Algorithm for Detection and Prevention of Email Date Spoofing

M. Tariq Banday
P.G. Department of Electronics and Instrumentation Technology
University of Kashmir, India

ABSTRACT
Different security protocols offer different levels of security to insecure Simple Mail Transfer Protocol. These protocols vary considerably in degree of efficiency and adaptability. E-mail system not only suffers from various well known message integrity problems like spamming, phishing, sender spoofing, etc., but it also experiences a lesser know problem of date spoofing. This paper briefly appraises date spoofing and threats it can cause to e-mail and other e-systems. It also illustrates processes to send and receive date spoofed e-mail messages. Further, it lists solutions to the problem of date spoofing and proposes a model including necessary algorithms to detect and stop transmission and reception of date spoofed e-mail messages.

General Terms
E-mail, Spoofing, Internet mail, Authentication, Algorithms, Network Security, E-mail Security.

Keywords
E-mail Date Spoofing, Security Protocols, E-mail Security, SMTP, Mail Transfer Process.

1. INTRODUCTION
Privacy, authentication, message integrity, non-repudiation, and consistency are the key parameters in achieving security in e-mail system. Various protocols (SMTP) [1] that either creates an encrypted channel between the sender's and the recipient's servers or use some domain validation standards are being used over the insecure Simple Mail Transfer Protocol [2, 3] to attain security in e-mail system. Techniques to achieve message integrity which ensure security against mail forgery suffer from a problem of date spoofing as these techniques do not detect or stop transmission or reception of e-mail messages that are forged in date. Date spoofing can cause manifold of problems to e-mail system and other e-systems that use e-mail for communication.

The remaining text of the paper is organized as follows: section 2 describes date spoofing, its threats to email system, sending and receiving of date spoofed e-mails and their treatment by SMTP servers, section 3 describes the existing SMTP mail transaction process, and section 4 presents techniques towards the solutions to the problem of date spoofing. It also proposes a model and algorithms for detection and prevention of date spoofing, followed by conclusion in section 5.

2. DATE SPOOFING
Date spoofing emerged as an e-mail spoofing trick, wherein a spammer sends spam e-mails that contain forged send date to recipients. It keeps e-mails listed on top in recipient mailbox, thereby maximizing the chances of immediate attention by the recipient. The 'Date' header field in a date spoofed e-mail may contain a date which is ahead or before the actual date it was sent. Accordingly, a date spoofed e-mail may be either a pre-dated or a post-dated message. Date spoofing has been reported by Banday et al in [4], who have conducted a detailed study of handling of such e-mail messages by some commercial and corporate e-mail servers. It has been reported that: a) almost all studied e-mail servers accept e-mail messages spoofed in date, however, some reject post-dated e-mail messages spoofed in date by more than two days, b) on webmail systems which use send date as a sort field, post-dated e-mail messages remain on top in the inbox of the recipients, c) send date sorting and short date formats in some webmail systems make even learnt recipients difficult to suspect a mail being spoofed in date, and d) date-spoofed e-mails are not trapped by spam filters. Further, spam and spam filtering is known to considerable percentage of e-mail users but majority of e-mail users are unaware of date-spoofing.

2.1 Threats from Date Spoofing
Date spoofing aids spamming by increasing chances of opening spam e-mails and impediments its countermeasures due to increase in false positives [4]. The problem of date spoofing is not only limited to spamming, it can cause manifold of problems, more serious than spamming. Permitting submission, transmission or reception of either pre-dated or post-dated e-mail messages can not only lead to confusions and wastage of time of their recipients but also can inflict threats to several other electronic services and systems that use e-mail for communication, record and reference. These include e-commerce, e-tendering, e-evaluation, e-transactions, etc. In these, e-mail before or after a particular date and time is inacceptable as response within some stipulated time is mandatory. A user may trick these systems by sending response after the expiry of deadline by sending pre-dated e-mail. As a consequence, a dispute over the correctness of date can cause a protracted legal battle between the contending parties.

