3-12-2008

Email 'Message-IDs' helpful for forensic analysis?

Satheesaan Pasupatheeswaran
Edith Cowan University

DOI: 10.4225/75/57b2735e40cbe
6th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2008.

Email 'Message-IDs' helpful for forensic analysis?

Satheesaan Pasupatheeswaran

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia
spasupat@student.ecu.edu.au
ptheesan@yahoo.com

Abstract

Finding the source of spoofed email is a challenging task for forensic investigators. Header of an email has several fields that can be used for investigation. An investigator can easily understand the evidences embedded within most of the header fields of an email, except the message-id field. Therefore, there is a need to understand how message-ids are constructed and what useful information can be recovered from them. The immediate aim of the analysis is to find the message-id construction mechanism of 'Sendmail' mail transfer agent (MTA) version 8.14 and how the findings can be used successfully in forensic analysis. Source code of the 'Sendmail' MTA is made use of during analysis. This analysis will uncover several information that will help to find email source and validate other email header fields also. The drawbacks in message-id based forensic analysis also discussed here.

Keywords

E-mail header, message-id, msgid, sendmail forensics, e-mail forensics, e-mail header analysis, network forensics.

INTRODUCTION

An electronic mail consists of two parts, the header and the body. The header part carries information that is needed for email routing, subject line and time stamps while the body contains the actual message/data of an email. The header and the body are separated by a blank line. The header contains several mandatory and optional fields (Resnick, 2001). In order to uniquely identify each email all mail transfer agents (MTAs) use some sort of unique identifier. This identifier is referred to as 'Message-ID'. Message-ID field is inserted into a header either by mail user agent (MUA) or the first MTA. Even though the Message-ID is optional as per RF2822 it recommends using it. Sendmail is one of MTA that handles email delivery and relaying process. Sendmail uses message-id for tracing emails and for logging process ids (Costales, Janse, Abmann, & Shapiro, 2007, p1160). Sendmail recommends including message-id in emails and also it recommends setting relevant macros in its configuration file in order to implement compulsory checking of message-ids (Costales et al, 2007, p776). Unlike spoofing other fields in the header, spoofing message-id needs special knowledge. Only technical envy spammers can spoof the message-id cleverly. So deep analysis on message-ids may reveal some sort of information that will open a window to trace the source of an email. Also the message-id will help to find a particular email log entry within a log file of email server.

Like conventional mail service, when e-mail is routed from source to destination all intermediate relay servers (SMTP) insert their stamp at the beginning of the header. This stamping procedure helps to trace the email if such a demand arises. The stamp consists of three fields known as 'From', 'SMTP ID', and 'For'(Klensin, 2001) . Figure 1 shows an email header that passed through several MTAs. Each MTA inserted a unique-id in the header of email (Al-Zarouni, 2004). There are several IDs in the header field of an email that may help to trace the source of the email but this discussion is limited to sendmail message-id only. Analyzing intermediate SMTP IDs is beyond the scope of the discussion. However this paper briefly discusses intermediate SMTP-Ids also.

```
Received: from search.org ([64.162.18.2]) by sgiserver1.search.org with SMTP (Microsoft Exchange
Internet Mail Service Version 5.5.2650.21)
id K9HBB4C4; Mon, 21 May 2001 09:47:01 -0700
Received: from web14506.mail.yahoo.com ([216.136.224.69]) by SEARCH.ORG
with SMTP (IPAD 2.52) id 3579700; Mon, 21 May 2001 08:47:23 -0800
Message-ID: <20010521164640.85785.qmail@web14506.mail.yahoo.com>
Received: from [216.104.228.118] by web14506.mail.yahoo.com; Mon, 21 May 2001 09:46:40 PDT
Date: Mon, 21 May 2001 09:46:40 -0700 (PDT)
From: <can_do1@yahoo.com>
Subject: check out this e-mail header
To: todd@search.org
```

Figure1: Email header with several identifiers (ID)

Message-ID

RFC 2822 states that each email must have a globally unique identifier. This must be included into the header of an email. The RFC 2822 also defines the syntax of message-id. It should be like a legitimate email address and it must be included within a pair of angle brackets. According to RFC 2822, message-id can appear in three header fields. They are 'message-id header', 'in-reply-to header' and 'references header'. But message-id of the present email must be included against the 'message-id' header.

