

A Big Data Architecture for Security Data and Its Application for Phishing Characterization

Neha Narkhede¹, Shweta Chaudhari¹, Manoj Nagthane¹, Himanshu Joshi¹, A. G. Dongre²

Student, Department of Computer Engineering, PVG's COET Pune, SPPU, India.¹

Guide, Department of Computer Engineering, PVG's COET Pune, SPPU, India²

ABSTRACT: As the internet grows cyber security problems also arise with it. Different malicious activities are being carried out by the attackers so that they will be able to get the information of the victim. Using this information the attackers perform their illegal activities. Enterprises usually collect terabytes of security-relevant data, including network traffic, and software application events, among others. However, well established techniques, most of the time, are not scalable and typically produce many false positives when dealing with large amounts of data, degrading their efficacy. To face these emerging problems, big data analytics has attracted the interest of the security community. The use of big data frameworks for security solutions presents several benefits, such as the possibility of storing and using large quantities of security data. In this paper we design an architecture in which being built on the top of the Big Data frameworks that aims to mitigate the cyber security problem like phishing. It is being designed such that we are able to detect the phishing emails in a large data set and the information collected by the honeypot.

KEYWORDS: Spam and Non Spam Emails, cyber security, phishing, spam.

I. INTRODUCTION

Use of internet grows so the cyber security problems also arise. Different activities are done by attacker to gain sensitive information of the victim. Different malicious activities are being carried out by the attackers so that they will be able to get the information of the victim. Using this information the attackers perform their illegal activities. After gaining the information attacker perform illegal activities. For this we are proposing the system and the main objective of this project is to detect phishing emails from collected data in Honeypot. In this project we design an architecture in which being built on the top of the Big Data frameworks that aims to mitigate the cyber security problem like phishing.

Use of internet grows so the cyber security problems also arise. The cost of cybercrimes includes the damage to company performance and to national economics caused by cyber- attacks, and the effects of information theft. Different activities are done by attacker to gain sensitive information of the victim. Different malicious activities are being carried out by the attackers so that they will be able to get the information of the victim. Using this information the attackers perform their illegal activities. After gaining the information attacker perform illegal activities. To face such attacks, we need to be able to identify and characterize them. A complex factor is the enormous increase in the volume of available data. Almost 90% of data in the world today were created in the last two years. Security issues become critical due

to factors such as the large volumes and variety of data. Enterprises usually collect terabytes of security relevant data. For this we are proposing the system and the main objective of this project is to detect phishing emails from collected data in Honeypot. In this project we design an architecture in which being built on the top of the Big Data frameworks that aims to mitigate the cyber security problem like phishing. However, well established techniques, most of the time, are not scalable and typically produce many false positives when dealing with large amounts of data, degrading their efficacy. To face these emerging problems, big data analytics has attracted the interest of the security community. The use of big data frameworks for security solutions has several benefits, such as the possibility of storing and using large quantities of security data. In general, the traditional infrastructure keeps the data only for a limited period. Besides that, traditional techniques are inefficient when performing analytics and

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

complex queries on large, unstructured datasets, while big data platforms perform these operations efficiently. In this paper we present architecture for cyber security applications based on big data frameworks. Our architecture has the capability of collecting data from different sources, storing, combining, and processing them effectively. We implemented an application to process large volume of spam traffic collected from four computers. For example, sources like pcap files and other logs from a honeypot, data streams collected from black list sites can all be stored in our system. Our application collects data from honeypots located in the slave's computer and stores the messages in the mailboxes on top of HDFS. We locate a honeypot on one of the computer which will be connected to all the computers we are using. The data collected from the honeypot along with the network traffic pcap files, blacklist sites data and a data set which contains emails is being stored on the top of HDFS. These are given as input to the architecture. The main contribution of the application is its capability to identify phishing emails in a set of spam messages. We use an algorithm which differences the legal and illegal emails.

II. IMPLEMENTATION

A software implementation is a systematically structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user.

A. SYSTEM ARCHITECTURE

We proposed architecture for cyber security systems and used it to implement a spam email detection application. The architecture must satisfy important properties like scalability as the performance must be scalable as the input grows efficiency as the system deals with security threats, uniform programmability as the system deals with multiple kinds of data from different sources. The architecture is depicted in Figure 1. It is composed by parts: (i) data collection, (ii) storage, (iii) processing, (iv) execution, (v) detection and separation.

