ISSN No. 0976-5697

Volume 4, No. 2, Jan-Feb 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Countermeasure Technique for Email Spoofing

Manish Kumar*
Assistant Professor, Dept. of Master of Computer Applications, M. S. Ramaiah Institute of Technology, Bangalore and Research Scholar, Department of Computers Science and Applications, Bangalore University, Bangalore, INDIA
manishkumarjsr@yahoo.com

Dr. M. Hanumanthappa
Dept. of Computer Science and Applications,
Jnana Bharathi Campus,
Bangalore University,
Bangalore -560 056, INDIA,
hanu6572@hotmail.com

Dr. T.V. Suresh Kumar
Professor & Head, Dept. of Master of Computer Applications,
M. S. Ramaiah Institute of Technology, Bangalore-560 054, INDIA
hod_mca@msrit.edu

Abstract: Email spoofing is a technique used by hackers to hide their identities which plays a major role in phishing which poses to be an authentic or legitimate user. Phishing attacks use spoofed emails and fraudulent websites designed to fool recipients into divulging personal financial data such as credit card numbers, account usernames and passwords, social security numbers, etc. Major of inbound email threats such as spam, viruses, phishing and most email denial-of-service attacks can all be traced back to a single common cause – lack of authentication in the email protocol SMTP. Hence this shortcoming has to be addressed with great deal of effort which when un-attempted causes severe consequences. In recent years lot of research has been done in this direction but focus on informing the users whose email address is used for spoofing was given least or almost no priority. At present scenario somehow only the receiver comes to know that the email received is fake but users whose email address is used to spoof the email do not come to know about the incident unless and until receiver suspect the mail and cross examine it with the sender manually. In this paper we are discussing the technique which can detect and inform the individual whose email address has been used for spoofing the email. It will help the innocent users to take a counter measures.

Keywords: E-Mail Spoofing, E-Mail Authentication, SMTP (Simple Mail Transfer Protocol), DKIM (Domain Keys Identified Mail), SPF(Sender Policy Framework), MTA (Message Transfer Agent),Mail Submission Agent (MSA), or Mail User Agent (MUA).

I. INTRODUCTION

E-mail spoofing is the forgery of an e-mail header so that the message appears to have originated from someone or somewhere other than the actual source. Distributors of spam often use spoofing in an attempt to get recipients to open, and possibly even respond to, their solicitations.

E-mail spoofing is possible because Simple Mail Transfer Protocol (SMTP), the main protocol used in sending e-mail, does not include an authentication mechanism. Although an SMTP service extension (specified in IETF RFC 2554) allows an SMTP client to negotiate a security level with a mail server, this precaution is not often taken. If the precaution is not taken, anyone with the requisite knowledge can connect to the server and use it to send spoofed messages

Email spoofing may occur in different forms, but all have a similar result: a user receives email that appears to have originated from one source when it actually was sent from another source. Email spoofing is often an attempt to trick the user into making a damaging statement or releasing sensitive information (such as passwords)[1].

Whatever is the motivation, the objective of spoofed mail is to hide the real identity of the sender. A sender can use a fictitious return address or a valid address that belongs to someone else [2].

There are some techniques which are used to prevent phishing attack:

a. Strong website authentication
b. Mail server authentication
c. Mail authentication via digital signature

If authentication fails in any form then it will give rise to spoofing attacks which is the entry point for phishing attacks.

II. TECHNICAL ISSUES

A. Email Client Vulnerabilities:

It is easy to spoof email because SMTP (Simple Mail Transfer Protocol) lacks authentication. If a site has configured the mail server to allow connections to the SMTP port, anyone can connect to the SMTP port of a site and (in accordance with that protocol) issue commands (as show in the Figure:- 1) that will send email that appears to be from the address of the individual's choice; this can be a valid email address or a fictitious address that is correctly formatted.

SMTP and early email clients were designed for sending and receiving only text-based emails. Email clients have become more feature-rich as well, supporting scripting languages, address books, and integration with other desktop applications. Although certainly useful, these additional functions have also introduced vulnerabilities into mail clients

that have been exploited by viruses, worms, and other forms of malware.

B. Automating The Process:

As we can see, this process (Shown in Figure :-1) is pretty strait forward. Automating the process is quite simple and can be done by writing a script in any scripting languages. A script designed to send out mass mail can do so very quickly and efficiently. If you or your companies mail server were to be a target of email relay, it could cause you a lot of trouble. It may even overwhelm your mail server to the point of causing a denial-of-service attack.

