functions for latent class analysis
FB-NSA	Boundary-fixed NSA
FM	F-Measure
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HIDS	Hybrid Intrusion Detection System
honeyCAPTCHA	Honeypot + CAPTCHA techniques
HTML5	Hypertext Markup Languauge version 5
CSS	Cascading Style Sheet
IDPs	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IT	Information Technology

MIS	Mammalian Immune System
NIDS	Network Intrusion Detection System
NSA	Negative Selection Algorithm
OALFB-NSA	Online Adaptive Learning Boundary-Fixed NSA
OCR	Optical Character Recognition
PHP	Hypertext Preprocessor
PIN	Personal Identification Number
TN	True Negative
TP	True Positive

Definition of terms

Artificial Immune System (AIS): can be defined as a computational system inspired by the principles and processes of the Mammalian Immune System (MIS).

Bayesian Detection Rate (BDR): a Bayesian representation of Positive predictive Value (PPV), which is the probability of an intrusion when the IDS outputs an alarm.

CAPTCHA: distinguishes the human users and machines by giving a simple test to the human of which the machine might not be able to solve.

Caret : Is a package (short for **C**lassification **and Re**gression **T**raining) which set a function that attempt to streamline the process for creating predictive models.

Detection Rate (DR): The ratio between the number of correctly detected attacks and the number of the attacks.

e1071: is a function for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier

False Positive Rate (FPR: It is the ratio between the number of normal instances detected as attack and the total number of normal instances

honeyCAPTCHA: Is an enhanced Intrusion detection framework that make a genuine user access to a system easy, living bots with a cognitive CAPTCHA, where the solution to the CAPTCHA is fruitless.

Mammalian Immune System (MIS): Is a collection of organs; tissues, cells and enzymes all united under one goal; to protect the mammal body.

Negative Selection Algorithm: Is one of the main algorithms in AIS inspired from the negative selection of an adaptive immune system

OCR: Optical Character Recognition of human as a means of testing to enable access to a given resources.

IDPS: Intrusion Detection and Prevention are two broad terms describing application security practices used to mitigate attacks and block new threats.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Computer security is a field in IT that focuses on the protection of both computer hardware and software resource. The Internet as a major tool in IT needs to be secured for its enormous impact on our daily life. Hence, security on the internet is now becoming an appealing area of research. There is no doubt that Artificial Intelligence has come with a huge progress in our strive for technological advancement, although with the abuse of the Artificial Intelligence technology, it has now become a monster that stands to be a threat to security. Cyber-attack in the past was of concern to office holders and government, but today cyber-attack is of general concern to all as it can trigger war and political instability (Joseph**,** 2018). There is no defined feature that will qualify anyone to fall a victim of cyber-crime apart from being on the internet. Legitimate users of the internet can be attacked by web-bots in many ways some of which include: social engineering, malvertising, ransomware, phishing and spy phishing, malware, sql injection. While attacking users, bots cause severe harm to victims ranging from loss of victim's files, losing computer control, hardware destruction and possibly the victim's life. About 4.5 million identities were stolen in 2017 approximately more than the internet users. Cyber-criminals will continuously target identities and steal credential of internet users in 2018 (Joseph**,** 2018).

Attacks, cyber-crime, intruders, fraudsters as they may be called are generally regarded as internet threats which consist of all act of anomaly against computer security. Some of the known threats as listed by (Public Safety Canada, 2017) are as follows:

i. Botnets, which refers to the collection of software bots that creates an army of infected computers (known as zombies) that are controlled by the originator.

ii. The distributed denial of service (DDoS) attack, which refers to the flooding of a network with useless information by the zombie computers purposely to sabotage the website.

iii. Hacking, a process of getting an unauthorized access to a computer.

iv. Malware, a malicious software that infects victim's computer as a scareware to intimidate the victim, reformat his/her hardware leading to loss of files or stealing of information.

v. Pharming, a fraud that redirects a legitimate user to an illegitimate website on visiting a legitimate URL that was spoofed hence, luring victims to give out his/her vital information to scammers.

vi. Phishing are fake emails, text messages or websites sent by scammers that will lure victims to give out his /her information to them.

vii. Ransomware are types of malware that restrict victim's access to his/her computer or a vital file inside the system, with a display message of payment before the restriction will be removed.

viii. Spam are annoying junk emails that create a burden to communication service providers and businesses to filter electronic messages which can be used to phish a victim's information without his consent

ix. Spoofing are fake emails and websites that are created and look similar to real ones which are used in conjunction with phishing techniques to exploit victim's information.

x. Spamware or Adware are software products that collect a victim's information without his consent. It comes in the form of free download and is installed automatically with or without your consent.

xi. Trojan horse is an embedded software or disguise on legitimate software. It is an executable file that will install itself and run automatically once it's downloaded. Trojan horse can delete victim's files, use a victim's computer to hack other computers, watch a

victim's through his/her web cam, log a victim's keystrokes (such as a credit card number entered in an online purchase), record usernames, passwords and other personal information.

xii. Viruses: Malicious computer programs that are often sent as an email attachment or a download with the intent of infecting a victim's computer, as well as the computers of everyone in the victim's contact list. Just visiting a site can start an automatic download of a virus.

xiii. Worms are a common threat to computers and the Internet as a whole. A worm, unlike a virus, goes to work on its own without attaching itself to files or programs. It lives in a victim's computer memory, doesn't damage or alter the hard drive and propagates by sending itself to other computers in a network – whether within a company or the Internet itself.

xiv. Wpa2 handshake vulnerabilities: The Key reinstallation attack (or Krack) vulnerability allows a malicious actor to read encrypted network traffic on a Wi-Fi Protected Access II (WPA2) router and sends traffic back to the network. Krack can affect both personal (home users and small businesses) and enterprise networks.

Measures or techniques used to protect computer and internet resources against those attacks consist of:

i. Authentication: ensures that users and computers are who they claim to be by establishing proof of identity. This is usually accomplished based on one or a combination of something you are (a biometric e.g., such characteristics as a voice pattern, handwriting or a fingerprint), something you know (a secret e.g. a password, Personal Identification Number (PIN) or cryptographic key) or something you have (a token e.g. a credit card or a smart card).

ii. Encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot. *Encryption* does not itself prevent interference, but denies the intelligible content to a would-be interceptor.

iii. Firewall is a network security system that monitors and controls over all your incoming and outgoing network traffic based on advanced and a defined set of security rules.

iv. Intrusion Detection System (IDS) is a process used to identify intrusions. Intrusion detection techniques have been traditionally classified into one of two methodologies: anomaly detectionor misuse detection.

v. Intrusion Prevention System is an advanced IDS that is capable of preventing a known detective intrusion. One of the prominent IPS is CAPTCHA

Other techniques that are used as a compliment or even substitute of the re-known techniques listed above as described by Muhammad and MuhammadReza (2015) who categorizes them into three as: Interactive methods, Administrative method and Cheating bots.

i. Interactive method provides mechanism to distinguish between human users and bots in the form of request to do an action that bots could not usually perform such as motion (mouse) based operations or presenting human only assets like social security number, sms/email verification etc.

ii. Administrative method involves high level supervision task such as detecting spam content or preventing bots from attack. The use of third-party services for content analysis and filtering is one of the reliable tools used by administrators to detect spammers. Techniques used under these categories include: centralized sign-on, Limiting account, Logging, Server-side validation, Response time etc.

iii. Cheating bots is reverse techniques that identify bots by attracting them to do an action that human users could not do. Some techniques under these categories consist of honeypot, switching form fields and confirmation pages

1.2 Problem Statement

Despite effort by researchers to stabilize the tradeoff between security and usability in IDPs, problems in developing a unique user friendly and a secure framework for detection and prevention of all spambots still persist. An attempt by Souley and Abubakar (2018) to develop a CAPTCHA-BASED IDS left challenges that include:

i. Presenting CAPTCHA in the user interface. CAPTCHA is dreaded that people are frustrated by seeing it (Michael, 2017).

ii. The instruction: "do not type anything in this textbox" in Souley and Abubakar (2018) seems ambiguous to daily CAPTCHA users. Hence human may neglect it and fall a victim.

iii. Bots that skipped CAPTCHA test can easily gain access to the system.

iv. Their system only considers retrieving intelligent bots i.e. those capable of breaking the CAPTCHA neglecting those bots that did not attempt the CAPTCHA solution which are also threat to our system security.

v. More so their system has no positive fallback for genuine users to prove wrong the decision of the system by miss-threatening.

Our work will focus on providing a solution to these problems by:

i. Eliminating CAPTCHA in human interface, and replacing it with a decoy field that will detect and redirect all bots to a honeyCAPTCHA, as a means of observing their behavior. Hence our CAPTCHA will only be visible to the bots.

ii. The instructions will be eliminated to avoid confusion by users.

iii. All bots IP addresses will be captured and separated as intelligent and non-intelligent through their attempt toward solving the deceitful CAPTCHA.

iv. Use response time technique and Classical Negative selection algorithm to provide a fallback opportunity for the misclassified humans

1.3 Motivation

This research is motivated by the lingering tradeoff between security and usability (Mohammad & MohammadReza, 2015) of an Intrusion Detection and Prevention Systems (IDPS) turning the situation to a cat and mouse game. Hackers are getting faster whilst defenders are treading water. Cybercriminals are way ahead of the game against defenders without having to try anything new according to the latest edition of Verizon's benchmark survey of security breaches (John, 2016). Honeypots techniques were mainly employed for studying new threat due to the advancement of artificial intelligence. Using honeypot alone makes most systems vulnerable to attacks. Researchers have proven that the combination of honeypot and other known techniques like firewall, CAPTCHAs and other IDS will enable robust security and improve efficiency of the IDPS systems (Parita *et al*., 2016). Most of the designed IDPS systems left this problem unsolved.

1.4Aim and Objectives

This research aims at designing an enhanced intrusion detection framework by improving the CAPTCHA BASED IDS.

The objectives of the research are to:

i. Develop honeyCAPTCHA, an enhanced intrusion detection framework
ii. Implement the developed framework
iii. Evaluate the developed framework

1.5 Research Methodology

a) Designing honeyCAPTCHA intrusion detection framework which consist of:
    i. Login form with a decoy field hidden using JavaScript to detect bots.
    ii. Cognitive CAPTCHA that will receive the detected bots.
    iii. Dummy page that will receive intelligent bots that pass the CAPTCHA test
        a. Setting a time limit of 5 seconds to fill a Compose mail form in a dummy page.
        b. Creating a fallback opportunity for genuine users that were classified as bots using a Classical Negative Selection Algorithm considering the time set limits in the Compose mail form.

b) Implementation consist of:
    i. Hosting the framework on a similar site used by Souley and Abubakar (2018) to justify the data collection.
    ii. Retrieve all the visitors IP address, with IP address retrieval algorithm in Souley and Abubakar (2018).

c) Evaluation of the robustness and usability based on the visitors IP addresses collected from the two systems using the following metrics:
    i. Robustness: measured by False Positive Rate (FPR), Detection Rate (DR)

ii. Usability: measured with Bayesian Detection Rates (BDR) (Gu *et al.*, 2006) using caret and e1071 packages in R languages on.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the concepts of Intrusion detection and prevention system, the CAPTCHA, Honeypots, Response time and Negative Selection Algorithm, which are relevant techniques that are used in the development of the proposed system. It also discusses the Naïve Bayes classification algorithm that will be use to model our systems dataset. Also, prominent evaluation metrics that were used to evaluate the existing system and a single unique metric that is used in evaluating all IDS systems will be discussed in this chapter. The tools that will be use in designing and analyzing our system are also discussed. Shortcomings of IDS and CAPTCHA are briefly discussed with relevant literatures that led to the combination of honeypot with IDPS. Gaps identified in the literature that motivated our work were also discussed.

2.2 Intrusion Detection System

IDS detect and alerts on possible malicious attack even within networks. The IDS sensors can be place at various locations throughout the network. It is normally signature based, looking for predefined signatures of bad events. As soon as the known signature is detected it updates the firewall's filtering rules. Most of the intrusion detection systems (IDS) use one of the two detection methods: misused detection or Anomaly detection (Gupta, 2015).

IDS are described broadly according to the chart in Figure 2.1 which defines the types and the detection techniques.

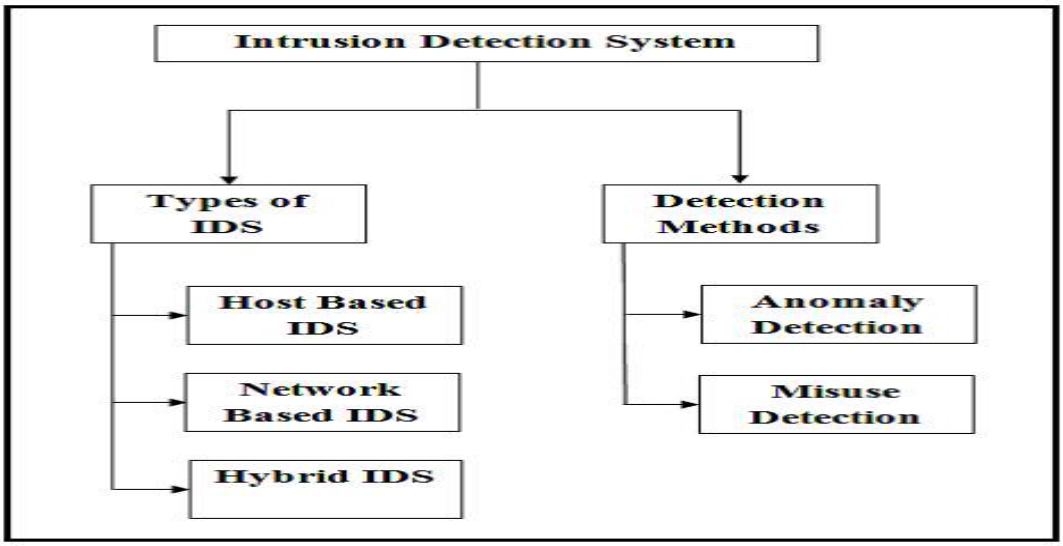


Figure 2.1: Classification of IDS (Sahasrabuddhe *et al*, 2017)

These types are discussed as follows;

i. Host based IDS: Data are collected from a single host or different host log records that include audit record of operation system, system logs, application programs information and so on.
ii. Network-based IDS (NIDS): Data collected from the network collective flow to network section such as: Internet packets.
iii. Hybrid IDS (HIDS): HIDS will monitor the network traffic for a specific host as well as it will monitor all network traffic like NIDS.

The Detection method can also be discussed as follows;

i. Anomaly-based IDS: It attempts to find malicious activities by measuring the traffic deviation from the expected baseline. It relies on machine learning and artificial intelligence to classify the network traffic into normal or abnormal.
ii. Signature-based IDS: Human analysts investigate suspicious traffic, extract features of known intrusions and use the predefined signatures to discover malicious packets.

2.3 CAPTCHA

CAPTCHA distinguishes the human users and machines by giving a simple test to the human of which the machine might not be able to solve. CAPTCHA helps in preventing the artificial intelligence software which are identified as bots (which imitate human users) from endangering web services by performing those malicious activities like the spamming and other frauds. Over the years, CAPTCHA have been deployed and employed in various organization as a preventive tool against bots.

CAPTCHA is considered pessimistic in security policy that exists in front end of the web as preventive functions. Mohammad and MohammadReza (2014) categorized CAPTCHA into three groups as shown in Figure 2.2.

The classification can be considered as a developmental trend of the CAPTCHA defense system based on security and usability.

The first CAPTCHA designed was based on OCR (i.e. text-based CAPTCHA) which considers the Optical Character Recognition of human as a means of testing to enable access to a given resources. OCR CAPTCHA was developed in different variations that consist of the normal text-based, Pessimal Print, Baffle Text, Scatter Type and GIMPYm, Ez-Gimpy and ASCII art CAPTCHAs. They were developments from the security robustness motives.

The second groups under Visual CAPTCHAs are the popular and user-friendly Image based CAPTCHAs which were developed due to issues of usability and security of the previous CAPTCHAs (i.e. some of the text-based). They consist of the 2D and 3D types which are found to be user friendly and secured alternatives to most of the text-based CAPTCHAs. But due to the dwindling challenges that face the internet security and the demand for easy accessibility from human users, CAPTCHA evolution moves on to meet other classical demand.

Moving Object CAPTCHAs were invented to tackle current challenges. They consist of the video and the animation types. But the challenges to those CAPTCHAs are mainly from users who see most of the CAPTCHAs under these categories as expensive and not user friendly.

The non-Visual CAPTCHAs came into being mainly to solve the prospective user's problem like the disabilities or impairment from cognitive, age or physical deformity.

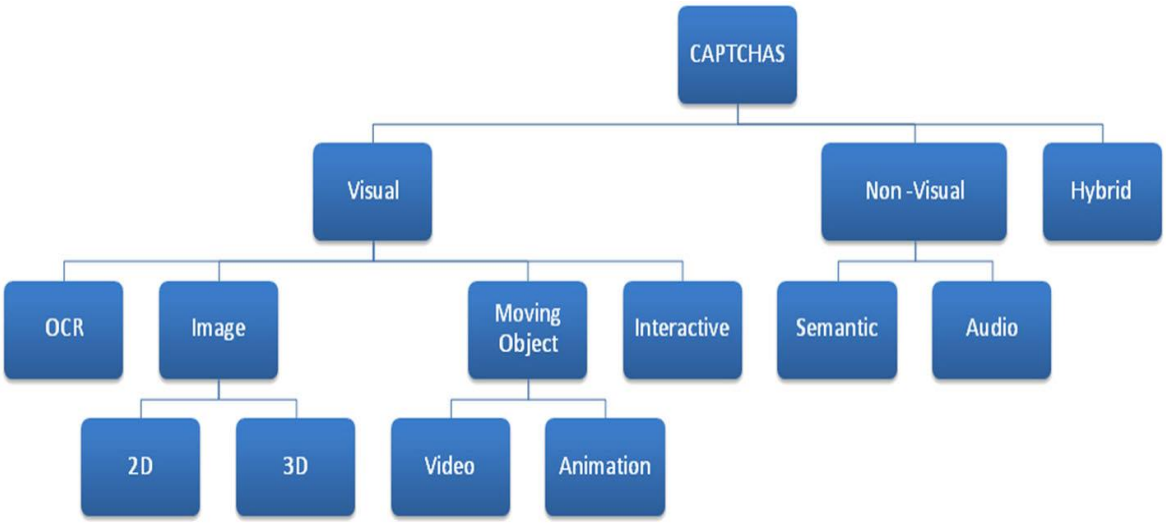


Figure 2.2: CAPTCHA Classification (Mohammad & MohammadReza, 2014).

