Forensic analysis of E-mail address spoofing

Conference Paper · September 2014

DOI: 10.1109/CONFLUENCE.2014.6949302

CITATIONS
11

READS
6,109

5 authors, including:

Emmanuel S Pilli
Malaviya National Institute of Technology Jaipur
135 PUBLICATIONS **1,732** CITATIONS
SEE PROFILE

Preeti Mishra
Malaviya National Institute of Technology Jaipur
31 PUBLICATIONS **726** CITATIONS
SEE PROFILE

Sumit Pundir
Graphic Era University
9 PUBLICATIONS **106** CITATIONS
SEE PROFILE

R. C. Joshi
250 PUBLICATIONS **2,989** CITATIONS
SEE PROFILE

Forensic Analysis of E-mail Date and Time Spoofing

Preeti Mishra, Emmanuel S. Pilli and R. C. Joshi
Department of Computer Science & Engineering
Graphic Era University,
Dehradun, India
preeti.mish22@gmail.com, emmshub@gmail.com, chancellor.geu@gmail.com

***Abstract*—There are no adequate and proactive mechanisms for securing E-mail systems. E-mail date and time spoofing is one of the major problems of E-mail security. The effects of E-mail spoofing can be limited by the appropriate configuration of E-mail servers and improved user awareness of the problem. The only real countermeasure is the use of digitally signed messages that allow a recipient to authenticate the identity of the sender. This paper presents E-mail forensics to detect E-mail Date and Time spoofing. We have created dataset of spoofed and legitimate E-mails. We propose an algorithm to perform the forensic analysis of E-mail time and date spoofing, by reading the header information and analyzing the fields related to date and time. We have given a policy to check sent-date and received-date fields of every E-mail. If the sent-date and sent-time differs from the received-date and received-time by some predefined margin, the E-mail has been spoofed. The algorithm is validated on the dataset created in our lab.**

Keywords- SMTP; E-mail Spoofing; E-mail Forensics; Date; Recieved;

I. INTRODUCTION

E-mail is one of the most widely used applications of Internet. E-mail security can be defined as the ability of the system to provide privacy, sender authentication, message integrity, non repudiation, and consistency of messages.

E-mail is vulnerable to both passive and active attacks. Passive threats include *Release of message contents*, and *Traffic analysis* while active threats include *Modification of message contents*, *Masquerade*, *Replay*, and *Denial of Service, spoofing etc* [1]. The Symantec Intelligence Report (February 2012) [2] reports that 68% of all mails are spam, one in 358.1 E-mails is identified as phishing mail, and one in 274.0 E-mails contained malware. All of these E-mails are spoofed and sent as if they are coming from a trust worthy source. E-mail date and time is very crucial in cases such as bank statements, stock broker's communications and notices from head / boss in terms of deadlines, etc.

Simple mail transfer protocol (SMTP) [3] which is the most widely deployed and primary protocol for E-mail transfer does not define any security and privacy policy.

The E-mail header is the envelope of the E-mail containing such information as: sender's E-mail address, receiver's E-mail address, subject, time of creation, delivery stamps, message author, cc, bcc, etc. The date field in a spoofed E-mail header may contain a date which is ahead or before the actual date it was sent. An attacker may also change the time at which the E-mail was sent. Time field of *Date:* header can also be manipulated by attacker and make the E-mail message to be sent on time different from actual time. This may produce vulnerable result, specifically for those receivers, whose servers' mails are sorted according to sending date and time. In this paper, we are performing the detection and analysis of *date* and *time* spoofed E-mails. We propose an algorithm to perform forensic investigation of date and time spoofed E-mails by comparing the dates in the header.

The rest of the paper is organized as follows: Section 2 describes E-mail headers and how the values are manipulated to effect E-mail spoofing. Section 3 describes the related work on E-mail spoofing detection and investigation. Our proposed technique is described in Section 4. Section 5 explains experiments performed, data set created & analyzed and results obtained. The paper concludes with future work in Section 6.