2.2 Sending and Receiving Date Spoofed E-mails
SMTP [2] is used for sending e-mail from the sender's client to the sender's server and it is also used for its transmission from the sender's SMTP server to the recipient's SMTP server. However, SMTP is not used for accessing the recipient's mailbox on recipient's SMTP server. This is to permit e-mail access using diverse technologies via online, offline and disconnected access models [5].

An e-mail message is send using an e-mail client computer running a client application like Outlook Express, Office Outlook, Eudora or other similar mail client application that connects to the sender's SMTP server which accepts the e-mail

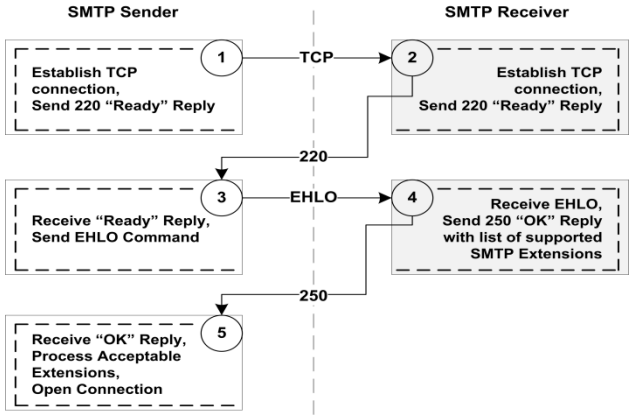
message using SMTP commands. The sender's SMTP server next transmits this message to the recipient's SMTP server using SMTP commands. In Web based e-mail services such as mail.yahoo.com and gmail.google.com clients and servers are combined and are integrated behind a web server. The web based e-mail is much easier to use than other online access methods, however, they are inflexible as the users do not have direct access to their mailboxes and can use only features implemented by the website provider. Sending SMTP servers do not check the authenticity of the date in an e-mail message arriving from the SMTP client and thus permit transmission of date spoofed e-mail messages.

Date spoofed e-mails can be sent from any e-mail client program either by directly altering the send 'Date' header field or by changing the clock of the computer running the e-mail client program. Custom e-mail programs or bulk e-mail tools may also be used to send date spoofed e-mails by directly altering the send 'Date' header field of the message. Programs like Outlook Express, Office Outlook, Eudora, etc. can be used to send date spoofed e-mail messages by temporarily altering the clock of the client computer before sending the message which generates the spoofed 'Date' header for the message from the altered clock of the client computer. It is also possible to establish a direct connection with sending or receiving SMTP servers through Telnet and send e-mail messages spoofed in 'Date' header field by directly issuing SMTP commands. Web based e-mail services do not allow inclusion of sender controlled 'Date' header field and instead take date from the system clock of the server and as such cannot be used to send date spoofed e-mail messages.

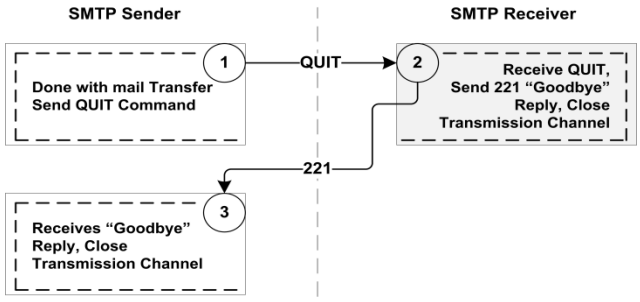
An e-mail message is received by the SMTP server of the recipient and is stored in recipient's mailbox on that server. The recipient can use online, offline or disconnected access models to retrieve the mail to his local mailbox using protocols like POP3 [6] or IMAP4 [6]. Besides these, a user can also use webmail interfaces to check e-mail messages stored on his SMTP server. Different SMTP servers treat incoming date spoofed e-mails differently. Most of them accept both pre and post dated e-mails and do not classify them as spam while other do not accept date spoofed e-mails spoofed beyond certain offset from the current date or some classify them as spam. Different mail access programs and webmail interfaces list date spoofed emails differently. Some programs and web mail interfaces like MS Outlook, Yahoo, Zapak, HUSH, GMX, GAWAB, etc. besides other field use both send and received date as sorting fields, while others like Inbox, Gamil, Mail, OVI, etc. besides other fields use only send or received date as sorting field. Among those which do not use send date fields as sorting field in listing e-mails, some display the send date field on opening the e-mail message. Further, e-mail retrieval protocols do not check for the correctness of date field(s) in e-mail messages.

