Design a Blacklist based Cloud E-mail Spam Filtering Service

Md. Mahmudul Hasan, Md. Al-Amin, Md. Niaz Imtiaz

Lecturer, Department of Computer Science and Engineering, Pabna University of Science and Technology, Bangladesh

Email: mahmudul.cse@pust.ac.bd

Abstract— E-mail service is one of the popular inventions in modern communication and technology. Users from every corner of the world are now connected via email services for sending messages, news and sharing confidential information. Emailing to friends and family, national and internationals, official and non-official purposes, connecting to other services – everywhere we use email services. Spamming is one of the major threats in digital world that causes a millions of technology forwarded users be suffered and felt untrusted. Users are not aware of spam; in most cases they have no idea of the pattern of spam as they appear as a normal text, links, images or could be anything attractive. Exploring those items leads to an instant lose of online identity and property before they can even guess about it. In this paper, we aim to identify spam characteristics, consolidate them and build a cloud based spam filtering service that act as a firewall to email messages and easily incorporated into any email service client.

Keywords— spam filter; blacklisting; cloud email; cloud API; spam properties; email client; cloud service.

INTRODUCTION

In modern life, the dependency on internet is increasing day by day. The majority of the people communicate with each other, store and share personal information in verities of online sites over the internet. Not every person is very technical; they know very little about internet technologies and very often they are not aware of what they are doing. As a result, people are experiencing unethical incidents around them. People with more emotion are suffering much more in the network than other groups [1]. Spamming is one kind of such deceptions that looks lucrative, plays with human minds and sympathy, and raise serious threats to the user identity and resources. Social networking sites such as Facebook, MySpace, Twitter and particularly E-mail services are consistently among the top 20 most-viewed web sites are attracted by spam [2]. In 2006, more than 70% of the total messages that are sent over the internet are spam [3]. In the end of 2016, though it has reduced in a reasonable amount, still spamming in email happens more than 60% of times (in Fig. 1) [4]. When user accesses the spam contents, they may lose their id, password and private information that could lead a serious damage to their private property. Therefore, detection and removal of spam content from the network is a serious matter of concern.

People around the world are taking huge benefits from various cloud services including E-mail service. It has become the essential part of our personal communication and business [5]. Users are more encouraged to store, share and communicate via Email because of the almost free of cost, high availability, reliability and high performance of the services. A typical example of Email service providers are Google, Yahoo, AOL, Hotmail and so on which provides millions of terabytes storage and effective spam filtering services with free of cost. However, these services are public and often not appropriate for personal business. Business organizations tend to use their own organizational E-mail ID from the hosting company. In this situation, the business owners are not interested to invest lots of money to implement their own spam filter service and eventually express their attention towards an external spam detection and removal agency that can take care of the spamming. In this paper, we aim to build a cloud based spam filtering service (SFS) and an Email client (EMC) that consumes the SFS in real time. A basic pricing model is also considered in SFS for clients like EMC.