II. E-MAIL DATE AND TIME SPOOFING

E-mail applications are vulnerable to various risks such as viruses, junk mail, unauthorized software, offensive text or pictures, unauthorized disclosure of sensitive information, forged messages, legal liabilities, date/time spoofing etc. E-mail date and E-mail address spoofing are the two important forms of E-mail spoofing. E-mail date spoofing, as shown in Fig.1, occurs when someone changes the sending date. The messages identified by boxes 'A' and 'B' in Fig.1 were sent on March 31, 2012 and at the same time. The

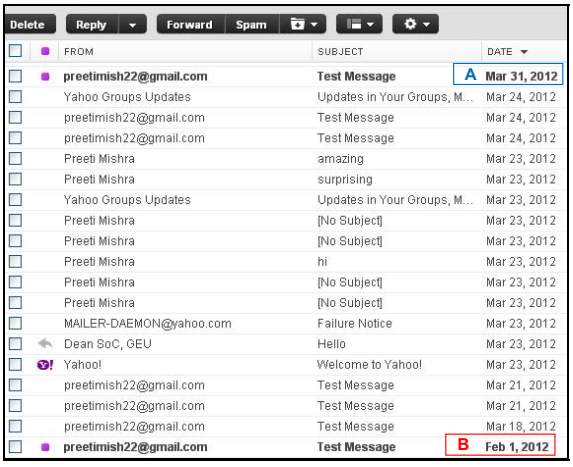

Figure 1. Email Date Spoofing

978-0-7695-4872-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCCT.2012.69

mails appear reordered in the inbox as mail 'A' is legitimate and mail 'B' is date and time spoofed.

E-mail address spoofing refers to sending mail which pretends to come from someone else. This can be done by altering the *From:* field in the E-mail header. This is more difficult as the Sender Policy Framework (SPF) [4] verifies sender IP addresses before the server relays the E-mail.

The E-mail header contains the following fields: From, X-Apparently-To, Return–Path, Received–SPF, X-Originating-IP, Authentication-Results, Received: from, Received: by, DKIM-Signature, Date, From, To, Message-ID, Subject, MIME-Version, Content-Type, Content–Transfer-Encoding, Content-Length etc. The *From:* field at the top indicates sender's address and the date and time of sending. The field X-Apparently-To is relevant when mail has been sent as a BCC and contains the address as in To field. The Return-Path field is the E-mail address which sends E-mail back to sender. The mail server will send a message to the specified E-mail address if the message cannot be delivered. Received: SPF is used to describe what mail server is allowed to send messages for a domain. DKIM-Signature security field may be used to validate the domain part of the sender's or author's E-mail address. The trace information is added by Originator, Relay including MTA in the form of Received: fields. Date: contains the sending date and time. All the Received: fields contain date and time information of E-mail at the time of receiving mail and domain / IP address of receiving server. Path can be traced by reading the header bottom to top and by analyzing all Receiving: headers (the top Received: header contain the last server information). From: gives the sender's address. The Date:, first Received: and last Received: fields are considered.

III. Related Work

Bob Radvanovsky [5] provides a fundamental understanding about how to read and interpret electronic mail headers, and what tools and methods may be utilized to interpret if they are legitimate or artificial. The paper explains a single example, presents analysis, drawing possible conclusions as to how to decipher the analysis.

Gaurav Malik [6] provides ways in which spam and spoofed E-mails are being tackled and also make suggestions on how confidence can be raised by the use of hybrid approaches. The approach of domain based analysis (including SPF – Sender Policy Framework) is discussed.

Major work in this area was done by Tariq Banday. Banday [7] analyzes E-mail date spoofing and reports how date-spoofing is done, lists the implications of date-spoofed E-mails on E-mail servers and surveys E-mail user behavior. He also discusses detection of date-spoofed E-mails to enable forensic examinations of date forged E-mails and presents possible technical solutions to stop date-spoofing.

Banday [8] projects the need for E-mail forensic investigation and has also carried out a detailed header analysis of a spoofed E-mail message. He has also discussed the various possibilities for detection of spoofed headers and identification of its originator. Further he has also discussed the difficulties faced by investigators during forensic investigation of E-mail with possible solutions.

Banday [9] shows how E-mail date spoofing can be detected and prevented. He also illustrates the processes to send and receive date spoofed E-mails. Sending SMTP servers can enforce a policy to check send Date and Resent-Date fields, if present, of every E-mail message received from SMTP client before their onward transmission. In case send or resend date differs from the current date (date from the clock of the server) by some predefined margin, the mail may be discarded with a notification to the sender, otherwise, the mail is transmitted. Like sending servers, receiving servers can also enforce a policy to check send Date, Resent-Date and Received fields against the system clock before accepting mail for transmission.