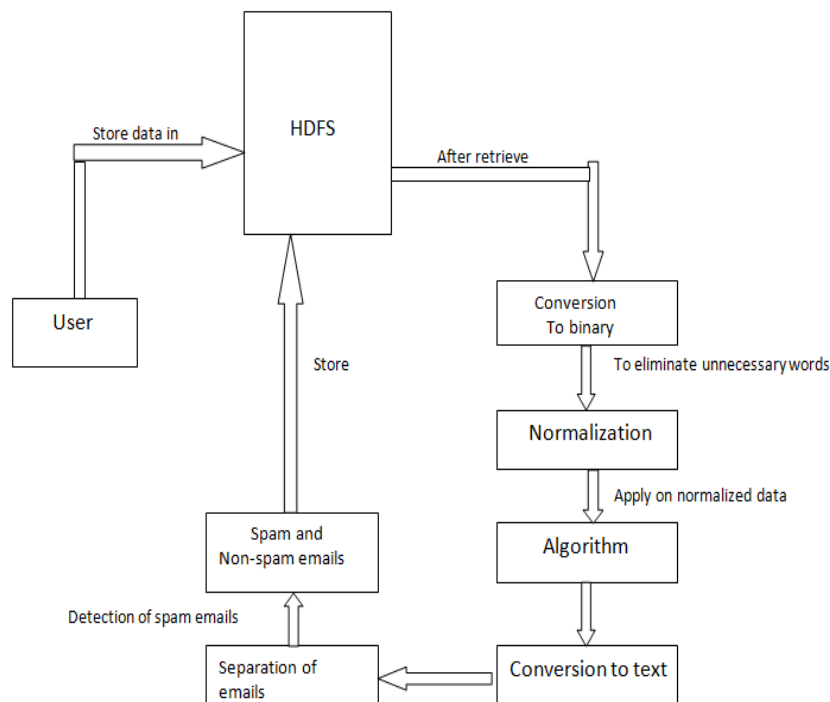


Figure 1. System Architecture

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Explanation-

(i) Data Collection

The first step of our architecture is the collection of data. The most important aspect of this step is to identify relevant data sources. Due to advances of techniques used by attackers, it is necessary to use distinct sources of data in order to mitigate cyber security threats. Our input sources are like pcap files and other logs from a honeypot, data streams collected from black list sites. In our system the goal is to analyze the spam emails so we used honeypot which will be installed on one computer which will be known as master. The other connected three computers will act as slave. For each connection of the honeypot we store all the content of each message and additional network information extracted at the time of connection from the slave computer. These all collected data will be stored in the mailboxes.

(ii) Storage

Every data source has different characteristics and is produced in a different volume and velocity, thus the data information must be considered in order to store the data in the best possible way. One of the most used data storage for storing a large amount of data is HDFS (Hadoop Distributed File System). It is reliable and scalable distributed file system and it provides high performance access to data. One important factor of HDFS is fault tolerance, it is obtained through replication. Replication means keeping multiple replicas of Each data block in different nodes. It is directly accessible from most Big Data processing frameworks.

(iii) Processing

Once the data is properly stored, we must enable the processing tool to process the input data. The input data is being processed by the MapReduce algorithm. The MapReduce algorithm contains two important tasks namely Map and reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Reduce task takes the output from the map as its input and combine those data tuples into smaller set of tuples. Map Reduce programs execute into three stages- map stage, shuffle stage and reduce stage. The mapper's job is to process the input data given to the system. The provided input data is in the form of file or directory and is stored in Hadoop File System (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data. The reduce stage is the combination of shuffle stage and the reduce stage. The job of reducer is to process the data that comes from the mapper. After processing, it produces a new set of output which will be stored in the HDFS. The data is then converted into binary in the form of 1 and 0 so it goes easy for the machine to read the data. As the main processing is to detect the spam emails, there are several words which are specific which are used by the phisher to attack to the victims system. These words are searched and the rest of the data which is not necessary is being eliminated. Thus the normalization process is being carried out on the input data to eliminate the non-necessary words. It just eliminates the general words like he, she, and, or, like etc. Thus after this step we now have only the required data which may be useful to detect the spam emails.

(iv) Execution

After eliminating the non-necessary words the required data is being available with us for the further processing. For execution of the application we used Naïve-Bayes algorithm. Naïve Bayes classifiers are a popular statistical technique for email filtering. Naïve Bayes classifiers work by correlating the use of tokens and then using Bayes theorem to calculate a probability that an email is or is not a spam. Bayes theorem is used several times in the context of spam: first time to compute the probability that the message is spam, knowing that a given word appears in this message, second time to compute the probability that the message is spam taking into consideration all of its words, sometimes a third time to deal with rare words. Popular words have particular probabilities of occurring in spam emails and legitimate emails.