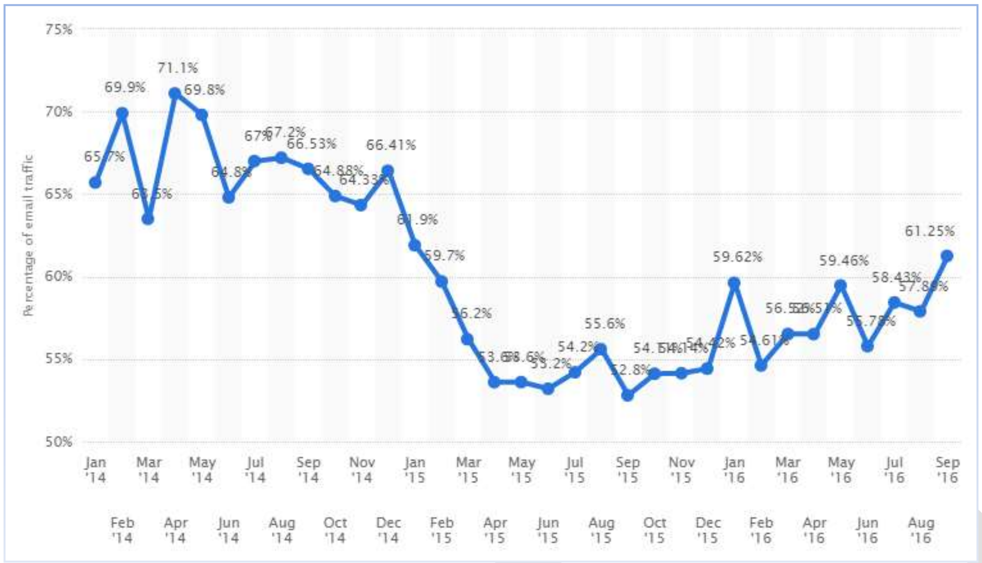


Fig. 1- Statista Spam statistics: spam e-mail traffic share 2016

In the remainder of this paper, at first we conduct a comprehensive survey on the characteristics of spam and email services. Then we investigate potential attack behaviors in Email server for understanding spam. After that we review existing solutions for removing identified behaviors and potential attacks. Later, we describe our proposed system architecture followed by the system implementation and performance measurement. Finally, we highlight the limitations of the proposed system and recommend various aspects to the future researchers.

SPAMMING IN EMAIL

Number of users in E-mail services is rapidly increasing day by day. More and more people store their information and communicate with one another by E-mail service. When benefits of E-mail service are widespread, unwanted messages are appearing problematic; often it contains unwanted advertisement messages. These unwanted messages are called spam [6]. In other words, spam is usually considered to be electronic junk mail or junk newsgroup postings. Some people define spam even more generally as unsolicited email. We often receive unwanted information from a variety of electronic systems mainly through emails, electronic boards and messengers, called spam [7].

Spamming attacks

Likejacking attacks

This attack refers to the action carried by the victim and initiated by the attackers. The idea behind this attack is to attract targeted users using interesting posts that motivate users to do some action on it. This often happens in advertising various products with offer. Users who click the links becomes the collaborator to the attacker inadvertently because the malicious scripts automatically re-posts the links, images or videos on their contacts' premise. A more popular version of this attack causes user profiles to "Like" a Facebook page without their consent [8]. Another popular example is an attacker in a web page designs overlay UI elements of like button on offer pages that says 'claim your free iPad' to attract users claiming the iPad. When user clicks to claim they are unintentionally click on like button and hence their friends and contacts see the attacker's page and they do the same mistakes [9].

Rogue application

Facebook allows developer to build applications for users to play games, to add custom profile settings and to do more stuff. It is an open platform for everyone to develop and submit applications for potential users. Cybercriminals take this advantage and create rogue applications that may be used for spamming post [8].

Attacks via Facebook chat

This is another media where spammer posts links to user message window with attachments or external links that either cause harms in their local machine if downloaded or they are taken to the attackers' page that exactly looks like the social media pages. Most of the time people are unaware of the web site URLs in the browser and provide sensitive information to the attackers [8].

Spammed tweets

Cybercriminals are enough smart to analyze alternative ways to post unethical but interesting posts to social networks such as Twitter. The shortened hashing technique for categorizing post types and text limit in posting cannot avoid the spammers their posts are short but convincing enough that the user follows the links that spammers posted. Free vouchers, job advertisement posts and testimonials for effective weight loss products are popular examples [8].

Properties of Spam in E-mail

According to the statistics [4], E-mail has been the primary area for the spammers to conduct their operation. We investigate various sources and consolidate common properties of spam.

Hidden recipient address

It is often seen that the recipient address in email signature is empty. Spammers do this in order to hide the fact that the mail was sent to a mass number of recipients, and most likely not to publish their email list to the original recipient [10].

Message contains many tags

Some spammers use lots of HTML comment tags to avoid content filters within the email body text. In this way, content filters unable to differentiate comments tags and spam words and so as the recipients [10].

HTML body in message

Many spammers lean to prepare HTML messages without the plain text body part and send that to recipients email. While reading the HTML version of the message, the images and other contents are visible automatically and allows user to click the contents for spreading spam [10].

Remote images in message

In order to avoid spam messages from being blocked by word filters, spammers include an image in their message that cannot be filtered for words. In addition, upon opening the email message the image is downloaded from the spammer's website. Since each message contains a unique ID, the spammer will know exactly which recipient has viewed the mail [10].

