Spring 2015

Cheating Detection in Online Examinations

Gaurav Kasliwal
San Jose State University

Cheating Detection in Online Examinations

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Gaurav Kasliwal

May 2015

The Designated Project Committee Approves the Project Titled

Cheating Detection in Online Examinations

by

Gaurav Kasliwal

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

May 2015

Dr. Mark Stamp	Department of Computer Science
Dr. Thomas Austin	Department of Computer Science
Fabio Di Troia	Università del Sannio

ABSTRACT

Cheating Detection in Online Examinations

by Gaurav Kasliwal

In this research, we develop and analyze a tool that monitor student browsing activity during online examination. Our goal is to detect cheating in real time. In our design, a server capture packets using KISMET and detects cheating based on either a whitelist or blacklist of URLs. We provide implementation details and give experimental results, and we analyze various attack strategies. Finally, we show that the system is practical and lightweight in comparison to other available tools.

ACKNOWLEDGMENTS

I would like to take an opportunity to thank Dr. Mark Stamp, for his continuous guidance and support provided during the tenure of this project. Also, I would like to thank the committee members Dr. Thomas Austin and Fabio Di Troia for monitoring the progress of the project and their valuable time. Special thanks to Professor Kathleen O'Brien for her guidance.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Online exams [24, 32] are more proficient and simple to keep up when contrasted with paper based exams, for both teacher and students. Online exams are eco–agreeable as they are paperless. The Professor can gather the answers online and students get their outcomes in a brief time when contrasted with paper based exams. The Professor can distinguish if any student is cheating and make the online exam more robust.

These days, professors in the college have started giving online tests and exams in class. Students are permitted to utilize their laptops and the internet during exams. This makes the test accessible online to the students and submit their tests online too. As the internet is accessible during exams, student can misuse the internet for cheating [26] by utilizing online information. It is easy to find cheating if students are asking answers to their peers directly, but it will be difficult to find cheating if they are using online resources for finding answers. In this research, we build a tool that look for student browsing session during online examination. This tool captures the network traffic independent of the operating system running on students laptop. There is no need of installing any software to the student's or professor's laptops for the test, which makes this tool very easy to handle and use. In this research, we designed an online examination tool and executed distinctive strategies for cheating identification in online examination. We consider diverse strategies and measure viability of each. We also discuss different methods student used for cheating and how to overcome those using our tool.

This paper is organized in multiple chapters and an appendix. In Chapter 2, we provide background information of online exam and impact of cheating. Different methods available for cheating detection in online examination and how they work is described in this chapter. Chapter 3 focuses on previous work and different methods available for cheating detection in online exam. Different tools are available for detecting cheating in online exam and we are giving brief information of these tools in this chapter. Chapter 4 outlines the idea about project setup and different tools we are using. We are building online test monitoring tool to detect cheating in online exam. The project setup is very easy to build and maintain. In Chapter 5, we discussed about different methods by which we can restrict student from cheating during online exam and reduce impact of cheating for online exam. Next, we discuss three different methods applicable for cheating detection. Specifically, we consider Blacklist URL, SVM Machine Learning Approach and URL Frequency Analysis method. Our experimental results present in Chapter 6. We conducted an experiment of in–class online examination in some classes at San Jose State University. In Chapter 7 we are looking at enhancements that can be done in this project and consideration for future work. Chapter 8 contains our conclusion. In appendix A we have given screen–shots related to Kismet server setup and user interface. It also has screen–shots for online test system and shows different steps involved during test.

CHAPTER 2

Background

2.1 Online Examination

Online exams are becoming more and more popular these days. There are advantages and disadvantages of online examinations. The primary advantage is that, it is online and easy to use. Assessment of answers can be completely mechanized for different type of questions. It does not cost much when contrasted with paper based exam, as there is no paper work included like printing the question and answer paper. The expense of online exam is less when contrasted with conventional paper based exam. The disadvantage of the online examination is, student have access to the internet and it enables possibility of doing cheating during exam. In this project we are planning to replace traditional paper based exam system by online internet based exam system. In real life, we can see many example of online exam. Numerous specialized organizations effectively began online exams for screening the candidates. In not so distant future we will see online exam framework to be executed everywhere in school and college level.

2.1.1 Need of Online Exam

Examinations are used since long time to check the potential of students and their learning capacity. These days students need to submit assignments and tasks given by professor and they are checked by grader or professor himself. Numerous students give exams during end of the semester and professor needs to give result back in a week or so. This tool will give provision for making, conducting and evaluating examinations in very less time. The Internet open doors for making

examinations both more reliable and less expensive than they are as of now.

2.2 Cheating

There are many cheating strategies [20] students come up during examination. A good testing setup is beneficial to all students. A disciplined and good testing standard will give equal chance to all the students and makes sure the integrity of the examination system. Our goal is to minimize cheating during online exam and detect [15] different methods of cheating. The quality of education depends upon secure examination without any cheating.

2.3 Machine Learning

Machine learning is a field that investigates the development and investigation of calculations that can gain from preparing information. These machine learning calculations work by building a model from preparing information and utilizing that model to settle on forecasts or choices. Machine learning uses computational measurements for forecast making of test data. Machine learning is a sub–field of software engineering research in the field of artificial intelligence. There are two primary classes of machine learning calculation as supervised and unsupervised learning. They vary just in the causal structure of the model.

In Supervised Learning technique, we are given sample inputs and their outputs, and the objective is to take in a general decision that maps inputs to outputs. The model characterizes the arrangement of perceptions, called inputs and has another arrangement of perceptions, called outputs. At the end of the day, the inputs are thought to be toward the starting and yields toward the end of the causal chain. The models can incorporate interceding variables in the middle of

inputs and yields, e.g. spam filtering. In Unsupervised Learning technique, no marks are given to the learning calculation, abandoning it all alone to discover structure in its data. In Unsupervised learning, all the perceptions are thought to be brought on by dormant variables, that is, perceptions are thought to be toward the end of the causal chain. Models for administered learning leave the likelihood for inputs undefined. This model is not required as long as inputs are accessible, however in the event that a percentage of the information qualities are missing, it is unrealistic to take decision about the outputs. e.g. data mining clustering algorithms.

2.3.1 Supervised Learning Method

Supervised learning [30] is the machine learning technique for deriving a function from labeled training data. The training data comprise of an arrangement of training illustrations. In supervised learning, every sample is a couple, comprising of an information article called as vector and a desired output value. A supervised learning calculation breaks down the training dataset and produces a model, which can be utilized for mapping new dataset. An ideal situation will take into account the calculation to accurately focus the class names for concealed occasions.

2.3.2 Support Vector Machine

In machine learning, support vector machines are Supervised learning models with related learning calculations that examine information and perceive designs. It is utilized for grouping and regression investigation. Given training examples, every stamped as fitting in with one of two classifications. SVM training calculation fabricates a model that classify or assigns test information into one class

or the other, making it a non–probabilistic binary linear classifier. SVM model is a representation of samples as focuses in space, mapped so that the illustrations of the different classes are partitioned by clear gap that is as wide as possible. New cases are then mapped into that same space and anticipated to have a place with a classification in light of which side of the gap they fall on. In addition to performing linear classification, SVM can efficiently perform a non-linear classification using what is called the Kernel Trick, implicitly mapping their inputs into high-dimensional feature spaces.

2.3.2.1 Edit distance

Edit distance [34] is method to find out how similar two input strings are to one another by calculating minimum number of operations required to transform from one input string to another string. Edit distance [25] values are helpful in natural language processing for finding and correcting the spelling errors by comparing values in dictionary that have a low distance to the word input. Assume, there are two strings a and b. The edit distance $d(a, b)$ is minimal series of edit operations that converts a into b. There are three ways we can edit [25] the string and change it another string as follows:

1. Insertion of a single symbol

   If $a = uv$, then inserting the symbol x produces uxv. This can also be denoted $\epsilon \rightarrow x$, using ϵ to denote the empty string.

2. Removal of a single symbol

   It changes uxv to uv $(x \rightarrow \epsilon)$.

3. Replacement of a single symbol x for a symbol $y \neq x$ changes uxv to uyv

$(x \rightarrow y)$.

The Edit Distance algorithm is defined in Algorithm 1.

Algorithm 1 Calculate Levenstein Edit Distance

```
1: S1 = Input string1
2: S2 = Input string2
3: length1 = Length of string1
4: length2 = Length of string2
5: T = Two dimensional matrix of size length1+1 * length2+1
6: editDistance = Edit Distance between two input strings
7: for i = 1 to length2 do
8:    T[i][0] = i
9: end for
10: for i = 1 to length1 do
11:    T[0][i] = i
12: end for
13: for i = 1 to length2 do
14:    for j = 1 to length1 do
15:       if S1[j − 1] == S2[i − 1] then
16:          D = 0
17:       else
18:          D = 1
19:       end if
20:       T[i][j] = min(T[i − 1][j − 1] + D, T[i − 1][j] + 1, T[i][j − 1] + 1)
21:    end for
22: end for
23: editDistance = T[length1][length2]
24: return editDistance
```

Edit distance with non–negative cost fulfills the sayings of a metric, giving rise to a metric space of strings, with few conditions given as follows. Every edit operation has positive expense. For each operation, there is a reverse operation with equivalent expense. Considering these properties, the metric axioms are fulfilled as follows:

$d(a, a) = 0$, if two strings are equal, since each string can be insignificantly changed to itself using exactly zero operations.

$d(a, b) > 0$ when $a \neq b$, as this would require no less than one operation at non-zero expense.

$d(a, b) = d(b, a)$ by fairness of the expense of every operation and its opposite.

CHAPTER 3

Previous Work

Generous work has been done for creating strategies for cheating detection in online examination. This chapter gives an outline of the past work.

3.1 Useful Tools

Distinctive tools are accessible for cheating detection in online examination. Information of diverse tools accessible is portrayed in this section.

3.1.1 Computer Based Test

Computer based test [8] is a product by which you can give a test. The test is having a settled example and disconnected from the net i.e. no internet accessible during the test period. Student cannot open any tab apart from tab on which test is going on. It permits teacher or test taker to offer a more steady test conveyance, speedier scoring and reporting and secure environment. Testing focus can utilize desktop PCs or laptops, least necessity fulfilled for test setup. Computer based testing is turning into a general standard for giving the test. It eliminates the need of paper [37] and makes it advanced.

3.1.2 Respondus Lock Down Browser

Respondus [28] is a custom program that secures the testing environment inside test environment. At the point when students are utilizing this program, they are not able to print, duplicate, go to another URL, or access different applications. At the point when test is begun, students are locked until they complete the

test. Respondus is programming by which you can make online evaluations. Online assessments can be made right inside course, however Respondus can serve as our inquiry database, conveying inquiries to a few online courses. Secure Browser and Respondus Monitor [28] are changing instruction by giving organizations more adaptability in offering online exams.

3.2 Network Traffic Analysis

A packet analyzer is an application or system that can intercept and log traffic passing over a network. As information streams over the system, the sniffer catches every packet and unravels the packet information, demonstrating the estimations of different fields in the packet. There are tools like Kismet [13] and Wireshark [35], which comes truly helpful for catching packets inside system. Packet capture is the procedure of capturing and logging activity.

3.3 Online Application Monitoring Tool

In this project [2], Sathya built up a tool utilizing client–server structural planning. This tool used to show the student host name and the site visited by student during browsing activity. Professor had the capacity to see all the applications opened by student and logs for monitoring activity. Keeping in mind the end goal to run the tool effectively professor and student needs to fulfill few necessities. Professor and student needs to have windows OS running on their laptop or PC. Windows firewall and anti virus software needs to be turn off. In network places, turn on the alternative of file sharing inside the system. The work—group of computer should be MSHOME. Professor and students should have Java installed on their machine. Professor and students should run the software

given to them for effectively utilizing the test device.

3.3.1 Possible Attacks

Students have access to internet, so they can make use of internet for cheating [26] in diverse ways. Student can disconnect from the SSID given by the professor and join with an alternate SSID accessible, check for answers and interface with old SSID again. The device is getting the applications or sites which are running from the task manager. In the event that the student changes name of the application, then task manager will likewise have the same name that was given by the student. For instance, a student have changed the name of program "Mozilla" or "Chrome" to some other name, for example, "Notepad" or "Paint", then the task manager will also have the updated name for that task.

3.3.2 Limitations

This tools meets expectations for the Microsoft windows operating system and not helpful for all other operating systems, for example, Linux and Mac OS. Student needs to install some software before taking the test, which will consume a lot of time for setup.

3.4 Online Test Monitoring

In this project [14], Sumit composed and executed a test tool that professors can use to monitor online movement of student during in class online examination [32]. The objective of this project [14] was to assemble an online test monitoring tool in which, student can take exams using their laptops. Students were permitted to get to just a couple of sites that are whitelist. Sites, other than

whitelist destinations will not be permitted. The monitoring tool should recognize if any student tries to cheat by opening any site which is not on the whitelist. At present, student can cheat by opening blacklisted websites [1], which are not permitted, or they can associate with remote system to open those sites. Regardless, they would be opening a limited site on their laptop.

This tool utilizes a wireless sniffer [13] to capture and classify packet. If student endeavors to get to a site that is not permitted (one from blacklist), the professor will be informed by means of an Android application or through Internet. Distinguishing a student who is cheating is challenging since numerous applications send parcels without client intercession. This project gave exploratory results from reasonable test situations to represent the achievement of proposed methodology.

CHAPTER 4

Project Setup

In this project, we are building online test monitoring tool to detect cheating in online exam. Project setup is easy to build and maintain. The project setup details are described in following sections.

4.1 Initial Setup

Initial setup need 2 laptops. One laptop should have Ubuntu operating system running on it for Kismet server. We will need a WiFi router [31] with following specifications: TP–LINK TL–WR841N Wireless N300 Home Router, 300 Mpbs, IP QoS, WPS Button, Two 5 dBi antennas to increase the wireless robustness and stability. Kismet server [13] (packet capture server) is running on one laptop. Test is hosted on test server on another laptop. WiFi router allow students to connect to test. Test server and packet capture server are connected to WiFi router by LAN cable. Figure 1, shows the basic architecture of the test system [24] we have implemented and different component it is using.