R. Hadjidj et al. [10] proposed an integrated analysis platform, IEFAF, in which a security analyst can perform a variety of tasks related to E-mail analysis. IEFAF is programmed in Java. The Java Mail API is used to parse E-mails in several file formats and extract relevant information. IEFAF is composed of Inter-database browser, Statistics explorer, Data mining explorer, Weka sub module and E-mail explorer. The framework offers different functionalities ranging from E-mail storing, editing, searching, and querying to more advanced functionalities such as authorship attribution and E-mail account localization.

EmailTracer [11] is a tool to track E-mail sender's identity. It analyzes the E-mail header and gives the complete details of the sender like IP address, which is key point to find the culprit and the route followed by the mail, the Mail Server, details of Service Provider etc.

IV. Proposed Technique

Our proposed technique calculates the threshold or margin which is the usual time taken to receive an E-mail. This margin is used to detect E-mail date and time spoofing in an E-mail. All the date and time fields are converted to UTC (Universal Time Coordinated) time before comparing their differences with the margin.

A. Proposed Technique to Calculate Margin

The margin or threshold was experimentally found to be few seconds to a maximum of twelve minutes, in the case of popular mail servers, when there is no arbitrary delay. We have also proposed an algorithm to calculate the threshold by examining the maximum of differences in time between the sending time and last server time. The threshold or margin plays an important role in deciding date and time spoofed E-mails. We consider a number of legitimate E-mail datasets and find out the delay in the delivery of each E-mail. The maximum delay is calculated and taken as the margin. This value can be updated as frequently as possible by considering more legitimate datasets for accurate threshold.

```
set margin=0, diff=0
set the margin value from the previous run
(margin is set to 0 in first run)
while (mail headers are available)
{
    Read date / time / offset of first
    Received: and Date:
    Convert into UTC
    Calculate difference between Lastser_Time
                          and Sending_Time
    diff = Lastser_time - Sending_time
    if(diff>margin)
       margin=diff
}
Write margin to the margin file
```

Figure 2 Algorithm to calculate Margin

The algorithm Fig. 2 takes input a normal E-mail header file and margin file (which initially contains zero as initial margin value). Further it extract three fields: *Date:* field (containing sending date / time / UTC offset), the last *Received:* field from the top (containing first server's E-mail receiving date / time / UTC offset) and the first *Received:* field from the top (containing last server's E-mail receiving date / time / UTC offset). These three fields are explained in Table I. Then convert the above three fields into UTC time zone so that the values are uniform across various servers. Various time zones are shown in Table II.

We find out the difference between sending time and last server E-mail receiving time. If the difference is greater the margin; it writes the difference to margin file. Each time a new E-mail header is processed the difference between

TABLE I. E-MAIL HEADER FIELDS USED FOR SPOOFING

Name of Field	Description
Date:	*Date:* contains sending date and time
Received:	The last *Received:* field in the E-mail header (from the top). It contains domain name / IP address of server receiving the mail from sender and its receiving date and time.
Received:	This first *Received:* field in the E-mail header (from the top). It contains domain name / IP address of the last server receiving the mail from sender and its receiving date and time.

TABLE II. RELATION BETWEEN UTC AND OTHER TIME ZONES

Time Zone	UTC equivalent
Central European Time (CET)	UTC + 01:00
Eastern European Daylight Time (EEDT)	UTC + 03:00
Indian Standard Time (IST)	UTC + 05:30
China Standard Time (CST)	UTC + 08:00
Australian Central Standard Time (ACST)	UTC + 09:30
New Zealand Standard Time (NZST)	UTC + 12:00
Uruguay Summer Time (UYST)	UTC – 02:00
Eastern Daylight Time (EDT)	UTC – 04:00
Central Standard Time (CST)	UTC – 06:00