Different addresses in From and Reply-to field

This is a common property of spam emails. This kind of emails often comes from legitimate email addresses that are valid and never marked as spam. For example, an attractive email such as 'US DV Lottery Won' has arrived to recipient email address. People without reviewing the sender email address share their identity to the spammers. In such many cases, the sender names are written as the original name relevant to the subject but the email address is different that actually does not bear any identity of the sender or senders organization [10].

Number or character sequence in sender's address

In some cases we observe that sender email addresses contain numerical name or a sequence of unusual characters that does not resemble the usual human name or organizational syntax. For example, TerUndUcV@hotmail.com or son9348534@gmail.com both are valid email addresses. However, they do not bear any meaningful context rather it looks an automated script generated name [11].

Message body is based64 encoded

One encoding procedure, base64 is used sometimes by spammer to encode the message headers and body so that spam filters are unable to read the spam content and perform any filtering [11].

Invalid, repetition or empty data in address locations

Sometimes we see that the send and recipients have the same email address, sometime the recipient field is empty and sometimes there is email name only without the '@' sign. This resembles that the email was sent from an automated program and can be assumed that to be circulated for spamming [10].

Profitable offers

Another way of spamming is to send profitable offers to targeted users. Sharing assets from bank accounts, millions of dollar wining from an email draw etc has now become regular and greedy users are sharing their identity unconsciously without realizing that they cannot be that lucky.

In a survey, Red Earth Software analyzed the headers of 500 spam messages and found that the following spam characteristics were mostly found by their ratios [10]. Table 1 shows the kinds of spam behavior happens more and less frequently.

Table 1 - Percentage of Emails found based on kinds of Spam Characteristics

Spam characteristics	**% of found E-mails**
Recipient address not in **To:** or **Cc:** field	64%
To: field is missing	34%
To: field contains invalid email address	20%
No message ID	20%
Suspect message ID	20%
Cc: field contains more than 15 recipients	17%
From: is the same as the **To:** field	6%
Cc: field contains more between 5-15 recipients	3%
To: field contains more between 5-15 recipients	2%
To: field contains more than 15 recipients	1%
Bcc: field exists	0%
To: field is empty	0%
From: is blank or missing	0%

Anti-Spam Algorithms

Several studies have been conducted in detecting and removing spam from the network. Majority of the works have been related to text based filtering as contents are written in text very often. For example, emailing, posting data to social networks; URLs of web pages are also texts. We investigate available spam detection techniques are described following.

1. Naive Bayes spam filtering technique

Naive Bayes is popular statistical technique for e-mail spam filtering. It is used to identify features of spam e-mail. The technique is developed on stack of Bayesian classifier [12] which has the nature 'naive' that means it simplifies the computation involve in a particular classifying process. Bayesian classifier is based on Bayes' theorem. Naive Bayesian classifiers follows the effect of an attribute value on a given class is independent of the values of the other attributes. In spam detection technique, a sequence of words is considered in the place of attributes in Naive Bayes classifier. Naive Bayes classifiers try to correlate the sequence of tokens or words for separating spam and non-spam e-mails and detect an email is spam or not based on the probability calculated from Bayesian inference [13].

The main advantage of Bayesian spam filtering is that it can be trained on a per-user basis and constantly self adapting. It’s simple to implement and its false positive or negative rate is low. It has drawback too. It requires large number of data that need to be trained properly in order to make it work at their most effective outcome. The training leads more time.

2. Blacklist

Blacklist is the form of rule based filtering that uses one rule to decide which emails are spams. It involves lists of various kinds of information that is used to check the incoming content in a network. Some of the blacklist text types and sample values are presented in Table 2. Blacklist can be used for on both large scale and small scales spam filtering application. The main advantage of this technique is that it can block substantial amount of email though it blocks a range of blacklist items instead of individual item.

Table 2 – Sample blacklist items with categories

Blacklist Category	**Sample values**
gambling	fun, free, online, gold, gratuits, mobile, Samsung, casinos.
Hosting	managed, vps, web, joomla, dedicated, reseller, windows, linux, a2, wordpress
Injection	%url%, %BLOGTITLE%
Medication	viagra, cialis, propecia, Zoloft, clomid,
Merchandise	air max, air Jordan, Nike Huarache, jimmy , choo, jordan shoes, dre beats
Miscellaneous	#https?:\/\/[^\/]+\.pl
Polish Text	Strona, Warszawa, koszenia, ginekolog, pogrzebowe, notariusze, krakow
Services	Alternatefuel, online dating, reverse phone, lookup, cash advance, online divorce
Software	eMule, product key, cdkey, key sale, upgrade key
Trading	Anyoption, Optionbit, Optionfair, iOption, Onetwotrade
URLs	free-casino-bonus.com, womensclothescheap.com, familiekock.nl