4.2 Important Steps

Students will connect to WiFi provided to them (SSID and secret key is given).Student can begin exam by connecting to given WiFi. When they are joined with WiFi, they have to enroll for the exam. Student will enlist with their details (student name, student id and email address etc). Our tool will catch IP address and MAC address of the Student portable PC and store them into database by mapping it to student id (primary key). Now student will begin taking exam and

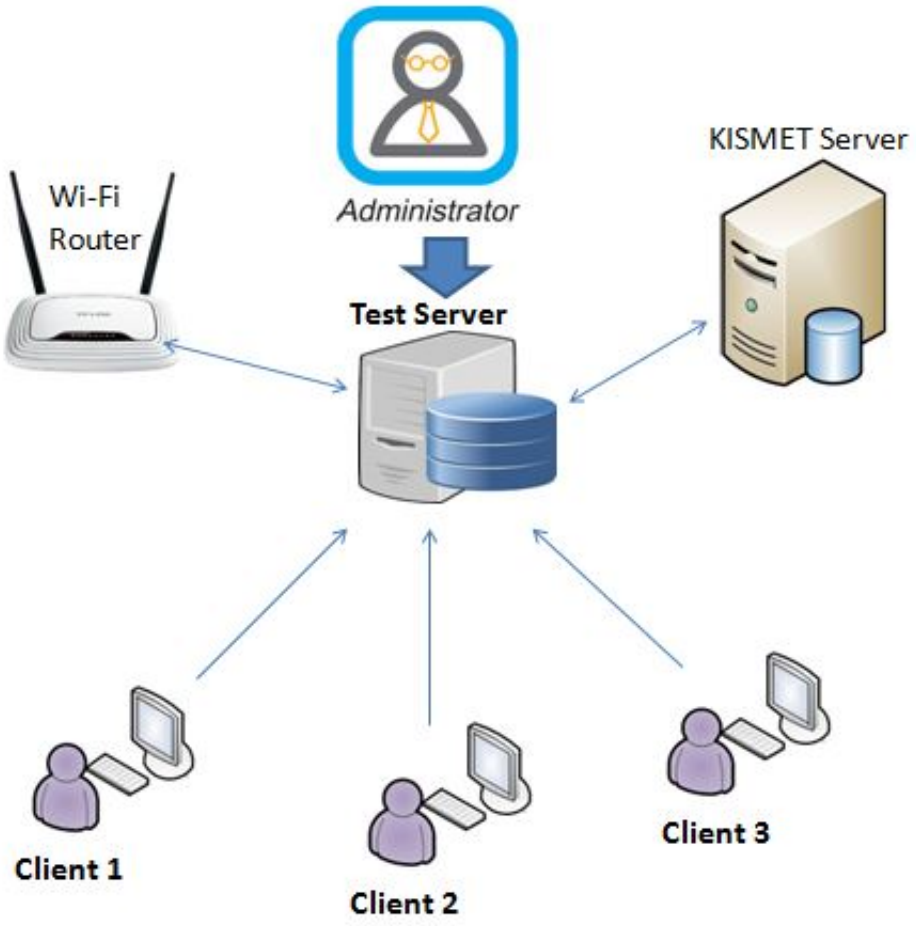

Figure 1. Architecture Diagram of Online Examination

Kismet server [13] will begin catching packets. When exam is over, we will have pcapdump (packet capture). A python script will parse every one of these packets and make a .csv record with important information of our interest like HTTP URL (HTTP : hypertext transfer protocol, URL : uniform resource locator), User Agent and so on from the packet.

4.3 Online Test System

Online test is hosted on test server. The test layout looks like as in Figure 2.

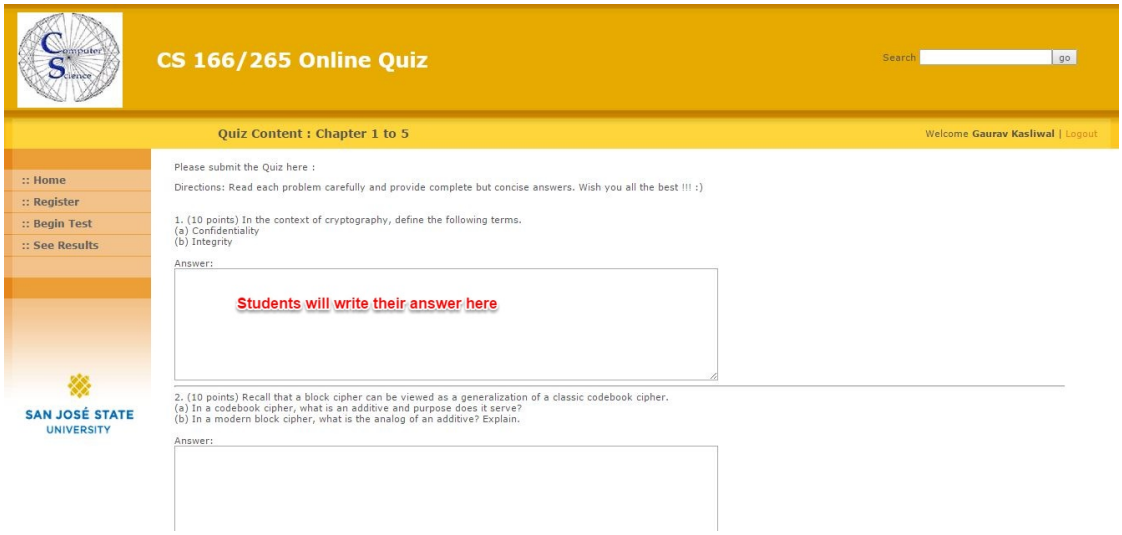

Figure 2. Test Layout of Online Exam

Steps to give online exam:

1. Student will connect to give WiFi using SSID and password given at the beginning of the test.
2. We will provide student with test URL like `http://192.168.1.2/onlinetest/`. Student will go to that URL and register online to login for online test.
3. After successful login into the test, student will start giving test.
4. Once they are done with all the questions, submit the test.
5. There is no extra overhead on client(student) machine to install any software.
6. All the information will get stored to a database. The database schema we are using for this project is shown in Figure 3.

We are utilizing the database to store the test outcome and data that student enter while doing enrollment online. We picked MYSQL database as it comes

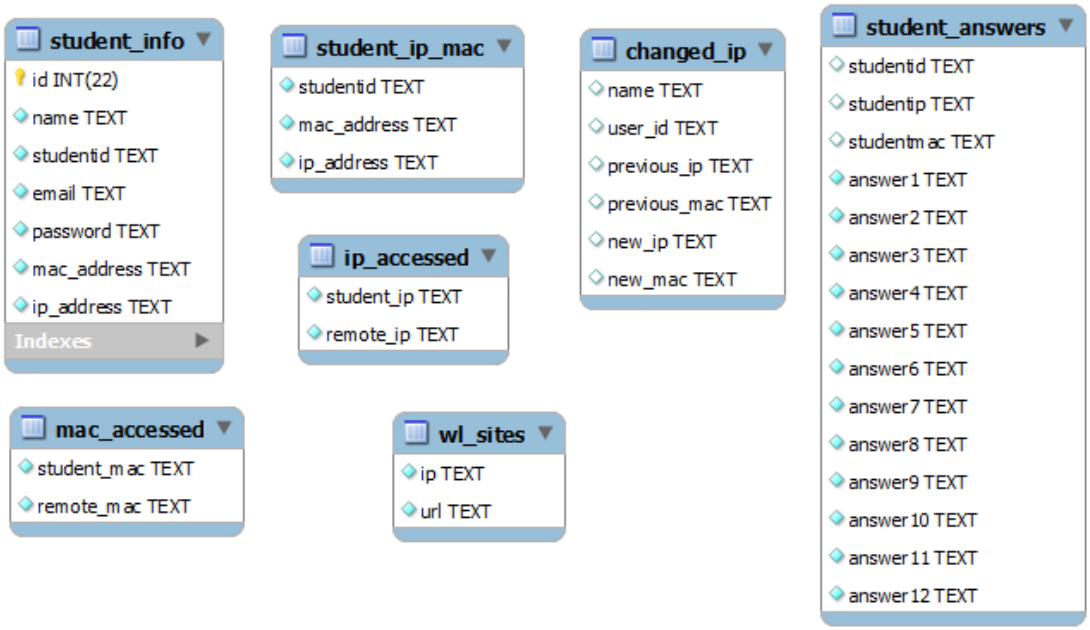

Figure 3. Database Schema for Storing Important Information

packaged with the XAMPP server. We are utilizing diverse tables to store the data.

Table Description:

student_info: Student information is stored at registration time. student_ip_mac: Student–id, MAC address and IP address of the student's machine. ip_accessed: It stores all the remote IP address accessed by particular student. changed_ip: If someone tries to change the IP address, we store mapping between student id and IP address information in this table.

wl_sites: White list of URL which student can use during exam.

4.4 KISMET

4.4.1 Introduction

Kismet implies "Fate" or "Destiny". Kismet [13] is the most imperative instrument for this project. It is a network detector, packet sniffer, and intrusion detection system framework for 802.11 remote LAN. Kismet will work with any wireless card which supports raw monitoring mode, and can sniff 802.11a, 802.11b, 802.11g, and 802.11n traffic. The program runs under Linux, Free BSD, Net BSD, Open BSD, and Mac OS. Kismet contrasts from different remote system locators in meeting expectations inactively. It meets expectations inactively, implies it does not send any bundles for logging reason. Kismet has the capacity identify the vicinity of both wireless access points and wireless clients, and to partner them with one another. It is likewise the most broadly utilized and cutting–edge open source remote observing apparatus. Kismet includes the capacity to log every sniffed packet and spare them in a tcpdump [23] or Wireshark [35] record design. Kismet can likewise catch "Per Packet Information" headers. More insights about setup and use are included with snapshots in Appendix A.

4.4.2 Configuration

With a specific end goal to arrange the wireless card [12] for monitor mode and begin catching bundles, Kismet need to have root access. We can have Kismet (packet capture server) server running on one portable workstation. Begin kismet [13] as root. Install it so that the control segments are situated to begin as root. Beginning kismet as root implies that Kismet will keep running as root. Installed frameworks regularly have a great deal less storage room and RAM, and frequently do not implement user/root division as entirely because of

these constraints.

4.4.3 Capture Sources

All packets in Kismet originate from a capture source [12]. Capture sources are ordinarily network cards on the local system, on the other hand they can likewise be a already recorded file or a remote capture framework running a Kismet drone. Capture sources may be included by means of the Kismet UI under the "Add Source" choice, in which case the alternatives may be included under the "Options" field, comma differentiated. They might likewise be characterized in the kismet .conf design record as the "ncsource=" alternative, such as : ncsource=wlan0:option1=foo,option2=bar

4.4.4 Logging

Kismet will log the .pcap file, GPS log, alerts, and network log in XML and plain text. Naturally, Kismet will attempt to log to .pcap documents utilizing the PPI every packet header. The PPI header is a well documented [12] header supported by Wireshark and different devices, which can contain range information, radio information, for example, flag and noise levels, and GPS information. PPI is just accessible with late libpcap forms. When it is not accessible, Kismet will fall back to standard 802.11 configuration with no additional headers.

4.4.5 Legal Issues

There is nothing unlawful about Kismet [12] (it is not quite the same as some other capture tool) but rather we should check our nearby laws first. Recording information from systems that you do not have authorization to may be viewed as

an unlawful. Utilizing systems you do not have consent to utilize may be viewed as “Theft of Service” and is illicit.

4.5 Useful Tools

Following tools were used for implementing this project. Tools used were valuable for catching the packets inside system, putting away data into the database, test server for facilitating the online test.

4.5.1 Wireshark

Wireshark [35] is a free and open–source packet analyzer. It is utilized for system investigating, examination, programming and correspondences convention improvement, and instruction. Wireshark is fundamentally the same to tcpdump, yet has a graphical front–end, in addition to some coordinated sorting and sifting choices. Wireshark permits the client to put system interface controllers that backing promiscuous mode into that mode, keeping in mind the end goal to see all activity unmistakable on that interface, not simply movement tended to one of the interface’s designed addresses and show/multicast activity.

4.5.2 TCPDUMP

Tcpdump [23] is a typical packet analyzer that runs under the command line. It permits the client to capture and showcase TCP/IP and different packets being transmitted or got over a system to which the PC is connected. Tcpdump is free programming and appropriated under the BSD permit. Tcpdump prints the content of network packets. It can read bundles from a system interface card or from an already made spared bundle record. Tcpdump can compose parcels to

standard yield or a record.

4.5.3 XMPP Server

Extensible Messaging and Presence Protocol (XMPP) [10] is a communication protocol for message–situated center product in view of XML (Extensible Markup Language). The protocol was initially named Jabber. We can have online test and utilization database for putting away data which we are getting over the system into the database. The Internet Engineering Task Force (IETF) [3] framed a XMPP working gathering in 2002 to formalize the center conventions as an IETF texting and vicinity innovation.

4.5.4 MYSQL

MYSQL [21] is a relational database administration framework (RDBMS), and come without GUI tools to control MYSQL databases or oversee information contained inside the databases. We are putting away all the data from the test into MYSQL database for further investigation.

CHAPTER 5

Cheating Detection and Avoidance Approaches

There are numerous ways in which we can limit student from cheating during online exam and diminish the effect of cheating [5, 26] for online exam. Online examination is by all accounts reliable however it is a welcome for doing cheating with accessible online assets. In this chapter, we talk about three distinct techniques for cheating detection. In particular, we consider Blacklist URL, SVM Machine Learning Approach and URL Frequency Analysis technique. In following sections, we are discussing the details of these approaches for our project.

5.1 Blacklist URL Method

Designed and developed simple blacklist and whitelist URL [9, 22] approach to classify HTTP URL and detect cheating. This is the most fundamental method for cheating detection by discovering the blacklisted URL used by any student during exam. We have to keep up rundown of blacklist [1] and whitelist URLs on test server. It is hard to keep up to date database for blacklisted and whitelist URL as they may change over the time of time. As we get the packet capture, bring the HTTP URL from the packet and think about if that URL is present in the blacklisted URLs and arrange them as blacklisted URL, on the off chance that it is introduce in the blacklisted URL database. The principle focal point of this system is we can have 100 % precision for our outcomes.

5.1.0.1 Blacklist URL

The blacklist [1] is a list of all the URL that students are not supposed to visit during the online test or quiz. e.g. for CS 265/166 test we had following URLs in blacklist.

Table 1. Blacklist URLs

Dr. Stamp's website
YouTube videos
Webopedia
Coursera website
Wikibooks
Google Drive
Dropbox
Stackoverflow website
Stackexchange
Udacity website

5.1.0.2 Whitelist URL

Whitelisted URL is a list of URLs that student can visit during the test or quiz. In the event that student visit these URLs during exam, it will not be considered as cheating. For CS 265/166 test we had URLs like Piazza and Wikipedia in whitelist.

5.1.0.3 HTTP Packet format

HTTP packet [4] comprises of HTTP header, Body and Trailer. The HTTP header comprises of an request or response line. HTTP request line contains a method, URL, and version. HTTP response line contains a version, status code, and reason phrase. A MIME header is contained zero or more MIME fields. A MIME field is made out of a field name, a colon, and (zero or more) field qualities. The qualities in a field are differentiated by commas. HTTP header containing a

request line is normally referred to as an request. The following example shows a typical request header.

HTTP Request Example
GET http://www.google.com/ HTTP/1.0
Proxy-Connection: Keep-Alive
User–Agent: Mozilla/5.0 [en] (X11; I; Linux 2.2.3 i686)
Host: www.google.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1, *, utf-8

The response header for the above request may look like as takes after:

HTTP Response Example
HTTP/1.0 200 OK
Date: Fri, 13 Nov 2009 06:57:43 GMT
Content–Location: http://mail.google.com/index.html
Etag: "07db14afa76be1:1074"
Last-Modified: Thu, 05 Nov 2009 20:01:38 GMT
Content-Length: 7931 Content-Type: text/html
Server: Microsoft-IIS/4.0 Age: 922
Proxy-Connection: close

HTTP message with an expanded HTTP header is shown in Figure 4.