3. MAIL TRANSFER PROCESS

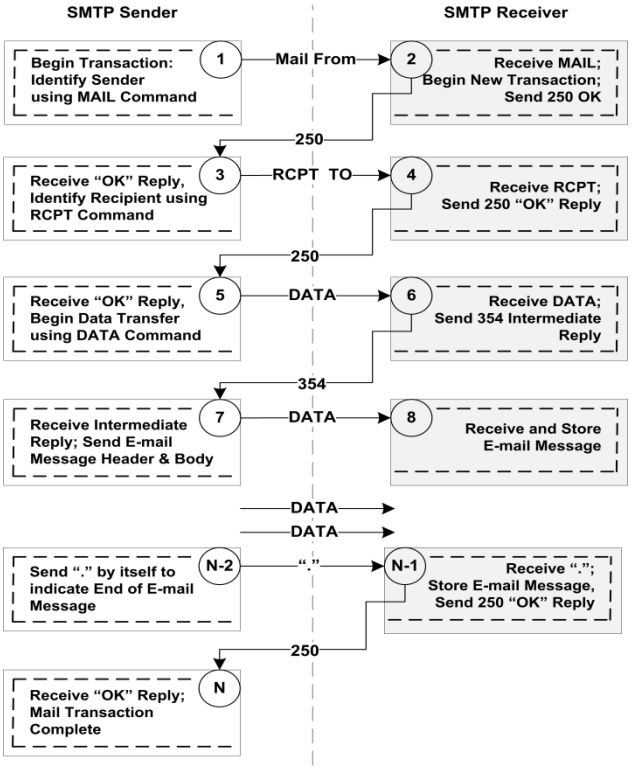
The process of sending e-mail from the sender's client to the sender's server or its transmission from sender's SMTP server to recipient's SMTP server consists of a) connection establishment, b) mail transactions and c) connection termination which are illustrated in figure 1.



(a) Connection Establishment



(c) Connection Termination



(b) Mail Transactions

Fig 1: Mail Transfer Process

Connection is established by the creation of a TCP connection on an ephemeral TCP port. The receiver sends connection acceptance reply using a code 220. The response also includes server information including full server name and the version of the SMTP server software. The client on receiving the connection ready reply issues HELO/EHLO in case of ESMTP which also includes the domain name of the client. The SMTP server after receiving the HELO/EHLO command, responses with service code 250 along with the supported ESMTP extensions. In case, the receiver does not support extensions, it replies with a service code 500. Mail transactions are performed to transfer the mail from the sender to the receiver. The MAIL command which includes the sender identification is issued by the sender for which the receiver responses with a go ahead service code 250. The receiver may validate the sender and also may reject e-mail reception for security reasons. On receiving the service code 250, the sender specifies recipients using one or more RCPT TO commands. Again the server responds with a go ahead service code 250 or may reject the e-mail reception. Some authentication mechanism may be indicated, to perform an authentication protocol exchange, optionally negotiate a security layer for subsequent protocol interactions during this session. Mail is transmitted through several transactions using DATA command after completion of transfer of e-mail envelope, Transmission of a period "." through DATA command marks the end of the transmission. The server stores the e-mail in the mailbox and issues a service reply code 250. The mail transaction is terminated by the sender and the receiver. Neither sender nor receiver ensures correctness of date field(s) during the mail transmission process and as such date spoofed e-mail messages are transmitted and received just like normal e-mail messages. QUIT command is used by the SMTP sender to terminate the SMTP session. The receiver on receiving the QUIT command, issue a service code 221 that indicates successful connection termination.