2.4 Honeypot

Honeypot is an imitated system that portrays the behavior of a real productive system aimed at luring an intruder to maliciously exploit the system with the motive of attacking the real system. Zakaria and Ansiry (2012), described honeypots as a honey that attracts beers. A honeypot is an information system resource that is covertly deployed inside the network and purposely designed to be scanned, attacked and compromised. Honeypot techniques are used to compliment some of the known security techniques like IDS, IPS or Firewall to provide a robust security with a high performance especially for new vulnerabilities (Baykara & Das, 2015). Honeypot, when exploited, monitors the activities of the attacker, which will be used to strengthen the defense system against the attacker in subsequent attacks, hence the goal of honeypots is to have the system probe attacks (Souley & Abubakar, 2018).

Kevat (2017) outlines some of the points that we consider when implementing honeypots. They include:

i. Purpose of using the honeypots either for early warning or forensic research?
ii. The interactivity levels, which maybe low, medium or high?
iii. Should the honeypot be deployed on real or emulated system?
iv. And lastly where should it be placed, either before or after the Firewall or in the DMZ (De militarized zone)?

2.5 Response Time

Is a trick that is used on a web form to calculate the time, during which a form is filled and submitted. Real users take a few moments to read all the information and fill in the required fields. However, robots can fill it all almost instantaneously. By setting a minimum time limit for

page submissions, sites can catch out spambots without any impact on genuine users (Moth, 2013).

2.6 Negative Selection Algorithm

Artificial Immune System (AIS) can be defined as a computational system inspired by the principles and processes of the Mammalian Immune System (MIS). It uses ideas from the operation of the MIS and applies them to computational problems. Negative Selection algorithm (NSA) is one of the main algorithms in AIS inspired from the negative selection of an adaptive immune system (Ramdane & Salim, 2017). The classical NSAs consist of two stages: Firstly, the NSAs generate a detector set in the non-self-space. A candidate detector is generated in the whole space. If the detector does not match with the known self-states. It becomes a mature detector and is added to the detector set. In the second stage, the unknown states are tested with the detector set. If an unknown state is matched by any mature detector, the NSAs assert that an anomaly has occurred (Forrest *et. al.,* 1994).

2.7 Naïve Bayes Classifier

A Naïve Bayes classifier technique is based on the so-called Bayesian theorem particularly suited for situations where the dimensionality of the input is high. It is categorized as supervised learning as well as a statistical approach to classification (Stevens, 2016). Naive Bayes is considered a very simple classification that outperforms most popular classification algorithms. It is named after Thomas Bayes the proposer of the theorem (1702-1761).

Bayesian Theorem is stated mathematically in the following equation:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}, \quad (1)$$

Where P (A) and P (B) are probability of an events and P (B) $\neq 0$

P (A|B) is a conditional probability: the likelihood of event *A* occurring given that *B* is true P(B|A) is also a conditional probability: the likelihood of event *B* occurring given that *A* is true(A) and P(B) are the probabilities of observing *A* and *B* independently of each other. This is known as the marginal probability. Naïve Bayes classifiers are used in many ways which include: text classification, spam filtering, online application, real time prediction, recommendation system etc.

2.9 Technologies used in Evaluation of the System

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, and statistical inference to name a few. Most of the R libraries are written in R. The primary use of R is and will always be statistic, visualization, and machine learning. R is a great tool to explore and investigate the data. Elaborate analysis like clustering, correlation, and data reduction are done with R. This is the most crucial part, without a good feature engineering and model; the deployment of the machine learning will not give meaningful results (What is R Programming Language? Introduction and Basics, 2019). Most popular R packages that we used in this research work are caret and e1071.

Caret package (short for **C**lassification **And Re**gression **T**raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for: data splitting, pre-processing, feature selection, model tuning using resampling, variable

importance estimation as well as other functionality (*Max, 2019*). Caret can perform various data mining functionalities easier and more in a more user-friendly way than scikit-learn of python (Zhuoheng, 2018).

An e1071 is a function for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier (Domantas, 2019).

2.10 Related work

For security enhancement, several works from different perceptions have been carried out to make security robust against malicious attacks. Our review will focus mainly on the development series that lead to the development of our existing system.

According to the universal usability concept, information and communication systems should be designed in a way that could be used by a broad range of users including those with some disabilities (Muhammad & MohammedReza, 2014).

Usable accessibility incorporates usability measurements of efficiency, learnability, memorability, tolerance for error and user satisfaction into the inclusive nature of accessibility testing, ensuring that a product is usable and accessible to as wide a range of individuals as possible (Foley, 2012).

Generally, traditional IDS are passive in such a way that they detect and report attacks based on predefined rules. Traditional IDS focus mainly on how to detect attacks based on a given rule of either assignment or abnormality (Faysel & Syed, 2010). That means a new attack that is not defined in the system will not be detected, also some interactions with genuine human may ambiguously be considered a threat.

Modi *et al.* (2013) reviewed most of the soft computing techniques used in IDS development. They comprise the biologically-inspired techniques like the genetic algorithm and some major machine learning tools like Fuzzy logic, support vector, artificial neural network. On their conclusion, they emphasized that's of techniques application in IDS and IPS will optimally improve system security.

Elie and Steven (2012) Developed 'DeCAPTCHA' software which enables the computer to render the CAPTCHA code legible by cleaning up text. The software was successful in cleaning up 66% of Visa's Authorize.net CAPTCHAs, 70% of Blizzard's Entertainment and 73% of CAPTCHA.com's captcha.

Yan *et al*., (2012) designed a generic method to break all text-based CAPTCHAs, which is considered the best in generic attack, with the success rate of 5% to 77%, which can solve a puzzle at less than 15 seconds average speed in a standard desktop. Text-based CAPTCHAs are more of human friendly but vulnerable to attack.

Gaurav *et al*., (2014) designed fCAPTCHA, which consists of multiple images of human faces and non-face image with varying degree of distortion. Users get access by matching faces belonging to single individual.

Muhammad and MohammedReza, (2014) reviewed different CAPTCHAs and categorized them into three i.e. the visual, non-visual and hybrid. They evaluate them and suggest some alternatives, based on given criteria that will be considered when prioritizing the selection and implementation of the CAPTCHAs. The criteria consist of the cost, efficiency, robustness and effect on usability. They identify security and usability as the major barrier in CAPTCHA deployment. The suggested alternatives are Administrative, Interactive and cheating bots which

were tested by the same criteria with the CAPCTHA and found an improvement in both the usability and security in their deployment.

Dempsey (2014) classified methods used to exploit CAPTCHA into three, the Optical Character Recognition (OCR), Learning which used machine learning techniques to break CAPTCHAs and Farming that exploit CAPTCHA by exposing it to humans to solve based on a certain reward, known or unknown to the solver.

After a comparative analysis of different CAPTCHAs and their alternative, Parita *et al*. (2016) arrived at a conclusion with a suggestion, that honeypots and CAPTCHA have their respective weaknesses and drawbacks that makes them independently less effectively, but by integrating and removing the weaknesses they will form a viable defense against our system

But, Powell, *et al.*, (2017) designed a novel image-based CAPTCHA, employ by object recognition, inspired by negative selection algorithm of biological immune system. A two-phase filtering algorithm, which ensures that the CAPTCHA is resilient to automated attack while remaining easy for human users to solve. The image CAPTCHA is not convenient to user's and the system will be boring (Josh, 2019). Hence a need to completely eliminate the CAPTCHA should be considered.

To eliminate traditional CAPTCHA, Google introduced reCAPTCHA that makes verification simple for users by only clicking on a checkbox while making it harder for bots. The reCAPTCHA work using an advanced risk analysis that comprises of browsing history of the genuine user already tracked by google cookie just to determine the difficulty of challenge that is presented to the user, explore how aspect of the browser environment effects the risk analysis, canvass rendering techniques to fingerprint user across machine and browsers, to identify how

user-agent influence the user's reputation and the timing of movement and movement pattern of mouse to decide what type of challenges will be presented to the user (Suphannee *et al*., 2016).

An advanced no CAPTCHA reCAPTCHA, which is invisible to human, was introduced in 2017 by Google due to the trouble with reCaptcha that drives users to the extreme edge of sanity. The invisible CAPTCHA shows no challenge to user instead it returns probability scores between 0.0 (100% bots) and 1.0 (100% human). (Google no Captcha + INVISIBLE reCaptcha, 2019).

The limitation of the Google reCAPTCHA in general is that a user looks suspicious with reason not directly connected or even clear to him, like having an outdated version of browser or browsing engine, browser version does not correspond to the actual environment of experience or user-agent is mis-formatted. With that user will be presented with the dreaded CAPTCHA for advanced verification and machine can solved the image reCAPTCHA challenge almost 70.78% (Suphannee *et al.*, 2016).

Also, the latest no CAPTCHA reCAPTCHA users have no false positive fallback and opportunity for actual humans to prove wrong the decision of the system in case you receive a low score (Google no Captcha + INVISIBLE reCaptcha, 2019). Also, Google reCAPTCHA used their long-acquired database for the analysis, which restricts the technology only to them. Our systems will look similar to the invisible Google reCAPTCHA system, with no challenges for users and a decoy field that detect only bots, to pass it to the honeyCAPTCHA.

Some of the IDPS that employ Negative selection algorithm are reviewed to set our foundation to improve our system using the Negative selection techniques. The authors focus on the improvement in the detection accuracy and algorithm efficiency, through covering a non-self-

space with fewer detectors, and cover the holes by using detectors with a smaller radius (Ramdane & Salim, 2017).

Those proposed models are:

(Jinquan *et. al.,* 2009) proposed a self-adaptive NSA that uses a novel technique to adjust adaptively the self-radius and evolve non self-covering detectors, to build an appropriate profile of the system only by using a subset of self-samples, aims at reducing the number of elf elements and resolve the problem of adaptability in classical NSA. The results on Iris show that the system is an efficient solution to anomaly detection with high detection rate, low false alarm rate, self-learning and adaptation.

(Li *et. al.,* 2016) proposed a new NSA, named boundary-fixed NSA with online adaptive learning under small samples (OALFB-NSA) is proposed. In this work detectors are generated into two steps: In the first one the Boundary-fixed NSA (FB-NSA) generates a layer of detectors, which are around the self-space. These detectors are only related to the training samples, and have nothing to do with the training times. In the second step OALFB-NSA detectors can adapt themselves to real-time variety of self-space during the testing stage. Experimental comparison among proposed algorithms, V-detector and other anomaly detection algorithms on Iris datasets and biomedical dataset shows that the FB-NSA and OALFB-NSA can obtain the higher detection rate and lower false alarm rate in most cases.

Abdolahnezhad and Banirosta (2016) proposed email detection based on the modified classical NSA. The model improves the random generation of a detector in NSA with the use of both the spam and non-spam spaces. Two set of detectors are generated one for spam and other for non-

spam detectors. The experimental results in spam base datasets show that the detection performance of the model is higher than the conventional NSA.

Response time is another trick that is use to control spam inversion into our information content. The idea behind this tricky approach is to calculate the time during which forms are filled and submitted. Although it takes little time for users to complete forms, bots are almost instantaneous. System could determine bots if the form is filled out in a predefine amount of time (Muhammad & MuhammadReza, 2015). Real users take a few moments to read all the information and filled in the blanks, however robots can fill it all at instant. By setting a low minimum time limit for page submissions site can catch bots without any impact on genuine users. Unfortunately, some spammers will be wise to this trick and find a way around it, but it will at least catch out some unwanted visitors (Moth, 2013). By estimating the average time spent on a comment, one could define certain rules. For example, if a submission takes less than five seconds, which is virtually impossible for a human but just enough time for a bot to do its job, you could ask the user to try again (Bushell, 2011).

Some IDPS that employ honeypot to strengthened security and improve user's efficiency of the systems include:

Linora *et al.,* (2014) designed an intrusion detection system that depends on honey pot. They built the models of normal behavior for multitier web applications considering both front-end requests and backend database queries. It provides a container-based IDS with multiple input streams to produce the alerts and can identify a large number of attacks with the minimal false positive rate. But the system limitation was IDS works on assumption for an abnormal behavior. The IDS did not indicate the mechanism it will employ for detecting whether it is malicious activities or not.

Koniaris *et al.,* (2014) use two honeypot systems to setup a malware catcher on one hand and a malicious connections logger on the other. But only using honeypot makes the system to be weak and only the activities of intruders were logged.

Malay, *et al.,* (2015), Proposed a system that handles multiple clients using the concept of honeypot, Intrusion detection system (IDS) monitor whole network and looks for intrusion, when detected, honeypot will be activated and divert the traffic to dummy/virtual servers and backtrack the source (IP address). But the system put IDS in front of the honeypot, means the passiveness of an IDS still exist in the system

Yesugade (2016) combined the features of IDS, IPS and Honeypot. The system analyzes and tracks the behavior of the attacker by monitoring the network and capturing the log. The propose system has the sophisticated framework for investigating intruders as well as intrusion events. Detection ratio and Response time to measure the system. But the entire incoming packets are logged by the honeypot. Also, the gateway contains added verification on username/password thereby increasing user's verification process.

Resende *et al.,* (2018) used an adaptive approach based on genetic algorithm to select features for profiling and parameters for anomaly-based intrusion detection methods. Additionally, two anomaly-based methods are introduced to be coupled with the proposed approach. The system was evaluated using FPR, Detection Rate with the limitation in depending on the anomaly of the existing IDS.

Vivekan and Rajbhar (2018) designed a portable Java-based real time packet capturing with intrusion detection and prevention honeypot for windows-based operating system. But the limitation of the system is that the attack must be on the system. Rule-based IDS and the default

Firewall of the OS were used. Require a lot of computation because it implements a real time IDPS, so the performance will depend on the processing power of CPU.

Based on Parita *et al*., (2016) suggestion of combining CAPTCHA and honeypot, Souley and Abubakar (2018) developed a CAPTCHA-BASED IDS, which on their view CAPTCHA becomes IDS instead of been IPS. The system work in a way that an active CAPTCHA is presented in the gateway with a placeholder tag "DO NOT TYPE ANYTHING IN THIS TEXTBOX". Human users were tasked to read, comprehend and abide by the instruction. While an intelligent bot which targets the system will solve the CAPTCHA and pass in with the zeal of bypassing the security, but unfortunately to him will be redirected to a dummy page where his information will be tracked and instantly be blocked. The main aim of the system is to identify those intelligent bots that attack our systems, study their behavior for further research and production.

Also, the CAPTCHA-BASED IDS framework is seen in Figure 2.4, that demonstrate how the system works. From the framework the system generated a loose CAPTCHA codes, then visitors can be bots when filling the empty field or genuine users by skipping the field. All bots the field in the empty field were regarded as intelligent bots, which their IP addresses will be retrieved and they will be blocked instantly from accessing the system.

Naïve Bayes Classification techniques will be used in this work to classify our collected dataset, due to its usage in the work of Souley and Abubakar (2018) and other several works related to spam classification like the work of Gianvecchio, *et al.,* (2011) that used Entropy and Bayesian Classifiers to classify Chat bots on one of the most popular and large commercial chat networks Yahoo! Chat. The entropy classifier is to identify new chat bots and add to the Chat bot corpus

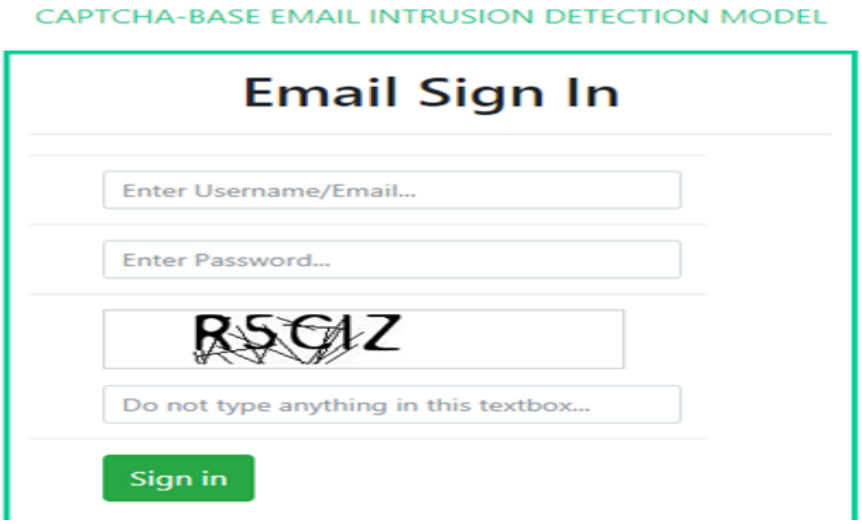


Figure 2.3: A gateway for Souley and Abubakar (2018) System.