It is simple and effective strategy. This strategy have basic calculation and quick preparing. It gives great Performance and 100% Accuracy. Yet, we can't

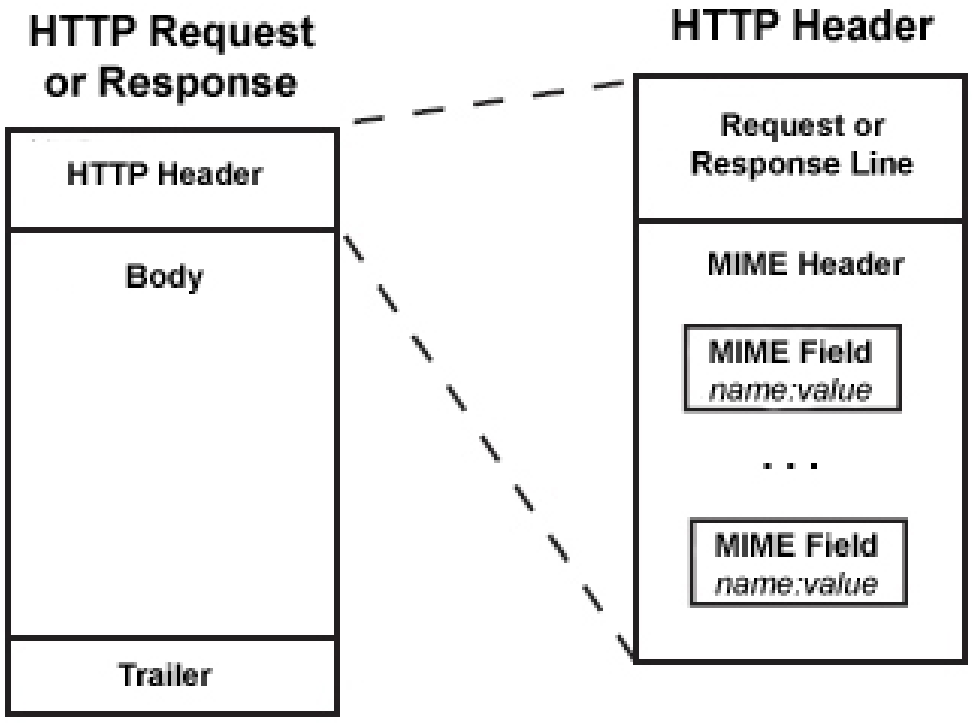

Figure 4. HTTP Request/Response and Header Structure

identify URL outside the blacklist. Blacklist [1] needs to be kept up to date from time to time. There is additional overhead for upgrading the database of blacklisted URL.

5.2 Machine Learning Approach

We are using Lib SVM [16] (support vector machine) method for machine learning algorithm and analyze the URL dataset we get after processing packet capture from kismet server. This is a useful approach to overcome the downside of the blacklist URL approach. We create a model based on training dataset of URL. For training purpose we have used DMOZ [29] dataset which is largest, most comprehensive human–edited directory of the Web. It was historically known as

the Open Directory Project (ODP). It contains a categorized list of Web URLs. We used the Computer data dataset from DMOZ as a train data (1547974 URLs, including sjsu.edu). We are building the model based on DMOZ URL dataset. We collect the test data (URL dataset from kismet server) and apply this model created in previous step to test dataset. The main advantage of this method is, it can classify unknown URLs as well. We are using edit distance property to find how similar two strings are. The edit distance [25] between two strings $s = s_1 \dots s_n$ and $t = t_1 \dots t_n$ is given by d_{mn} defined by the recurrence. We find edit distance between two strings by calculating the number of transformation required to transform one string to another. We can use copy, substitute, insert, delete operations to transform one string to another.

Let d_{ij} = score of best alignment from $s_1 \dots s_i$ to $t_1 \dots t_i$. Then

$$d_{ij} = \begin{cases} d_{i-1,j-1} & \text{if } s_i = t_j \text{(copy)} \\ d_{i-1,j-1} + 1 & \text{if } s_i \neq t_j \text{(substitute)} \\ d_{i-1,j} + 1 & \text{(insert)} \\ d_{i,j-1} + 1 & \text{(delete)} \end{cases} \tag{1}$$

Define

$$d(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Then using Equation 2, we can reduce 1 to

$$d_{ij} = \begin{cases} d_{i-1,j-1} + d(s_i, t_j) & \text{(copy or substitution)} \\ d_{i-1,j} + 1 & \text{(insert)} \\ d_{i,j-1} + 1 & \text{(delete)} \end{cases} \tag{3}$$

We can simulate the algorithm by taking a small example with two strings "PARK" and "SPAKE". Initial state using Equation 3 for these two strings is shown in Table 2.

Once we do all the iterations for all the characters in the string, we get the final state as in Table 3. The final cell in the table, i.e. cell value in last row and

Table 2. Initial State for Edit Distance

		P	A	R	K
	0	1	2	3	4
S	1				
P	2				
A	3				
K	4				
E	5				

last column shows the edit distance between two strings. As per Table 3, the edit distance between two string "PARK" and "SPAKE" is 3.

Table 3. Final State for Edit Distance

		P	A	R	K
	0	1	2	3	4
S	1	1	2	3	4
P	2	1	2	3	4
A	3	2	1	2	3
K	4	3	2	2	2
E	5	4	3	3	3

5.2.0.4 LIBSVM for string data

In machine learning, support vector machines are supervised learning models with associated learning calculations that examine information and perceive patterns, utilized for classification and regression analysis. SVM is powerful machine learning instrument for non string inputs. We utilized Lib SVM [16] for applying machine learning way to deal with string based information.

In machine learning and data mining, a string kernel [16, 33] is a kernel function that works on strings, limited successions of symbols that need not be of the same length. String portions can be instinctively seen as capacities measuring the comparability of sets of strings, more comparable two strings a and b are, higher the estimation of a string kernel $K(a, b)$ will be. Utilizing string part with

kernelized learning calculation, for example, support vector machine permit such calculations to work with strings, without needing to make an interpretation of these to settled–length, genuine esteemed highlight vectors.

$$\text{Kernel Function } = \exp(-\gamma \cdot \text{edit}(x, y)^2)$$

where, edit(x,y) is the edit distance between two strings and

$$\gamma = 1/\text{number of features}$$

Here, we are listing all the URLs from the captured packet and convert it into test dataset. Run the model on this test dataset and calculate accuracy. This method gives good result. We run the SVM model via command line and commands are available for training the model and predicting the results based on the model we get after running SVM on training data.

We did experiment using DMOZ dataset in training dataset and URL captured during online test as test dataset. We used kernel type as edit distance. Graph in Figure 5 shows the result graph. We are having the training dataset size on x–axis and accuracy in percentage on y–axis. We can see that the accuracy increases as test dataset size increases. The growth is greater from 1000 to 4000 (size of training dataset) but increase in accuracy is less from 4000 to 10000 (size of training dataset). More results with different size of train and test dataset are given in Appendix A.

5.3 URL Frequency Analysis

This is a useful approach to find out the frequency of URL accessed by students and find out highly accessed URL and very rarely accessed URL. If one of the URL is getting accessed more frequently, we can suspect that this URL might be

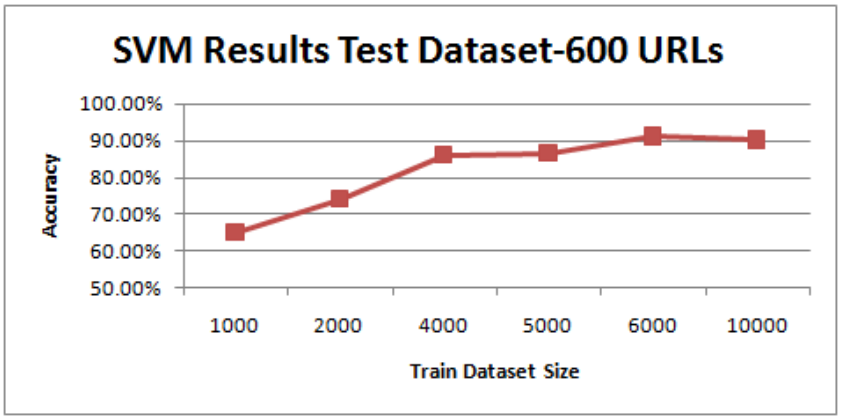

Figure 5. SVM Results for Train Dataset of Size 600

suspicious. If one of the URL is getting accessed rarely, we can also look for that URL being suspicious. Algorithm for URL frequency is as follows:

1. Extract and list out all the URLs student visit during exam.
2. Calculate the overall frequency of the URL accessed during the test.
3. Find out the frequency of the URL accessed by each student or user of the system.
4. Assign weight to each URL based on the frequency of the URL. URL, that is accessed most number of times will be assigned higher weight.
5. Calculate score as per the weight given to each URL in previous step for URL accessed by each student.
6. Decide a threshold score and all the students with score above threshold score will be classified as cheater.

Graph in Figure 6 shows the number of times a given URL is accessed. If given URL is not in blacklist and access count is high, it can be suspicious and we can add it in blacklist.

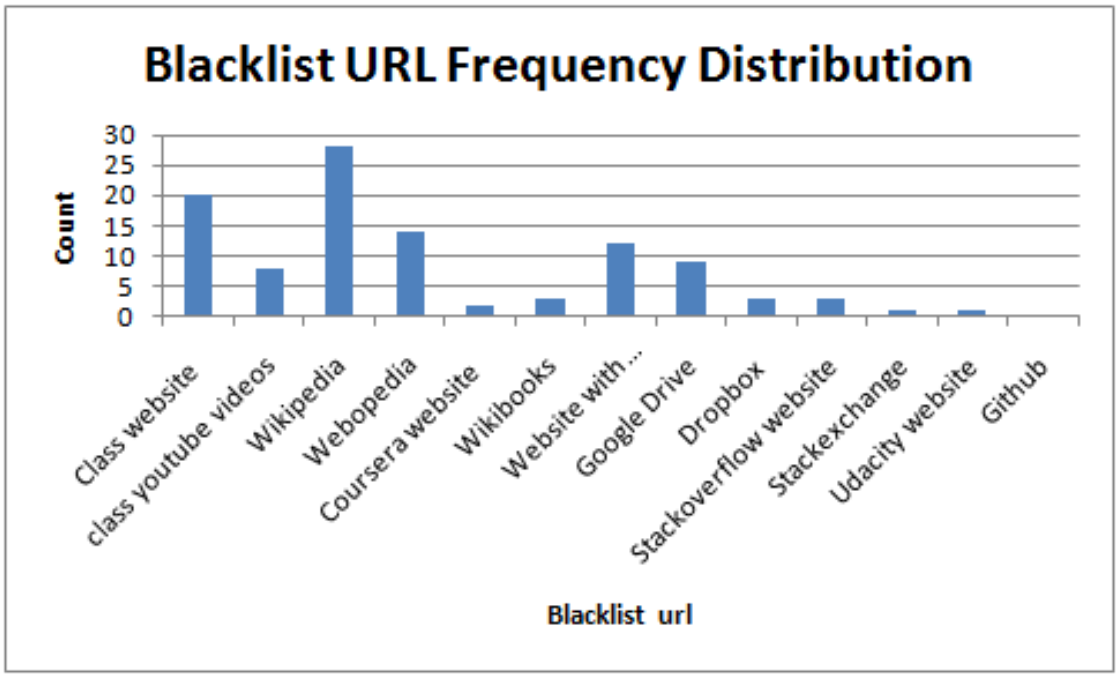

Figure 6. Blacklist URL Frequency Distribution

CHAPTER 6

Experiments

This chapter discusses about the experimental setup, methods used in the experiment and results obtained from these experiments. Experiments involves implementation and testing of different methods as described in upcoming sections.

6.1 Test Setup for Experiment

There are multiple ways in which student can cheat [26]. We conducted an experiment of in–class online examination. We took a test for 2 classes, CS 265 and CS 166 in San Jose State University CS department. There were 60 students who gave online exam [11]. We created a test and hosted it on test server in class. Students were having different types of OS on their laptop. More specifically, there were 15 MAC, 41 Windows and 4 Linux user. Students were able to give test smoothly without installing any software on their laptop. As students were having access to the internet, they tried cheating in multiple ways. In following sections we will discuss the main observations from the test.

6.2 Results

6.2.1 SVM Results

We ran multiple test cases for getting SVM [16] results. We trained the model using train dataset and created a model. Then, used this model to get results for test dataset. Size of test dataset and train dataset we used for this experiment is as follows:

Here are the graphs showing the comparison of the SVM results on different

Table 4. SVM Results for Different Size Dataset

Test dataset size	**Train dataset size**	**Accuracy**
100	1000	62.00%
100	2000	77%
100	4000	82%
100	5000	83%
100	6000	89%
100	10000	91%
200	1000	61.00%
200	2000	76.00%
200	4000	81%
200	5000	83%
200	6000	85.00%
200	10000	92.00%
400	1000	61%
400	2000	75.00%
400	4000	89.00%
400	5000	87.00%
400	6000	91.00%
400	10000	91.00%
500	1000	67.00%
500	2000	77.00%
500	4000	86.00%
500	5000	87.00%
500	6000	89.00%
500	10000	90.00%
600	1000	65.00%
600	2000	74.00%
600	4000	86.00%
600	5000	86.50%
600	6000	91.00%
600	10000	90.00%

size of train and test dataset. From the graph we can see that, accuracy increases sharply from 60% to 90%, but then it increases slowly. The graph becomes stable near 90% accuracy, even if we increase the size of train data, accuracy does not increase significantly.

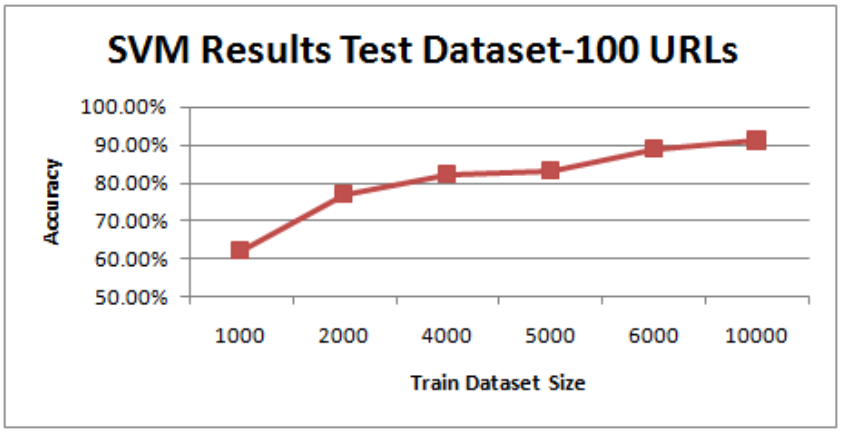

Figure 7. SVM Results for Train Dataset of Size 100

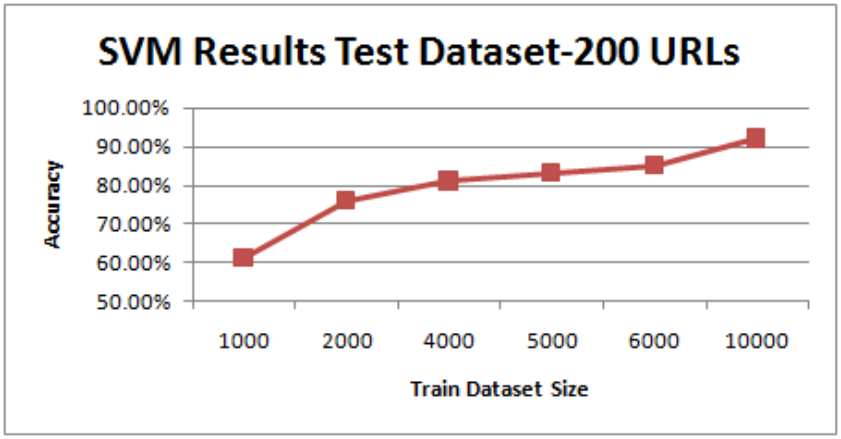

Figure 8. SVM Results for Train Dataset of Size 200

6.2.2 Cheating Methods Student Used

We conducted experiment for in–class test and we found following scenarios for cheating [26]. The points below discusses different methods for cheating and how we detect them.