(v) Detection and Separation

After the execution of the Naïve Bayes algorithm the emails which consists of spam detection words are detected and the data which is in binary format is converted in text format. After these the emails detected are separated from the non-spam emails and the spam emails are being stored in spam folders and all the emails are

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

again being stored on the top of HDFS. The result of our application is that it detects the spam emails and separate them from the non-spam emails. It should be able to detect the spam emails in less stipulated to time. The system should work and give result as per the accuracy.

B. ALGORITHM

a) Naïve Bayes Algorithm:

The formula used by the software to determine that is derived by,

$$\Pr(S|W) = \frac{\Pr(S|W) \cdot \Pr(S)}{\Pr(S|W) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)}$$

Where:

Pr (S|W) – is the probability that a message is a spam, knowing that the word is in it. Pr(S) – is the overall probability that any given message is spam.

Pr (W|S) – is the probability that the word appears in spam messages.

Pr (H) - is the overall probability that any given message is no spam.

Pr (W|H) – is the probability that the word appears in ham messages.

b) Algorithm for Spam detection

Input: user input data from email sender, spammer training dataset Ti, spammer list L

Output: data classification as spammer, hammer and normal

Step 1: User sends a mail with some random data

Step 2: receive by receiver

Step 3: for each record from $Rec = \sum_{k=0}^m \binom{m}{k} Rec^k$

Step 4: for each spammer list (l from Li) event++;

Step 5: if (cur_Sender.equals(l)) or if(cur_Sender.equals(k)) spammer++;

Step 6: initialize threshold T =5.0

Step 7: if(event<=T)

Return jammer;

Step 8: else if(event>T)

Return spammer;

Step 9: else

Return Normal;

Step 10: step 3 and 4 when reach null;

III. RESULTS

The overall snapshot of this project is given below:

HOME PAGE

When you enter our site you can see this page as a homepage from which you can go on Login if account is already created else user can go for registration.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

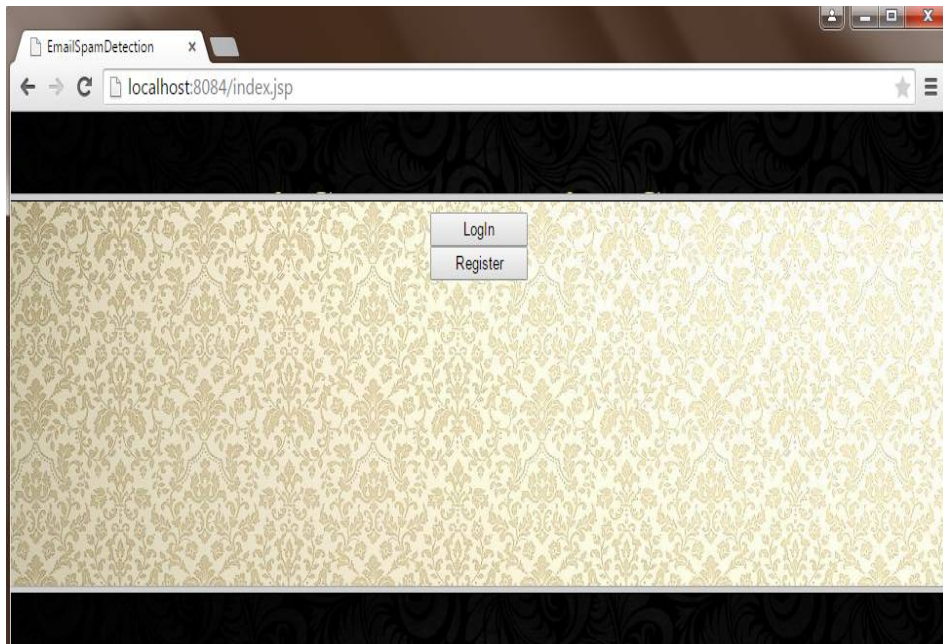


Figure2. HOMEPAGE

USER-LOGIN INTERFACE

Below Fig shows that project Login Interface from which each user access accounts and also create new accounts.