4. SOLUTIONS TO DATE SPOOFING

Banday et al in [4] have shown how header analysis can be carried out on a suspected e-mail message to detect possible date spoofing in it, however, they have neither proposed any model nor have given its implementation details. They have also mentioned some techniques that may be used to stop e-mail date spoofing. These are: a) not to trust the send date field and/or not to use it at all in e-mail programs, b) sending and receiving servers check the correctness of send date field before transmission or acceptance of the e-mail message, c) mail programs use both send and received date fields while listing e-mail messages, d) make existing security protocols especially DKIM sign date header field, and e) use some trusted time stamping service.

These solutions suffer from several potential disadvantages. Significance of send date field is lost if it is not used or not trusted by recipients. Correctness of date can be validated if all servers and clients are time synchronized. Using both send and received date fields in mail programs will not stop transmission of date spoofed fields, however, will help recipients in locating date spoofed e-mails. For including trusted date in e-mail message, date and time requires to be fetched from some third party time server. Further, both sending and receiving servers must trust same time server.

4.1 Proposed Model for Detection of Date Spoofing

Date spoofing can be effectively checked at: a) sending server by stopping the transmission of date spoofed e-mails, and b) receiving servers by discarding the reception of date spoofed e-mail messages. In addition to these, mail programs and webmail interfaces can use both send and received date fields while listing e-mails to make its detection easy for recipients.

An E-mail message while being transmitted from the sender's client to recipient's mailbox is handled by several intermediate nodes, which may add date to the message through one or more received also called trace fields. Thus, date and time is included in an e-mail message in 'Date', 'Resent-Date' if the e-mail is resent after some initial delivery failure and 'Received' header fields. The details of these fields are given in table 1.

Table 1: Description of fields containing date and time

Field Name	Set By	Field Description
Date:	Originator	It holds date and time when the message was made available for delivery.
Resent-Date:	Mediator	When manually forwarding a message, resent date field refers to the forwarding, not to the original message.
Received:	Originator, Relay, Mediator, Destination	The "Received:" field contains a (possibly empty) list of tokens followed by a semicolon and a date and time specification. It contains trace information that includes originating host, Mediators, relays, and MSA host domain names and/or IP addresses.

Each date and time entry includes day of month, month of year and year in multiple formats besides optional name of day of week. The date and time specification as per the RFC 5332 is given in table 2:

A date-time specification must be semantically valid. That is, the day-of-week (if included) must be the day implied by the date, the numeric day-of-month must be between 1 and the number of days allowed for the specified month (in the specified year), the time-of-day must be in the range 00:00:00 through 23:59:60 (the number of seconds allowing for a leap second; and the last two digits of the zone must be within the range 00 through 59 [7]. As detailed in previous sections, the process of sending an e-mail message from sender's client to the sender's server or its transmission from sender's server to the recipient's server comprises of three processes. In this section, a modified mail transaction process is proposed to detect date spoofed e-mail message at the sender's server and at the recipient's server. The connection establishment and the connection termination processes remain unchanged in this proposed model.

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.

Table 2: Date and Time Specifications

Field/Token Name	Format	Remarks
date-time	[day-of-week ","] date time [CFWS]	The day-of-week is optional and CFWS denotes "Comments and Folding White Spaces" that indicate paces where header folding can take place.
day-of-week	([FWS] day-name)	The day-name is three letter abbreviation of day of week. FWS denotes "Folding White Spaces" that indicate paces where header folding can take place. This can take any value from the set: "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun".
date	day month year	The date includes day, month and year. The day format is ([FWS] 1*2DIGIT FWS). It is an allowable numerical day of month. The month is short name of month and can take any value from the set: "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" / "Jul" / "Aug" / "Sep" / "Oct" / "Nov" / "Dec". The format for year is (FWS 4*DIGIT FWS). It is any numerical year 1900 or later.
time	time-of-day zone	The format for time-of-day is hour ":" minute [":" second]. It specifies the number of hours, minutes, and optionally seconds since midnight of the date indicated. Hour, minute and second are any valid two numerical digits.
zone	(FWS ("+" / "-") 4DIGIT)	The zone specifies the offset from Coordinated Universal Time (UTC, formerly referred to as "Greenwich Mean Time") that the date and time-of-day represent. The "+" or "-" indicates whether the time-of day is ahead of (i.e., east of) or behind (i.e., west of) Universal Time. The first two digits indicate the number of hours difference from Universal Time, and the last two digits indicate the number of additional minutes difference from Universal Time.