Incognito mode issues us local privacy [27] and not network privacy. Numerous students tried using incognito mode for cheating. It is a security feature included in web browser to incapacitate perusing history. This permits an individual to peruse the Web without putting away neighborhood information that could be

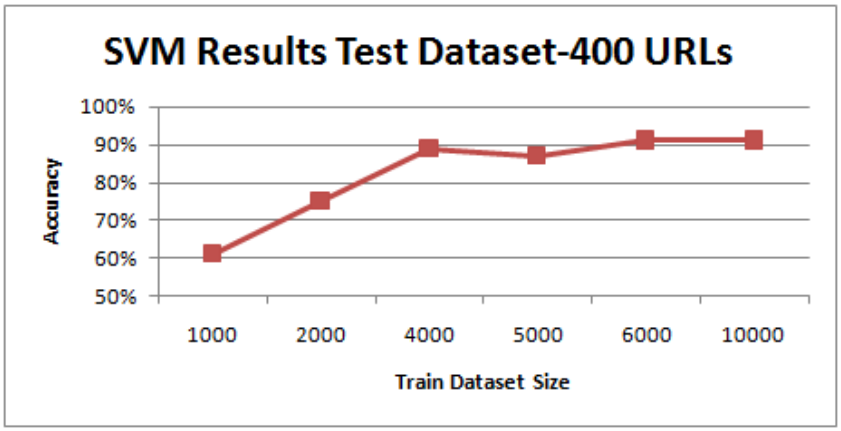

Figure 9. SVM Results for Train Dataset of Size 400

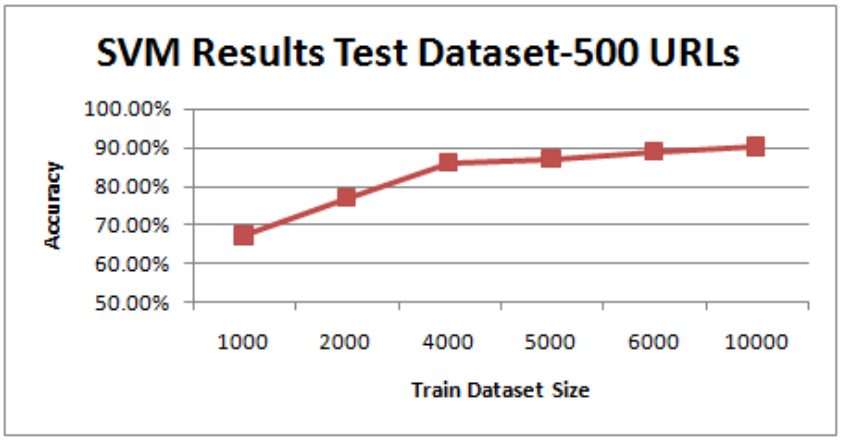

Figure 10. SVM Results for Train Dataset of Size 500

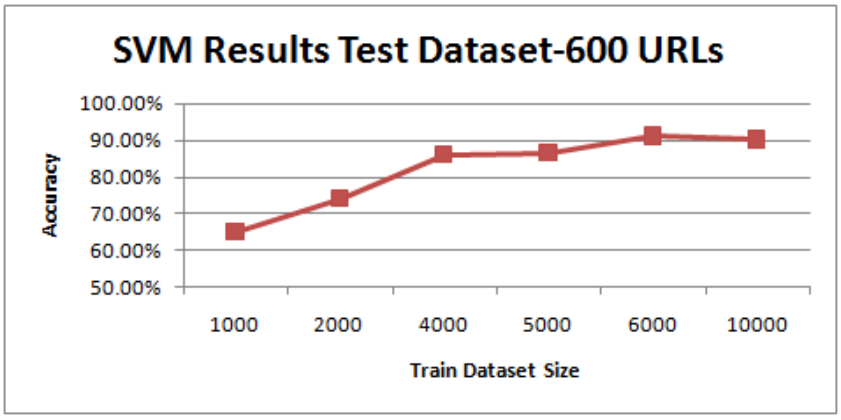

Figure 11. SVM Results for Train Dataset of Size 600

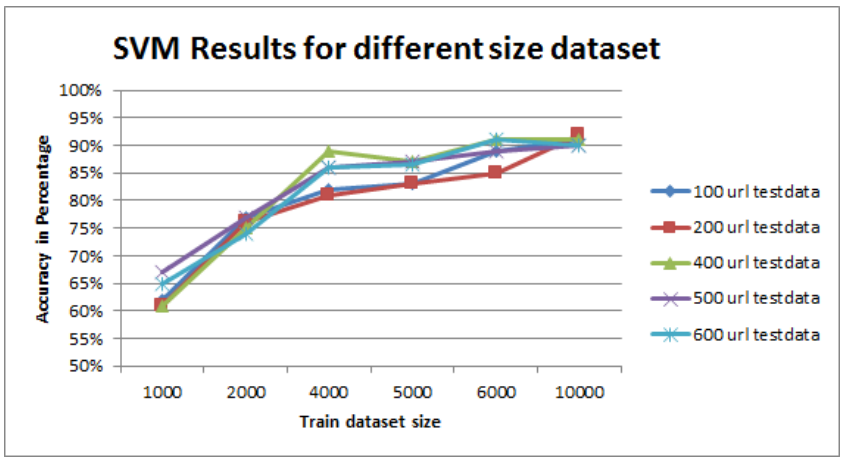

Figure 12. SVM Results Comparison for Different Size of Test and Train Dataset

recovered at a later date. This protection security is just on the local processing gadget and it does not prevent a test server from catching a system activity student is attempting. Few students utilized virtual box to open up a virtual machine and skimming on the Internet from program in virtual machine. For the test, we provide a SSID for connecting to test server and then only student will have access to the test server. Our system can detect if student changes their WiFi by continuously polling, to which SSID the student IP address is connected to.

ZenMate [36] is extension in browser for opening the web and issues you free security and scrambles the majority of our program movement. Change our area with our VPN (Virtual Private Network) to get to access in the web that are not accessible in specific regions. Access destinations in the web that are not accessible in specific regions, quicker than utilizing a basic intermediary administration.

Students tried tethering [6] during exam. Tethering is uniting one gadget to another. Tethering permits offering the Internet connection of the phone with different gadgets like tablet, laptop etc. Connection of the telephone or tablet with different gadgets could be possible over remote LAN (WiFi), over Bluetooth or

by physical association utilizing a link, for instance through USB. In the event that tying is done over WiFi, then it is Mobile Hot spot association. The Internet associated cell phone can go about as a compact remote access point and switch for gadgets joined with it. In the event that student tries to associate with an alternate WiFi other than one for the test, then our framework can distinguish this by consistently surveying, to which SSID the student IP location is joined with. Student tried using a Team viewer connection. Team Viewer is a restrictive PC programming bundle for remote control, desktop imparting, online gatherings, web conferencing and record exchange between PCs. Team Viewer can be utilized without charge by non–business clients. RDP [18] is an exclusive convention grew by Microsoft, which furnishes a client with a graphical interface to unite with another PC over a system association. The client utilizes RDP customer programming for this reason, while the other PC must run RDP server programming. One can have entry to the greater part of our projects, documents, and system assets, generally just as we were before our PC. This methodology can just bolster one client at once.

Students found online file of textbook using Google search. We can add that sources into blacklist and detect if student are using these URLs. Students stored their power–point slides and notes on Google drive and Dropbox. We added the Drop Box and Google drive into the blacklist. We need to list out the websites that provide online storage and add them into blacklist. Students tried using browser other than one on which they are giving the test. Still, our system can capture the traffic and nothing stops us from detecting cheating in this case. We discussed different methods student tried for cheating during exam. Graph in Figure 13 shows the percentage wise distribution of different cheating methods [20] students

tried. We can come up with corrective action depending upon the method most of the students are using for cheating.

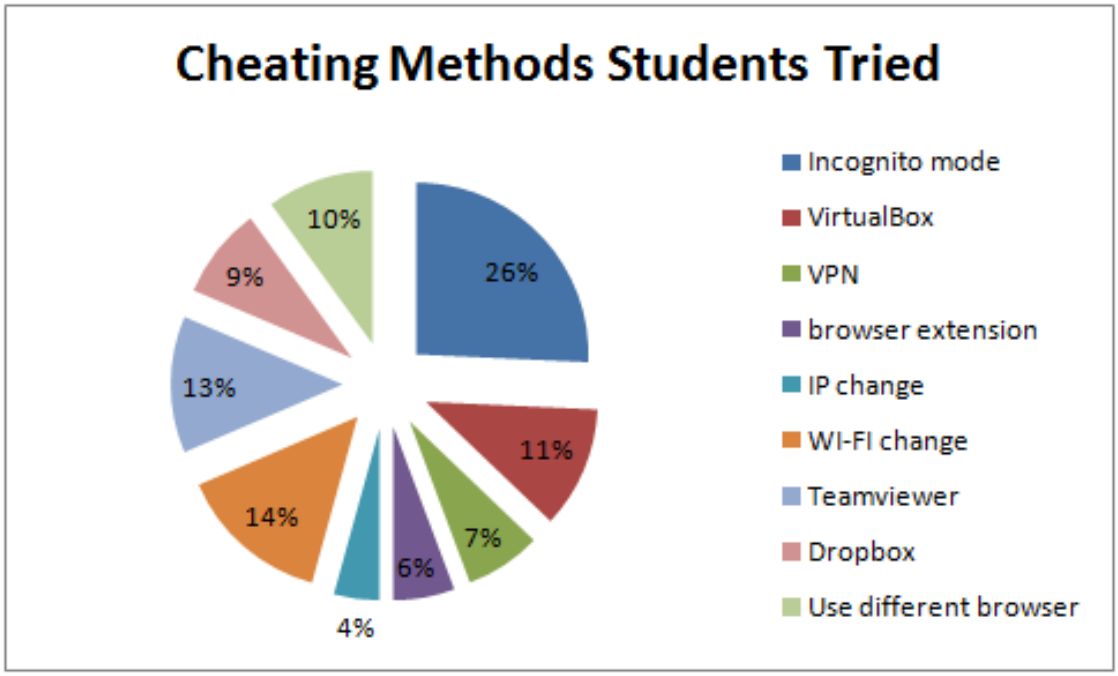

Figure 13. Cheating Methods Student Tried

6.2.3 Cheating Methods Difficult for Detection

We took feedback after exam, from students about cheating methods they tried. Some of the methods are mentioned below. A student can connect via RDP to his server at home and browsed the web from there. Since traffic over RDP is encrypted and even if packets are captured, we will only see TCP.port=3389 traffic and we will not be able to detect cheating of this type. Student can talk to a classmate through Google hangouts, as Google hangout uses HTTPS traffic. Typing in a message and encrypting it using a written Caesar's cipher program and a shared email with a classmate. Some student used Google Document to communicate between them. Student had access to the email so they saved the

encrypted message as a draft and the other person decrypted it using the same program. So only the encrypted message was being sent over the network.

6.2.4 Cheating Methods Cannot be Detected

During In–class online examination, students used their personal laptop. Students were already having all the power point slides, class notes, homework stored locally on their laptop. If we can provide students with a laptop/desktop from the lab, without any local storage, this type of cheating can easily be removed. Students may ask answers to a question to another student. This activity cannot be detected. This can easily be removed, if we provide examiner monitoring student during exam.

6.2.5 ROC Curves

In this section, we are drawing receiver operating characteristic(ROC) curves. In ROC. We plot true positive rate(TPR) vs false positive rate(FPR) by varying threshold through the range of scores. That is, FPR on x–axis, TPR on y–axis. Proportionally, we can say that it is sensitivity vs specificity. Receiver Operating Characteristics (ROC) curve is a graphical representation for accuracy of classification system. It is made by plotting genuine positives out of the aggregate real positives (called as TPR = genuine positive rate) versus the portion of false positives out of the aggregate genuine negatives (called as FPR = false positive rate). TPR and FPR are otherwise called sensitivity of classification system. For plotting ROC curve, we figure probabilities for genuine positive and false positive. A curve is created by plotting the cumulative distribution function of the genuine positive probabilities on the y–axis versus the aggregate conveyance capacity of

false positive probabilities on the x–axis. The Area Under the Curve (AUC) of ROC gives a measure of accuracy of characterization. True Positive (TP) is an endeavor to cheat that is recognized and is real cheating. In the event that student attempted to cheat and get recognized, it will be considered TP. On the off chance that student utilize any URL from the blacklist URL, our tool will recognize them. e.g. student used stackoverflow.com during test and our tool detected him as a cheater. False Negative (FN) is an endeavor to cheat, however there is been no recognition of duping. On the off chance that student attempted to cheat and did not get identified it will be considered FN. In the event that student utilize one of the technique for cheating which can not be distinguished by our tool, they will cheat in exam being undetected. e.g. Use of VPN connection during exam. True Negative(TN) is student did not attempt to cheat and there is been no recognition. The student did not attempt to cheat and there is no recognition, Students are utilizing ordinary test server URL is TN. URL caught during student is skimming through the whitelist URL or simply giving test without doing any sort of cheating, will be viewed as True Negative. e.g. URL captured while giving test i.e. online test URL in packet capture(which is very likely to come in packet capture) or URL captured while using piazza during (it is URL from whitelist) test. False Positive (FP) is student did not attempt to cheat but rather identified as cheater. Student got to some URL not from blacklist, we thought that it was suspicious and recognize as cheating is considered False Positive (FP). e.g. URLs like proxfree.com or utilization of Hola Chrome Extension.

The curves are drawn as show in Figures 14 and 15 based on the URL count and whether it is True Positive, True Negative, False Positive, False Negative. Figure 14 shows that area under the curve is 0.95040, means we get 95.04% accuracy

and Figure 15 shows that area under the curve is 0.936, means we get 93.60% accuracy.

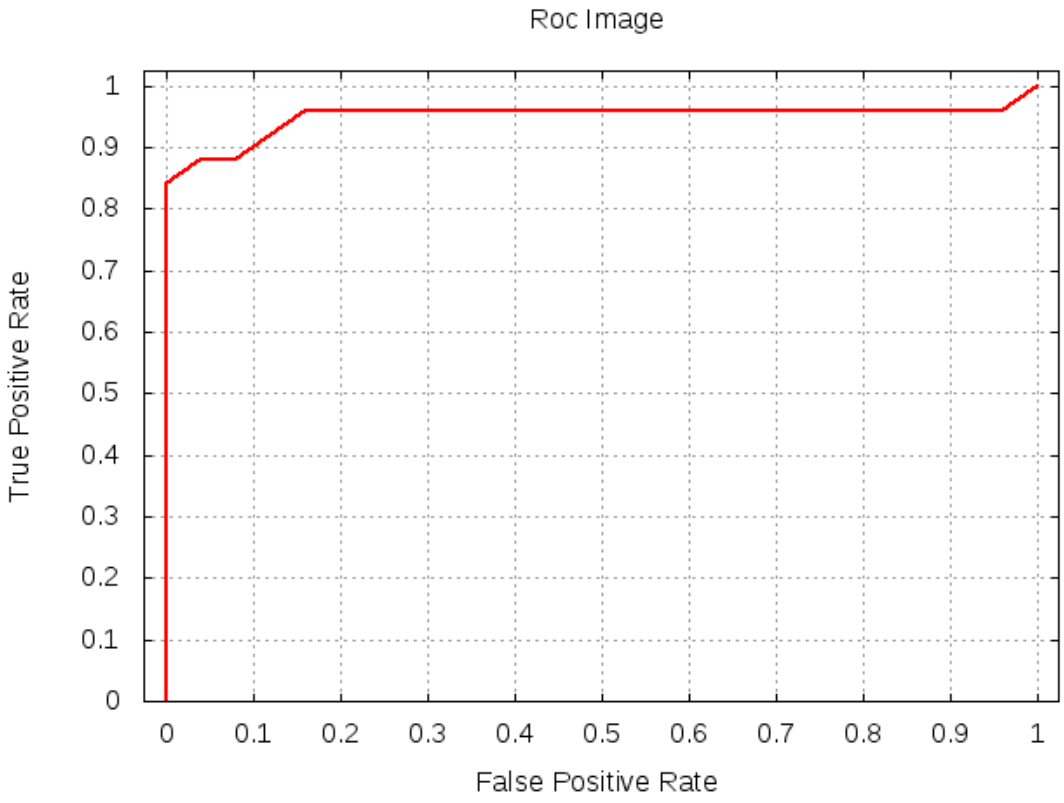

Figure 14. URL Based ROC Curve for Grad Class Test

6.2.6 Word Cloud Graph

A Word Cloud is utilized for visual representation for content information. It is normally used to depict keyword metadata form text. Tags are single words extracted from the long text and the importance of each tag is shown with font size or color or both of them. This arrangement is valuable for rapidly seeing the most conspicuous terms and for finding a term in order to focus its relative unmistakable quality. We found out some commonly used words from the packets we captured and cheating methods students tried. The world clouds are drawn as