3. Greylists

A relatively new spam filtering technique, Greylist takes the advantage of the fact that many spammers only attempt to send a batch of junk mail once. Under the Greylist system, the receiving mail assumes all the users as threat to the system and discards messages from unknown users in first attempt. It sends back a failure message to the originating server. If the mail server attempts to send the message second time- a step most legitimate server will take the Greylist assumes the message is not spam and allows it proceed to the recipient’s inbox. At the same time the recipient’s email or address will be added to Greylist as legitimate senders. Greylist filter requires less system resources than some other types of spam filters. However, they require more time in mail delivery that is problematic when we are expecting time sensitive messages.

4. Honeypot technique

In this approach, spammers who attempt to operate automated program on the sites or in email services are failed due to a loophole is created for them. They will find hosts and attempt to send mail through it, wasting their time and resources and potentially revealing information about themselves and the origin of the spam they're sending to the entity that operates the honeypot. In various sites, form applications are developed by developers where they put an additional field for the robots. As an instruction, human are said not fill the field, but robots fill the information and get trapped by the developer and simply blacklisted by their IP. The main advantage of this technique is that it is very simple to implement but applicable only small scale applications.

5. Pattern Detection Technique

Pattern detection approach consists of a large database of messages worldwide to detect spam patterns. This method works well when the message has no content or has only attachments. This method is more automated than most of other techniques because the service provider maintains the comparative spam database instead of the system administrator. The main advantage of this technique is that it is able to stop spam in real time before it gets to the end user.

6. URL filtering technique

Most spam or phishing messages contain an URL that they entice victims into clicking on. So a popular technique since the early 2000 consists in extracting URLs from messages and look them up in databases such as Spamhaus' Domain Block List (DBL).

SYSTEM ARCHITECTURE

In this study, we design and implement two systems: one is the spam checker that is analogous to CSP (Cloud Service Provider) and an E-mail client that is similar to vendor in cloud computing concept. They communicate via an Application Programming Interface (API). The overall structure has been presented in Fig. 2. The EMC has its own user and consumes the spam filtering service, SFS. So, the proposed system architecture two different systems: EMC and SFS. As a spam filtering technique we choose blacklist as it contains individual tokens that are compared to incoming text to detect spam. Also, it works real time with any existing codebase.