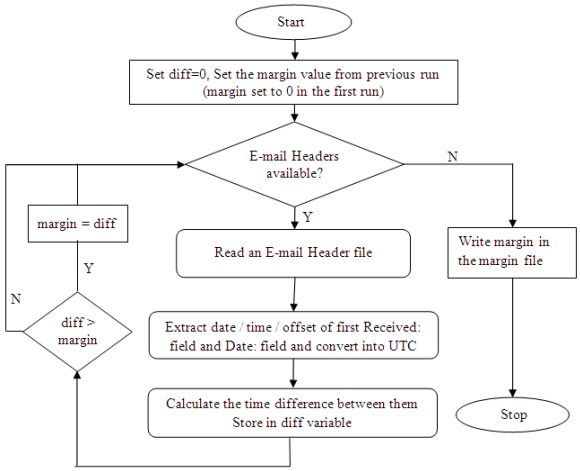

Figure 3 Flowchart to calculate Margin

sending time and first server time is calculated and compared with the margin. If difference is greater than margin, margin is updated. After processing the dataset of Legitimate E-mail Headers, margin value is written to the margin file. Here seven attributes extracted from above three fields of an E-mail message are stored in following variables: `LastSer_Date`, `LastSer_Time`, `FirstSer_Date`, `FirstSer_Time`, `Sending_Date`, `Sending_Time`, and `Diff`. The flowchart for the algorithm is shown Fig. 3.

B. Proposed Technique to Detect Date and Time Spoofing

The algorithm for detecting date and time spoofing is given in Fig. 4. It reads concerned three fields from the email header and compares them to determine whether the E-mail has been spoofed for date and time or not. There are three cases for any E-mail which is delivered to the recipient: (1) E-mail is not delivered on the same date of sending (2) E-mail is delivered on the same date but with a large variation in time and (3) E-mail is delivered on the same date and time (within an acceptable margin). The first two cases may be reasons for E-mail date and time spoofing. The third case is where the mail is legitimate.

```
if(sending_date and sending_time is not
   according to the usual semantics)then
        date error notification

elseif (sending_date == lastser_date)
{
    if (lastser_time – sending_time < margin)
        the mail as legitimate    // (Case 3)
    else
        the mail as time spoofed  // (Case 2)
}

elseif (sending_date == firstser_date)
        the mail as legitimate    // (Case 3)

else
        the mail as date spoofed // (Case 1)
```

Figure 4. Algorithm to detect Date and Time Spoofing

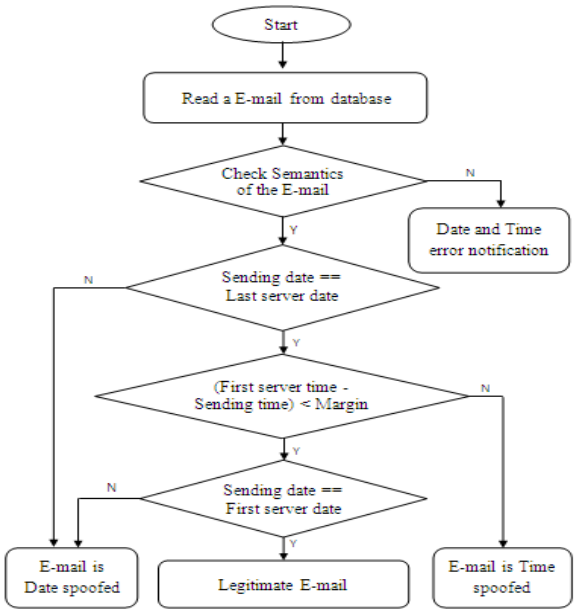

Figure 5. Flowchart to detect spoofing

Our algorithm first checks the semantics of date and time fields in the E-mail header. It generates an error message if the semantics are improper (if the hacker could not set the semantics in his or her mail client program) and proceeds further, otherwise. It then checks whether sending_date and lastser_date are same. If the dates are same, it checks whether the difference between sending_time and lastser_time is less than a set margin or threshold. If the difference is less than the margin threshold, then the E-mail is found to be legitimate (case 3) and spoofed in time, otherwise (case 2).

If the dates are not same, the algorithm checks whether the sending_date and firstser_date are same. If they are same, the E-mail is not date spoofed, but may have been delayed because of a server breakdown or over load on some intermediate servers relaying the E-mail (case 3). If the sending_date and firstser_date are not same, then the E-mail is date spoofed (case 1). The flowchart for the proposed technique is shown Fig. 5. In case if first SMTP server is temporarily unavailable then E-Mail sending error will come. We have implemented this algorithm in 'C' programming language.

V. EXPERIMENTS

The following experiments are performed to calculate the margin and detect date and time spoofing:

- Creation of spoofed E-mail using a mail-client program.
- Creation of dataset of spoofed mails using Java program.
- A margin calculating program reads several headers of legitimate E-mails to calculate margin and write to a file.
- An E-mail classification program that uses this margin and takes header of an E-mail as an input and decides whether an E-mail is date/time spoofed or legitimate.