A web server that allows hosting of PHP scripts may be used to send spoofed emails to any email user. This type of attack uses the mail method provided in PHP to launch spoofing attack. A html page is used as index to fetch the details of fake email to be used. The data is pushed onto the php script which executes the sending of the fake email [5].

Command: telnet gmail-smtp-in.l.google.com 25
Response : 220 mx.google.com ESMTP 4si5132501pbm.48
Command: **MAIL FROM**
Response : 503 5.5.1 EHLO/HELO first. 4si5132501pbm.48
Command: **HELO**
Response: 250 mx.google.com at your service
Command: **MAIL FROM: <manishkumarjsr@yahoo.com>// (can be any email address e.g. xyz@zbc.com) //**
Response : 250 2.1.0 OK 4si5132501pbm.48
Command: **RCPT TO: <manishkumarjsr@gmail.com> // (can be any email address e.g. abc@xyz.com) //**
Response : 250 2.1.5 OK 4si5132501pbm.48
Command: **DATA**
Response : 354 Go ahead 4si5132501pbm.48

Date: 8/8/2007 4:15:30
To: manishkumarjsr@gmail.com
From: manishkumarjsr@yahoo.com
Subject: This is test mail
Hi Manish This is just a test mail
.
Response : 250 2.0.0 OK 1328620872 4si5132501pbm.48
Command: **quit**
Response : 221 2.0.0 closing connection 4si5132501pbm.48

Connection to host lost.

Header of Email Received

```
Delivered-To: manishkumarjsr@gmail.com
Received: by 10.223.86.15 with SMTP id q15cs155556fal;
        Tue, 7 Feb 2012 05:21:13 -0800 (PST)
Received: by 10.68.200.65 with SMTP id jq1mr58568991pbc.54.1328620872650;
        Tue, 07 Feb 2012 05:21:12 -0800 (PST)
Return-Path: <manishkumarjsr@yahoo.com>
Received: from  ([202.122.18.70])
        by mx.google.com with SMTP id 4si5132501pbm.48.2012.02.07.05.19.26;
        Tue, 07 Feb 2012 05:21:12 -0800 (PST)
Received-SPF: neutral (google.com: 202.122.18.70 is neither permitted nor denied by
domain of manishkumarjsr@yahoo.com) client-ip=202.122.18.70;
Authentication-Results: mx.google.com; spf=neutral (google.com: 202.122.18.70 is
neither permitted nor denied by domain of manishkumarjsr@yahoo.com)
smtp.mail=manishkumarjsr@yahoo.com
Message-Id: <4f312548.8401440a.21fc.5d5fSMTPIN_ADDED@mx.google.com>
Date: 8/8/2007 4:15:30
To: manishkumarjsr@gmail.com
From: manishkumarjsr@yahoo.com
Subject: Tjhis is test mail
Hi Manish This is jiusust a test mail
```

Figure1:- Spoofed Email Attack through Telnet SMTP

III. PREVENTION TECHNIQUES

A. Email Authentication Systems:

Email spoofing is probably one of the biggest current web security challenges. Without authentication, verification, and traceability, users can never know for certain if a message is legitimate or forged. Email authentication systems may provide an effective means of stopping email spoofing. The three main contenders for authentication are Sender Policy Framework (SPF), SenderID, and Domain Keys [3].

```
Received-SPF: neutral (google.com: 202.122.18.70 is neither permitted nor denied by
domain of manishkumarjsr@yahoo.com) client-ip=202.122.18.70;
Authentication-Results: mx.google.com;
spf=neutral (google.com: 202.122.18.70 is neither permitted nor denied by domain of
manishkumarjsr@yahoo.com) smtp.mail=manishkumarjsr@yahoo.com
```

Figure 2:- Email Header Highlighting SPF status for Spoofed Email

```
Received-SPF: pass (google.com: best guess record for domain of
manishkumarjsr@yahoo.com designates 121.101.151.224 as permitted sender) client-
ip=121.101.151.224;
Authentication-Results: mx.google.com; spf=pass (google.com: best guess record for
domain of manishkumarjsr@yahoo.com designates 121.101.151.224 as permitted sender)
smtp.mail=manishkumarjsr@yahoo.com; dkim=pass (test mode) header.i=@yahoo.com
```