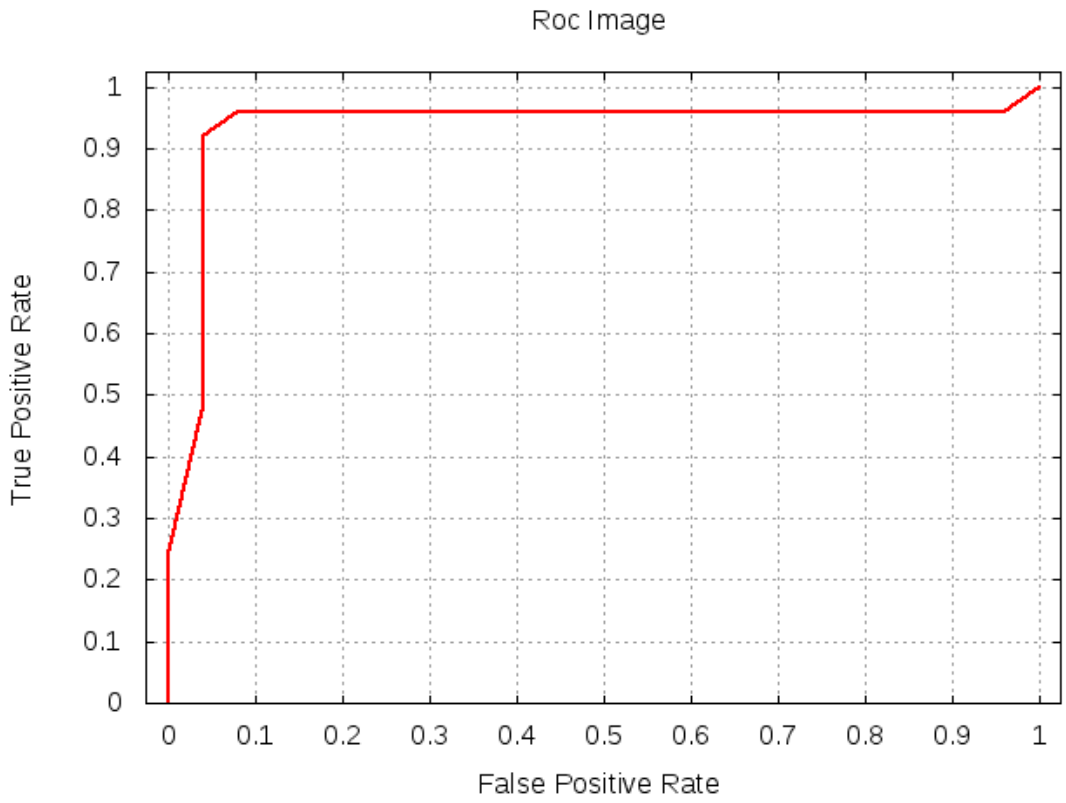

Figure 15. URL Based ROC Curve for Under–Grad Class Test

show in Figures 16 and 17 based on the URL count and whether it is True Positive, True Negative, False Positive, False Negative.

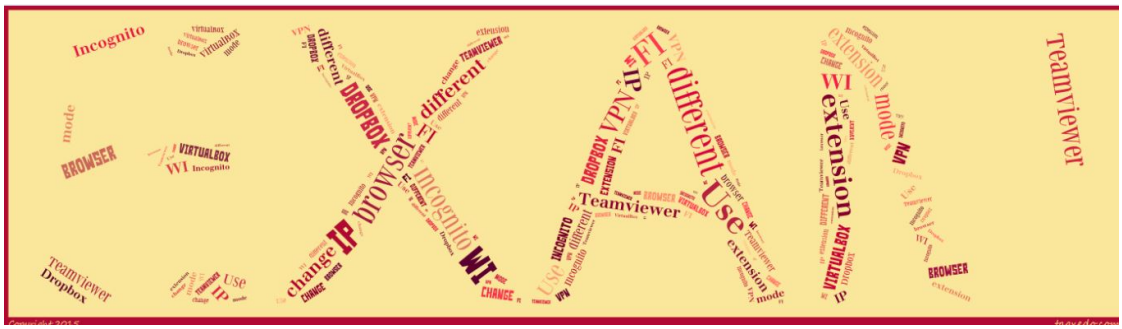

Figure 16. Word Cloud of Cheating Methods Students Used in Exam

In Figure 16 we are showing different cheating methods students used during exam. From the word cloud, we can see that, student are using methods like WiFi

change, using another browser more as compared to other methods. In Figure 17 we are showing different URLs students used during exam. From the word cloud, we can see that, students are accessing URLs like Webopedia, You tube more as compared to other URLs.

Figure 17. Word Cloud of URL Students Used in Exam

CHAPTER 7

Future Work and Enhancements

7.1 Online Exam Setup Enhancements

Our online exam [11, 37] setup involves test server and packet capture server [13]. We could have single server handling both the things. We are capturing the network traffic while online exam is going on using kismet server and then analyze this traffic to get the results. We can build the system, which will analyze the network traffic [13] and give real time results. In this project, we are capturing the network traffic during exam time by kismet server and processing it offline after exam. We could develop a tool to analyze the traffic in real time and give results immediately on some online portal.

7.2 Analyzing HTTPS Traffic

For HTTP traffic, we have access to HTTP URL and other data as well. For now, we are using HTTP URL only. We get useful information if we read the packet in more detail. We can look into much detail for information in HTTP packet but if traffic is HTTPS, we cannot go deep into the packet and get the packet details as it is encrypted. HTTPS is a secure version of HTTP that uses the Secure Socket Layer or Transport Layer Security (SSL/TLS). Normally, it is utilized for administration that runs over HTTP and obliges security. HTTPS meets expectations with the TCP port 443. We are not able to break down the HTTPS system traffic [7] in this project. We could not process secure HTTPS system activity and break down packet included in applications like Gmail, Google Chat, Google Hangout, online commute and so on. We could discover somehow to manage HTTPS traffic.

We can decode SSL and TLS activity utilizing the Wireshark system protocol analyzer. In Wireshark, the SSL dissector is completely practical and backings propelled highlights, for example, decoding of SSL, if the encryption key is given. However, getting the encryption key is by and large in–feasible.

CHAPTER 8

Conclusion

8.1 Conclusion

We designed and developed a cheating detection tool [24] for online examination.The previous work in this area consists of using pre–defined blacklist URL and compare them with URL accessed by student. Apart from blacklist URL [17, 19, 22] method, we came up with SVM machine learning and URL frequency distribution approach and compared the results. These two approaches can detect new URL, which is not in blacklist as a malicious URL and we could update the blacklist by adding that URL into the blacklist. The current effort aims to use support vector [16] machine learning technique for unknown blacklist URLs.

We conducted experiment for cheating detection in online examination by hosting the test in class and got results. We collected the results and used three different methods for detecting cheating in examination. We also took feedback from the students after the test about what methods they tried for doing cheating during online exam. The test system we implemented is easy to use and there is no overhead of installing any software on students. We found different ways, student used for cheating and how we can detect them. There are some methods for cheating, which we cannot detect using above three detection techniques. We listed out those methods in detail. During our experiments, we observed that blacklist and whitelist approach give us the exact result, but this process is not useful to detect cheating with URL not in blacklist URL. The machine learning techniques can classify and predict new URL to be good or bad URL.

LIST OF REFERENCES

[1] M. Akiyama, T. Yagi and T. Hariu, Improved blacklisting, Inspecting the structural neighborhood of malicious URLs, *IT Professional*, 2013, 15(4), pp. 50–56 doi:10.1109/MITP.2012.118

[2] S. Anandan, Online application monitoring tool, Department of Computer Science, San Jose State University, Master's report, Paper 7, 2010, `http://scholarworks.sjsu.edu/etd_projects/7`

[3] P. S. Andre, Extensible messaging and presence protocol (XMPP) core, Jabber Software Foundation, Network Working Group, October 2004, `https://www.ietf.org/rfc/rfc3920.txt`

[4] Apache traffic server documentation, Programmer's guide, Functions for manupalating HTTP header, May 2014, `https://docs.trafficserver.apache.org/en/latest/sdk/http-headers.en.html`

[5] D. Cabrera, Faculty development and instructional design center, Tips to reduce the impact of cheating in online assessment, March 2013, `http://facdevblog.niu.edu/onlinecheating`

[6] X. Chen and J. Tan, An adaptive mobile robots tethering algorithm in constrained environments, Intelligent robots and systems, 2009, *IEEE/RSJ International Conference*, pp. 1377–1382, doi:10.1109/IROS.2009.5354800

[7] K. Cheng, M. Gao and R. Guo, Analysis and research on HTTPS hijacking attacks, *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference*, pp. 223–226, doi:10.1109/NSWCTC.2010.187

[8] Computer–Based Testing, GED testing service, Programs and services, May 2010, `http://www.gedtestingservice.com/educators/what-is-cbt`

[9] S. Egan and B. Irwin, An evaluation of lightweight classification methods for identifying malicious URLs, *Information Security South Africa*, pp. 1–6, Johannesburg, August 15–17 2011

[10] Extensible messaging and presence protocol (XMPP) standards foundation, Jabber Software Foundation, February 2011, `https://xmpp.org/xmpp-software/servers/`

[11] I. Y. Jung and H. Y. Yeom, Enhanced security for online exams using group cryptography, *IEEE Transactions 2009*, 52(3), pp. 340–349. doi:10.1109/TE.2008.928909

[12] M. Kershaw , Current Kismet Readme, Kismet documentation, Kismet 2011, `https://www.kismetwireless.net/documentation.shtml`

[13] Kismet a 802.11 wireless network detector for capturing packets, Kismet 2011–01–R1,
`https://www.kismetwireless.net/`

[14] S. Kumar, Online monitoring using kismet, Department of Computer Science, San Jose State University, Master's report, Paper 243, 2009,
`http://scholarworks.sjsu.edu/etd_projects/243/`

[15] T. Lancaster, The application of intelligent context–aware systems to the detection of online student cheating, *Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference*, pp. 517–522. doi:10.1109/CISIS.2013.94

[16] C. J. Lin and C. C. Chang, Libsvm tools, LIBSVM : a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011,
`http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[17] M. S. Lin, C. Y. Chiu, Y. J. Lee and H. K. Pao, Malicious URL filtering, A big data application, *2013 IEEE International Conference on Big Data*, pp. 589–596, Santa Clara, CA, October 6–9 2013

[18] C. Longzheng, Y. Shengsheng and Z. J. Li, Research and implementation of remote desktop protocol, service over SSL VPN, *Services Computing, 2004 IEEE International Conference*, pp. 502–505, doi:10.1109/SCC.2004.1358052

[19] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, Beyond blacklists, Learning to detect malicious web sites from suspicious URLs, Department of Computer Science and Engineering, University of California, San Diego, September 14 2009,
`http://cseweb.ucsd.edu/~savage/papers/KDD09.pdf`

[20] C. McWhirter and C. Porter, Measures to detect cheating proliferate, The Wall Street Journal, September 2014,
`http://www.wsj.com/articles/for-school-tests-measures-to-detect-cheating-proliferate-1411752291`

[21] MySQL 5.7 reference manual, MySQL, Open source database, April 2015,
`http://dev.mysql.com/doc/refman/5.7/en/`

[22] L. A. Nguyen, B. L. To, H. K. Nguyen and M. H. Nguyen, Detecting phishing web sites, A heuristic URL–based approach, *Advanced Technologies for Communications, International Conference,* pp. 597–602, Ho Chi Minh City, 2013

[23] Official website of Tcpdump and Libpcap, December 2013, `http://www.tcpdump.org/`

[24] C. C. Pan, K. H. Yang and T. L. Lee, Secure online examination architecture based on distributed firewall, *E–Technology, E–Commerce and E–Service IEEE International Conference*, pp. 533–536, Taipei, Taiwan, March 28–31 2004

[25] E. S. Ristad and P. N. Yianilos, Learning string–edit distance, Pattern analysis and machine intelligence, *IEEE Transactions 1998*, 20(5), pp. 522–532, doi:10.1109/34.682181

[26] Y. Rong, The research on the problem of ideological and political college based on cheating in the exam, *Electronics, Computer and Applications, 2014 IEEE Workshop*, pp. 51–53, doi:10.1109/IWECA.2014.6845554

[27] H. Said, N. A. Mutawa, I. A. Awadhi, and M. Guimaraes, Forensic analysis of private browsing artifacts, *Innovations in Information Technology (IIT), 2011 International Conference*, pp. 197–202, doi:10.1109/INNOVATIONS.2011.5893816

[28] D. Smetters, Respondus LockDown Browser, Assessment tools for learning systems, Octomber 2013, `https://www.respondus.com/products/lockdown-browser/`

[29] A. Tarek, DMOZ URL dataset, Categorized list of Web URLs, March 2014, `https://github.com/gr33ndata/dmoz-urlclassifier`

[30] V. D. Thanh, P. M. Tuan, V. T. Hung and D. V. Ban, Text classification based on semi–supervised learning, *Soft Computing and Pattern Recognition (SoCPaR), 2013 International Conference*, pp. 232–236, doi:10.1109/SOCPAR.2013.7054133

[31] TP–LINK TL–WR841N Wireless N300 home router, 300Mpbs, IP QoS, WPS button, August 2011, `http://www.tp-link.us/products/details/?model=tl-wr841n`

[32] A. Trivedi, A relevant online examination system, *Technology for Education (T4E), 2010 International Conference*, pp. 32–35. doi:10.1109/T4E.2010.5550114

[33] B. Vanschoenwinkel, F. Liu and B. Manderick, Weighted kernel functions for SVM learning in string domains, A distance function viewpoint, *Machine Learning and Cybernetics, Proceedings of 2005 International Conference*, (7):4226–4232, August 18–21 2005

[34] L. Wei, D. Xiaoyong, M. Hadjieleftheriou and C. Beng, Efficiently supporting edit distance based string similarity search using B–trees, *IEEE Transactions 2014*, 26(12), pp. 2983–2996. doi:10.1109/TKDE.2014.2309131

[35] Wireshark and Tshark, a wireless sniffer, May 15, 2012, `http://www.wireshark.org/docs/manpages/tshark.html`

[36] ZenMate security and privacy VPN, February 2015, `https://chrome.google.com/webstore/detail/zenmate-security-privacy/fdcgdnkidjaadafnichfpabhfomcebme?hl=en`

[37] G. Zhang and K. Haifeng, SQL paperless examination system design, *Computer Modeling and Simulation, 2010, Second International Conference*, pp. 475–478. doi:10.1109/ICCMS.2010.468