A. *Creation of Spoofed E-mail Dataset*

A Mail Client program was coded in Java. The date header of E-mail is manipulated and sent through the Mail client program and certain E-mail servers display the spoofed E-mails, sorted according to sending date. We have created spoofed and legitimate E-mails using a java program that is created using Java Mail API Packages (javax.mail.* and java.util.Date). Some of the classes that were used are Session, MimeMessage, InternetAddress, Transport. Some of the methods of MimeMessage were also used: message.setReceipient( ), message.setFrom( ), message.setSentDate ( ) etc. The headers of one legitimate, one date and one time spoofed E-mails are shown in Fig. 6, 7 and 8 respectively:

```
Received: from 127.0.0.1  (EHLO mail-gx0-
f181.google.com) (209.85.161.181)
by mta1432.mail.mud.yahoo.com with SMTP;
Fri, 23 Mar 2012 05:13:12 -0700
Received: by 10.182.141.105 with HTTP;
Fri, 23 Mar 2012 05:13:12 -0700 (PDT)
Date: Fri, 23 Mar 2012 17:43:12 +0530
```

Figure 6. Legitimate E-mail Header

```
Received: from 127.0.0.1  (EHLO mail-pb0-
f52.google.com) (209.85.160.52)
 by mta1220.mail.mud.yahoo.com with SMTP;
Fri, 23 Mar 2012 22:43:42 -0700
Received: from GEU-PC ([115.113.125.178])
 by mx.google.com with ESMTPS id
f7sm7373949pbr.3.2012.03.23.22.43.40
 (version=SSLv3 cipher=OTHER);
Fri, 23 Mar 2012 22:43:41 -0700 (PDT)
Date: Sun, 18 Mar 2012 06:13:24 +0530 (IST)
```

Figure 7. Date Spoofed E-mail Header

```
Received: from 127.0.0.1  (EHLO mail-iy0-
f177.google.com) (209.85.210.177)
by mta1104.mail.sk1.yahoo.com with SMTP;
Sat, 24 Mar 2012 05:15:12 -0700
Received: from GEU-PC ([115.113.125.178])
by mx.google.com with ESMTPS id
d6sm7506095pbi.23.2012.03.24.00.06.15
(version=SSLv3 cipher=OTHER);
Sat, 24 Mar 2012 05:15:12 -0700 (PDT)
Date: Sat, 24 Mar 2012 16:45:00 +0530 (IST)
```

Figure 8. Time Spoofed E-mail Header

B. *Dataset Creation*

Publicly available Dataset of E-mail headers is not found. We coded Java program to create the dataset of E-mail headers. The program takes the credential as username and password and reads the whole inbox folder. It reads the E-mails and generates the corresponding header and writes them onto separate files. Java Mail API packages javax.mail.*, javax.util.*, javax.io.* etc were used. Some of the classes used are Session, Enumeration, File, Header etc. Some methods of Enumeration like getAllHeaders ( ), hasMoreElements ( ), etc are used.

C. Margin Calculating Program

This program calculates the threshold to receive an E-mail. We process several legitimate headers and extract date and time information of required Received: field. The values are converted to UTC. The margin is calculated according to algorithm in Fig. 2.

We have created various samples of legitimate E-mails. The Table III shows the three local time (last server time, first server time and sending time as per the given order of rows of each E-mail) with their offset and UTC equivalent and difference in sending time and last server E-mail receiving time. Calculated margin = 13 sec.

TABLE III. UTC CONVERSION OF DIFFERENT FIELD VALUES OF LEGITIMATE E-MAILS

File	Local Time	Offset	UTC time	Diff
1	23 Mar 2012 05:13:12	-0700	23Mar 2012 12:13:12	00
	23 Mar 2012 05:13:12	-0700	23 Mar 2012 12:13:12	
	23 Mar 2012 17:43:12	+0530	23 Mar 2012 12:13:12	
2	23 Mar 2012 01:38:34	-0700	23 Mar 2012 08:38:34	01
	23 Mar 2012 01:38:33	-0700	23 Mar 2012 08:38:33	
	23 Mar 2012 14:08:33	+0530	23 Mar 2012 08:38:33	
3	26 Mar 2012 04:06:02	-0700	26 Mar 2012 11:06:02	02
	26 Mar 2012 04:06:00	-0700	26 Mar 2012 11:06:00	
	26 Mar 2012 16:36:00	+0530	26 Mar 2012 11:06:00	
4	19 Apr 2012 22:00:29	-0700	20 Apr 2012 05:00:29	13
	19 Apr 2012 22:00:28	-0700	20 Apr 2012 05:00:28	
	20 Apr 2012 10:30:16	+0530	20 Apr 2012 05:00:16	

D. Conversion to UTC

We extract the date, time, and offset from all three fields (two received and one date) and convert them into UTC time according to Table II. The conversion of date and time from many time zones to UTC involves subtracting the time zone offset with '+' symbol from the given time and adding the time zone offset with '–' symbol to the given time. The conversion of time zone offset to UTC for the three mails is shown in Table IV, V and VI respectively.