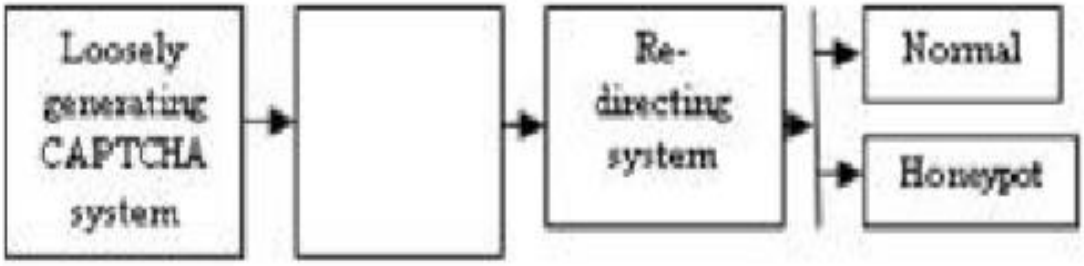


Figure 2.4: CAPTCHA-BASED IDS model framework (Souley & Abubakar, 2018)

based on a certain metrics (message sizes and inter message delay) while the Bayesian used the bots and human corpora to learn text pattern of bots and humans and then quickly classify bots based on the pattern**.** Also, Rusland *et al*., (2017) used Naïve Bayes to classify email spam tested on a different data set which they concluded that the algorithm's quality performance is based on the dataset used. Dataset with few instances of email and attributes gives good performance for Naïve Bayes Classifiers. The performance of the datasets is evaluated based on their accuracy, recall, precision and F-measure, which they concluded that the Naive Bayes classifier also can get highest precision that will give highest percentage spam message manage to block, if the dataset is collected from a single e-mail account. This is the motivated us in employing Naïve Bayes in training models to predict accurate classification.

2.11 Gap in the Literature

Despite effort by Souley and Abubakar (2018) to tackle those issues by providing a combination of Honeypot and CAPTCHA techniques in their system, much need to be done in tackling the usability and security of their system. Their limitations are what we proposed to solve in our new system. These limitations include:

i. It is language dependent, which may discourage users due to the barrier leading to an increase in error rates.
ii. CAPTCHA is becoming obsolete that most users don't want to see it. By displaying it becomes a threat and disgusting to users.
iii. Increase false positive alerts: among some reasons of using honeypot is to reduce false positive alert, in this system where a user mistakenly filled in the textbox will be considered a robot due to his interaction with the unclear system.

iv. CAPTCHA is IPS but it is used here as an IDS, where intruders that ignore it can be authenticated as a genuine user (false negative).

v. No fallback opportunity for the misclassified humans

vi. In general, the trick can be easily uncovered and that may cause more harm to the system.

CHAPTER THREE

METHODOLOGY

3.1 Design Overview

The proposed honeyCAPTCHA is intended to be design based on the following constraints.

i. The login page which is the gateway will be designed using HTML, CSS and PHP, providing only two visible fields (i.e. username and password fields). A decoy field that is hidden with a JavaScript function will be use to trap bots.

ii. A cognitive CAPTCHA is designed also with HTML, CSS and PHP, that will receive the redirected bots from the decoy field, which is dummy and deceitful.

iii. A dummy page will be designed with HTML, CSS and PHP to receive those intelligent bots that pass the CAPTCHA test, as a deceptive mechanism.

iv. IP address retrieval mechanism will be implemented with PHP function to retrieve genuine user, attempt, skilled and non-skilled bots IP address.

v. Time limit for composing a mail in the dummy page will be set to create a positive fallback opportunity for a genuine user classified as a bot.

vi. A field classical negative selection algorithm will be used to select the real bots and pass it to the admin for blocking and deleting the recorded IP addresses of the genuine user classified as a bot in the intruders' table.

vii. The Backend is to be designed using MySQL which is a database management that fully supports relational database model.

3.2 Input Interface Design

The main aim of the input design is to devise a gateway into a secure website as a genuine user and to create an obstacle by redirecting automated software from harming our valued resource. All users have one interface that receives and verifies their supplied login details. Where a genuine user will successfully be logged onto the system without attempting any security verification, while the bots will be detected by filling a hidden field, masqueraded with a JavaScript. The first page will be seen by all, and screened automated software from the genuine humans. The detected bots will be redirected to a dummy CAPTCHA page while its IP address will be retrieved. If the automated software succeeded in solving the honeyCAPTCHA, it will be recorded as intelligent bots and redirected to a dummy page, where its interaction will be timed and sent back to the login page if it exceeds the stipulated time or left to continue as a bot. If it fails to solve the CAPTCHA, its IP address will be recorded as non-intelligent, all using an IP address retrieving function in PHP (Souley & Abubakar, 2018).

3.3 Process Design

The major activities in the process design of the proposed system has to do with general algorithm of detecting and differentiating between automated software as intelligent and non-intelligent bots. Also, among the process are designing a honeyCAPTCHA for the bots and retrieving IP address of the detected bots.

3.4 Designed Algorithm

As mentioned above, five algorithms are designed to attain this research work. They include the general system algorithm which is as follows

1. Start
2. Supply login details

a. If (decoy field = 1) {
    i. Redirect to CAPTCHA challenge
    ii. IP Address mark as abnormal
        1. If (CAPTCHA = 1) {
            a. Redirect to Dummy Page.
            b. IP address mark intelligent
            c. Record access time
                i. If (access time > 5 sec) {
                    1. Goto step 2
                    2. Delete IP Address
                ii. } else {
                    1. Refresh the page
                }
        2. } else {
            a. Goto Step 2
            b. IP Address mark as non-intelligent
        }
b. } else {
    i. Redirect to real page
    ii. IP Address mark as normal
}

3. Stop

The **CAPTCHA challenge design process** with the algorithm as follows:

a. Start
b. Declare Variables and operators
    1. First numbers as random number from 1 to 10
    2. Second number as random number from 1 to 10
    3. Operators are +, -, *.
    4. Rotate between the operators as operator
c. Initiate the answer= 0;
d. Switch between operators
    1. Case +: answer = first number + second number
    2. Case - : answer = first number – second number
    3. Case *: answer = first number * second number
e. If answer = user answer; make access
f. End

The IP address retrieval function, with the algorithm as follows

i. Start
ii. Initiate variable ip to hold IP address, which will be empty
iii. Check IP address from share internet
iv. If it is not empty
    a. Assign it to ip
v. Else if empty you check IP address pass from proxy if not empty
    a. Assign it to ip
vi. Else assign remote IP address to ip
vii. End if
viii. Stop

Response time Algorithm ()

i. Start
ii. Set time out function (5 seconds)
    a. If 5 second elapsed
    b. Redirect visitor to login page
    c. End if
iii. Stop

Classical Negative Selection Algorithm to detect and forward IP address to the Admin

i. Start
ii. Initiate time sets > 5
iii. If IP address time set matches initial time sets
    a. Delete IP address from intelligent bots' table
iv. Else Detect and forward to admin page to block.
v. End if
vi. Stop

3.5 The proposed system Framework

The system framework is a modification of the CAPTCHA-Based IDS model. The framework forms a modified version of the existing system which in turn is regarded as a development in the security and usability of the verification-based systems. Figure 4 below brings out the full architecture of the proposed system. In the system, all visitors come to the login page to access the resources of the system. An attacker that fills in the form from the source code will be detected by a masqueraded JavaScript field, which will instantly be directed to the CAPTCHA challenge as deceit verification. Where the attacker succeeds in breaking the CAPTCHA, it will

still be redirected to a dummy page which is conspired to retrieve its IP address. System admin can block the IP address to avoid further confrontation, while the attacker is stored based on his advancement, either attempt bot, intelligent or non-intelligent bot. Basically at the dummy page a time frame of 5 seconds will be set to determine the actual bots that fills form instantaneously. Indeed, humans that were misclassified here can be opportune to start login afresh, in case of any error occurrences. Where their IP address can also be checked and deleted while submitting the actual bots trapped to admin for necessary action. This is done using a Classical Negative Selection Algorithm (NSA).

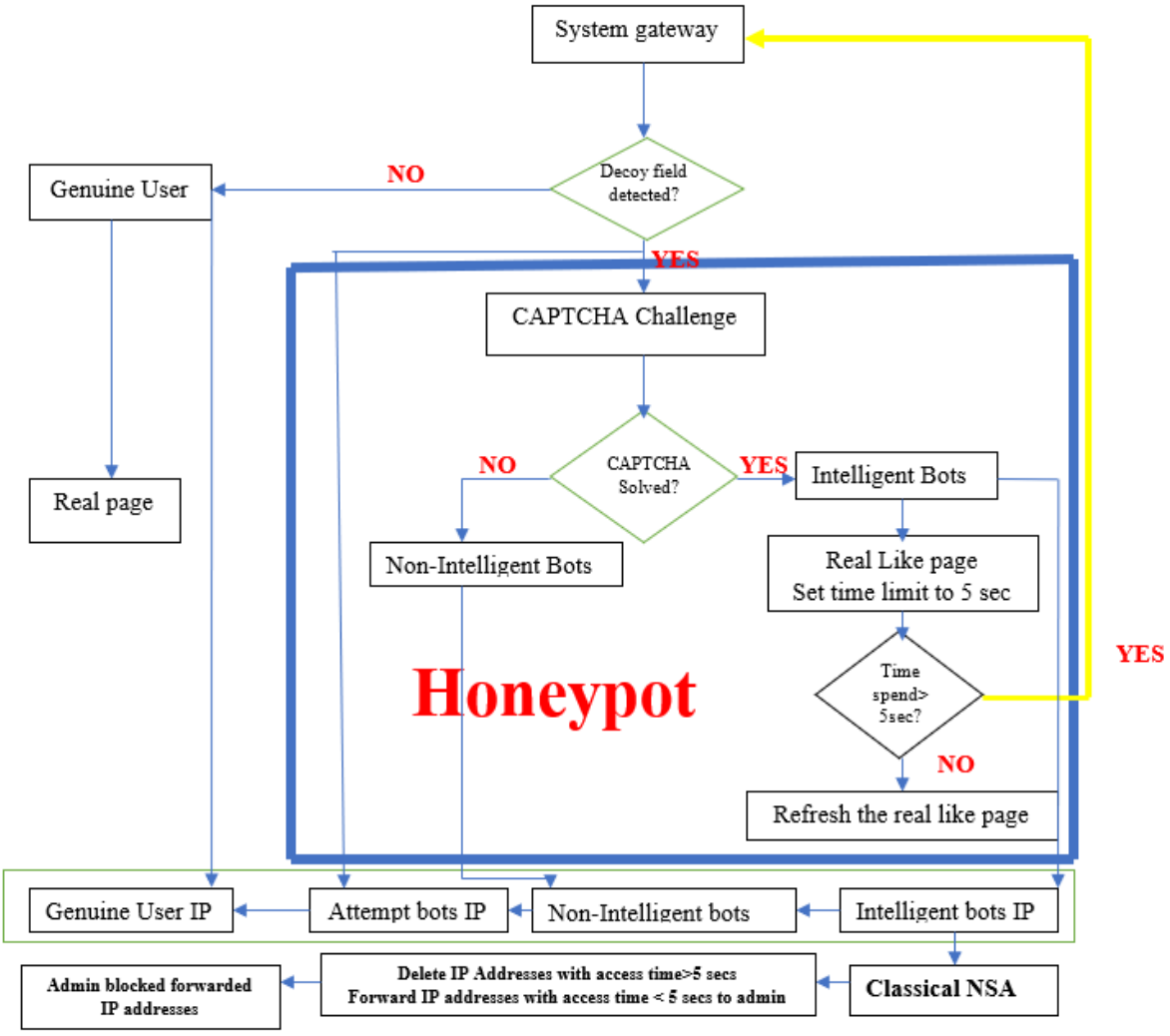


Figure 3.1: The proposed system Framework

3.6 The Flow Chart of the proposed system

Generally, the system can be directed easily using the flowchart in Figure 3.2 below.

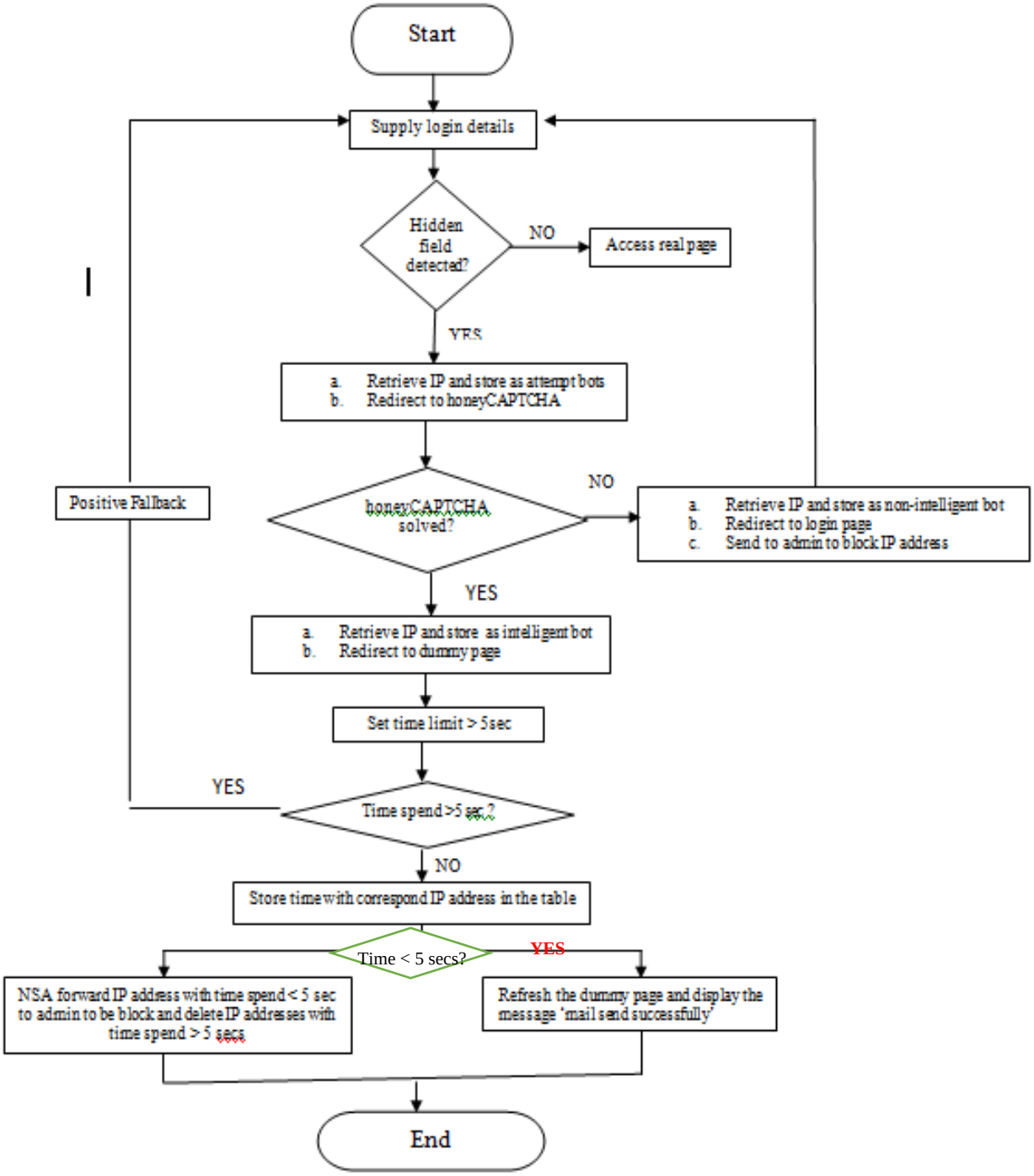


Figure 3.2: Flow Chart of the proposed system

3.7 How the Proposed system works

The proposed system as discussed earlier is mainly designed to ease user access to the verifiable resources in the internet at the same time increasing the security robustness of the system by making it harder for bots. In that regard, the proposed system comes with a unique feature of allowing a genuine user to have access to the resources by only supplying his username and password without any further verification, while at the other hand obscuring access to a detected attacker. This happen through modifying our existing system by removing the CAPTCHA test entirely to users view, while a hidden detector will redirect the attacker to a fruitless, cognitive honeyCAPTCHA to lured the attacker away from the actual system resources. A positive fall back opportunity that does not exist in the existing system will be an added advantage to misclassified humans, where an opportunity will be given to them to fall back to the gateway against earlier decision of the system. Negative Selection Algorithm is also used in our system to enable proper selection of the genuine bots identified by their time spent in their activities. These help the genuine user not to be blocked permanently from accessing the system. After the design of the system, it will be implemented on a real website to attract visitors, from all over the world, just as to identify the uniqueness and the global treatment of the system. The system robustness will be considered best on the confusion metrics of the IP addresses of the visitors taken from the two systems which will be relatively compared to summarize the performance of our system. Though the existing system only consider collecting the intelligent bots capable of breaking the CAPTCHA test. Our system will also in addition considers both the intelligent and non-intelligent bots, since we are not underrating our enemy. While comparing, we consider how robust is our system compared to that of the existing system by considering the percentage of the bot's penetration of our system to that of the existing system.

3.8 Evaluation Metrics for IDS

There are two ways in which IDS can be evaluated: Its Efficiency- an evaluation of the resources used and then its effectiveness, which describes the classification accuracy of the IDS (Tavallaee, 2011). Confusion metrics stands to be the best way to represent classification result of the IDS (Kumar, 2014). It is the metrics that represents result of classification which can be identified as either true or false category as seen in Table 2.1 below.