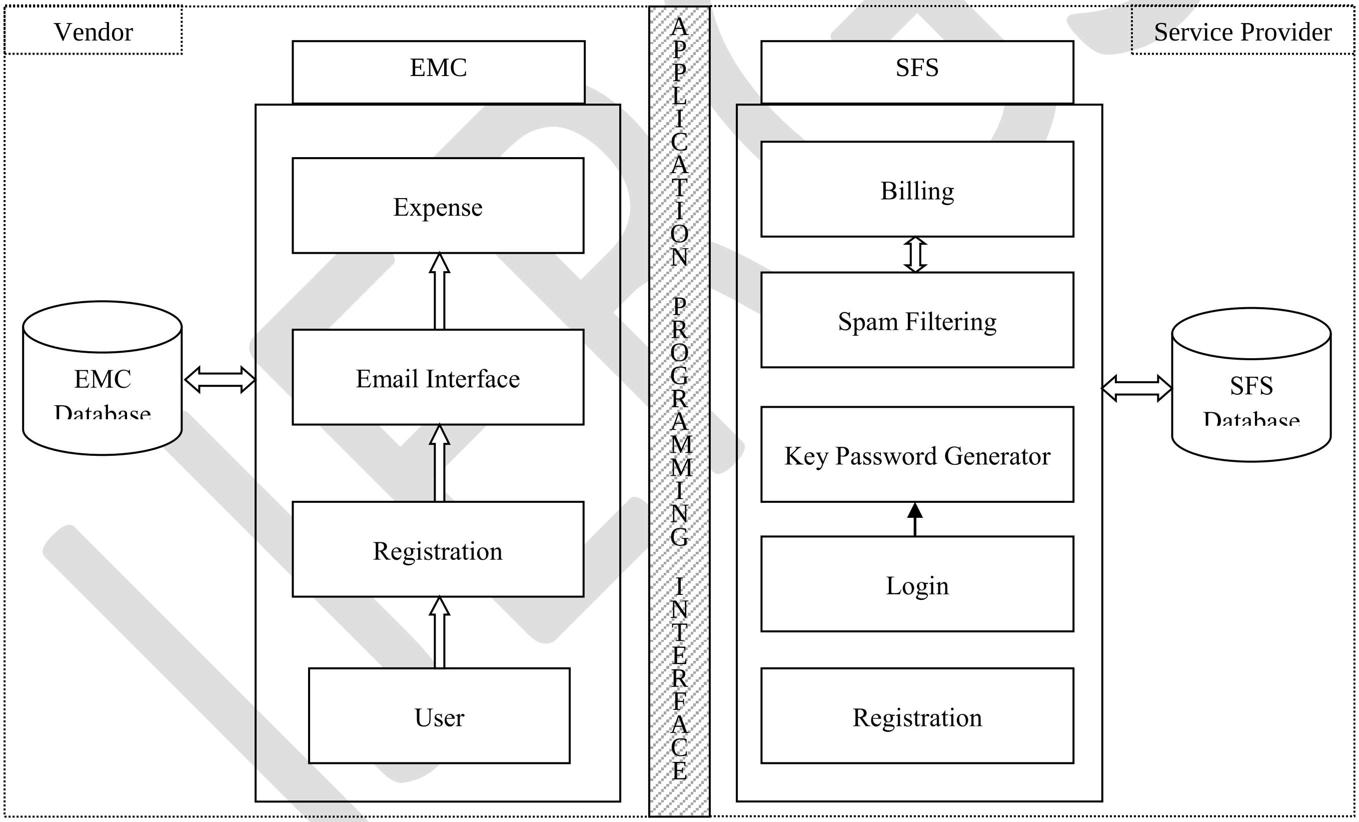


Fig. 2- Overall System Architecture

Spam Filtering Service (SFS)

In this service, we aim to build a system that securely provides spam filtering services to vendors. The system provides an API for the vendors to communicate and take services via secure channel. The API consists of four high level modules: registration, login, spam filtering and billing. Each module is connected to a database; we call this the SFS database.

At first, the system allows vendors to be registered using their credentials. Each of the potential vendors enters their details in a form and accesses the SFS vendor's profile. SFS enables vendors to generate a key-password combination from corresponding profile. The spam filtering module takes service request from the vendors via API authentication, process the request and provides a status of email whether it is spam or not. The authentication process of SFS is operated by this key-password combination. SFS generates a token against the authentication for each EMC that sends their first request to get the service. The token is a time period for which EMC are allowed to access the service without authenticating repeatedly for each request. The EMC only requires re-authenticating once the token expires. There is a billing section which keeps track of payments for each request from vendors depending on the pricing policy between SFS and EMC.

Email Client (EMC)

EMC is a cloud based E-mail messaging service that serves the communication platform among its users. EMC provides a registration form and uses its own database to store information of users who want a secure emailing service. For confidentiality and safety of the users EMC consumes a cloud based spam filtering service, SFS. EMC has three broad modules: registration, email interface, and expense details. The registration module allows user to sign up to EMC with their details information and credentials. EMC provides an email interface for sending and receiving messages via online. An expense details can also be monitored from the EMC that is directly obtained from SFS in real time via API.

Application Programming Interface (API)

SFS implements a RESTful (Representational state transfer) API for vendors to access the service. This API uses HTTP requests to complete operations such as GET, PUT, POST and DELETE data. In many cloud platforms including IBM cloud is using RESTful APIs [14]. The traditional RESTful requests are presented in Table 3.

Table 3 – RESTful Requests

Request Type	Description
HTTP GET	Requests are used to retrieve the representation of a resource or a list of resources.
HTTP POST	Requests are used to create new resources.
HTTP PUT	Requests are used to update existing resources.
HTTP DELETE	Requests are used to delete resources.

EMC requests SFS API for authentication using a GET request by sending the credentials and receiving a token. SFS verifies the EMC credentials and generate a token that expires after certain timeline. In next requests, EMC always uses this token to obtain other services. For example, once EMC has the verified token, it sends a POST request to SFS containing the token and email contents along with other relevant parameters. SFS verifies the token and process the spam filtering operation and return the response. SFS also measures the costing for processing the spam filtering. For this system, we consider a standard of $0.0001 for each word to check. Users can view the expenses from the EMC interface that are retrieved from SFS via API GET request.