Figure 3:- Email Header Highlighting SPF status for Normal Email

a. SPF (Sender Policy Framework or Sender Permitted From): Checks the "envelope sender" of an email message—the domain name of the initiating SMTP server. Sender path authentication [4] that helps recipients identify the authorized mail servers for a particular domain, and validate that emails they received has originated from these authorized sources.
b. SenderID: A path-based authentication technology that authenticates the sending domain, based on the network path the email took. Network path is defined by source IP address. It checks after the message data is transmitted and examines several sender-related fields in the header of an email message to identify the "purported responsible address."
c. DKIM (Domain Keys Identified Mail)- A crypto-based authentication technology that authenticates the sending domain, based on a cryptographic signature contained within the email. DKIM provides a "cryptographic signature" (or "key") of multiple email header fields and the body of a message. In its DNS record, a Web domain protected by DKIM publishes the public key (or "domain key") that corresponds to its self-generated private signing key. Email recipients can use that key to verify that the message header and body match the identity of the sending domain – helping them determine whether the email is likely to be a phishing or other malicious message. Domain Keys [6] is an attempt to give email providers a mechanism for verifying both the domain of the email sender and the integrity of the messages sent. Once the domain can be verified, it can be compared to the domain used by the sender in the From: field of the message, to detect forgeries. Domain Keys uses public key encryption technology at the domain level to verify the sender of email messages. There are tools such as PGP and S/MIME for encrypting and signing of email messages. A recipient of a signed message can verify the original sender based on the cryptographic signature.

B. Mail Server Authentication:

This technique uses enhanced DNS (domain name system) capabilities to verify the IP (internet protocol) address of sender's email server.

C. Mail Server Authentication via Digital Signature:

This technique uses existing industry standard S/MIME digital signatures to sign outbound mail to provide signature verification at the gateway or email client. S/MIME is an asymmetric cryptography technique which is used to authenticate sender and provide strong signature semantics.

The followings are the stages to authenticate e-mail using S/MIME digital signature:

a. Authority in the trusted public certificate, such as VeriSign, Thawte, GlobalSign, publishes digital signature for each e-mail addresses.
b. Each sent e-mails will be inserted a digital signature with private key. This digital signature provides a way to prove the authentication of "From:" address.
c. The recipient will be equipped with S/MIME protocol whose function is to verify the digital signature. If it is valid then "From:" address is expressed as a valid as well. Therefore, the recipient can trust the email content.

IV. EMAIL SPOOFING DETECTION TECHNIQUE

The sender domain authentication technology is used to detect whether sender's mail address is not pretend other domains. This is divided two. One is an electronic signature-based "Domain Keys" and other is the Internet Protocol (IP) address-based "SPF". Domain Keys is used in Yahoo! and Gmail. In our research, we have used Gmail as a free mail because users of Gmail have been increasing in recent years. We also used Gmail and Yahoo Mail as a recipient server and showed that it is possible to pretend a sender mail address. Moreover, we verified whether receiving side can distinguish whether received mail address is pretending other mail address when we used the IP address-based authentication technology. Since the spoofing some one else email address is illegal activity so for demonstration purpose we have used our own email address [7].