Table 2.1: The Confusion Metrics Table

Actual Classification	**Predicted Classification**	
	Attack	**Normal**
Attack	TP	FN
Normal	FP	TN

The Table can be explained as follows:

i. True Positive (TP): Intrusion successfully detected by the IDS.
ii. False Positive (FP): Normal non-intrusive behavior that is wrongly classified as intrusive by the IDS.
iii. True Negative (TN): Normal/non-intrusive behavior that is successfully labeled as normal/non-intrusive by the IDS.
iv. False Negative (FN): Intrusion missed by the IDS, and classified as normal/non-intrusive.

There exists no known benchmarking metrics for IDS and as a result the IDS validation is performed by measuring benchmark based on the dataset benchmark followed by their comparison with other existing representative techniques in the field. Although, confusion metrics is powerful in classification, some extended metrics that use the confusion metrics

variables will be more useful in terms of IDS. These extended metrics use some numeric variables that can be easily computed and compared for different IDS (Kumar, 2014). Matric that comes from the general confusion metrics includes:

1. Classification rate (CR): It is defined as the ratio of correctly classified instances and the total number of the instances.

$$CR = \frac{Correctly\ classified\ instances}{Total\ number\ of\ instances}$$

$$CR = \frac{TP + TN}{TP+TN+FP+FN} \quad (2)$$

2. Detection rate (DR): The ratio between the number of correctly detected attacks and the number of the attacks.

$$DR = \frac{Correctly\ detected\ attacks}{Total\ attacks}$$

$$DR = \frac{TP}{TP+FN} \quad (3)$$

3. False positive rate (FPR): It is the ratio between the number of normal instances detected as attack and the total number of normal instances.

$$FPR = \frac{Number\ of\ normal\ instances\ detected\ as\ attacks}{Total\ number\ of\ normal\ instances}$$

$$FPR = \frac{FP}{FP+TN} \quad (4)$$

4. Precision rate (PR): Fraction of data instances predicted as positive, which are actually positive. $$PR = \frac{TP}{TP+FP} \quad (5)$$

5. Recall: It measures the missing part from precision rate, which is known as the percentage from the real attack covered by the classifier. This is the same as detection rate.

6. F-Measure (FM): Is the harmonic mean of the precision and recall at a given threshold. It is mostly preferred when only one accuracy metrics is needed as evaluation criterion. $$FM = \frac{2}{\frac{1}{PR} + \frac{1}{Recall}} \quad (6)$$

7. Area under ROC curve (AUC): Originated from signal detection theory (Tavallaee, 2011), which is used to compare the efficacy of a models. It model's true performance by considering all probability cutoff. Area under a ROC curve is used as a summary statistic.
8. Sensitivity is defined as P (Pred = positive |True = positive) and is approximated by the fraction of true positives that are predicted as positive (this is the same as a recall). Specificity is P (Pred = negative |True = negative). It is approximated by the fraction of true negatives predicted as negatives.
9. Bayesian Detection Rate Is a Bayesian representation of Positive predictive Value (PPV), which is the probability of an intrusion when the IDS outputs an alarm.

$$BDR = \frac{B(1-\gamma)+(1-B)\alpha}{B(1-\gamma)} \quad (7)$$

$\boldsymbol{B}$ = Known as **base rate**; it is the measure of environment in which IDS operates (When B = 0 means 100% normal and when B = 1 it is 100% intrusion)

$\boldsymbol{\alpha}$ **= False positive rate**; Probability that an IDS output alarm when there is no intrusion.

$\boldsymbol{\gamma}$ **= False Negative rate**; Probability that an IDS do not output alarm when there is an intrusion. This metric is very important from the usability point of view because IDS alarms are useful to an intrusion response system only if the IDS have high PPV (Gu *et al*., 2006).

The proposed system in comparative studies with the CAPTCHA-BASED IDS will be tested in two ways: First, the robustness of the two system will be tested using the default evaluation metrics used in the existing system, i.e. Detection Rate, False Positive Rate, Precision, Recall, F-Score.

Also, the usability point of view will be quantitatively measured with Bayesian Detection Rate (PPV) (Gue *et al.*, 2006).

CHAPTER FOUR

SYSTEM IMPLEMENTATION, RESULT AND DISCUSSION

4.1 Introduction

This chapter presents the implementation of the proposed system and also, discussion on the results obtained as compared to the existing system. The implementation as recommended earlier is on an attractive website similar to that of the existing system. The IP addresses of the visitors of the system were captured and stored as attempt bots for those that were detected by our hidden field in the login form, intelligent for the successful bots that break the proposed honeyCAPTCHA and the non-intelligent for those that failed the test.

4.2 Implementation Requirement.

The implementation of the honeyCAPTCHA system is on a designed email service provider, which is similar to that used by the existing system. We used R data analysis tools to analyze our collected data from the database. Then compared side by side, our system with the existing system.

4.3 System Overview

The System was designed and hosted on a website since December, 2018 with domain name www.servicesmails.com. Visitors started trooping at the same month and we started collecting the stored IP addresses categorically in April, 2019, then, the analysis as can be seen in Figure 4.1 shows the system interface, through which the genuine user will interact with the system. The interface demands only the Username/email and Password of the visitor. Genuine users will go straight forward and supply the needed requirement. A hidden field set to trap an intruder is set below the Password field, which is obscured using a JavaScript code. An intruder that crawls

through the code will fill in the hidden field as seen in Figure 4.2, which qualified it to be redirected to a honeyCAPTCHA as a bot.

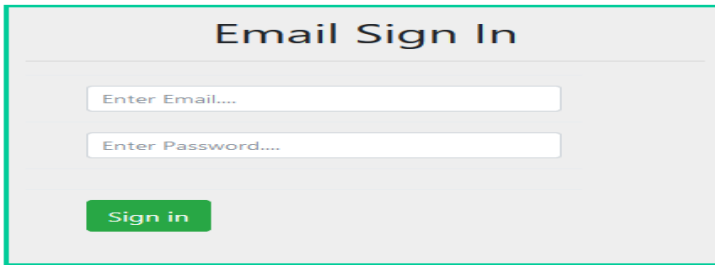


Figure 4.1: Login page of the system

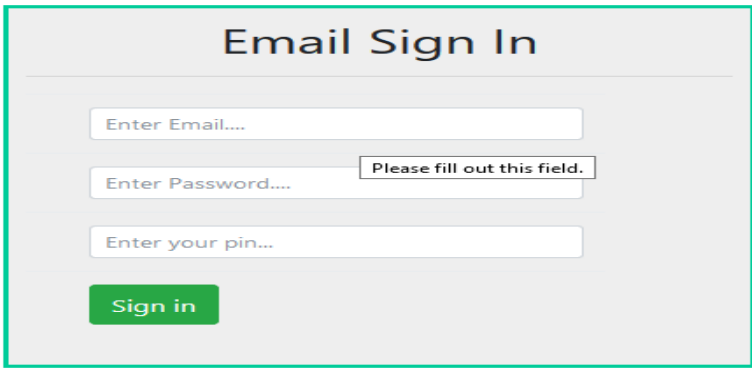


Figure 4.2: Login page of the system with the hidden field displayed

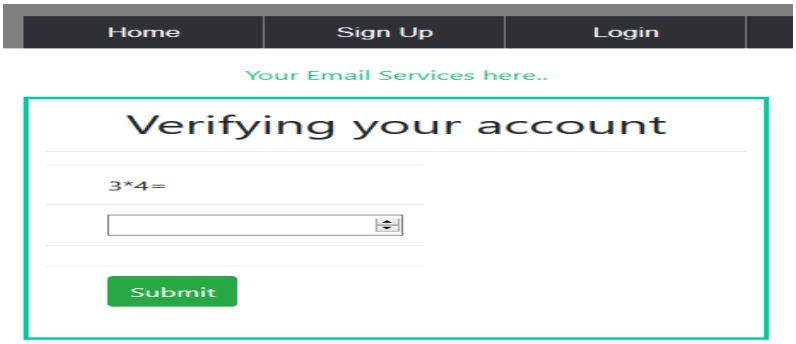


Figure 4.3: honeyCAPTCHA that receive the redirected bots

Figure 4.3 is a dummy CAPTCHA that receives the redirected bots as a lured security to the intruders from the actual system. The CAPTCHA will stand as a security gateway that will screen the bots and classify them as either intelligent or non-intelligent bots. If the bots successfully break the CAPTCHA, it is identified intelligent and will be redirected to a dummy page that looks like a real page of the system as shown in Figure 4.4. Here it can perform all the normal activities in the site as genuine user, but it will have no impact to the system, while the probable compose mail form filling is timed to differentiate real bots and the misclassified humans. Where a visitor exceeds the assigned five seconds will be detected and automatically be called back to the gateway. This is to avoid victimization of humans.

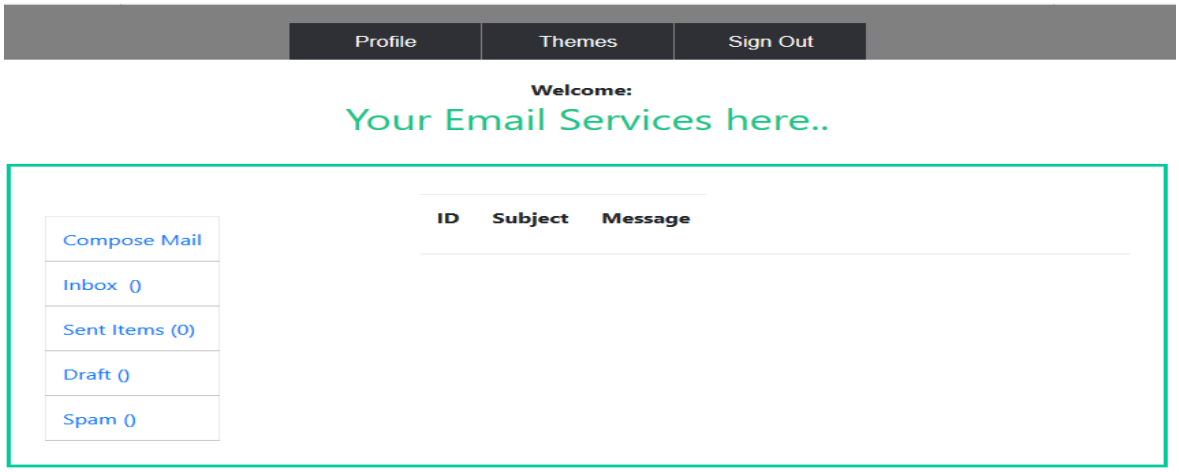


Fig 4.4 Dummy page that receive intelligent bots.

Figure 4.5 shows a snapshot of genuine users IP addresses, which have a general feature of a number combination. The Fig 4.5 gives an obvious identity of genuine users different from the automated programs that is also displayed in Figure 4.6, characterized by appearance of some suspicious letters with number combination from their hostnames. Those letters are like crawl, bots, spiders, static etc. These guides us in categorizing the misclassified IP address on four tables. The attempt bots were further categorized as Intelligent and the unintelligent bots, based

on their attempt toward CAPTCHA solution. Those bots that successfully break the CAPTCHA were grouped in a table as intelligent bots. Figure 4.7 shows the snapshot of the non-intelligent bots that failed to solve the CAPTCHA. Figure4.8 shows the snapshot of the intelligent CAPTCHA grouped, which were far less in the list in number due to the revert condition that select out those IP addresses that are not supposed to be there.