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 to recipient's mailbox. If the difference between send date/resent date and the first received date or difference between the first received date and second received date and so on or if the difference between the last received date and the current date (date from the clock of the server) is beyond some predefined margin, the mail may be discarded with a notification to the sender, otherwise, mail is stored in the mailbox of the recipient. This is shown in figure 2. In case the relevant date and time contained in the e-mail message does not agree with the system clock of the receiver, the receiver will send an error code indicating permanent failure and end the mail transaction. This permanent failure indicating message integrity problem can be resolved by correcting date in the e-mail message [8].

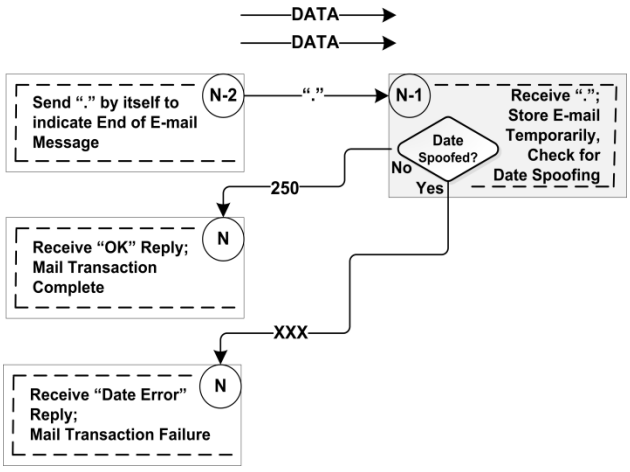


Fig 2: Modified Mail Transaction Process

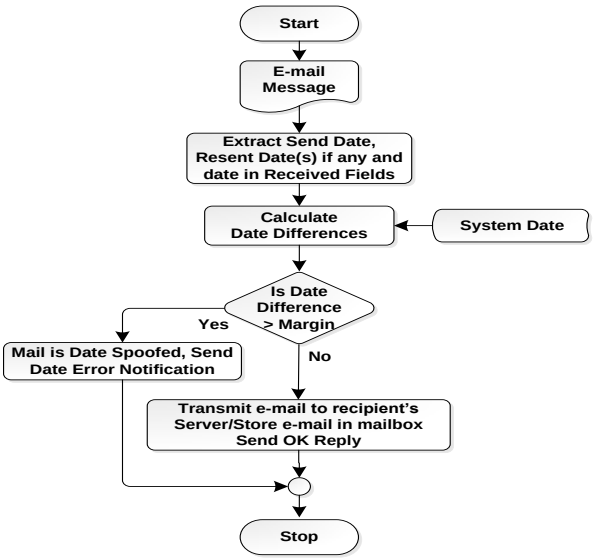


Fig 3: Algorithm to detect date spoofing

The margin may be decided on the type of conversation, e.g. it can be a few hours for scheduled activities e.g. tendering systems, evaluation reports, RFP submissions, Form submissions, etc. where response within a stipulated time is mandatory and a few days for non-scheduled activities.

The algorithm to determine correctness of send date at the receiving client and server is depicted in figure 3.