TABLE IV. CONVERSION TO UTC (LEGITIMATE)

Local Time	Time Zone	UTC Time
23 Mar 2012 05:13:12	-0700	23Mar 2012 12:13:12
23 Mar 2012 05:13:12	-0700	23 Mar 2012 12:13:12
23 Mar 2012 17:43:12	+0530	23 Mar 2012 12:13:12

TABLE V. CONVERSION TO UTC (DATE SPOOFED)

Local Time	Time zone	UTC Time
23 Mar 2012 22:43:42	-0700 PDT	24 Mar 2012, 05:43:42
23 Mar 2012 22:43:41	-0700 PDT	24 Mar 2012, 05:43:41
18 Mar 2012 06:13:24	+0530 IST	18 Mar 2012, 00:43:24

TABLE VI. CONVERSION TO UTC (TIME SPOOFED)

Local Time	Time zone	UTC Time
19 Apr 2012 22:00:29	-0700	20 Apr 2012 05:00:29
19 Apr 2012 22:00:28	-0700	20 Apr 2012 05:00:28
20 Apr 2012 10:30:16	+0530	20 Apr 2012 05:00:16

Legitimate mail shows mostly the same date and time (sometimes with a delay less than the margin). The date or time spoofed E-mail shows a difference in date and time. The first row and second row specifies E-mail receiving date time information of last server and first server respectively. The last row specifies the E-mail sending date time information at the sender's end.

E. Classification of E-mails

We consider a legitimate, a date spoofed and a time spoofed E-mail header and classify the E-mails through the E-mail classification program. The first sample falls under case 3 as E-mail is delivered on same date since last server receiving and the first server recieving dates are equal to the sending date. Therefore E-mail is not date spoofed. The difference between last server receiving time and sending time is calculated. The difference is 00:00:00 i.e. 0 sec < 13 sec (margin). Therefore it is not time spoofed as well. The sample is shown in Fig. 9 and Table VII.

```
Received: from 127.0.0.1(EHLO mail-gx0-
f181.google.com) (209.85.161.181)by
mta1432.mail.mud.yahoo.com with SMTP;
Fri, 23 Mar 2012 05:13:12 -0700
Received: by 10.182.141.105 with HTTP;
Fri, 23 Mar 2012 05:13:12 -0700 (PDT)
Date: Fri, 23 Mar 2012 17:43:12 +0530
```

Figure 9. Header of Legitimate E-mail sample

TABLE VII. LEGITIMATE E-MAIL HEADER

Sending Date	First Server Date	Last Server Date	Sending Time	First Server Time	Last Server Time
23 Mar 2012	23 Mar 2012	23 Mar 2012	12:13:12	12:13:12	12:13:12

The second sample falls under case 1 as E-mail is not delivered on same date since the last server receiving date is not equal to the sending date. We compare sending date and first server receiving date. We observe that these are not equal. Therefore E-mail is date spoofed as shown in Fig. 10 and Table VIII.

```
Received: from 127.0.0.1(EHLO mail-pb0-
f52.google.com) (209.85.160.52)
 by mta1220.mail.mud.yahoo.com with SMTP;
Fri, 23 Mar 2012 22:43:42 -0700
Received: from GEU-PC ([115.113.125.178])by
mx.google.com with ESMTPS id
f7sm7373949pbr.3.2012.03.23.22.43.40
(version=SSLv3 cipher=OTHER);
Fri, 23 Mar 2012 22:43:41 -0700 (PDT)
Date: Sun, 18 Mar 2012 06:13:24 +0530 (IST)
```

Figure 10. Header of Date spoofed E-mail sample

TABLE VIII. DATE SPOOFED E-MAIL HEADER

Sending Date	First Server Date	Last Server Date	Sending Time	First Server Time	Last Server Time
18 Feb 2012	24 Mar 2012	24 Mar 2012	00:43:24	05:43:41	05:43:42