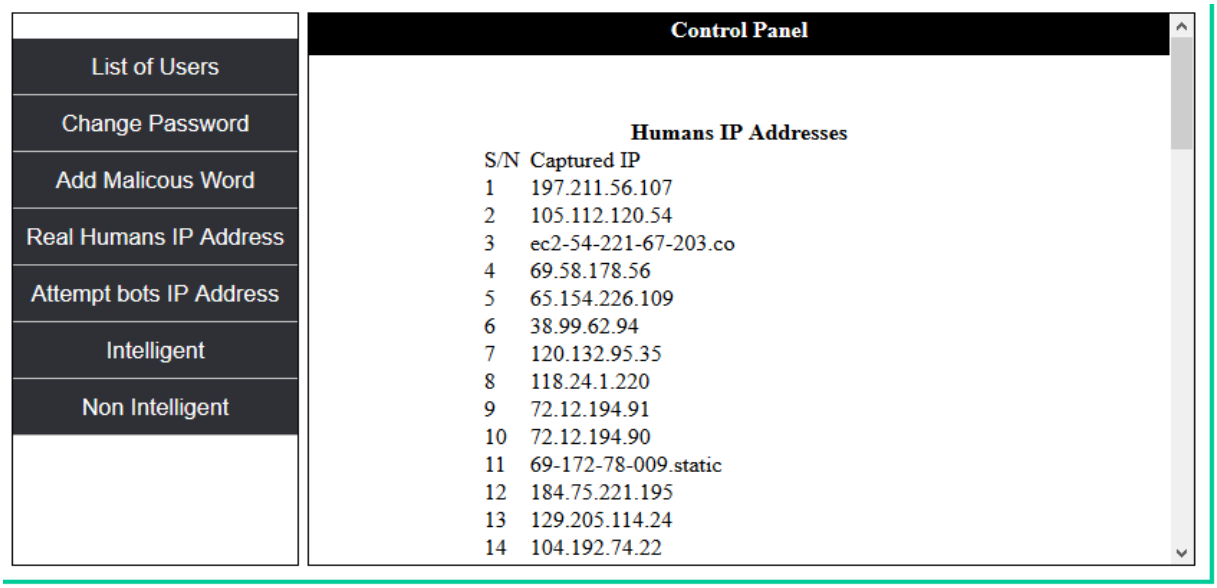


Figure 4.5: A Snapshot of a genuine human IP addresses

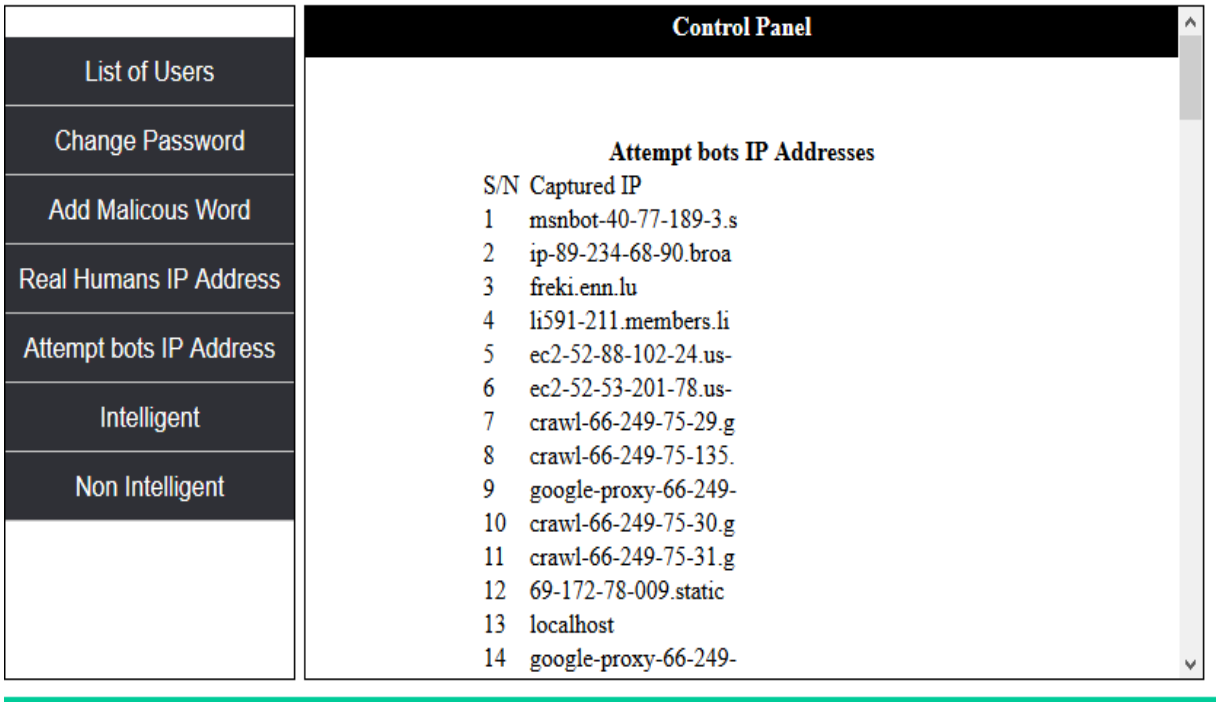


Figure 4.6: A Snapshot of an Attempt bots IP addresses

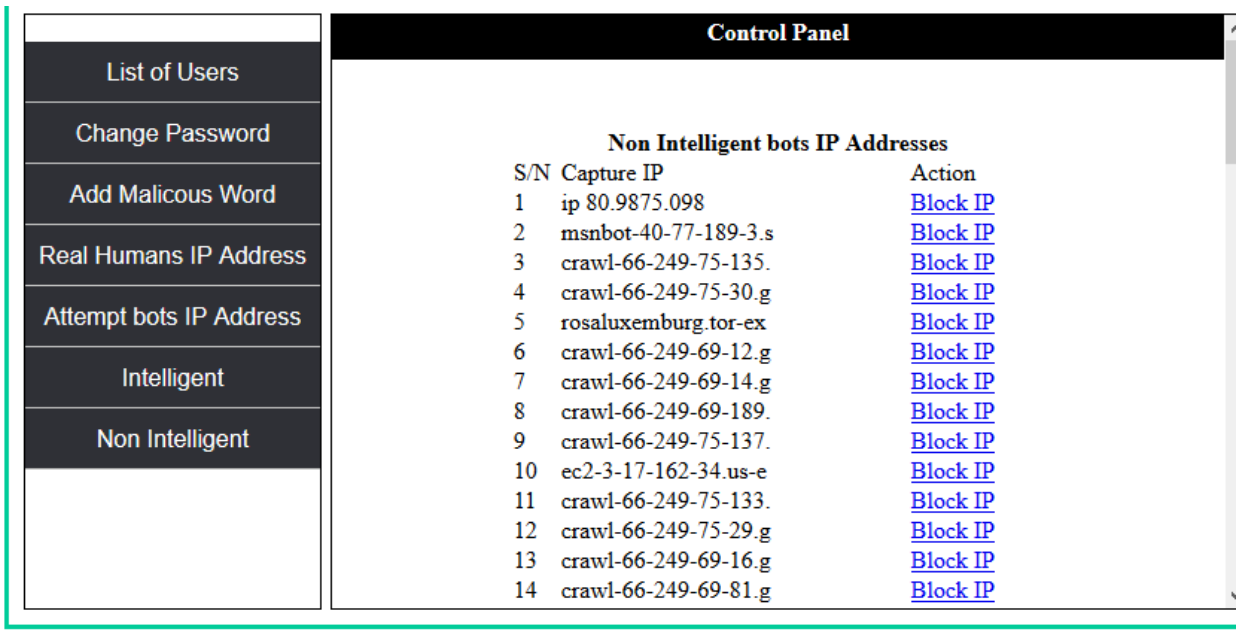


Figure 4.7: A Snapshot of a non-intelligent bots IP addresses

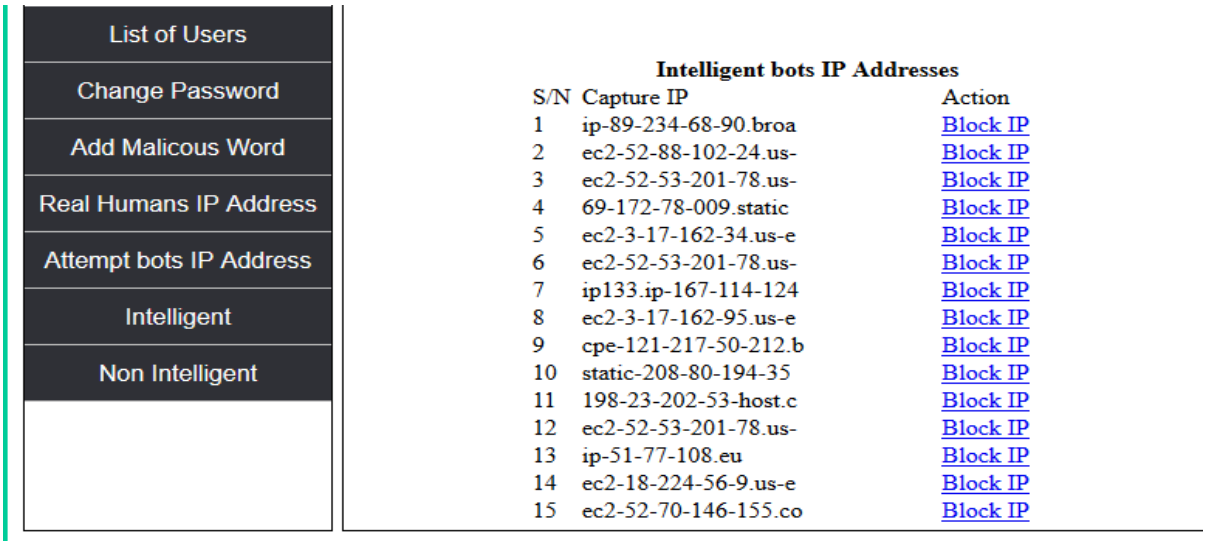


Figure 4.8: A Snapshot of an intelligent bots IP addresses

4.4 Evaluation

The performance of this system and that of Souley and Abubakar (2018) were evaluated using the metrics from Souley and Abubakar (2018) and a single unified, intuitive and appealing metric, grounded by information theory from the work of Gu *et al*., (2006). Both the robustness and the efficiency of the system were evaluated. Table 4.1 shows the analysis of the five-month IP addresses of all one hundred and ninety-two (192) visitors from December, 2018 to April 2019, which were categorized into Human, Attempt bots that comprises of the intelligent and non-intelligent as categorized by the honeyCAPTCHA system. Figure 4.9 shows the graphical representation of the visitors according to its representation in Table 4.1.

Table 4.1: The entire visitors of the proposed system categorized in month

Month	Total visitors	Human	Attempts bots	Intelligents	Non-intelligent
Dec 2018	**34**	17	17	4	13
Jan 2019	**39**	18	21	1	20
Feb 2019	**40**	16	24	5	19
Mar 2019	**38**	9	29	3	26
April 2019	**41**	16	25	6	19
	192	**76**	**116**	**19**	**97**

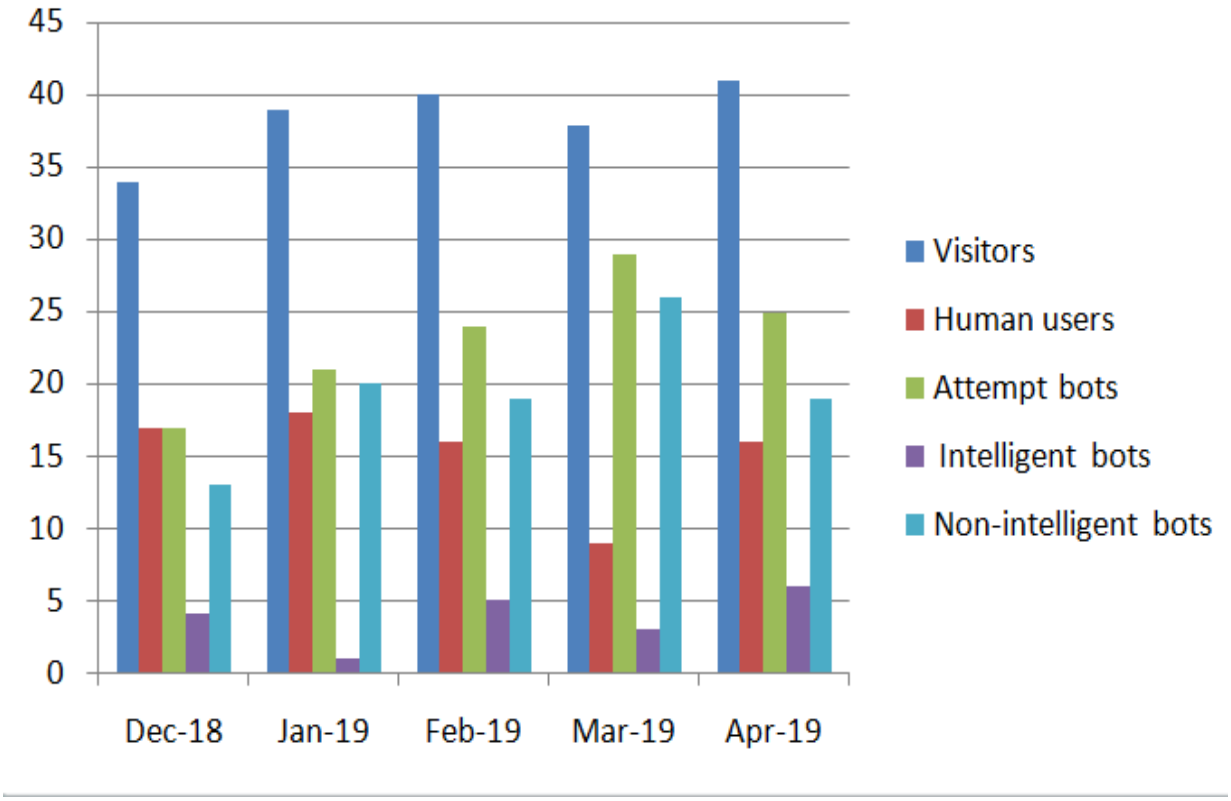


Figure 4.9: The Graphical representation of 192 visitors of the proposed system.

Analyses of the two systems were conducted in R using the caret and e1071 packages as discussed earlier. Generally, we consider using R due to the availability and richness of its classification and evaluation metrics (Zhuoheng, 2018). Figure 4.10 shows the snapshot of the analysis of the existing system from RStudio which brings out all the available metrics inbuilt in caret R package.

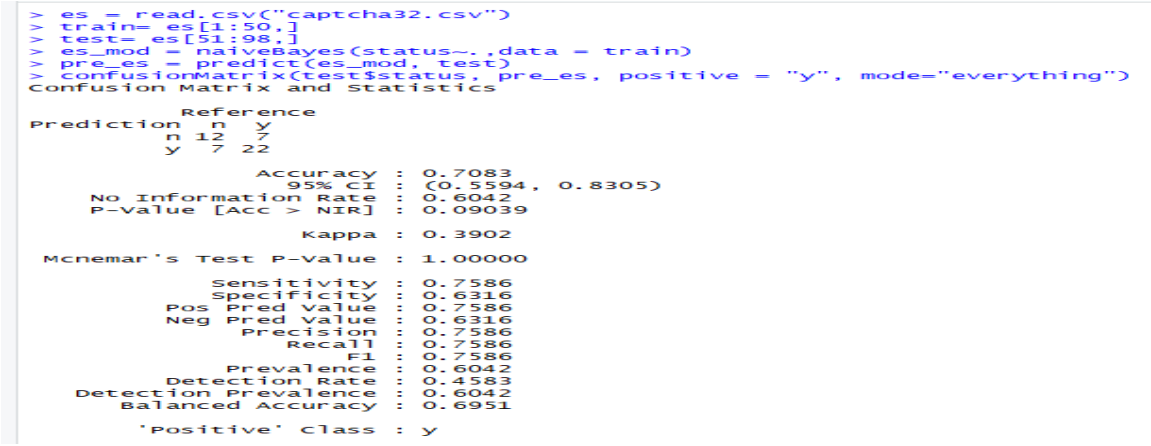


Figure 4.10: The Snapshot of the Analysis of the existing system performance in RStudio

Similar analysis was carried out on the proposed system in the same system which is shown in the snapshot of Figure 4.11.

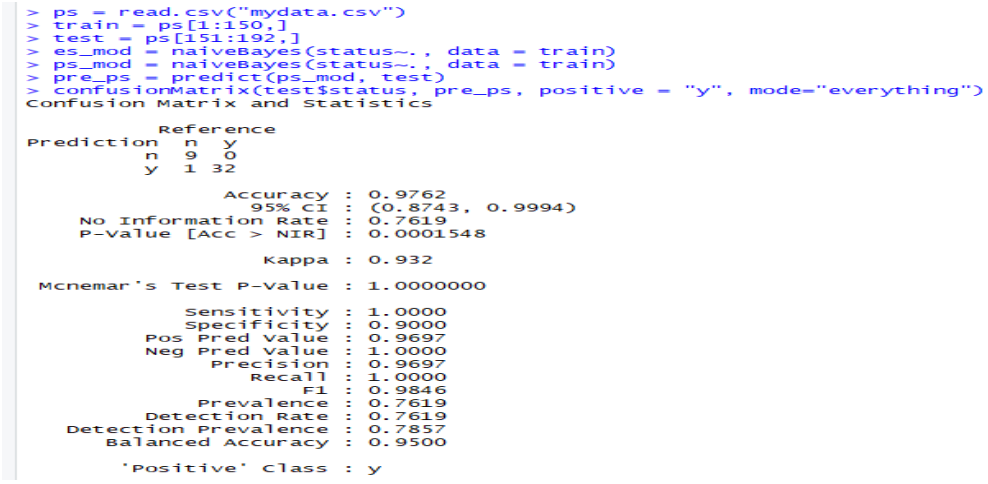


Figure 4.11: The snapshot of the analysis of the proposed system in RStudio

The proposed system and that of Souley and Abubakar (2018) robustness based on the Detection Rate, False positive Rate and True Positive Rate were compared in Table 4.2.

Table 4.2: Evaluating of robustness of the proposed system and that of Souley and Abubakar (2018).

SN	Evaluation Metrics	Existing System	Proposed System	Performance Differences
	Number of datasets	98	192	-
1.	Detection Rate	0.45	0.76	0.31
2.	False Positive Rate	0.36	0.10	0.26
3.	True Positive Rate	0.75	1.00	0.25

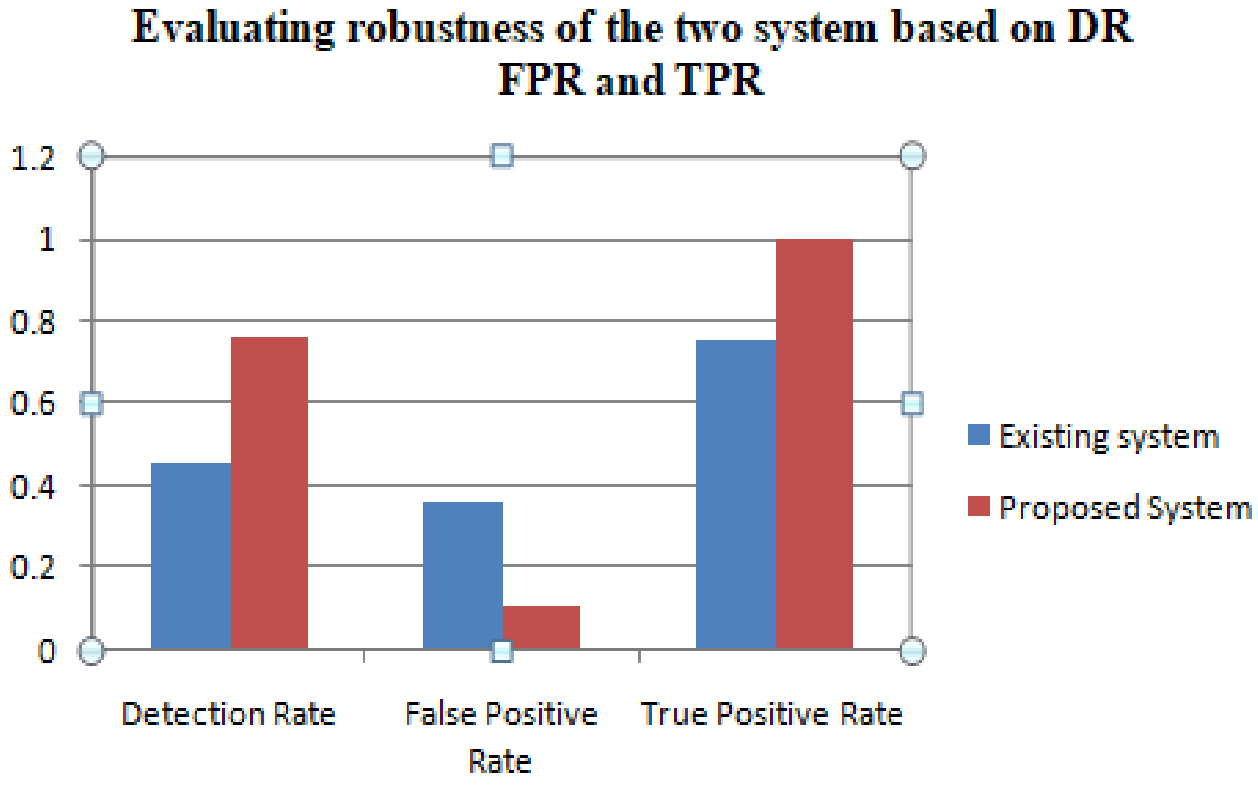


Figure 4.12: Graphical representation of the system performance w.r.t to DR, FPR and TPR

Table 4.3 describes the analysis of our proposed system with that of Souley and Abubakar (2018) based on other alternative measures that include the Precision, Recall and the F-measures, which were used in Souley and Abubakar (2018). The values of the metrics were all obtained from the snapshot in Figure 4.10 and Figure 4.11. They are graphically represented in Figure 4.13.

Table 4.3: Evaluate the robustness based on Precision, Recall, F-Measure and Accuracy of the two systems.

SN	Evaluation metrics	Existing System	Proposed System	Performance Difference
1.	Precision	0.75	0.96	0.21
2.	Recall	0.75	1.00	0.25
3.	F-Measure	0.75	0.98	0.23
4.	Accuracy	0.70	0.97	0.27

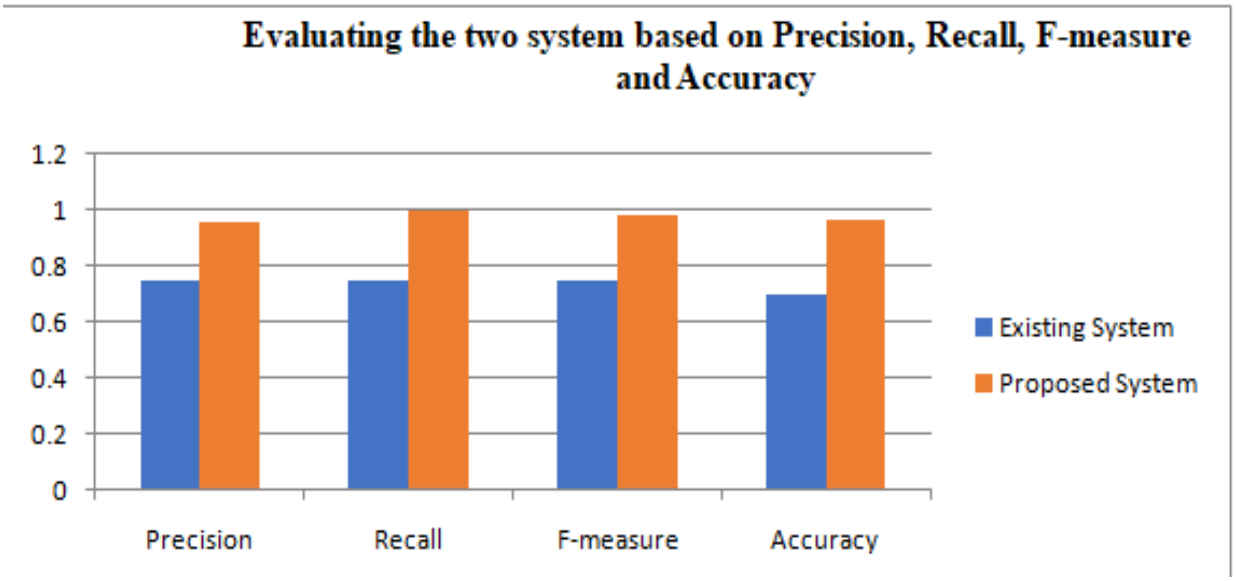


Figure 4.13: Graphical representation of the robustness based on Precision, Recall, F-measure and Accuracy

The System usability was measured using Bayesian Detection rate which was stated to be the most recommended metrics in the usability point of view (Gu *et al*., 2006). Table 4.3 shows the Usability of our proposed system visa a vis Souley and Abubakar (2018). The value is also obtained from the snapshots of the existing system analysis in Figure 4.10 and that of the proposed system analysis in Figure 4.11.