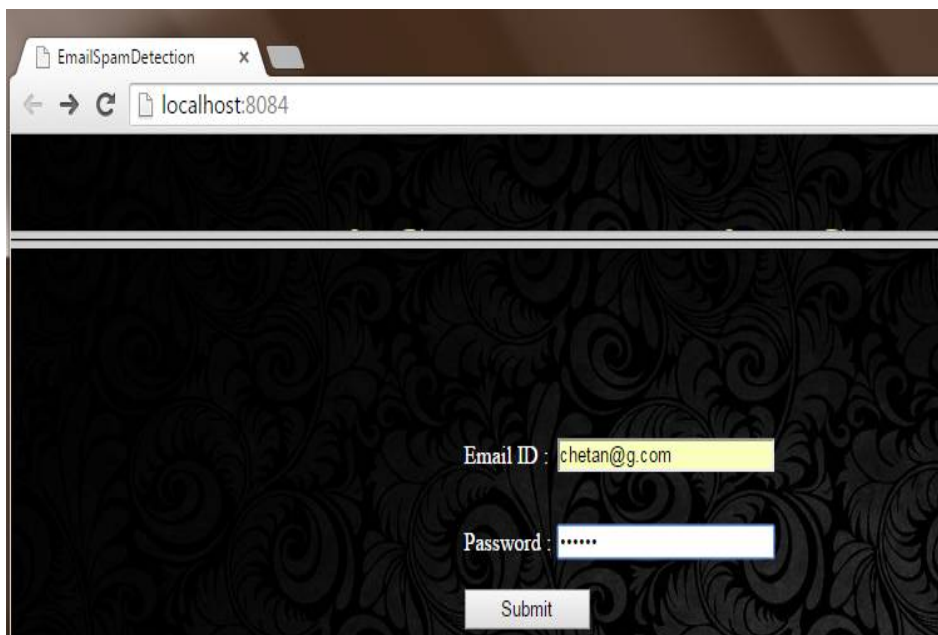


Figure 3. USER LOGIN PAGE

Figure 3 represents the page of user to log on to master server. Unique credentials are maintained for the master server and this user is solely responsible to monitor and configure the network statistics. There are two fields containing User

International Journal of Innovative Research in Science, Engineering and Technology

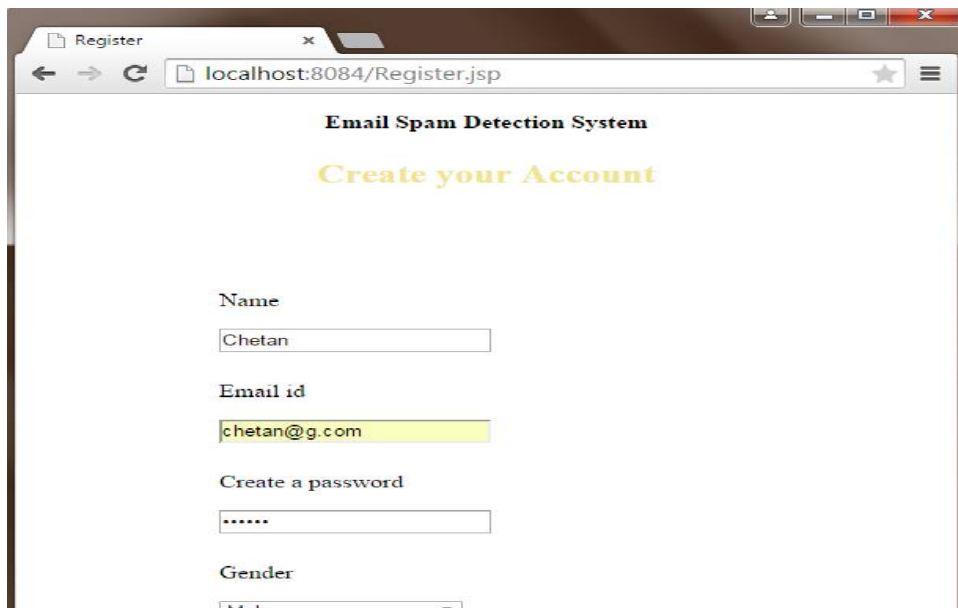
(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

ID and password.