After the reception of a period "." through DATA command which marks the end of the transmission, the e-mail message is temporarily stored in the receiving client or server which is transmitted to the receiving server or stored in the mailbox of the recipient on the receiving server only if the mail is not spoofed in date. The algorithm for calculating the date difference between various dates found in the e-mail message is shown in figure 4.

```
DateTime DateTimeDiff (Mail M)
    Get system date/time in SysDT
    if (Received Filed is present in M) do
            RecentRecDT=0
            while (EOF (M)) do
                    Get date/time from Received Field in RecDT
                    if (RecentRecDT < RecDT) then RecentRecDT= RecDT
            Calculate date/time difference between SysDT and RecentRecDT in DTDiff
            Return DTDiff
    else if (Resent Filed is present in M) do
            RecentResDT=0
            while (EOF (M)) do
                    Get date/time from Resent Field in ResDT
                    if (RecentResDT < ResDT) then RecentResDT= ResDT
            Calculate date/time difference between SysDTand RecentResDT in DTDiff
            Return DTDiff
    else
            Get date/time from Send Date Filed in SenDT
            Calculate date/time difference between SysDT and SenDT in DTDiff
            Return DTDiff
```

Fig 4: Algorithm for calculating date difference

This algorithm requires conversion of time from time zones, which obeys the following relationship: ***Time in Time Zone A – UTC Offset from Zone A = Time in Zone B – UTC Offset for Zone B.***

Thus, we can rearrange the above relationship to get Time in Zone B. ***Time in Zone B = Time in Zone A – UTC Offset for Zone A + UTC Offset for Zone B.***

Thus, if an e-mail message has been send from Delhi in India having an UTC offset of +05:30 at time 20:00, it should be received by an e-mail server in Los Angles having an UTC offset of -08:00 at ***Time in Delhi – UTC Offset for Delhi + UTC Offset for Los Angles + Transmission Time***, which is equal to ***06:30 + Transmission Time***. The Transmission Time is the time taken by an e-mail message to reach the destination. This time depends upon several parameters that include transmission speed and the delays introduced by intermediate nodes.

For the operability of this scheme without allowing some marginal difference between send date and times, it is essential that the sending server, receiver server and all other intermediate nodes ensure correctness of their clocks. However, it is very risky to implement such a solution because an e-mail message travels through several intermediate nodes that run different software's, have different time delays and all of them may not be time synchronized.

5. CONCLUSION

E-mail date spoofing is a message integrity issue of e-mail systems which existing security protocols and procedures have failed to address. This form of spoofing can cause problems in various other network services that directly or indirectly use e-mail system. E-mails spoofed in date can be transmitted using e-mail client programs by altering the 'Date' header field or by changing the system clock. Almost all existing e-mail systems accept date spoofed e-mail messages. Existing mail transmission process does not ensure correctness of date in the mail being transmitted. In theory some solutions to the problem of date spoofing have been presumed. This paper proposes a model to detect date spoofed e-mail messages and stop their transmission. Algorithms for detection of date spoofed e-mail messages have also been proposed. These algorithms can be implemented in any programming language and incorporated in the existing mail system. The algorithm has been successfully tested by implementing it using Java library.

6. REFERENCES

[1] Banday, M.T., and Qadri, J.A. 2010. A Study of E-mail Security Protocols. eBritian, ISSN: 1755-9200, British Institute of Technology and E-commerce, UK, Issue 5, Summer 2010, pp. 55-60. Available online at: http://www.bite.ac.uk/ebritain/ebritain_summer_10.pdf.

[2] Klensin. 2001. Simple Mail Transfer Protocol. IETF RFC 2821, Apr 2001.

[3] Resnick, P. Ed. 2001. Internet Message Format. IETF RFC 2822, Apr 2001.

[4] Banday MT, et al. 2010. Analyzing Internet e-mail date-spoofing, Digital Investigation, doi:10.1016/j.diin.2010.11.001.

[5] Banday, M. T. 2011. Design and Development of Efficient Techniques for Securing E-mail System from threats, PhD Thesis, University of Kashmir, India.

[6] Tzerefos, Smythe, Stergiou and Cvetkovic. 1997. A comparative study of Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) and X.400 Electronic Mail Protocols. In Proceedings of the 22nd Annual IEEE Conference on Local Computer Networks, pp. 545–554.

[7] Resnick, P. Ed. 2008. Internet Message Format. IETF RFC 5332, Oct 2008.

[8] Vaudreuil, G. 2003. Enhanced Mail System Status Codes. IETF RFC 3463, Jan 2003.