Table 4.4: The usability comparative analysis of the two systems

SN	Metrics	Existing System	Proposed System	Performance Difference
1	Bayesian Detection Rate (PPV)	0.76	0.97	0.21

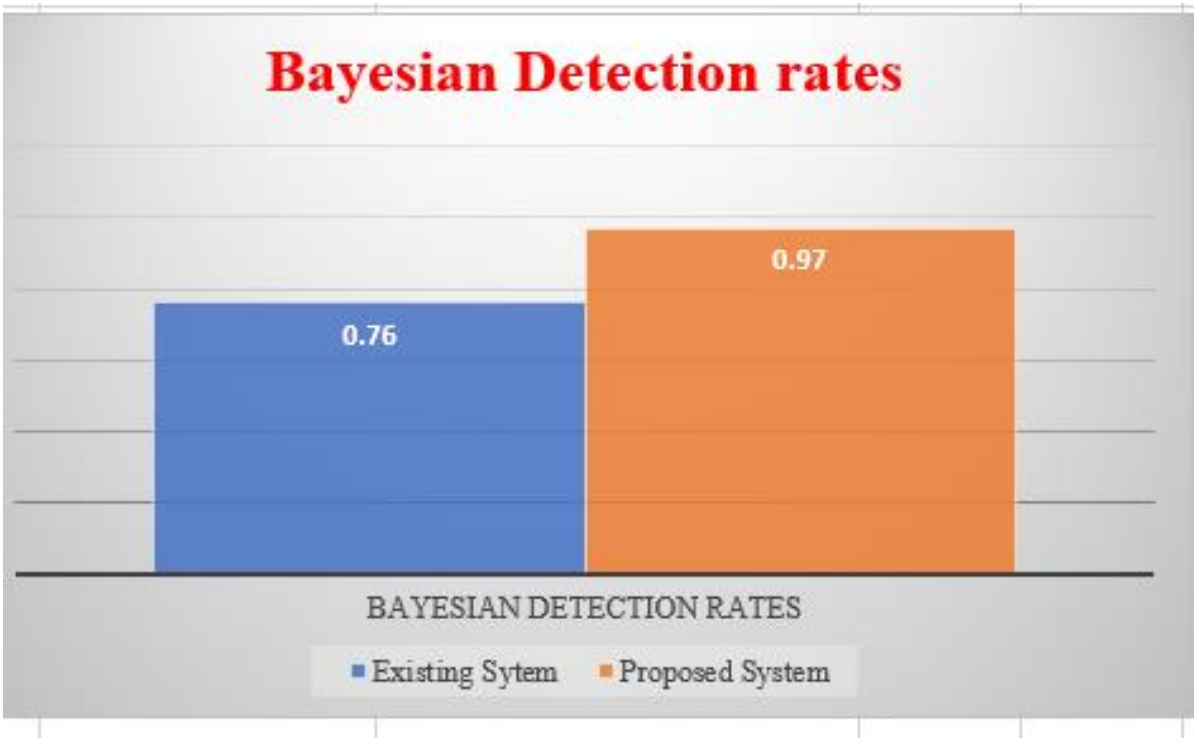


Figure 4.14: Graphical representations of the BDR of the two systems

4.4 Result Discussion

The metrics defined in section 2.8 show how standard is our proposed system measured, since it consists of many standardized metrics used in measuring the performance of IDS. Our system was measured and compared with Souley and Abubakar (2018) using these metrics: Detection Rate, False Positive Rate, True Positive Rate, Precision, Recall and F-measure to measure the robustness of the system, whereas the usability was measured using the Bayesian Detection Rate. All the above listed metrics have values ranging from 1 to 0 and performs better when the values go from 0 to 1 with the exception of false positive rate which takes the reverse. The proposed system has a Detection rate of 0.76 or 76% while that of Souley and Abubakar (2018) had Detection rate of 0.45 or 45%, false positive rate of 0.10 or 10% as against 0.36 or 36% of the existing system and True positive rate of 1.00 or 100% against 0.75 or 75% of the existing system. This means our system is 1.7 times better than that of Souley and Abubakar (2018) in detecting bots and has 10% less problems than the existing system with 36% issues in misclassifying human as bots. This is due to the possible JavaScript disability, though tackled by the sample response time detection used to create a possible fallback for normal visitors when they were classified as intruders. High scores for precision and recall showed that the classifier is returning accurate results (Scikitlearn documentation, 2018). From Table 4.3, both precision and recall for the existing system is 0.75 or 75% where as that of the proposed system are 0.96 or 96% and 1.00 or 100% respectively. This means the proposed system is 1.2 and 1.3 times the existing system respectively. This of course is anticipated due to the elimination of the security verification of the proposed system. The accuracy of classification of both the existing and the proposed system was also captured in Table 4.3, with 0.97 or 97% accuracy for the proposed system and 0.70 or 70% accuracy for the existing system. Here also, our proposed system is 1.39

more accurate than that of Souley and Abubakar (2018). This is due to the High rate of the positive prediction of our proposed system over that of the existing system.

The Usability of the system is treated based on the Bayesian Detection rate which is considered more sufficient in measuring IDS usability (Gu *et. al*., 2006). The proposed system has 0.97 or 97% BDR and that of Souley and Abubakar (2018) had 0.76 or 76%. It also means that the efficiency of the proposed system is 1.5 times that of Souley and Abubakar (2018). This will also not be unconnected with the free verification of the proposed system to users and ultimately IDS alarm is useful only if the IDS have high PPV (Gu, *et. al*., 2006). With this record, our proposed system out-performs that of Souley and Abubakar (2018), simply because the system eliminate any security verification in the gateway while employing the used of the decoy field inscribed by JavaScript, which is new in honeypot technology due to improvement by automated software to decode CSS and HTML and the provision of fallback opportunity for genuine humans that were misclassified and their respective IP addresses were deleted from the bots table.

We improved the combined techniques by hiding the CAPTCHA and using a decoy field as a redirector of any intrusions of the system to the CAPTCHA. Here, our system makes CAPTCHA only visible for bots while human will easily access the system without passing any security test. Previously, most of the hidden form field techniques were carried using CSS and HTML language. Due to the capability of intelligent bots, these fields were detected and bypassed (Narayan, 2015). With this set back, JavaScript became paramount in that regard. Our system improved on the existing system by using the JavaScript in masquerading detector field, response time to give a positive fallback opportunity for humans in the bots' corpus and to delete all IP addresses found to exceed the set time limit of five seconds. This development enables high Detection rate and Low False Positive rate. In Souley and Abubakar (2018), only intelligent bots

were targeted, neglecting other bots, which weakens the prediction of finding the capability of an intrusion. Ideally, if we are to conduct a forensic study on intrusion and intruders all bots should be put into consideration to maximize our prediction and further assumption. The positive fallback opportunity that was made available to misclassified humans and the user of the bio-inspired negative selection algorithm (NSA) reduces the false positive rate as well as increasing both the Detection rate and the system precision.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

Sequel to the lingering tradeoff between usability and the robustness of security techniques in IDPS state of the art, we contributed to this area of research by designing an enhanced Intrusion Detection Framework named honeyCAPTCHA, as a means to trick the intruders in a web-based application. The system was designed purposely as an alternative to CAPTCHA-BASED IDS that was designed by Souley and Abubakar (2018) to solve the perpetuated issues of web-based security. honeyCAPTCHA forms a gateway that seems easy for users to login while bots find it complicated due to the travail it undergoes in accessing the system. Detected intrusions were lured to a CAPTCHA challenge, which stands as a scapegoat for their possible attack. The decoy field obscured by JavaScript is the Detector, while the CAPTCHA challenge is the IPS that will separate the intelligent bots (i.e. intruders capable of breaking CAPTCHAs) and those that cannot. Response time and classical negative selection algorithm were used to safe the victimized humans from bots' treatment by providing an opportunity for them to retry their login. This entirely is used purposely to strengthen the security of our web-based application gateway and to enable an efficient interaction with genuine users in a honeypot way, by deceiving the intruders to collect their information which will be useful in upgrading our security techniques. The system combines IDS, IPS, response time, classical negative selection algorithm and honeypot technology to form a formidable framework for online applications. It serves as an alternative solution to those popular security gateways like CAPTCHA, IDS and other restricted honeypot skills used in protecting our online applications.

5.2 Conclusion

The used of the hidden Detector makes users free of any verification, at the same time redirecting intruders to a deceitful CAPTCHA. Solving the CAPTCHA qualified the bots to be identified as intelligent intruders which will be privilege to access a dummy page that resembles the real page. All possible activities performed in the real page will also be experienced in this dummy page under strict supervision using time setting in the compose mail form of the dummy page to distinguished confirmed bots and mistakenly victimized humans. This is to avoid most of the sophisticated attack by intruders, due to the development of artificial intelligence. In Artificial Intelligence world, it is believed that the Hackers have more skills than the information security personnel (Padderson, 2014), deceitful approach will be a welcome idea, since it will enable the attacker fill comfortable by breaking the security gate and discharging his illicit activities. While if the situation was a straight forward security check and regards to the preferences of the application to the hacker, the hacker may insist on discovering possible ways to compromise the security techniques used.

5. 3 Contribution to knowledge

The following are likely contributions of the work:

i. Introduction of decoy field technique to detect an automated program without human disturbance.

ii. Eliminating CAPTCHA entirely in human verification process to ease use of the application.

iii. Apart from intelligent bots that is retrieved from the existing system, non-intelligent and attempted bots will be retrieved in the proposed system.

iv. Providing a fallback to users that are mistakenly classified as bots and also tracing bots that skip the track from genuine login.

v. Making Farming method fruitless for Hack masters.

5.4 Recommendations for Future Work

The outcomes of this research have led to the following recommendations:

i. Further categorization of the intruders based on their activities on the dummy page will also improve the security check and boost the activities of the honeypot's techniques.

ii. The system makes emphasis on the bots' detection on just the honeypots side with less consideration in the aspect of real systems, probably to detect the possibility of a bots among the genuine users

iii. Using direct survey to enable user's interaction with the system will also measure some of the usability part like the learnability and memorability straight away.

REFERENCES

Abdolahnezhad, M. R., and Banirostam, T. (2016). Improved negative selection algorithm for email spam detection application. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, *5*(4), 956-960

Almgren, M., and Lindqvist, U. (2001, October). Application-integrated data collection for security monitoring. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 22-36). Springer, Berlin, Heidelberg.

Baykara, M., and Daş, R. (2015). A survey on potential applications of honeypot technology in intrusion detection systems. *International Journal of Computer Networks and Applications (IJCNA)*, *2*(5), 203-208.

Bushell, D. (2011, March 4). In search of the best CAPTCHA. Retrieved August, 20, 2019, from https://www.smashingmagazine.com/2011/03/in-search-of-the-perfect-captcha/

Chandan, M. P., Thacker, M. C., and Saxena, M. A Review Paper on Analysis of Decisive and Non-Intrusive Technique to Combat Form Spam.

Dempsey, A. (2014). ESCAPT: Easy Strategies for Computers to Avoid the Public Turing Test.Atkinson, L. (2000). *Core PHP programming: using PHP to build dynamic Web sites*: Pearson Education.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2008). Misc functions of the Department of Statistics (e1071), TU Wien. *R package*, *1*, 5-24.

Domantas, G. (2019), *Difference between HTML and HTML5*, Retrieved February, 4, 2019, from https://www.hostinger.in/tutorials/difference-between-html-and-html5

El Ahmad, A. S., Yan, J., and Tayara, M. (2011). *The robustness of Google CAPTCHA's*: Computing Science, Newcastle University.

Eslam M. H., Aida O. A., and Ahmed I. S. (2010). A hybrid intrusion prevention system (hips) for web database security. *International Journal of Engineering Science and Technology*. 2 (7), 2745-2762.

Faysel, M. A., and Haque, S. S. (2010). Towards cyber defense: research in intrusion detection and intrusion prevention systems. *IJCSNS International Journal of Computer Science and Network Security*, *10*(7), 316-325.

Foley, A. (2012). Biometric Alternatives to CAPTCHA: Exploring Accessible Interface Options. *Dublin Institute of Technology*.

Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994, May). Self-nonself discrimination in a computer. In *Proceedings of 1994 IEEE computer society symposium on research in security and privacy* (pp. 202-212). Ieee.

Gianvecchio, S., Xie, M., Wu, Z., and Wang, H. (2011). Humans and bots in internet chat: measurement, analysis, and automated classification. *IEEE/ACM Transactions On Networking*, *19*(5), 1557-1571.

Google no Captcha + INVISIBLE reCaptcha – First Experience Results Review (2019, Mar 10). Retrieved from https://tehnoblog.org/google-no-captcha-invisible-recaptcha-first-experience-results-review/

Gu, G., Fogla, P., Dagon, D., Lee, W., Skori, C. (2006) Measuring intrusion detection capability: An information-theoretic approach. In: Proc. of the 2006 *ACM Symposium on Information, computer and communications security*, pp. 90–101.

Gupta, M. (2015). Hybrid intrusion detection system: Technology and development. *International Journal of Computer Applications*, *115*(9), 5-8.

Gianvecchio, S., Xie, M., Wu, Z., and Wang, H. (2011). Humans and bots in internet chat: measurement, analysis, and automated classification. *IEEE/ACM Transactions On Networking*, *19*(5), 1557-1571.

Jinquan, Z., Xiaojie, L., Tao, L., Caiming, L., Lingxi, P., and Feixian, S. (2009). A self-adaptive negative selection algorithm used for anomaly detection. *Progress in natural Science*, *19*(2), 261-266.

Joseph, C. (2018, April, 17). Symantec Internet Security Threat Report 2018: TheTop Takeaways.[Blog post]. Retrieved from:https://thycotic.com/company//blog/2018/04/17/symantec-internet-security-threat-report-2018/. It is a snapshot of the page as it appeared on 26 Aug 2018 06:22:50 GMT.

Josh, D. (2019).Why CAPTCHAs have gotten so difficult. Demonstrating you're not a robot is getting harder and harder. [Blog Post]. Retrieved from https://www.theverge.com/2019/2/1/18205610/google-captcha-ai-robot-human-difficult-artificial-intelligence. Josh Dzieza@joshdzieza Feb 1, 2019, 11:00am EST

Jensen, S. H., Møller, A., and Thiemann, P. (2009, August). Type analysis for JavaScript. In *International Static Analysis Symposium* (pp. 238-255). Springer, Berlin, Heidelberg.

Koniaris, I., Papadimitriou, G., Nicopolitidis, P., and Obaidat, M. (2014, June). Honeypots deployment for the analysis and visualization of malware activity and malicious connections. In *2014 IEEE international conference on communications (ICC)* (pp. 1819-1824). IEEE.

Kumar, G. (2014). Evaluation metrics for intrusion detection systems-A study. *International Journal of Computer Science and Mobile Applications*, *2*(11), 11-17.

Kyrnin, J. (2018). 6 Modern Solutions to Protect Web Forms from Spam, LIFEWIRE. Retrieved from: *https://www.lifewire.com › How To › Web Design andDev › Basics.*

Li, D., Liu, S., and Zhang, H. (2016). A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection. *Engineering Applications of Artificial Intelligence*, *50*, 93-105.

Lie, H., W., (2005*). Cascading Style Sheets*. (PHD thesis). Faculty of Mathematics and Natural Sciences University of Oslo Norway.

Linora, J. A., andBarathy, M. N. (2014). Intrusion detection and prevention by using light weight virtualization in web applications. *International Journal of Computer Science and Mobile Computing*;3(3), 392-396.

Malav, S., Avinash, M. S., Satish, N. S., and Sandeep, S. C. (2015). Network security using IDS, IPS and honeypot. *Int. J. Recent Res. Math. Comput. Sci. Inf. Technol*, *2*(2), 27-30. .

*Max, K. (2019).*The caret Package, Retrieved from http://topepo.github.io/caret/index.html

Michael, C.(2017, May 1). On reCAPTCHA Dread. [Blog Post]. Retrieved from https://blogs.gnome.org/mcatanzaro/2017/05/01/on-recaptcha-dread/

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., and Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, *36*(1), 42-57.

Mohammad, M., andMohammadReza, K. (2014). CAPTCHA and its Alternatives: A Review. *SECURITY AND COMMUNICATION NETWORKS Security Comm. Networks*, 8: 2135–2156.