USER REGISTRATION



Register

localhost:8084/Register.jsp

Email Spam Detection System

Create your Account

Name
Chetan

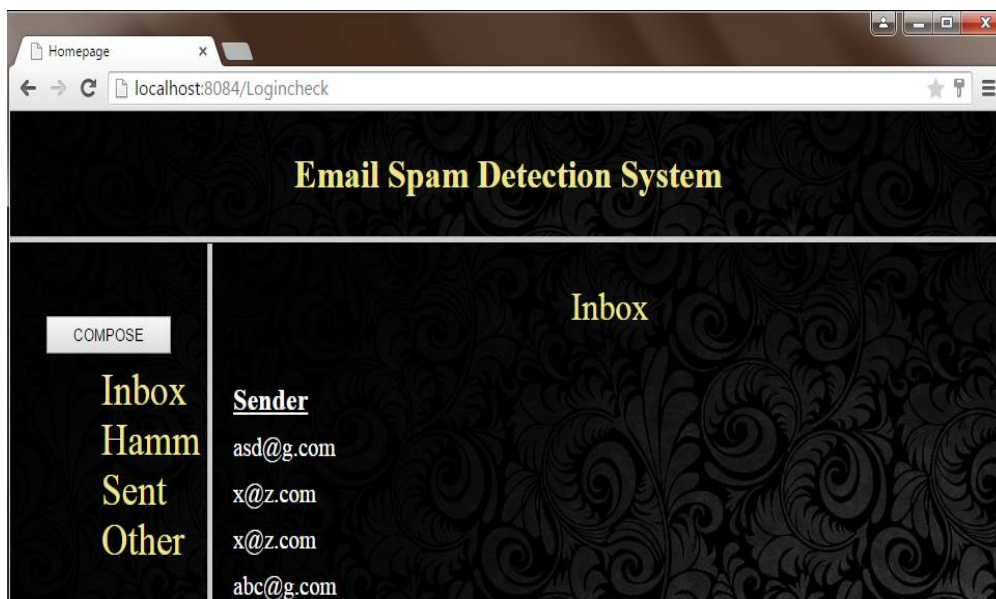
Email id
chetan@g.com

Create a password

Gender
Male

Figure 4. USER REGISTRATION

INBOX



Homepage

localhost:8084/Logincheck

Email Spam Detection System

Inbox

COMPOSE

Inbox

Hamm

Sent

Other

Sender

asd@g.com

x@z.com

x@z.com

abc@g.com

Figure 5. INBOX

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

COMPOSE MAIL

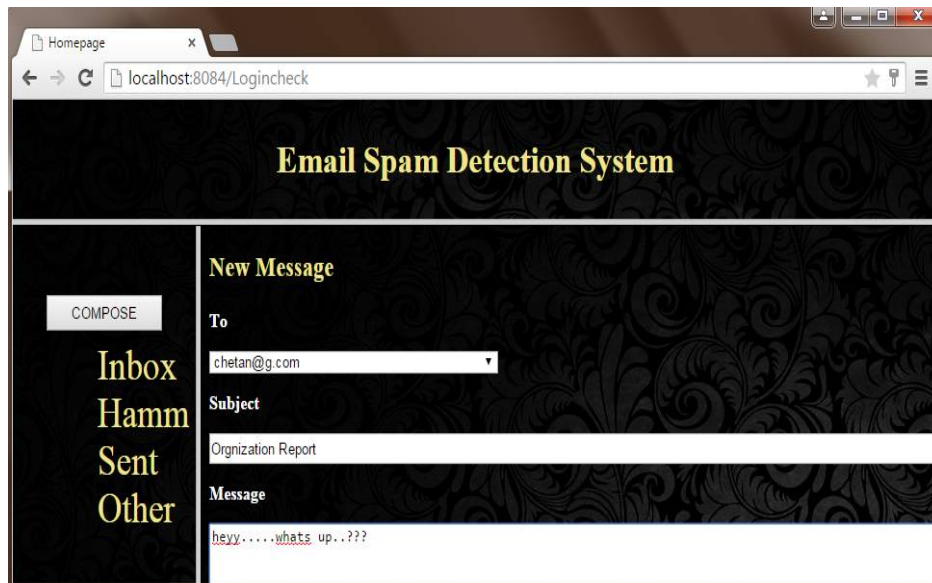


Figure 6. COMPOSE MAIL

IV. CONCLUSION

The proliferation of data sources and data collecting structures has led to a large increase in the data available for cyber security experts. To process such large volumes of data, scalable massive data processing solutions are needed. As mentioned in literature survey, the present work on the project uses the LSH algorithm which detects the spam emails but it gives accuracy of 98.1%. The system complexity is also high. Our system will reduce the complexity along with that the accuracy increases and the spam emails will be detected successfully in less time.

REFERENCES

- [1] Kotsiantis, S.B., Zaharakis, I.D., Pintelas, P.E.: Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 2006
- [2] Y. Yu, Y. Mu, and G. Ateniese, "Recent advances in security and privacy in big data," *j- jucs*, Mar 2015.
- [3] P. H. B. Las-Casas, V. Santos Dias, R. Ferreira, W. Meira, and D. Guedes, "A hadoop extension to process mail folders and its application to a spam dataset," in *International Symposium on Computer Architecture and High Performance Computing Workshop (SBAC- PADW)*, Oct 2014, pp. 108–113.
- [4] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security & Privacy*, 2013.
- [5] P. H. B. Las-Casas, V. Santos Dias, R. Ferreira, W. Meira, and D. Guedes, "A hadoop extension to process mail folders and its application to a spam dataset," in *International Symposium on Computer Architecture and High Performance Computing Workshop (SBAC- PADW)*, Oct 2014, pp. 108–113.