APPENDIX

Additional Screen–shots

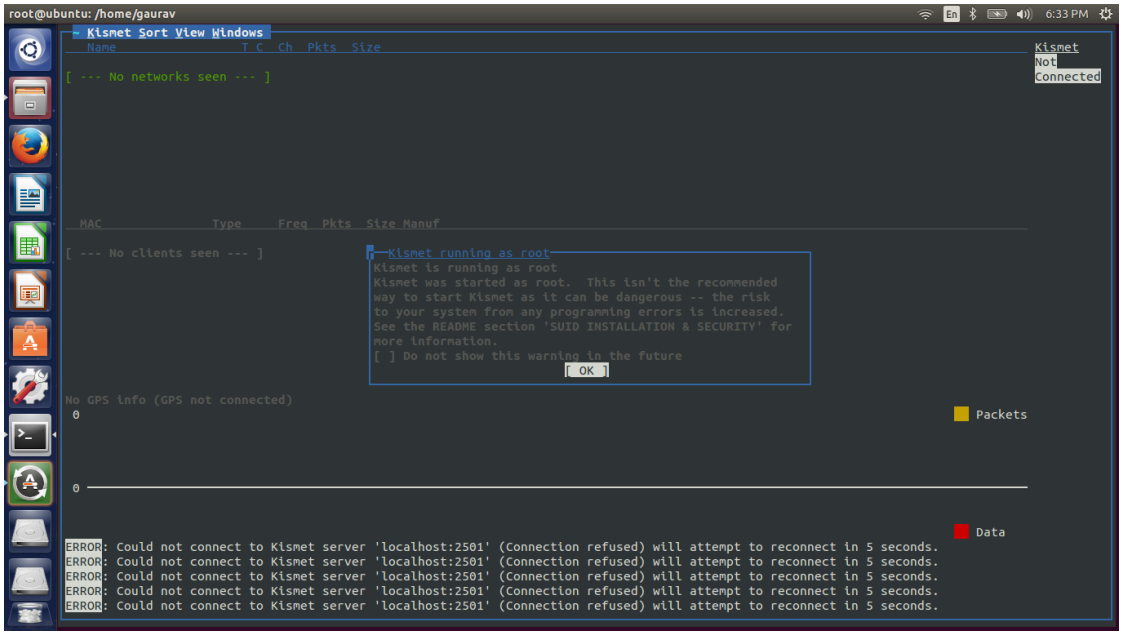

Figure A.18. Starting Kismet Server

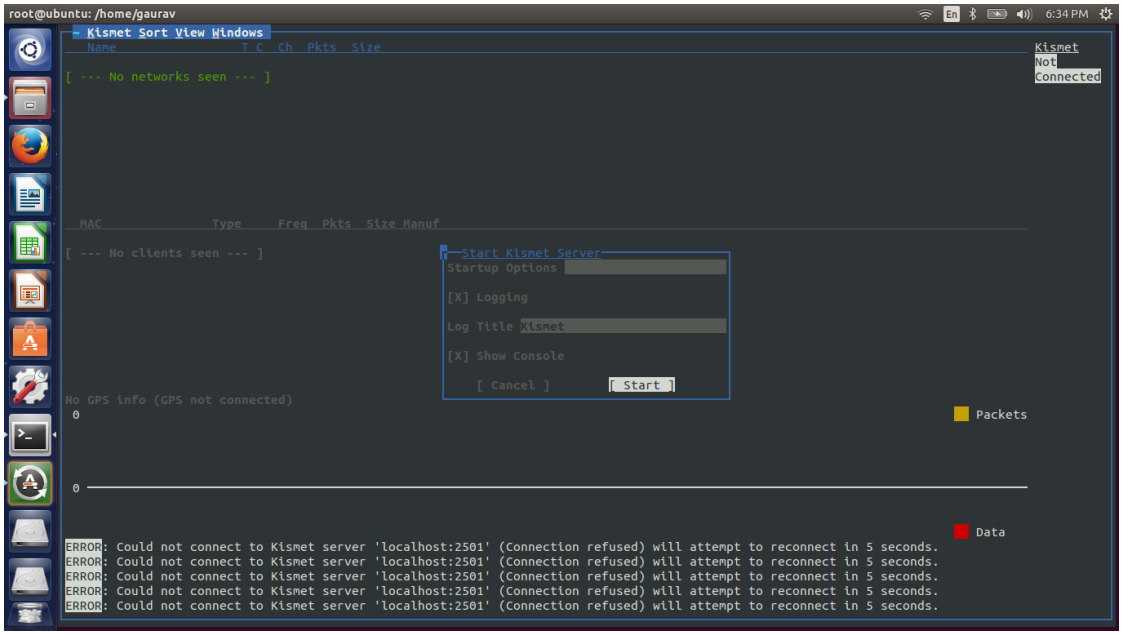

Figure A.19. Starting Kismet Server Options

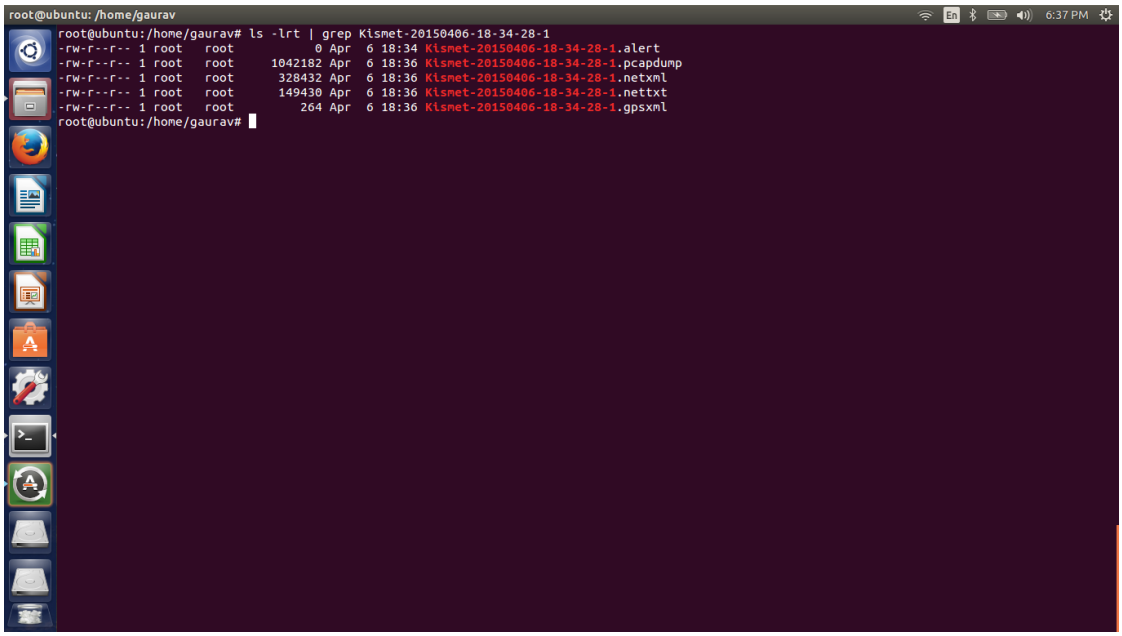

Figure A.20. Starting Kismet Server via Command Line

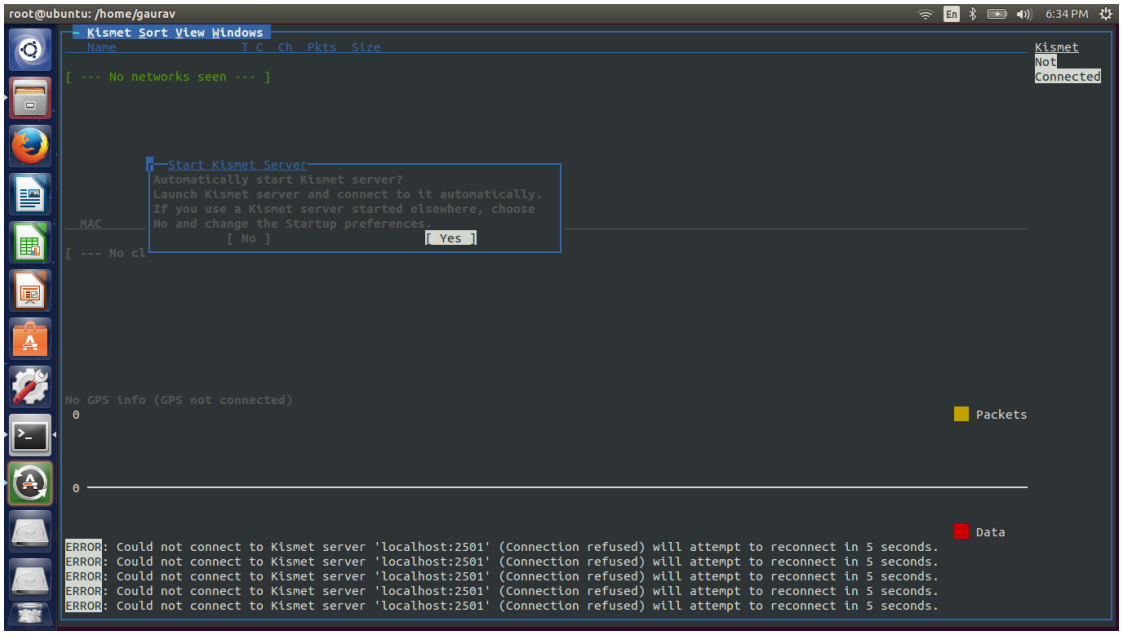

Figure A.21. Startup Preferences for Kismet Server

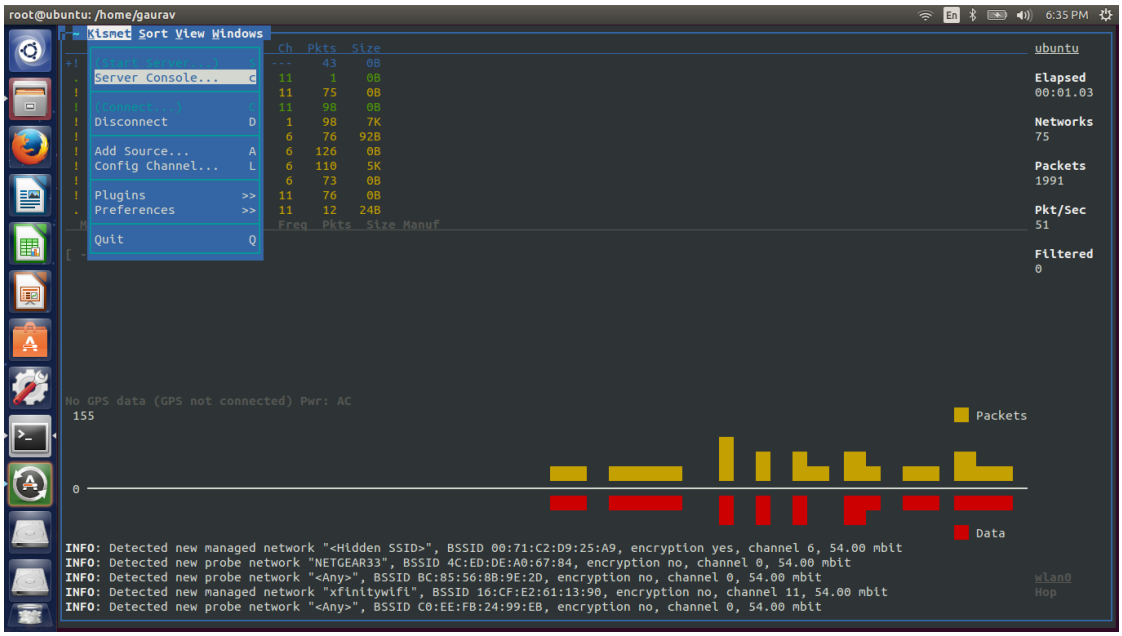

Figure A.22. Kismet Options Available

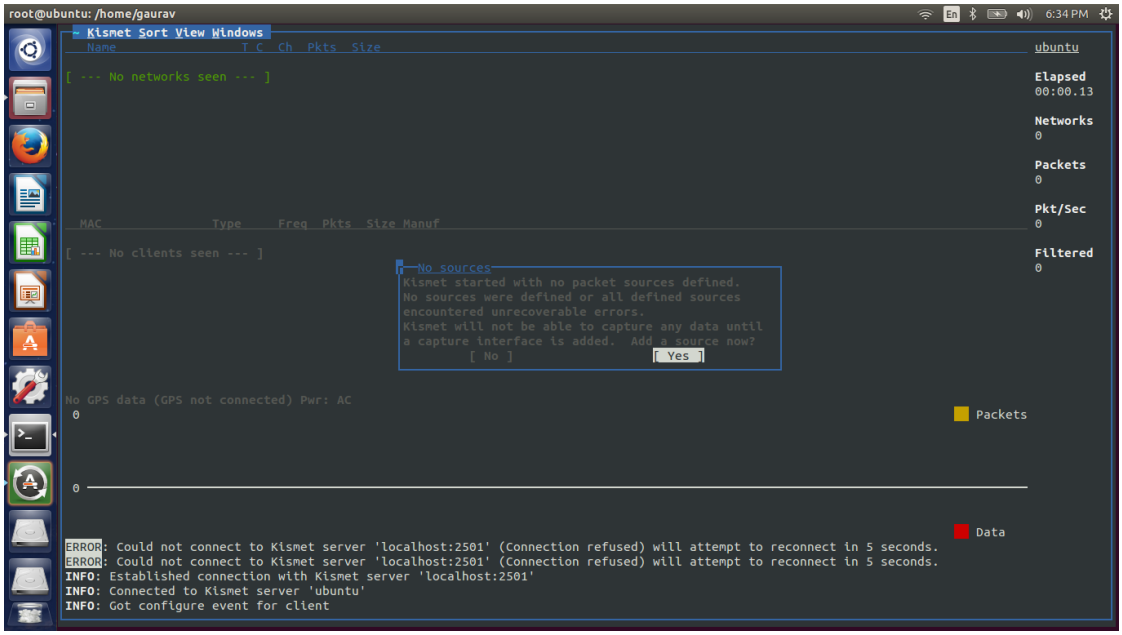

Figure A.23. No Sources are Defined for Capturing Packets