The third sample is an example for time spoofing (case 2) We calculate the difference between first server receiving time and sending time. The diff value is 01:00:12 = 72 seconds and 72 > 13 margin) and is time spoofed as shown in Fig. 11 and Table IX.

```
Received: from 127.0.0.1(EHLO mail-iy0-
f177.google.com) (209.85.210.177)
by mta1104.mail.sk1.yahoo.com with SMTP;
Sat, 24 Mar 2012 05:15:12 -0700
Received: from GEU-PC ([115.113.125.178])by
mx.google.com with ESMTPS id
d6sm7506095pbi.23.2012.03.24.00.06.15
(version=SSLv3 cipher=OTHER);
Sat, 24 Mar 2012 05:15:12 -0700 (PDT)
Date: Sat, 24 Mar 2012 16:45:00 +0530 (IST)
```

Fig.11. Header of Time spoofed E-mail sample

TABLE IX. TIME SPOOFED E-MAIL HEADER

Sending Date	First Server Date	Last Server Date	Sending Time	First Server Time	Last Server Tiime
24 Mar 2012	24 Mar 2012	24Mar 2012	11:15:00	12:15:12	12:15:12

The comparison between related work and proposed technique is shown in tabular form in Table X.

TABLE X. COMPARISON OF TECHNIQUES

Features	Related Work	Proposed Technique
Date checking	Comparing date and time of Received: and Date: fields as a single entity against margin	Comparing date and time of Received: and Date: fields as two different entities and difference is compared against margin
Margin calculation	No appropriate technique is given	Technique is given to calculate margin and the detection will be more accurate
Place of Investigation	Implemented in server	Post attack investigation is implemented at the victim
Intermediate server breakdown	Not considered	Considered
Updation of margin	Not considered	Updation of margin helps in getting better and accurate results
Implementation of algorithm	Not mentioned	Implemented in Java and C.

VI. CONCLUSION AND FUTURE WORK

E-mail is an important application and needs to be subjected to a number of security measures. E-mail date and time spoofing was analyzed in this paper. We created a spoofed E-mail dataset using Java Mail API classes. We have analyzed the header information of these E-mails. After performing the extensive header analysis of such E-mails, we have proposed an algorithm to detect date and time spoofing. We have also given an algorithm to calculate margin. The proposed algorithm is implemented over the dataset of E-mails. We are currently creating an application to automatically read the headers, convert to UTC and check for mail spoofing. Our future work is to find out whether the source address was spoofed and provide a mechanism to investigate the source of E-mails.

REFERENCES

[1] M. Toorani, "SMEmail - A New Protocol for the Secure E-mail in Mobile Environments," Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian , vol., no.,pp.39-44, Dec. 2008

[2] Symantec Analyst Relations, "Symantec Intelligence Report: February 2012," Symantec Corporation, 2012. http://www.symantec.com/connect/blogs/symantec-intelligence-report-february-2012 [Accessed April 10, 2012]

[3] J. Klensin, "Simple Mail Transfer Protocol," RFC 2821, April 2001.

[4] M. Wong and W. Schlitt, "Sender policy framework (SPF) for authorizing use of domains in e-mail," RFC 4408, April 2006.

[5] B. Radvanovsky, "Analyzing Spoofed E-mail Headers," *Journal of Digital Forensic Practice,* vol. 1, pp. 231-243, 2006

[6] G. Malik, "Tackling Spam and Spoof E-mail," in *First Conference on Advances in Computing and Technology* London: ICGES Press, 2006, pp. 65-71

[7] M. T. Banday, F. A. Mir, J. A. Qadri, and N. A. Shah, "Analyzing Internet E-mail date-spoofing," *Digital Investigation,* vol. 7, pp. 145-153, 2011

[8] M. T. Banday, "Analysing E-Mail Headers for Forensic Investigation," *Journal of Digital Forensics, Security and Law,* vol. 6, pp. 49-64, 2011

[9] M. T. Banday, "Algorithm for detection and prevention of E-mail date spoofing" *International Journal of Computer Applications*, vol. 06, pp. 7-11, 2011

[10] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, "Towards an integrated E-mail forensic analysis framework," *Digital Investigation,* vol. 5, pp. 124-137, 2009

[11] EmailTracer, Resource Center for Cyber Forensics – India, http://www.cyberforensics.in/OnlineEmailTracer/index.aspx