Moth, D. (2013, July 29). Six alternatives to using the dreaded Captcha images. [Blog Post]. Retrieved from https://econsultancy.com/six-alternatives-to-using-the-dreaded-captcha-images/

Narayan, P. (2015). How does Google's No CAPTCHA reCAPTCHA work? [Blog Post]

Parita, C., Chintan, T.,and Manish, S.(2016). A Review Paper on Analysis of Decisive and Non-Intrusive Technique to Combat Form Spam, *International Journal of Innovative Research in Computer and Communication Engineering*, 4(3).

Powell, B. M., Kalsy, E., Goswami, G., Vatsa, M., Singh, R., and Noore, A. (2017, May). Attack-resistant aiCAPTCHA using a negative selection artificial immune system. In *2017 IEEE Security and Privacy Workshops (SPW)* (pp. 41-46). IEEE

Public Safety Canada Departmental Plan, (2017), Common threats to be aware of. Retrieved August 12, 2017, from https://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/dprtmntl-pln-2017-118/index-en.aspx

Ramdane, C., and Salim C. (2017). Negative selection algorithm: recent improvements and its application in intrusion detection system. *Int. J. Comput. Acad. Res.(IJCAR)*, *6*(2), 20-30.

Resende PAA, Drummond AC.(2018) Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling. *Security and Privacy*; 1:4. Doi: https://doi.org/10.1002/spy2.36

Retrieved From :http://qnimate.com/how-does-googles-no-captcha-recaptcha-work

Rusland, N., F., Norfaradilla Wahid, N., Kasim, S. andHafit, H. (2017). Analysis of Naive Bayes Algorithm for Email Spam. *International Research and Innovation Summit*.

Kevat, S. M. (2017). Review on Honeypot Security. *International Research Journal of Engineering and Technology (IRJET)*, *4*(06), 1200-1203.

Stevens, I., D. (2016) Using machine learning to detect bots in World of Warcraft. *Transactions on networking*19 (5).

Suphannee, S., Jason, P., andResendes, D.(2019) I'm not a human: Breaking the Google reCAPTCHA. University, New York NY, USA

Souley, B., and Abubakar, H. (2018). A captcha–based intrusion detection model. *Int. J. Softw. Eng. Appl*, *9*(1), 29-40.

Sahasrabuddhe, A., Naikade, S., Ramaswamy, A., Sadliwala, B., and Futane, P. (2017). Survey on intrusion detection system using data mining techniques. *Int Res J Eng Technol*, *4*(5), 1780-4.

Tavallaee, M. (2011). *An adaptive hybrid intrusion detection system* (Doctoral dissertation,

University of New Brunswick, Faculty of Computer Science).

Vivekan and Rajbhar. (2018). Intrusion detectionand prevention using honeypot. *International Journal of Advanced Research in Computer Science*, 9(4),

What is R Programming Language? Introduction and Basics (2019, April 23). Retrieved from https://www.What is R Programming Language? Introduction and Basics .com/r-programming-introduction-basics.html

Yan, J. (2016). A simple generic attack on text captchas., ISBN 1-891562-41-X, retrieved from: http://dx.doi.org/10.14722/ndss.2016.23154.

Yesugade, K. D., Sanika, M. A., Sanika N. S., Charmi S. S., Malav, S. (2016). Infrastructure Security Using IDS, IPS and Honeypot. *International Engineering Research Journal* (IERJ), 2(3) Page 851-855.

Zakaria, W. Z. A., Kiah, M. L. M., Siew, H., Pooi, A., Bashir, U., Abbas, M., Ali, N. M. M. (2013). A review of dynamic and intelligent honeypots. *ScienceAsia 39S*, 1-5. DOI: 10.2306/scienceasia1513-1874.2013.39S.001.

Zhuoheng, X., Zhenghao, Y., Simon, J., Michael, R.,, Chris, R., Theerakorn, P., Matthew A.(2018), Caret Versus Scikit-learn A Comparison of Data Science Tools Lanham Purdue University Krannert School of Management, Retrieved From: http://matthewalanham.com/Students/2018_PURC_caretvsscikit.pdf

APPENDICES

Implementation of IP address blocking in php

```
if(isset($_GET['ipid'])){
                $ipid = $_GET['ipid'];
                include_once('connection/connect.php');
                $ipblock_query ="UPDATE instruder_ip SET ip_status= 1  WHERE id =".$ipid."";
                $query_run = mysqli_query($con,$ipblock_query);
                if($query_run){
                        function blockUsers($ipAddresses) {
                                $userOctets = explode('.', $_SERVER['REMOTE_ADDR']); // get the client's IP address and split
it by the period character
                                 $userOctetsCount = count($userOctets);  // Number of octets we found, should always be four
                                 $block = false; // boolean that says whether or not we should block this user
                                 foreach($ipAddresses as $ipAddress) { // iterate through the list of IP addresses
                                    $octets = explode('.', $ipAddress);
                                    if(count($octets) != $userOctetsCount) {
                                       continue;
                                    }
                                    for($i = 0; $i < $userOctetsCount; $i++) {
                                       if($userOctets[$i] == $octets[$i] || $octets[$i] == '*') {
                                          continue;
                                       } else {
                                          break;
                                       }
                                    }

            if($i == $userOctetsCount) { // if we looked at every single octet and there is a match, we should block the user
                                        $block = true;
                                        break;
                                    }
                                 }
                                 return $block;
                              }
                        echo'<script> alert("IP Blocked successfylly!") </script>';
                        echo'<script>window.open("captured_address.php","_self")</script>';
```

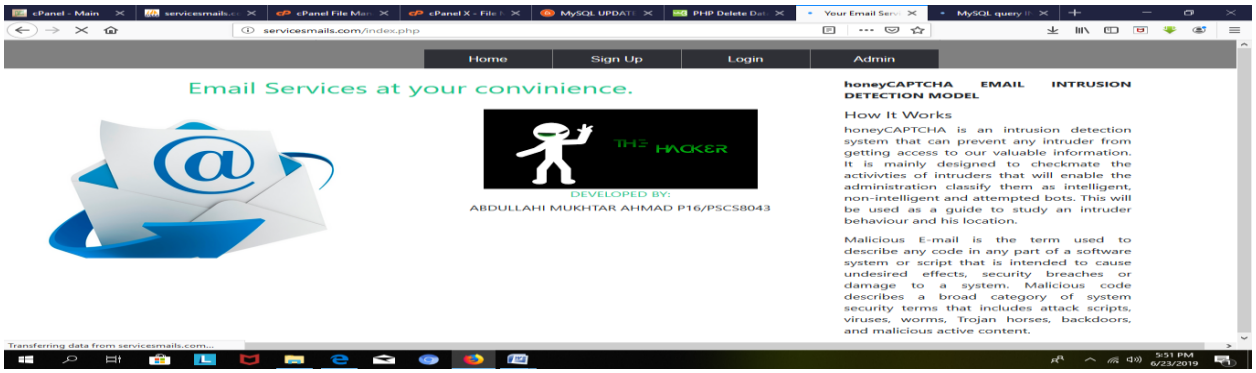
```
                }
        }else{
                header('Location:captured_address.php');
        }
?>
```

Implementing CAPTCHA design and verification

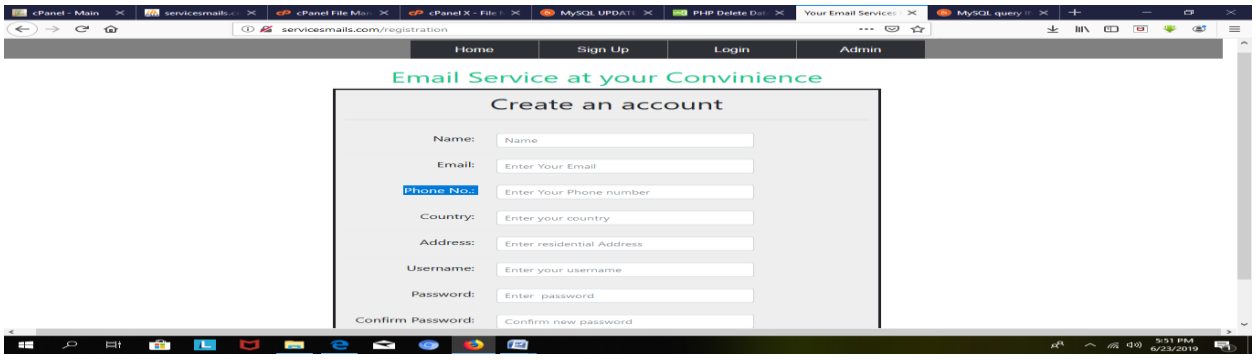
```
<?php
        session_start();
        $answer = $_SESSION["answer"];
        $user_answer = $_POST["answer"];
        include('mylib.php');
        $message ="";
        $servername = "mysql2001.my-virtual-panel.com";
        $username = "servi_user";
        $password = "G]o5pqCQk5rW";
        $dbname = "servi_honey_mail";
        $conn = mysqli_connect($servername, $username, $password,$dbname);
        //$dbselect = mysql_select_db($dbname);
/* Check connection
        if (!$conn){
   die("Connection failed: ".mysqli_connect_error());
                } *
if(isset($_POST['Submit'])){
                        $timestamp = time();
                        $ip = "";
        if ($answer == $user_answer){
                        if (!empty($_SERVER['HTTP_CLIENT_IP']))   //check ip from share internet
                                {
                                        $ip=$_SERVER['HTTP_CLIENT_IP'];
                                }
                        elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))   //to check ip is pass from proxy
                                {
                                        $ip=$_SERVER['HTTP_X_FORWARDED_FOR'];
```

```
                    }
                else
                    {
                        $ip=$_SERVER['REMOTE_ADDR'];
                    }
            $sql = "INSERT INTO intelligent_intr (ipaddr, date, iploc) VALUES ('$ip',NOW(), '$location')";
            $insert2 = mysqli_query ($conn,$sql) or die(mysqli_error($conn));
            if (!$insert2) {
                $error = "Unable to record new visitor\r\n";
            }
            header("location: user_net.php");
            }
        else{
                if (!empty($_SERVER['HTTP_CLIENT_IP']))   //check ip from share internet
                    {
                        $ip=$_SERVER['HTTP_CLIENT_IP'];
                    }
                elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))   //to check ip is pass from proxy
                    {
                        $ip=$_SERVER['HTTP_X_FORWARDED_FOR'];
                    }
                else
                    {
                        $ip=$_SERVER['REMOTE_ADDR'];
                    }
$txt = "An intruder with IP address ".$ip.". has accessed this application on ".date('Y-m-d');
$sendmail = mail("malamdogo2018@gmail.com", "Honey Port Mail Intrusion Report", $txt, "Hello Administrator");
$getIp_address = mysqli_query($conn,"INSERT INTO attemp_intruder_ip (ip_address) VALUES('".$ip."')");
            echo $ip;
            $sql = "INSERT INTO non_intelligent_intr (ip, ipdate) VALUES ('$ip',NOW())";
            $insert2 = mysqli_query ($conn,$sql) or die(mysql_error());
            if (!$insert2) {
                $error = "Unable to record new visitor\r\n";
            }
```

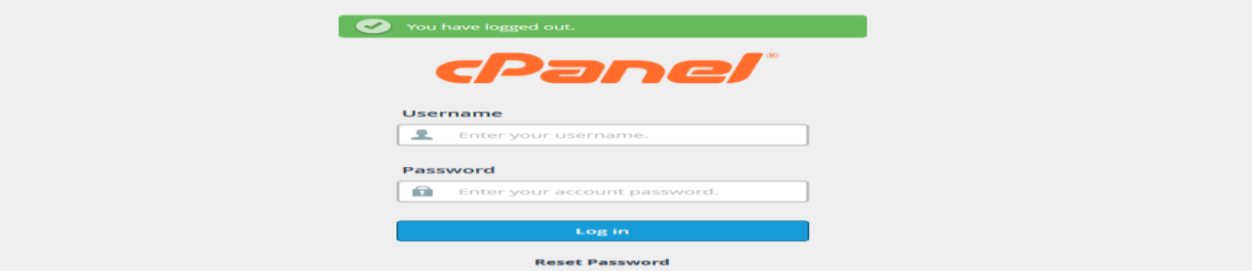
```
                header("location: index.php");
        }
}
?>
<?php
session_start();
$first_num = rand(1,10);
$second_num = rand(1, 10);
$operators = array("+","-","*");
$operator = rand(0,count($operators)-1);
$operator =$operators[$operator];
$answer = 0;
switch($operator){
        case "+":
                $answer = $first_num + $second_num;
                break;
        case "-":
                $answer = $first_num - $second_num;
                break;
        case "*":
                $answer = $first_num * $second_num;
                break;
}
$_SESSION["answer"] = $answer;
?>
```



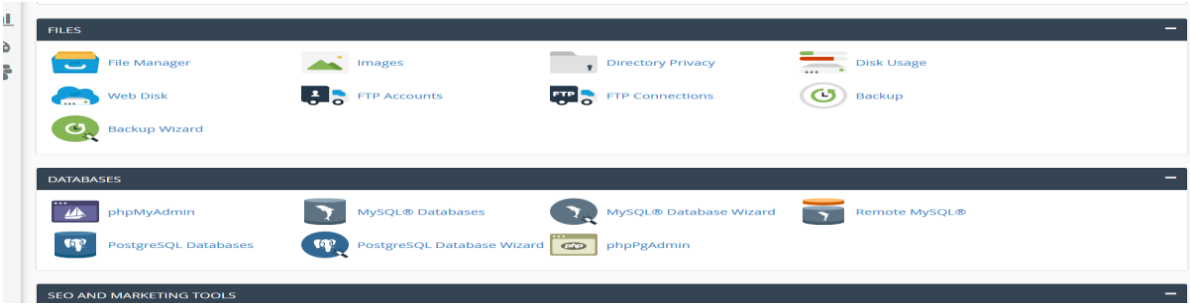
Home page of the Proposed system



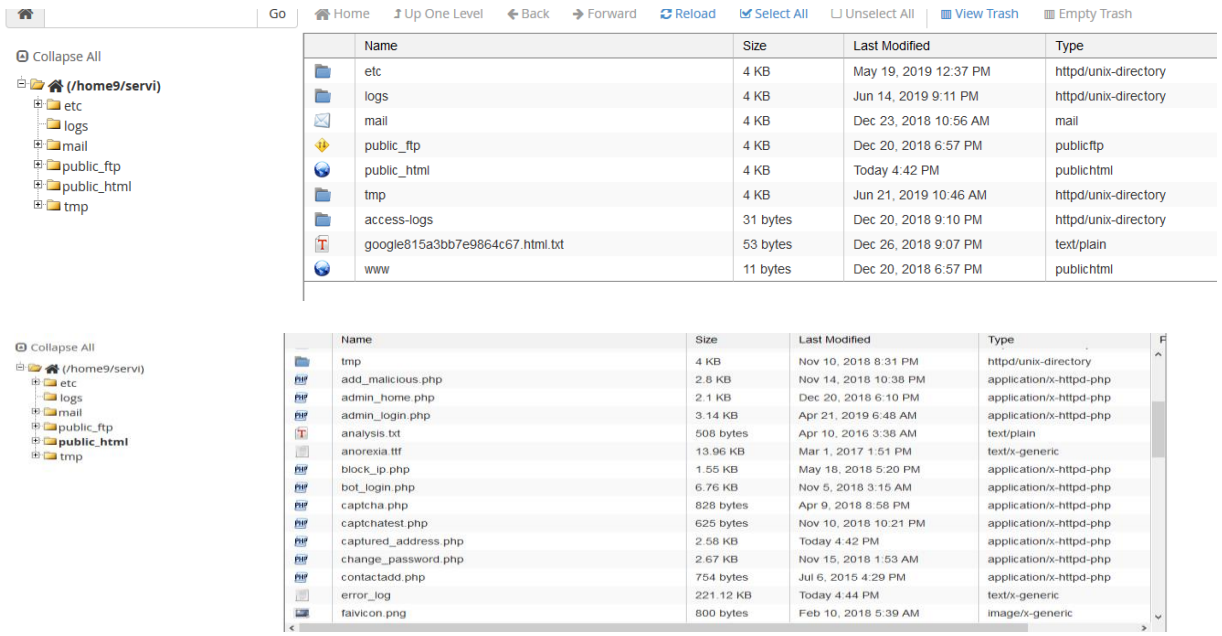
User Registration page of the proposed system



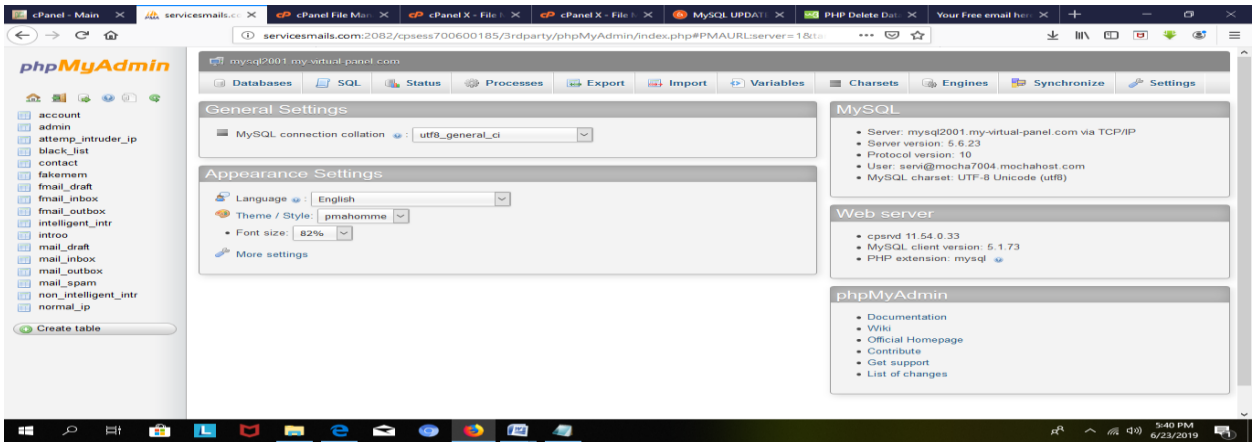
Control panel of the Proposed system Domain