Sendmail Message-ID

Sendmail Message-ID is formatted with two parts and they are connected by '@'sign. It looks like a legitimate email address. Right hand side (RHS) of the @ sign is a fully qualified domain name (FQDN) and left hand side (LHS) of the @ sign has two parts separated by '.'. The LHS part is created with date, time, process id and a few random numbers. Shown below is a sample message-id.

Message-ID: <200712141511.d872mLVW024467@cs.slt.edu>

Message-ID is always included within a pair of angle brackets. FQDN makes the MTA globally unique. The date and time with the combination of process id and special random numbers make the message unique in a particular MTA. This combination makes message-ids globally unique. Figure 1 shows sample sendmail header field (Costales et al, 2007, p7). Message-id is typed in blue bolded font.

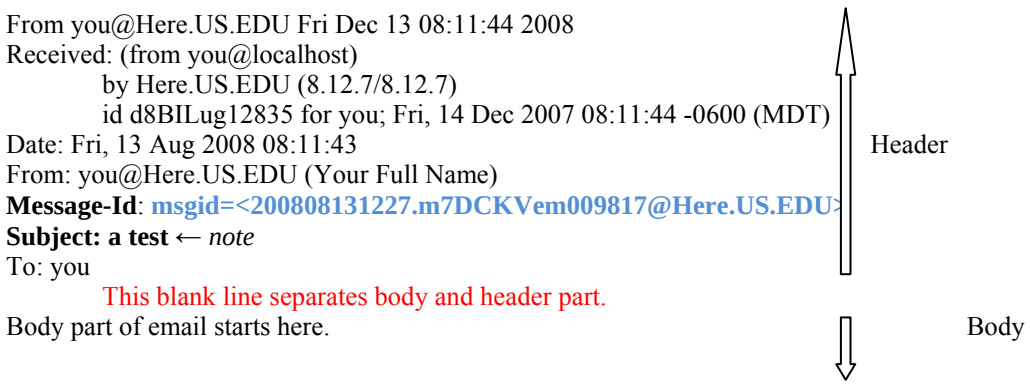
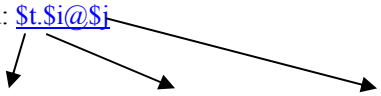


Figure2: Sample sendmail email structure

MESSAGE-ID GENERATION

Sendmail message-id is defined in the following format (Costales et al., 2007, p776).

Message-id: $t.$i@$j



E.g.: 200808131227.m7DCKVem009817@Here.US.EDU

Following paragraphs discuss each part of message-id.

$t

$t macro is a current UTC date and time. This is formatted in yyyymmddhhmm. It consists of 12decimal values. In the above e.g. the $t part is **200808131227**. If it is decoded the final results will be 2008-08-13 12:27. That means the email is handed over to delivery or delivered at 12:27 on 13-08-2008 UTC (Sendmail, 2007).

$i

$i is referred as a queue id. It is generated with a special algorithm. Queue id has three different formats with respect to sendmail versions. Queue id versions are categorised as 'before V8.6', 'starting with V8.6' and 'starting with V8.10'. Format of queue-id with respect to sendmail versions are given below (Costales et al., 2007).

Before V8.6 → AApid
From V8.6 → hourAApid
From V8.10 → YMDhmsSEQpid

Following paragraphs will present a brief description about components 'AA' and 'hour' and discusses sendmail V8.14 in detail.

AA

'AA' is a combination of English alphabet and other characters. RHS clocks from A-Z (26 characters) and LHS clocks from A- ~ (62 characters) until it generates a unique-id. This provides more than 1600 combinations (Costales et al., 2007, p397).

AA

AB

. *So on...*

AZ

.*So on...*

~Y

~Z ← *failure*

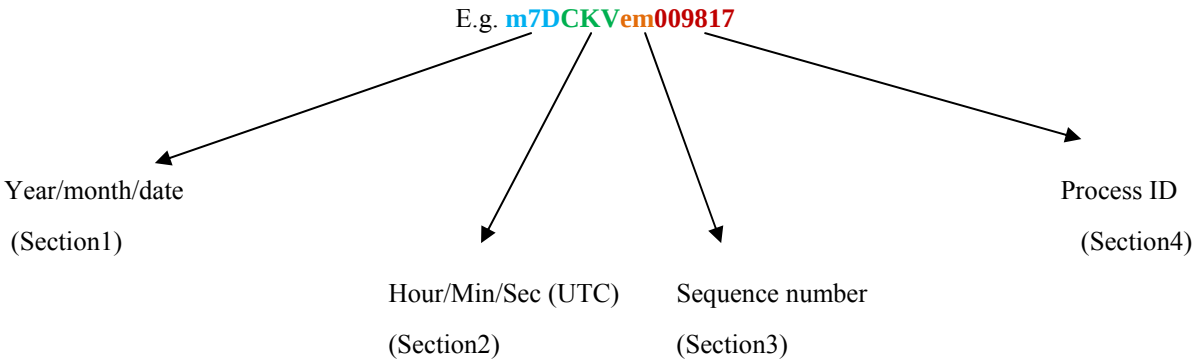
So on...