SYSTEM IMPLEMENTATION

Development Environment

We develop these systems in LAMP (Linux Apache MySQL PHP) stack. We use notepad++ IDE (Integrated Development Environment) for writing the source code and viewing the site in local PC. We use XAMPP v5.6.14 for the LAMP environment. As this has been developed in local PC we specify the site URL as http://localhost/sfs/ where 'localhost' is any domain name such as http://www.yourdomain.com/ and 'sfs' is the directory where the SFS web-service runs.

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. PHP and MySQL are used as the backend platform for the development. For the front end development, we use HTML5, CSS3, jQuery and Bootstrap v3.3.7.

SFS Development

The main objective of this system is to build an API with spam filtering engine. The API consists of several functional calls for various purposes. Any EMC can consume this service as web services. The response is in the form of JSON, a JavaScript notation for cross platform data exchange. To consume the web-service, a library package named cURL [15] is used in this study. The available API functional calls are shown in Table 4. A sample functional call for login module with description, corresponding API URL, cURL request sample and sample JSON response are following-

Vendor Login

Description: A vendor needs to login to SFS for accessing further operations. Vendor can login using the key and password from the SFS site.

Functional prototype:

```
function login($vendor_id, $key = null, $password = null){
        //connect to db
        //verify vendor against key, password and vendor id
        //return token if successful
}
```

API URL:

```
http://localhost/sfs/api/login
```

cURL request:

```
curl --user key:password http://localhost/sfs/api/login
```

API response:

```
{
        “result”: “success”,
        “token”: “8f5f759e961bf53eebf8d90dbd8c2ba9”
}
```

Table 4 – API Functional Call List

Functional Call	Description, parameters and URL
Spam Filtering	Request for spam filtering by vendors URL: http://localhost/sfs/api/filter_spam parameters: token, vendor_user_id, message method: POST response: result, is_spam, cost
Billing	Request for billing details for users in EMC URL: http://localhost/sfs/api/billings parameters: token, vendor_user_id, billing_priod method: GET response: result, charge_details

Proposed Spam Filtering Algorithm

We use blacklist technique in developing the SFS system. We collect a list of keywords from the web and make text files for each of the category mentioned in Table 2. The algorithm accepts the email message in real time and matches each word from the message with the keywords from the blacklist text files. We use regular expression for searching blacklisted keywords. For each service request we attach a cost factor for scanning the message through.

```
Algorithm SpamFilter(vid, cid, cm)
// vid is the vendor id, cid id client id, cm is client message
// blist is blacklisted data in single file
// K is blacklisted keywords
// M is a Boolean value whether we have match or not
// W is the number of words checked in the message
// P is the total price for the spam filtering

D ← blacklist directory
for each blacklist file blist in D do
     K ← get keywords from blist
endfor
Cl ← 0 // Cl is the current line in blacklist file
for each keyword k in K do
    Cl ← Cl + 1
    k ← remove space and comments before and
          after keyword  k
    if empty(k) then
        continue to next line of K
    endif
    M ← find keyword k in cm
    W ← Total words in the message
    P ← W * 0.0001
    return R(M, P) // return result to the vendor
endfor
```

EMC Modules

Front End

Front end includes the registration page, login page and the email interface. We build the pages using the bootstrap library. These pages can be viewed from Fig. 3. At first unregistered user enters into mail service website. They click on register button and fill up registration form with their personal information. After submission user's information are inserted into EMC database.