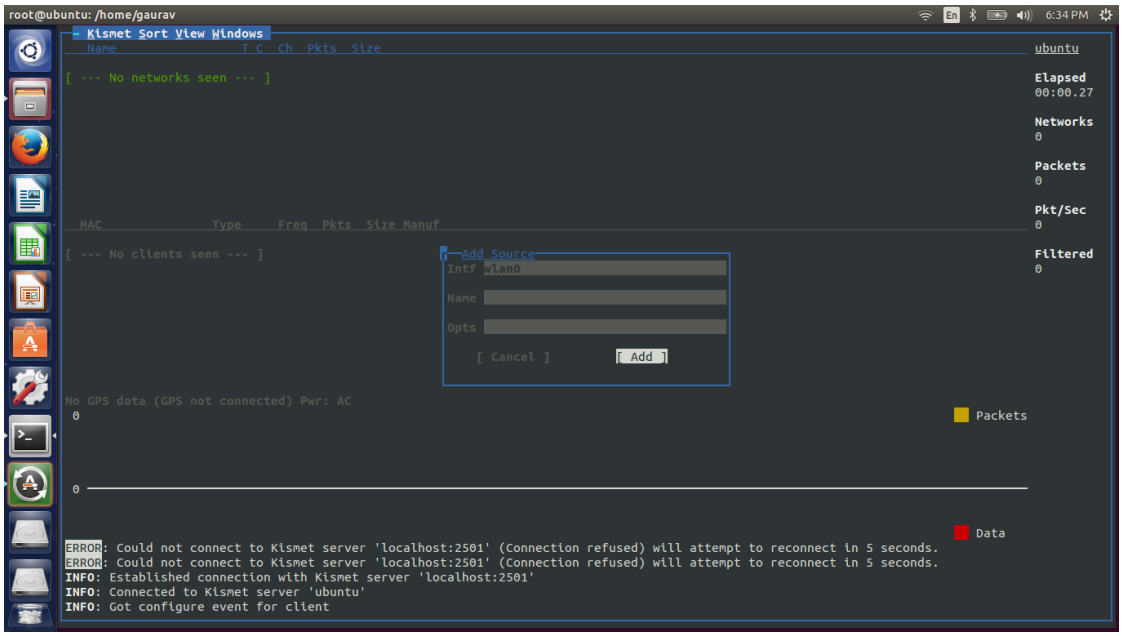

Figure A.24. Adding Source to Kismet Server

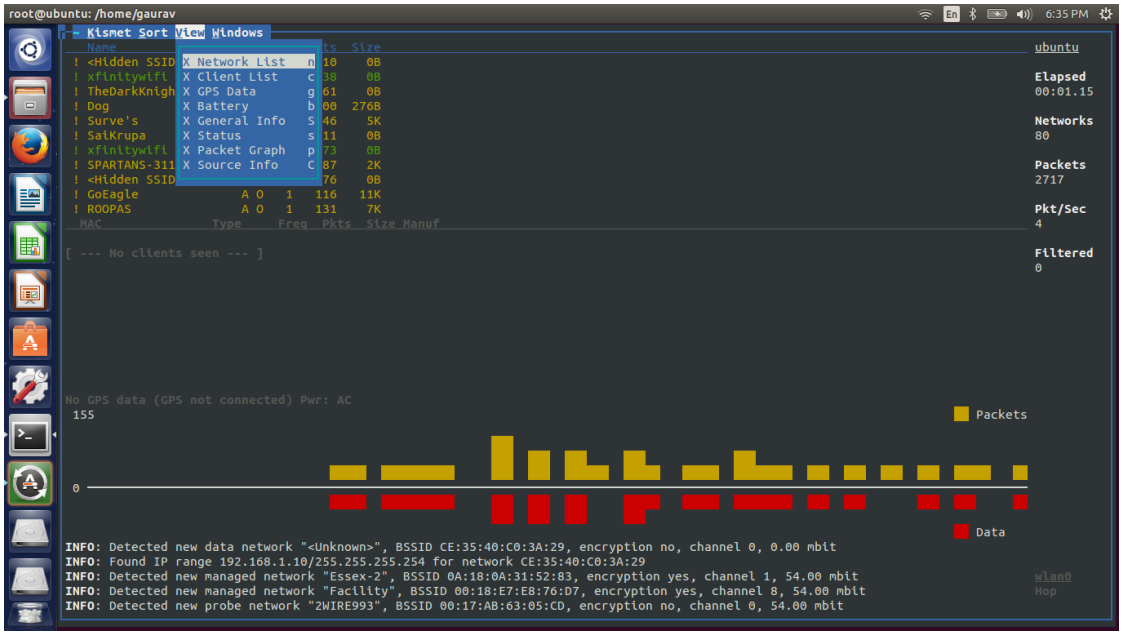

Figure A.25. Personalized View of Kismet Server

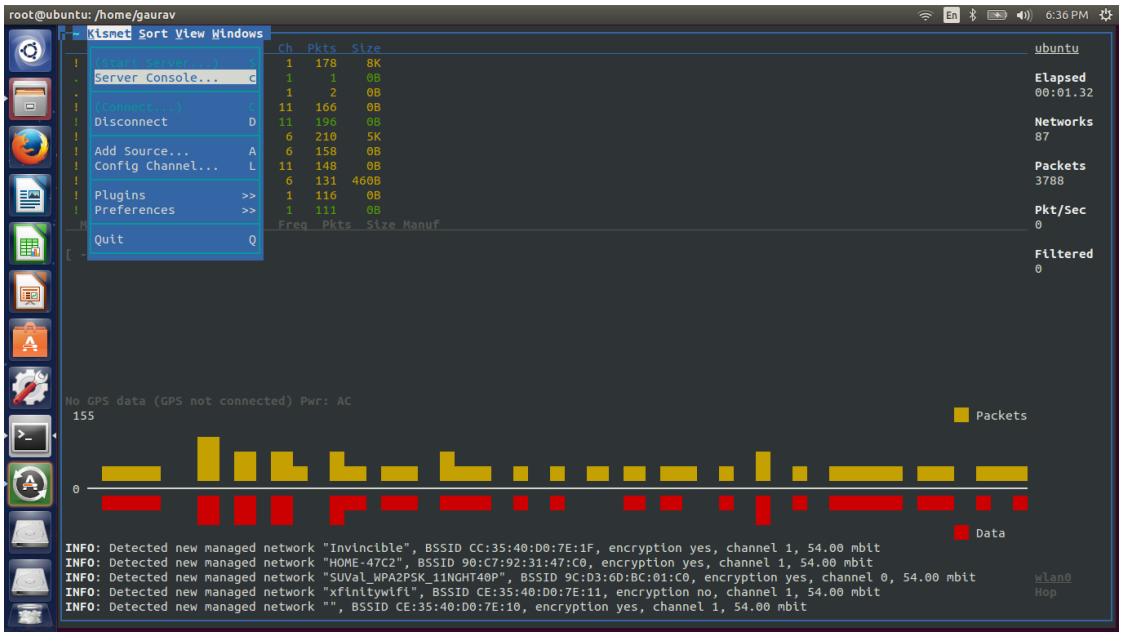

Figure A.26. Quit Kismet

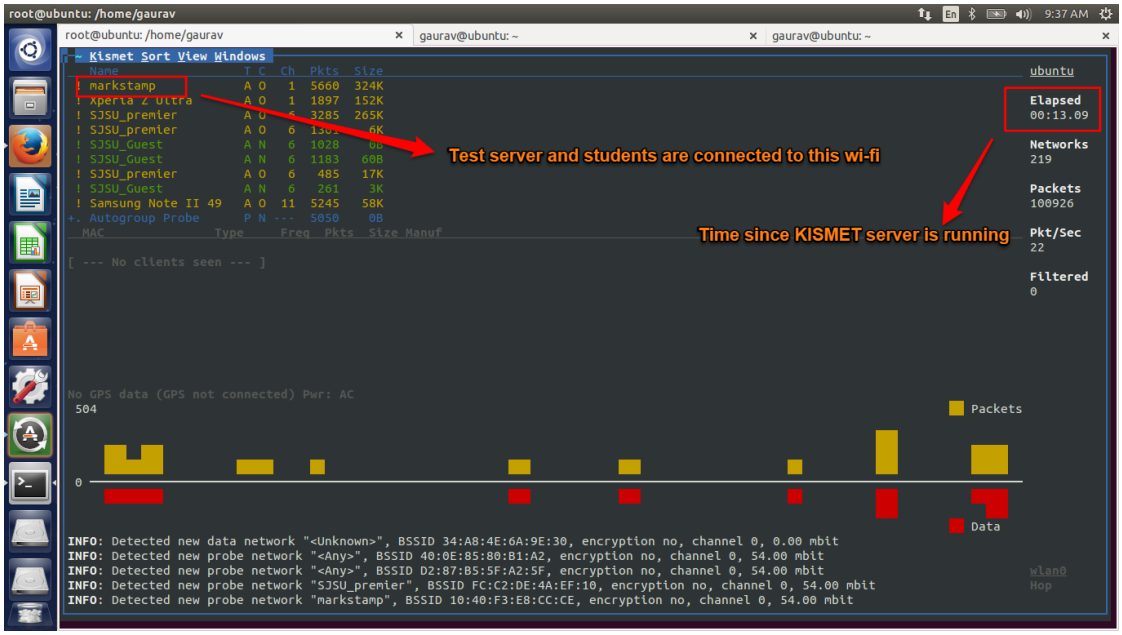

Figure A.27. WiFi and Running Time of Kismet

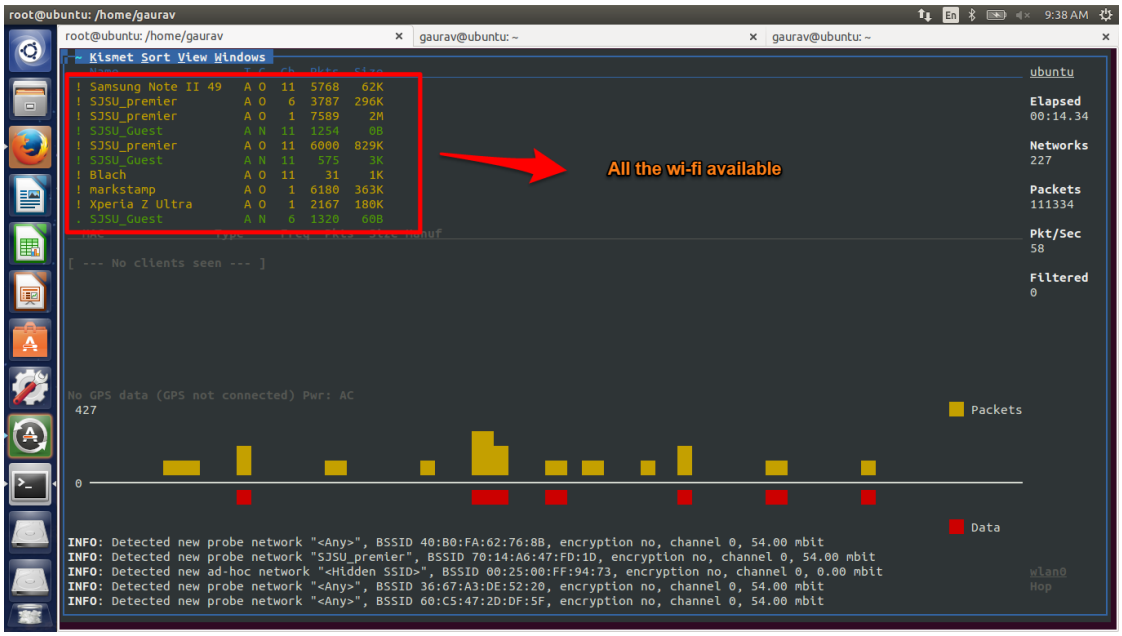

Figure A.28. List of WiFi Available

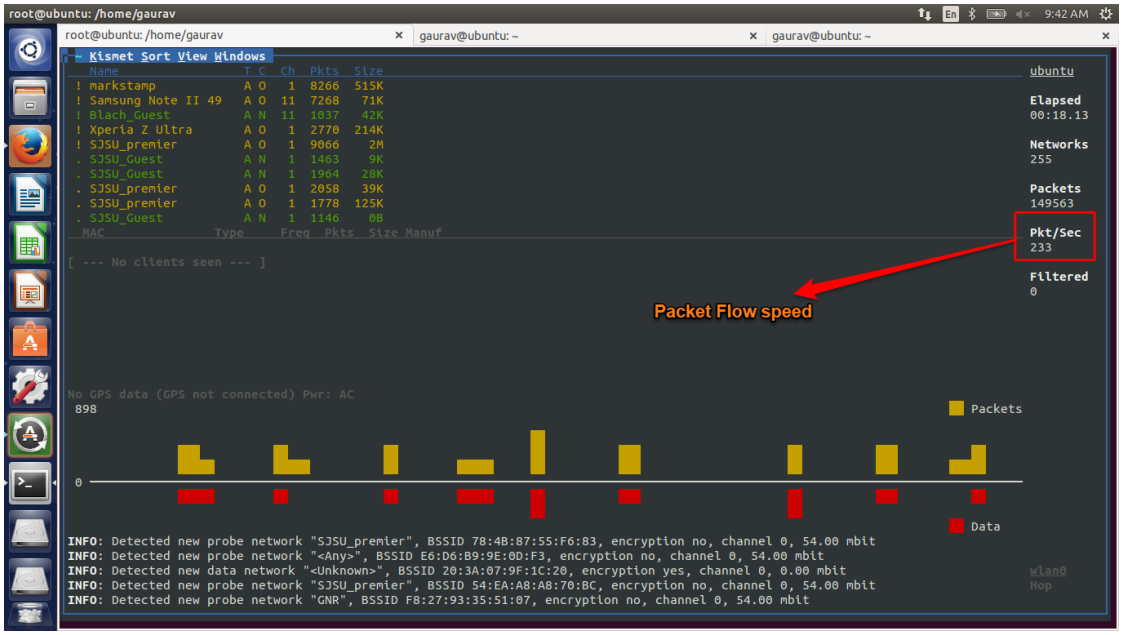

Figure A.29. Packet Flow Speed

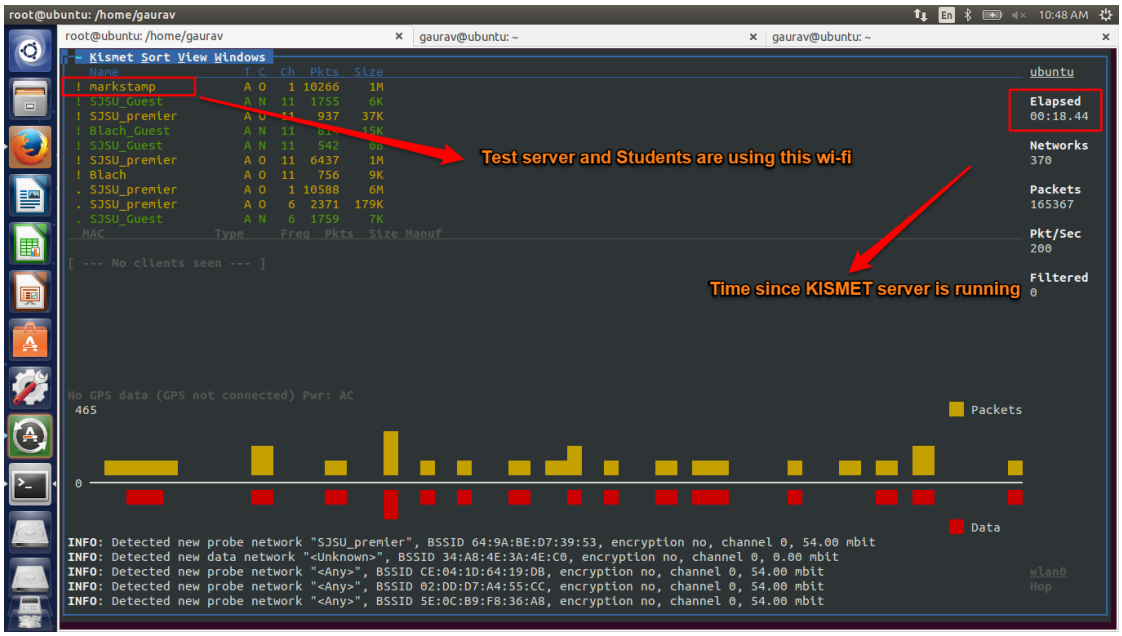

Figure A.30. Wifi and Running Time of Kismet

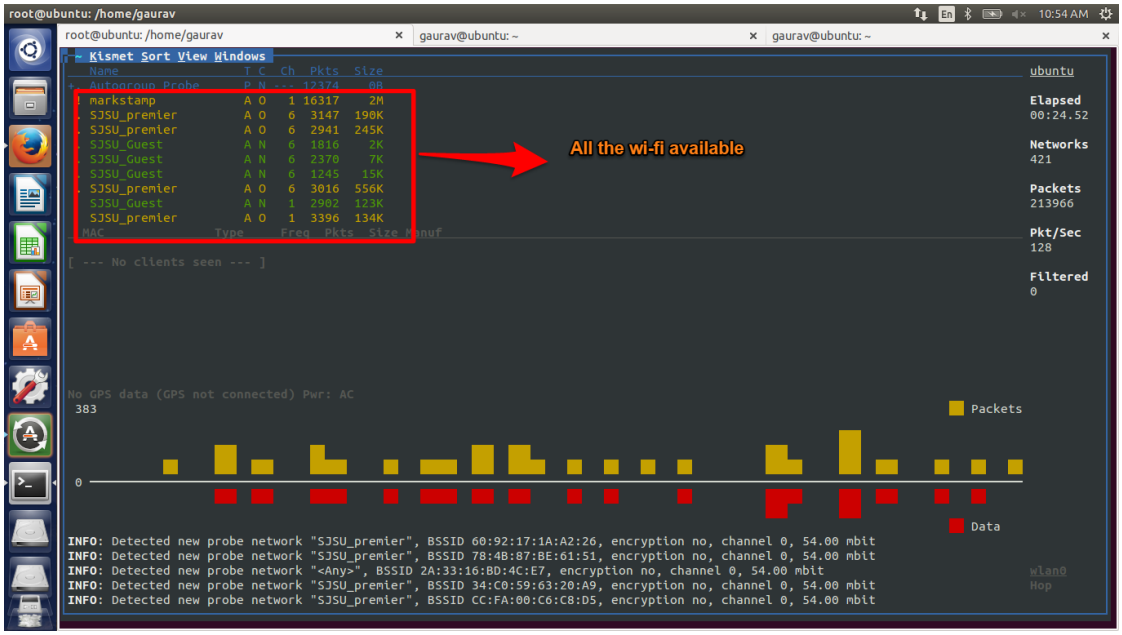