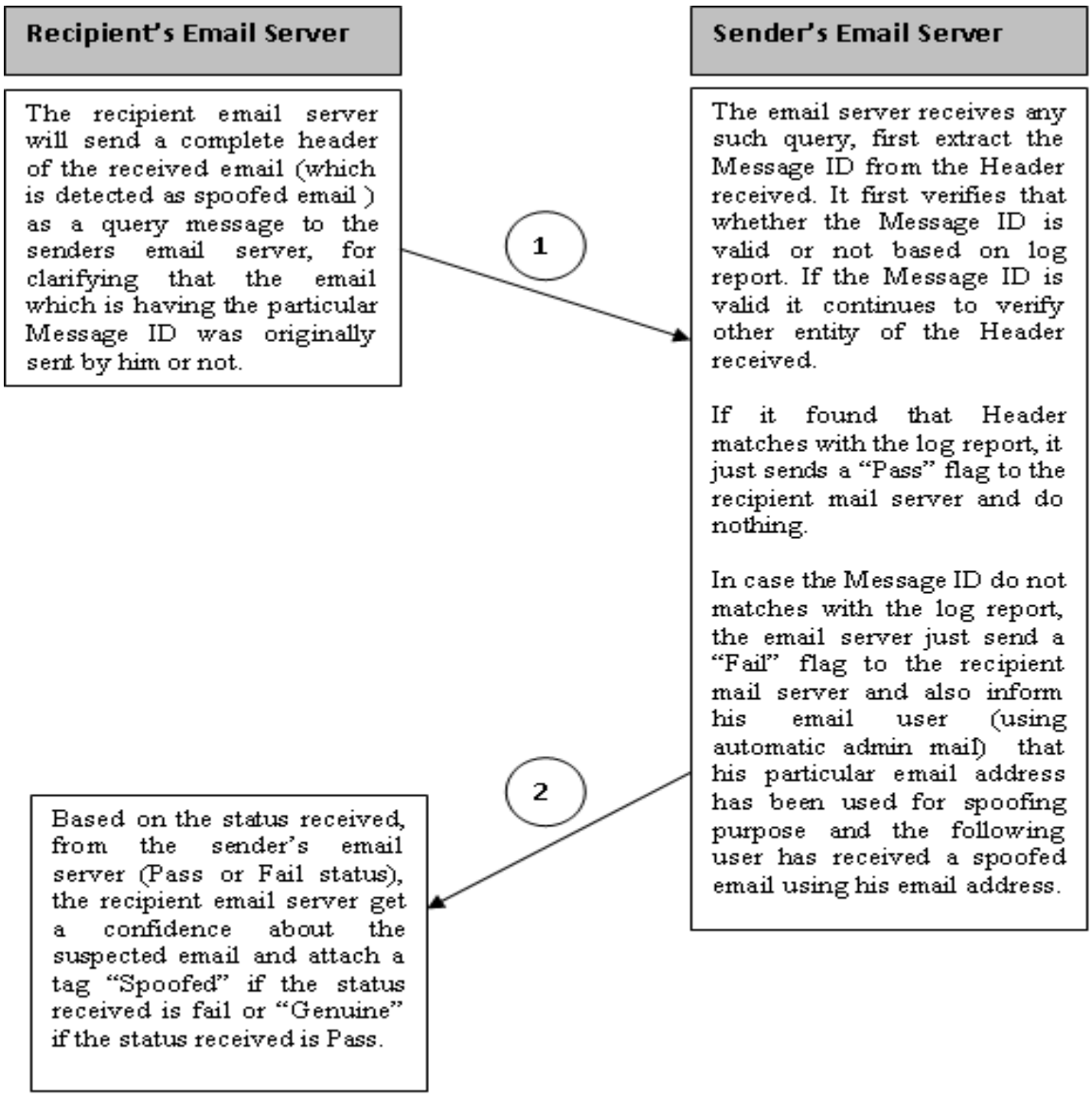

Figure 4:- Query process between the recipient's and sender's email address if the email is detected as spoofed at the recipient side

A. Evaluation of SPF:

Sender Policy Framework (SPF) is an email validation system designed to prevent email spam by detecting email spoofing, by verifying sender IP addresses. SPF allows administrators to specify which hosts are allowed to send mail from a given domain by creating a specific SPF record (or TXT record) in the Domain Name System (DNS).Mail exchangers use the DNS to check that mail from a given domain is being sent by a host sanctioned by that domain's administrators.

Now, we will explain how SPF is evaluated when email is actually received as an example of Gmail. When email is sent from the mail server that does not specify the SPF record at all, the lines found in the header field are similar to the Figure:-2.

The part of *"manishkumarjsr@yahoo.com"* is sender's mail address and as the result of the evaluation of SPF it is "neutral". "?all" is defined in the end of the record like "v=spf1 ?all" for instance, and when it does not match to other conditions and it is match to "?all", the SPF record of the sender domain becomes "neutral".

SPF do not immediately refuse the receiving of email even if the result is "neutral". On the other hand, when email is sent from the mail server that the SPF record is appropriately set, evaluation of SPF look similar to the lines shown in the Figure:-3, and the result of it is "pass".

This is the case that IP address in the sending side matches to the SPF record and succeeds in the authentication. Email is processed according to the evaluation of the sender domain because it is valid.