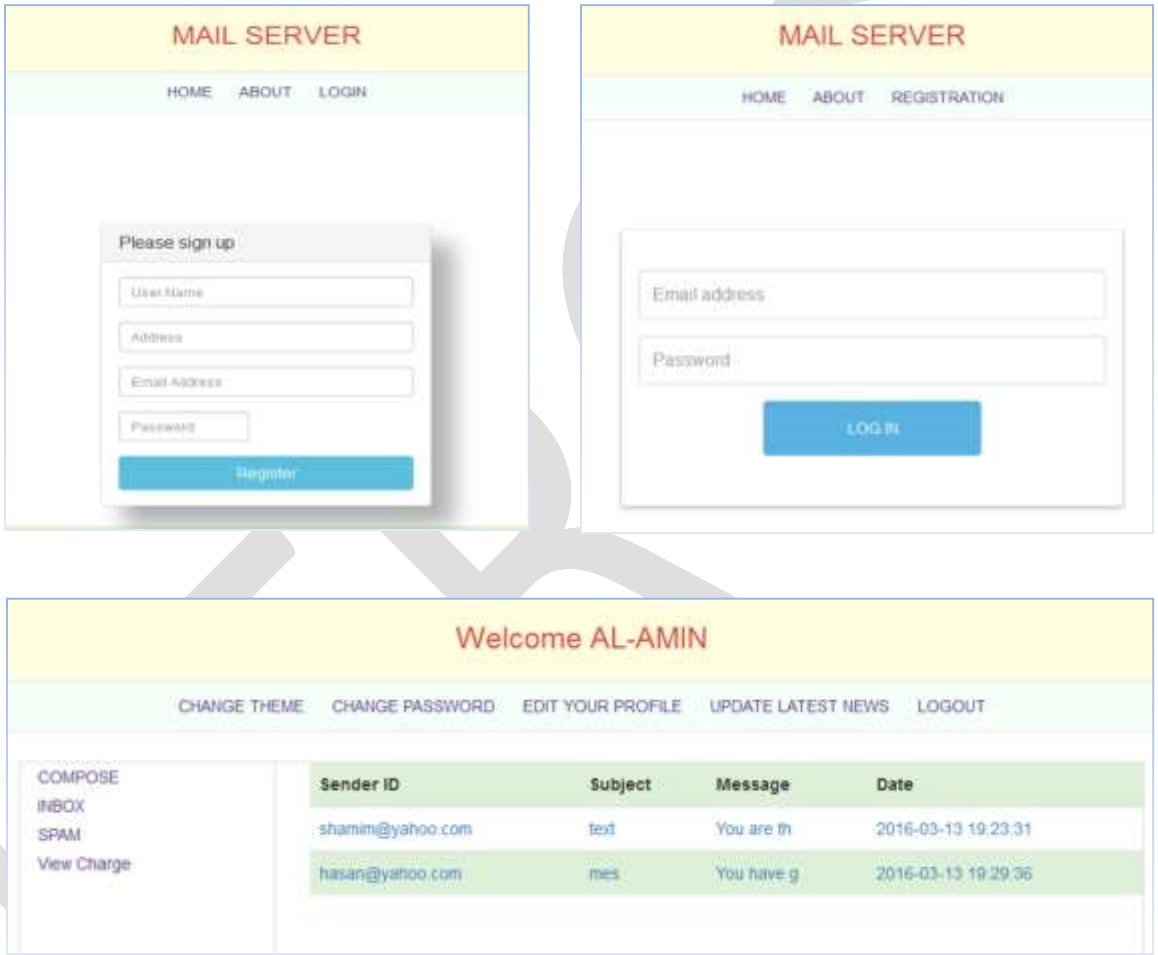
cURL Script

We use cURL for consuming the SFS as a web-service. We build a prototype that we follow for requesting all the functional calls allowed by SFS API. The prototype is following-

```
$url = 'http://localhost/sfs/api/login; // url from functional call
// parameters with the request
$fields = array(
            'vendor_id' => urlencode($vendor_id),
            'vendor_key' => urlencode($key),
            'vendor_password' => urlencode($password),

);
foreach($fields as $key=>$value)
{ $fields_string .= $key.'='.$value.'&'; }
rtrim($fields_string, '&');

// configuring cURL
curl_setopt($ch,CURLOPT_URL, $url);
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);
```

```
// execute cURL
$result = curl_exec($ch);
curl_close($ch);

// Get response as an array
$response = json_decode($result);
```

This way any EMC can consume the SFS. It is noted that json_decode() is the function which decodes any JSON response from SFS to array format for easier processing in the EMC backend such as storing data in the local database or showing the data in real time in users profile.