Domain page of the proposed system

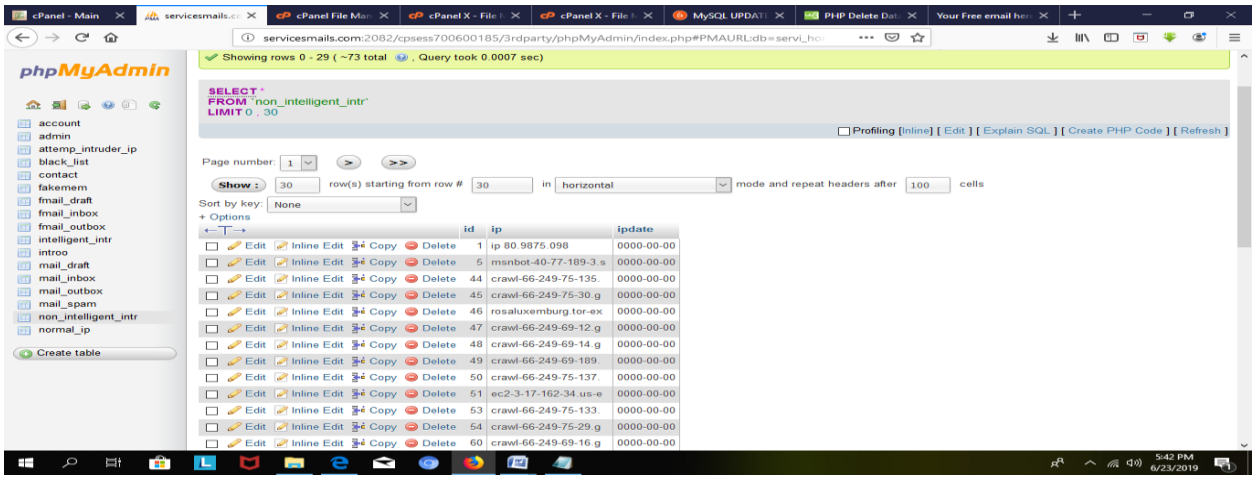


Pages Folder in the Domain Page

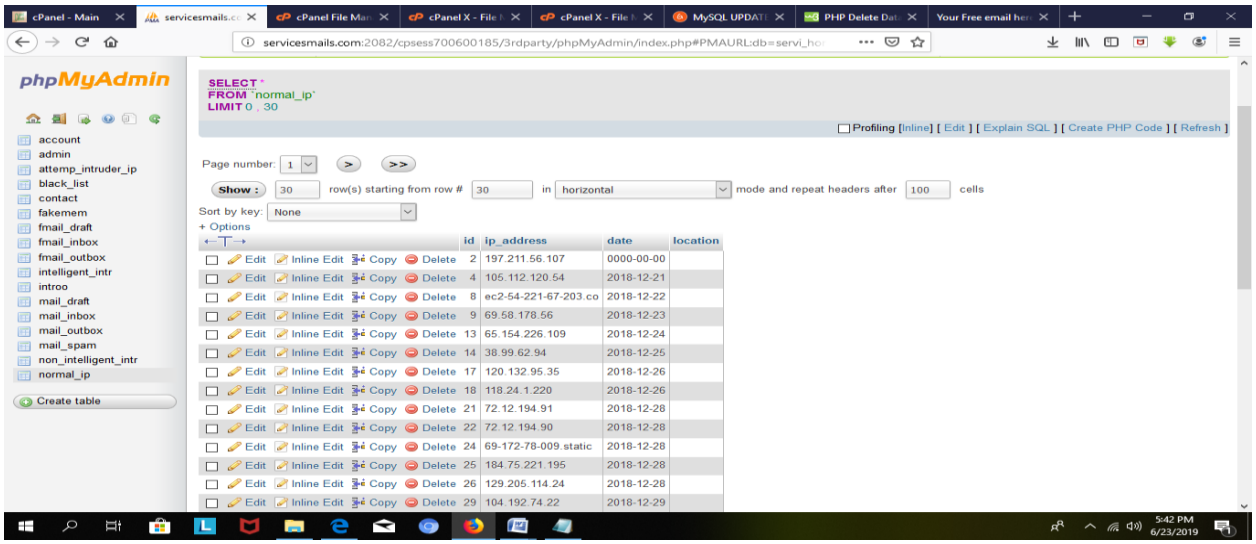


Backend Database of the proposed system

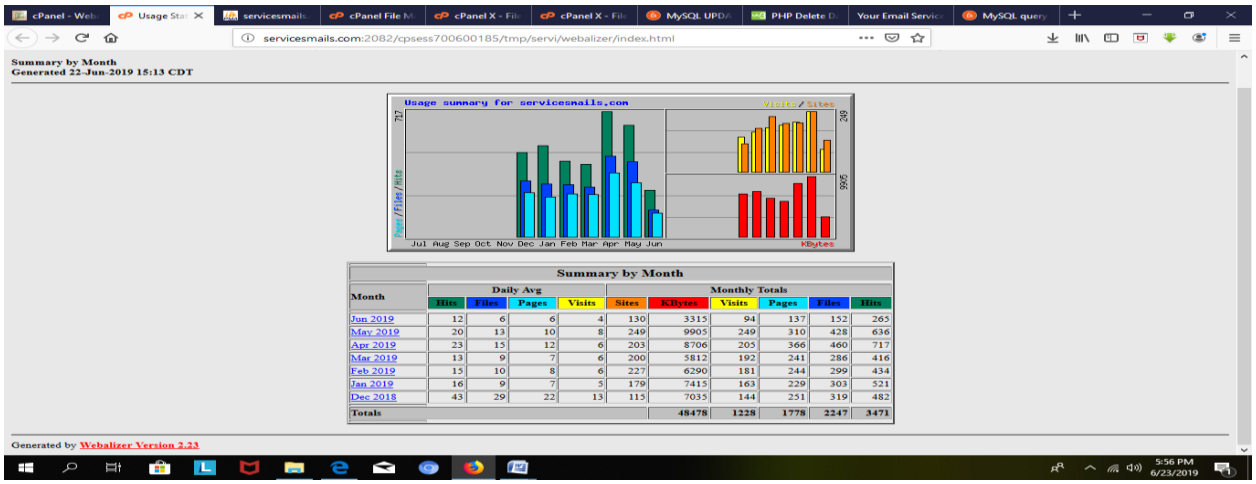
Captured Visitors in the Database



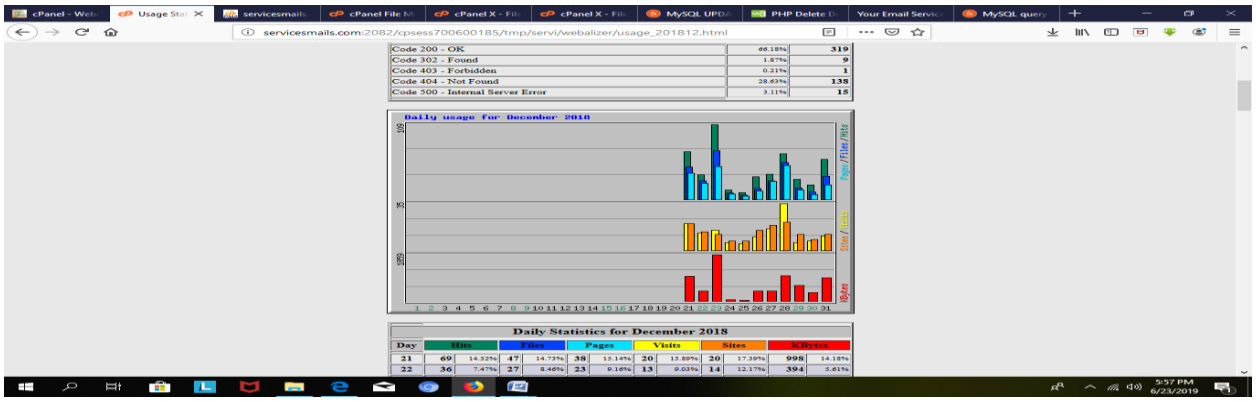
Nonintelligent bots Captured in the Database



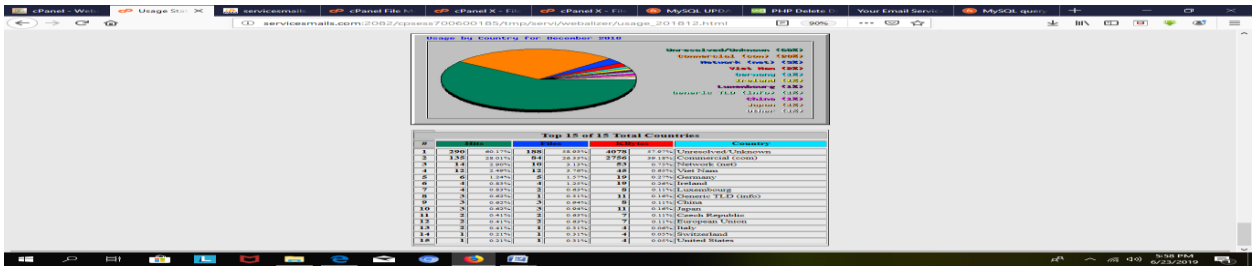
Genuine Human in the Database



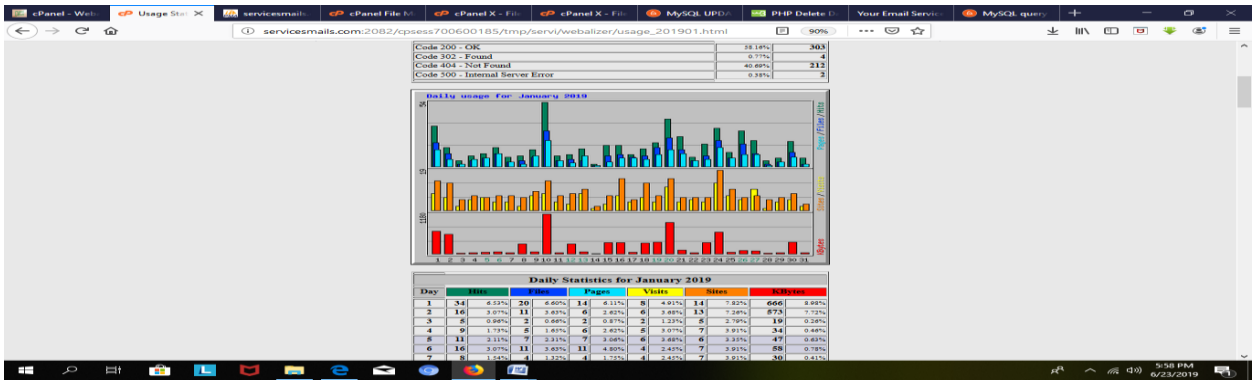
Statistical Analysis of the Visitors from December to June



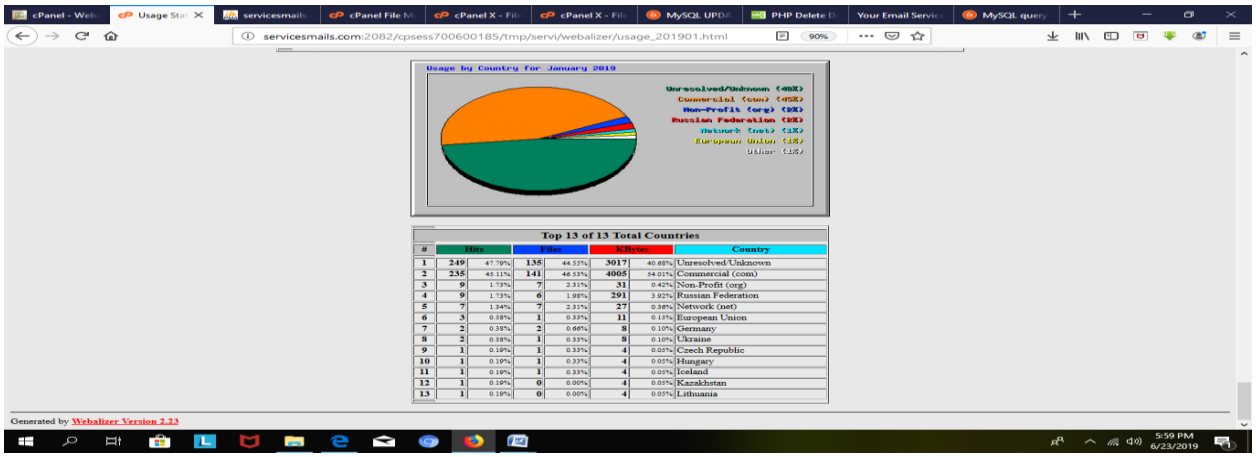
Statistical Analysis of December Visitors



Pie Chart representation of December's Visitors by Country

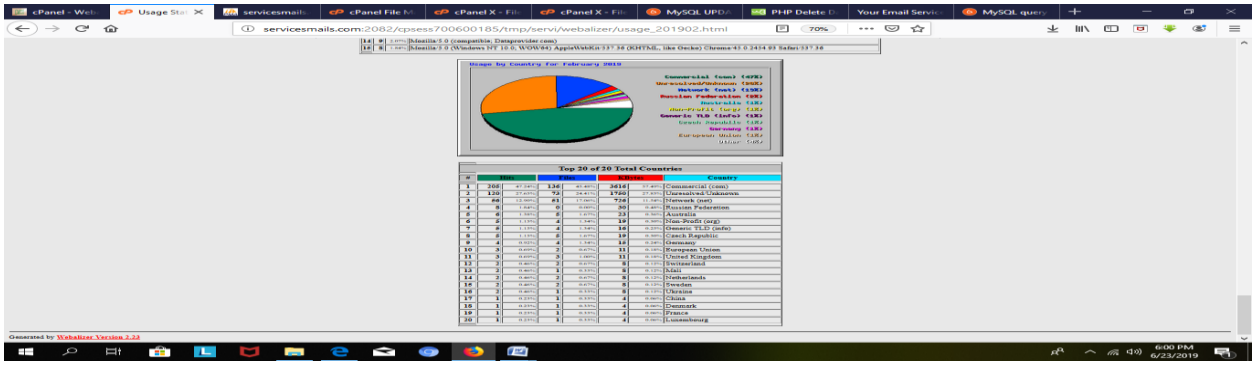


Daily visitor's representation of January



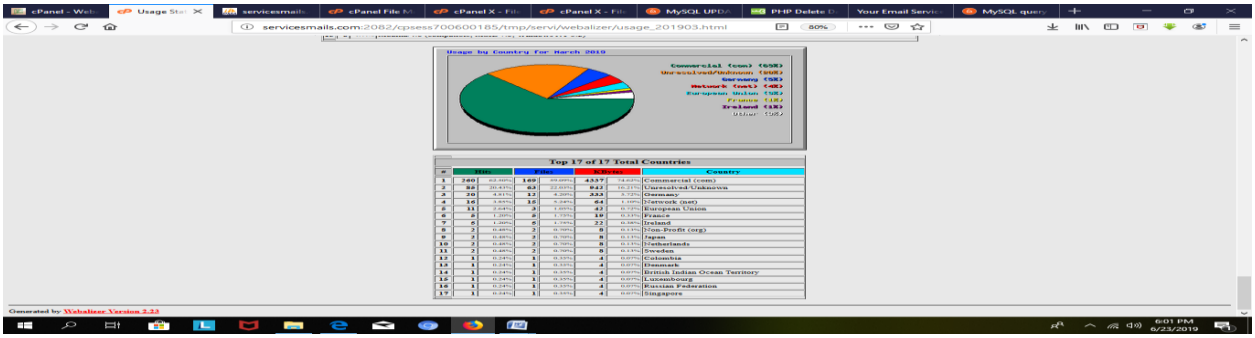
Pie Chart representation of January Visitors by Country

Daily visitor's representation of February



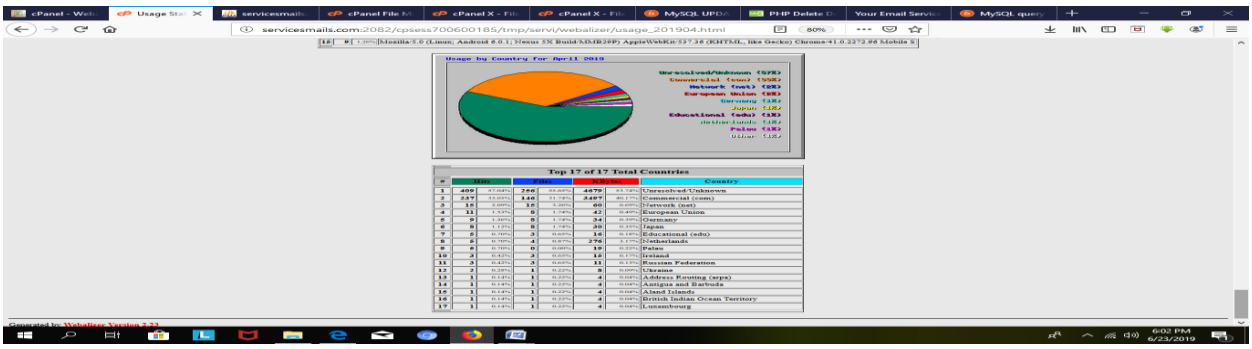
Pie Chart representation of February Visitors by Country

Daily visitor's representation of March



Pie Chart representation of March Visitors by Country

Daily visitor's representation of April



Pie Chart representation of April Visitors by Country