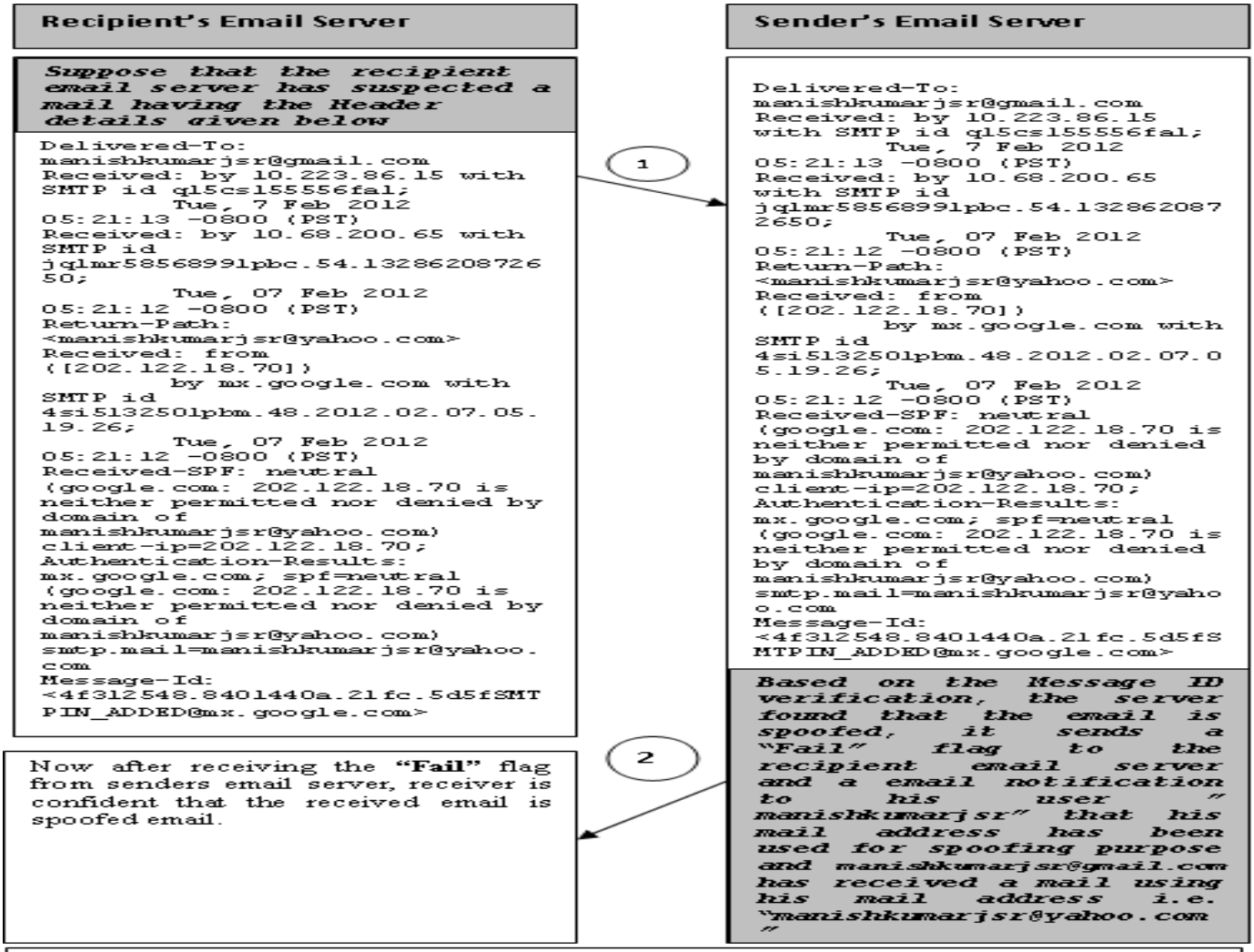

Figure 5:-Illutration of Query process between the recipient's and sender's email address if the email is detected as spoofed at the recipient side.

V. PROPOSED TECHNIQUE

Normally in email delivery a message transfer agent receives email from either another MTA (Message Transfer Agent), a mail submission agent (MSA), or a mail user agent (MUA). The transmission details are specified by the Simple Mail Transfer Protocol (SMTP). When a recipient mailbox of a message is not hosted locally, the message is relayed, that is, forwarded to another MTA. Every time an MTA receives an email message, it adds a Received trace header field to the top of the header of the message,[4] thereby building a sequential record of MTAs handling the message. The process of choosing a target MTA for the next hop is also described in SMTP, but can usually be overridden by configuring the MTA software with specific routes.