Figure A.31. List of WiFi Available

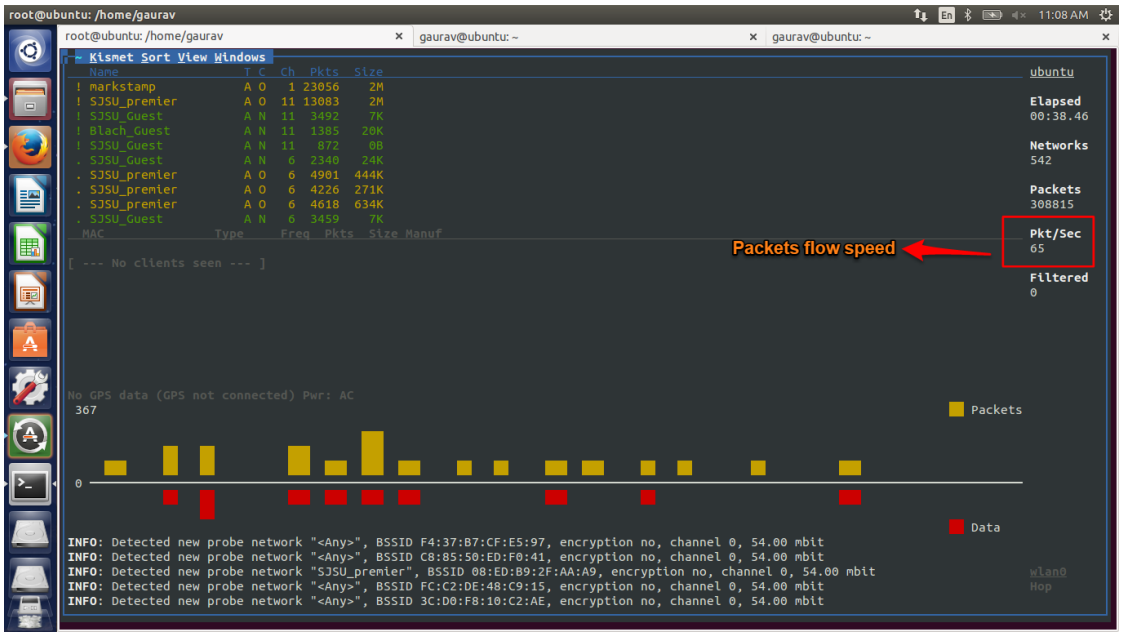

Figure A.32. Packet Flow Speed

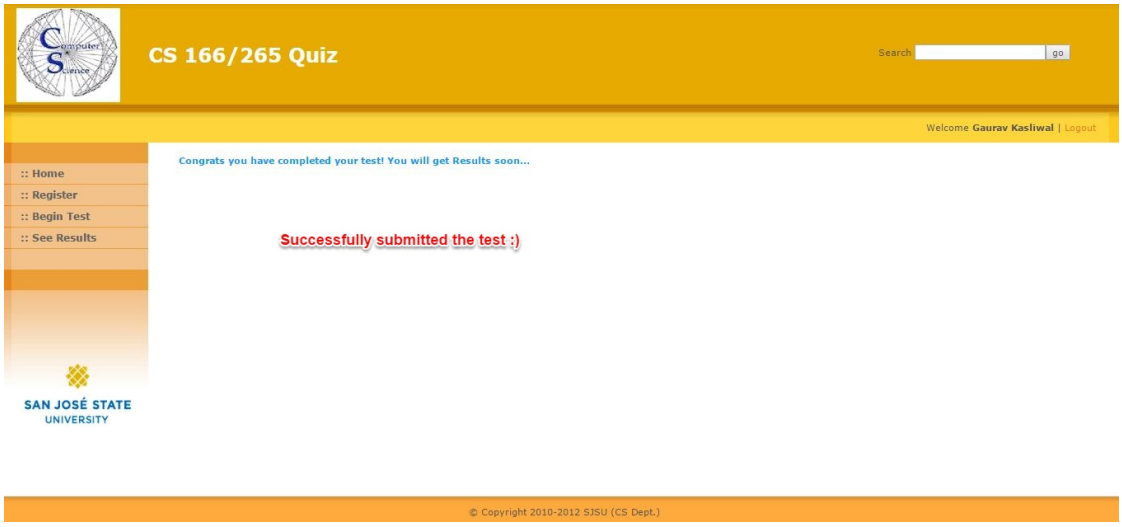

Figure A.33. On Submission of Test

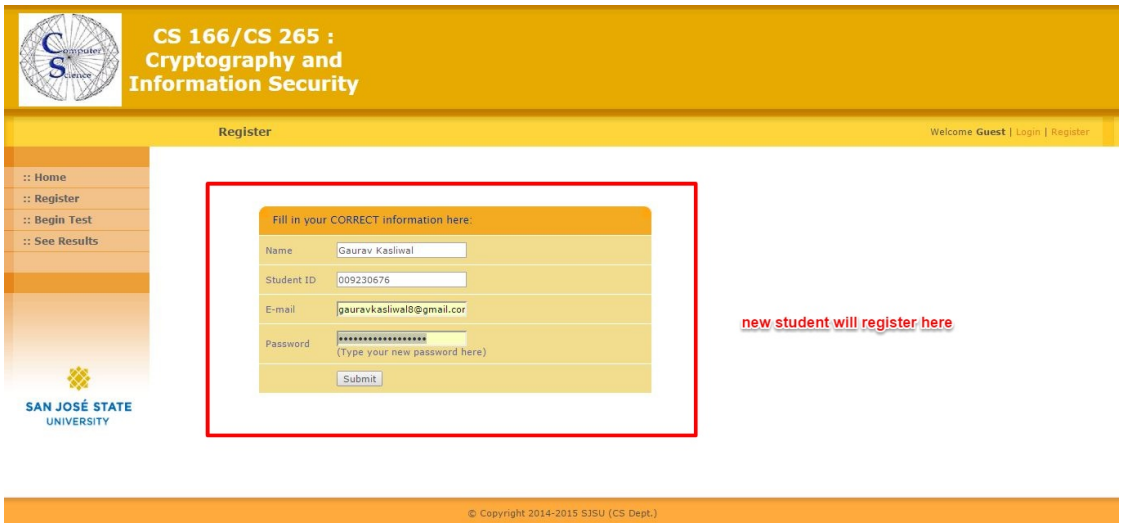

Figure A.34. Register for Test

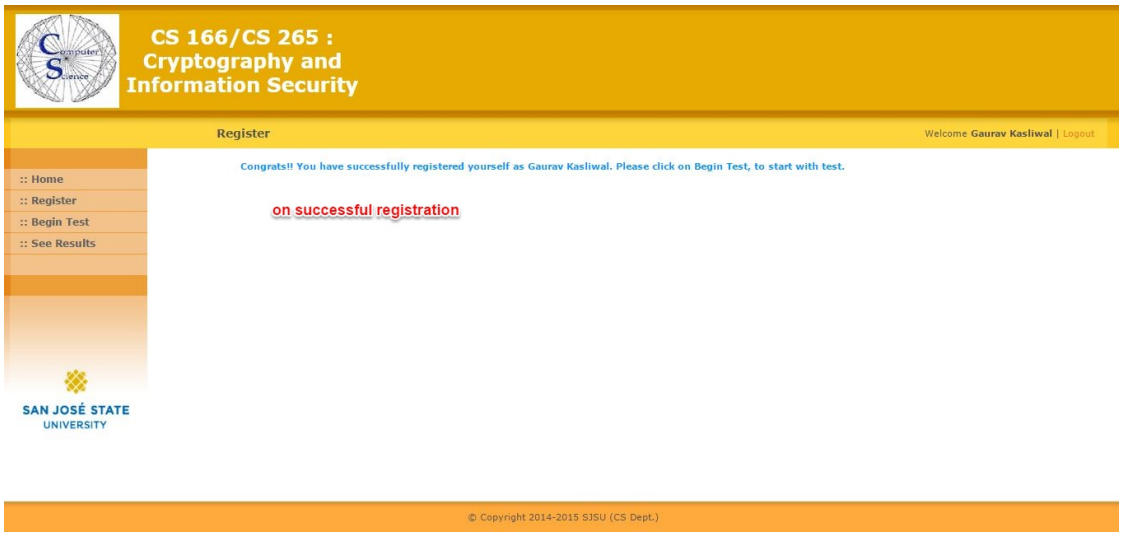

Figure A.35. Successful Registration

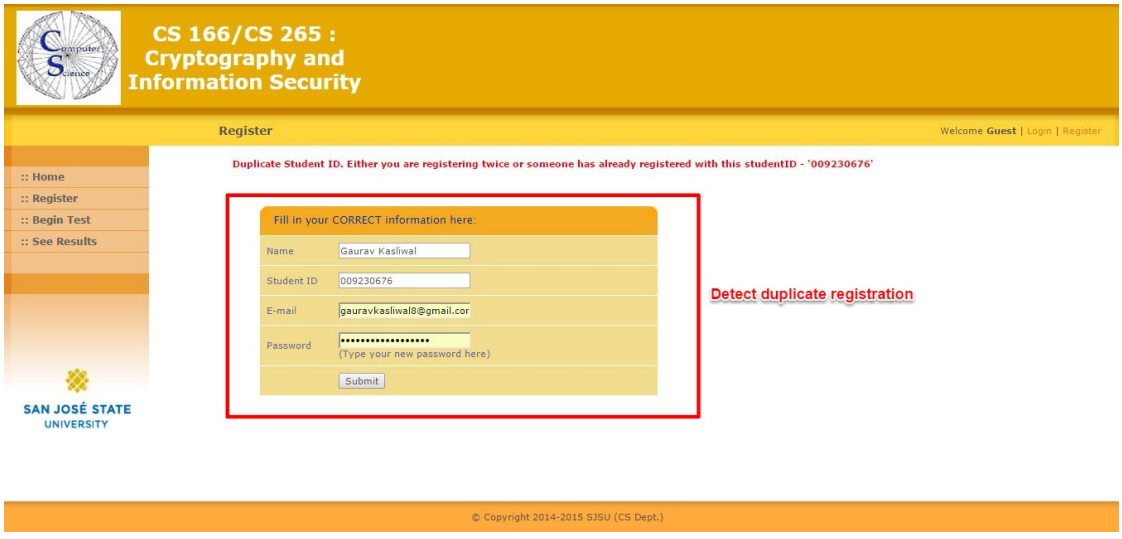

Figure A.36. Validating Registration

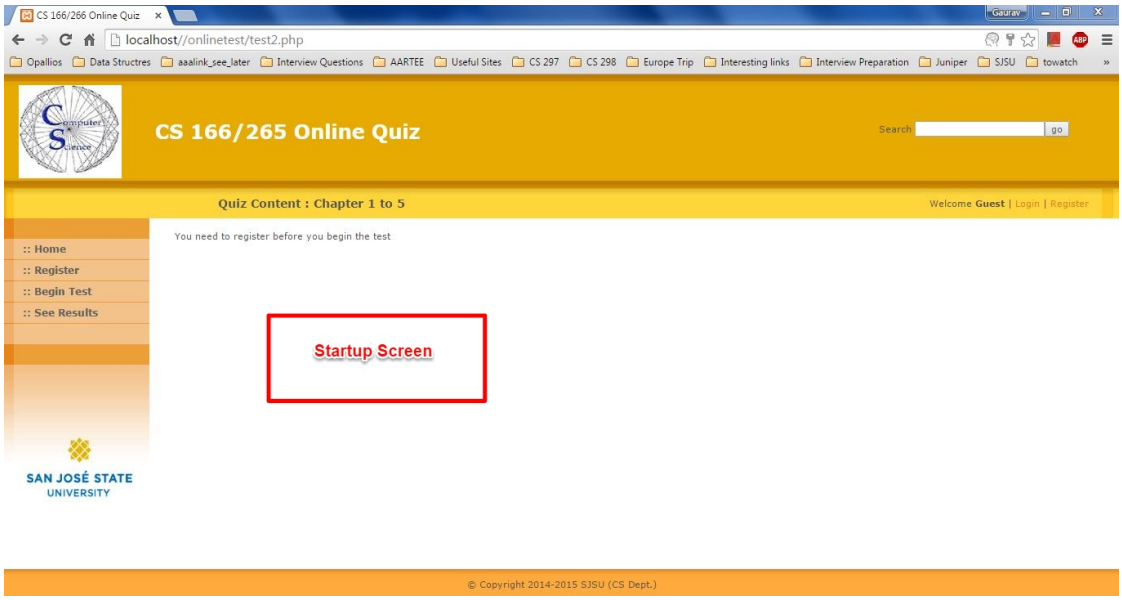

Figure A.37. Test Startup Screen

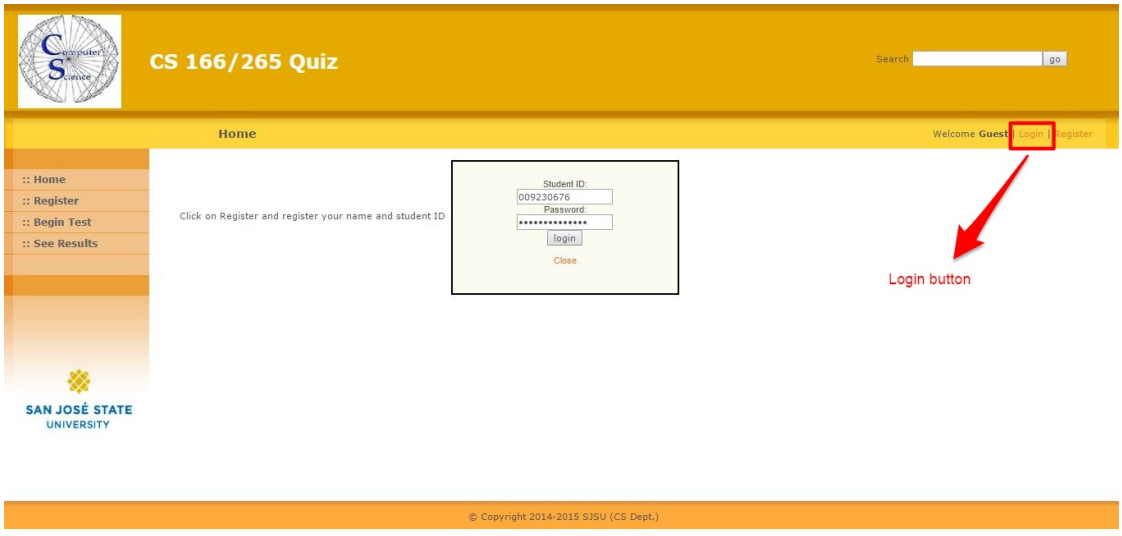

Figure A.38. Student Login