Sender ID	Subject	Message	Date
shamim@yahoo.com	text	You are th	2016-03-13 19:23:31
hasan@yahoo.com	mes	You have g	2016-03-13 19:29:36

Fig. 3- Front End Pages

LIMITATIONS AND RECOMMENDATIONS

The execution of the SFS is flawless and any EMC client can incorporate this service easily into their system. They only require registering to the SFS site, generating the keyword and easy use of cURL. The SFS system also provides secure service to the EMC by generating tokens for certain amount of period. However, Email processing requires fast outcome in terms of spam filtering as E-mail communication is an instant and interruptible service. The performance of this system may degrade if there is a potential delay in re-retrieving of tokens after its expiration. From the security point of view the proposed system can be acceptable in this context. The pricing model is in very basic mode in this study which requires further attention. Perhaps, a pricing plan category can be considered and implemented based on the client's choices.

We recommend this spam filtering service as baseline to other services which can be built and extended from this proposed architecture. Other spam filtering technique can also be built on this stack. There are available anti-spam services in web. This architecture with blacklist anti-spam technique can be a model for some other individuals or organizations to build such important system.

CONCLUSION

E-mail sites have millions of users from all over the world. People with less technical knowledge are frequently attacked by spammer in these days. Much sensitive information is leaked away without any intention of the users. They are being trapped emotionally; curiosity and sometime greediness push them to the threat and throat of the spammers. In this paper, we investigate spam properties and consolidated them into one place that may add knowledge to the community and save them from losing their property. We study existing anti-spam techniques and build a service that protects users from unwanted activities in email. The system is designed based on the cloud concept so that service providers and email clients both can be benefitted. Our proposed architecture can help e-mail servers to improve their security and detect malicious messages and further protects user private contents to be hacked. Finally, we can’t stop the spam completely but we can reduce.

REFERENCES:

[1] Avner Caspi and Paul Gorsky. “Online deception: Prevalence, motivation, and emotion” CyberPsychology & Behavior, vol 9, no.1, pp. 54-59, February 23, 2006.
[2] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. “Detecting spammers on social networks” Proceedings of the 26th Annual Computer Security Applications Conference, pp. 1-9, December 06-10, 2010.
[3] Kunal Mehrotra and Shailendra Watave, “Spam Detection - A Bayesian approach to filtering spam”, [Online] Available: http://www.cise.ufl.edu/~kmehrotr/filter/SpamFilterProjectReport.pdf, (February 15, 2016)
[4] Statista, “Spam statistics: spam e-mail traffic share 2016”, [Online] Available: https://www.statista.com/statistics/420391/spam-email-traffic-share/, (October 17, 2016).
[5] Sunil Paul, “Apparatus and method for controlling delivery of unsolicited electronic mail” U.S. Patent No. 6,052,709, April 18, 2000.
[6] Lee Codel Lawson Tarbotton, Daniel Joseph Wolff, and Nicholas Paul Kelly, “Detecting unwanted properties in received email messages” U.S. Patent No. 6,757,830, June 29, 2004.
[7] J.R Lee, Sang-Kug Ye, HDJ Jeong, “Detecting Anomaly Teletraffic Using Stochastic Self-Similarity Based on Hadoop” 16th International Conference on Network-Based Information Systems (NBiS), pp. 282 – 287, 2013.
[8] Valerie Ria Boquiron, “Spam, Scams and Other Social Media Threats” [Online] Available: http://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/75/spam-scams-and-other-social-media-threats, March 24, 2015.
[9] Lin-Shung Huang, Alexander Moshchuk and Helen J. Wang, “Clickjacking: attacks and defenses” 21st USENIX Security Symposium (USENIX Security 12), 2012.
[10] Mike Spykerman, “Typical spam characteristics”, [Online] Available: http://www.spamhelp.org/articles/Spam-filter-article.pdf, August 20, 2015.
[11] Policy Patrol, “Top 10 spam characteristics”, [Online] Available: http://www.policypatrol.com/top-10-spam-characteritics, August 19, 2015.
[12] Kevin P. Murphy, “Naive bayes classifiers” University of British Columbia, 2006.
[13] K. Ming Leung, “Naive bayesian classifier” Polytechnic University Department of Computer Science/Finance and Risk Engineering, 2007.
[14] IBM, “REST API Programming Guide”, [Online] Available: https://www-935.ibm.com/services/multimedia/API_Programmer_User_s_Guide_v1.4.1.pdf, (February 19, 2016).
[15] PHP, “PHP cURL”, [Online] Available: http://php.net/manual/en/book.curl.php, (February 20, 2016).