A MTA works in the background, while the user usually interacts directly with a mail user agent. One may distinguish initial submission as first passing through an MSA – port 587 is used for communication between an MUA and an MSA while port 25 is used for communication between MTAs, or from an MSA to an MTA; this distinction is first made in RFC 2476.

For recipients hosted locally, the final delivery of email to a recipient mailbox is the task of a message delivery agent (MDA). For this purpose the MTA transfers the message to the message handling service component of the message delivery agent. Upon final delivery, the Return-Path field is added to the envelope to record the return path.

As each email header contains lots of information and processed at every MSA, MTA , MDA and MUA gives a vital clue about the originality of the email. In general when a recipient server receives email, it can detect it as:

a. A Genuine mail from the authentic user or domain (based on various detection techniques discussed in Section III and Section IV) or
b. Spoofed Email.

We are concern about the mail which is detected as a spoofed mail. Once the mail is detected as a spoofed mail, we suggest that the users whose email addressed have been used for sending a fake email should come to know about the incidents that his email address has been used or falsified for some wrong purpose.

Continuing with the same example which we have shown in figure:-1, if we observe, each mail sent by the sender email server is having unique Message ID. In our example (Figure 1), the Message ID is "*Message-Id: <4f312548.8401440a.21fc.5d5fSMTPIN_ADDED@mx.google.com>"*

Message-ID is a unique identifier for a digital message, most commonly a globally unique identifier used in email. Message-IDs are required to have a specific format which is a subset of an email address and to be globally unique. That is, no two different messages must ever have the same Message-ID.

Now once the recipient email server has detected some mail as a spoofed mail, it can trigger the following steps shown in figure 4:

VI. CONCLUSION

The techniques suggested above can improve the spoof detection and will also help to inform the users whose email address has been used for spoofing purpose.

There is no methods that can perfectly overcome the spoofing attack until this time. The strong website authentication technique provides high level authentication, but it is not easy to be used by end user. The mail server authentication technique can verify a sender domain, but it does not authenticate "From:" address. Moreover, it also cannot accommodate e-mail forwarding. The technique, mail authentication via digital signature, accommodates a way to authenticate email address and prevent spoofing and spamming attacks. However, based on the cost, it is not an effective one.

In this research, we used TELNET to explain simply the procedure of email sender address spoofing, but if the setting of the email software is changed and own mail address is rewritten in other one, "spoofing" mail that pretended the sender can be transmitted. This is because there are no functions that certify the other party who transmitted correctly in SMTP. Therefore, the spoofing is prevented by using the sender domain authentication technology such as Domain Keys or SPF. Gmail used by the experiment did the authentication that used SPF in addition to Domain Keys. However, this information can't be watched as long as the receiving side does not watch the header field.

Moreover, when only the account part is pretended in the same domain, SPF can not detect. Therefore, recipient is not necessarily able to distinguish whether received mail address is misrepresented.

VII. ACKNOWLEDGEMENT

I would like to thanks MSRIT management, my colleagues and Dept. of Computer Science and Applications, Bangalore University, Bangalore for their valuable suggestions, constant support and encouragement.

VIII.REFERENCES

[1] Anti-Phishing Working Group. Phishing Archive.http://www.antiphishing.org/phishingarchive.html.

[2] D. Birk, M. Dornseif, S. Gajek, and F. Grobert, "Phishing phishers—tracing identity thieves and money launderers," Horst Gortz Institute for IT Security, Ruhr University Bochum, Tech. Rep. TR-HGI-01-2006.

[3] Dhanalakshmi Ranganayakulu , L. Kavisankar, C. Chellappan, "Enhanced E-Mail Authentication Against Spoofing Attacks To Mitigate Phishing" European Journal of Scientific Research, ISSN 1450-216X Vol.54 No.1 (2011), pp.165-175.

[4] E. Kirda, and C. Kruegel. ,"Protecting Users against Phishing Attacks", Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05), Edinburgh, UK, 2006, pp. 517-524.

[5] http://searchsecurity.techtarget.com/definition/email-spoofing

[6] J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, "DomainKeys Identified Mail (DKIM)," Internet Draft draft-allman-dkim-base-01,http://mipassoc.org/dkim/specs/draftallman-dkim-base-01.txt, October 2005.

[7] Toshiyuki Tanaka, Akihiro Sakai, Yoshiaki Hori, Kouichi Sakurai, "A countermeasure to email sender address spoofing", 2009 Joint Workshop on Information Security