hour

This maps 24 hour clock to uppercase alphabet. The time starts at midnight and midnight 12 is mapped as A. Then 1' o' clock is B and so on (Costales et al., 2007, p397).

Sendmail V8.14

The message ID of V8.14 consists of three parts. The below example clearly indicates each part (Sendmail, 2007). In order to make it more understandable each group of components are named as 'section x' where x=1, 2, 3, 4 within brackets directly below each description.



The first eight characters can be of any combination from the characters given in table 1(Costales et al., 2007, p397). The last 6 digits are process id.

Decimal Numbers	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Mapping character	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L

22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	b	a	b	c	d	e	f	g

44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	x	y	z

Table1: Mapping table

Section1

This part is current UTC time. Number of years is calculated from 1900 and then is divided by 60. The reminder is mapped to its relevant single character value (Costales et al., 2007, p397). See below example

Formula: Reminder (R) = (Current year- 1900) % 60
Reminder = (2008-1900) % 60 = 48
Map 48 in table 1→ 'm'

Months January through December is numbered from 0 to 11. Therefore number 7 must be August.

Date is represented by 'D'. From the map table it is 13th.

Hence the encoded year, month and date is 2008-08-13.

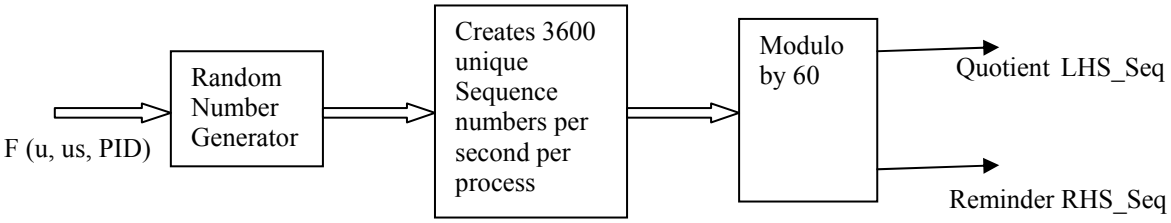
Section2

This is current UTC time. This is ordered as hour, minute and seconds. This is coded as hour, minute modulo 62 and seconds modulo 62 (Sendmail, 2007). Reverse mapping of each letter will decode the originating UTC time of the message.

C→ 12, K→ 20, V→ 31. So the email is originated or submitted for delivery at 12:20:31 UTC.

Section3

This is referred to as sequence number. These two are generated from a random number. Right hand side number is quotient and left hand side is reminder (modulo) of a random number. Seed of the random number is created with dynamic unique numbers in order to make the best possible random number. The time period since epoch to current time is calculated in seconds and microseconds. Then the total number of seconds, microseconds and process id is summed up. This sum is used as seed for the random number generator Figure 3 shows the sequence number generation process (Sendmail, 2007).



Where F (u, us, pid) = seconds (since epoch) + microseconds+ Process-ID

Figure3: Sequence numbers generator

Section4

This is a 6 digit process identifier (PID or Process-ID). This process ID is relevant to the process that attempted to deliver the email (Hunt, 2001). Sendmail tries to make Process-ID unique for each queuing process (Costales et al., 2007).

$j

This macro represents the fully qualified domain name (FQDN). This part starts with local host name followed by a dot and other parts of domain information (Costales et al., 2007). Domain names are globally unique. In our previous example the $j part was **Here.US.EDU**. Local host name is 'Here' and the local domain name is 'US.EDU'.

TRACING E-MAIL

In Message-ID generation section we found that the factors used to construct message-id themselves carried important information that can be used to trace source of an email. The following paragraphs discuss how this will help trace the source of an email message.

$j: Fully qualified domain name

FQDN contains local host name, from where the email was originated or the first sendmail MTA, and other domain information. In our previous example Here.US.EDU the first part, preceding the first dot is the local host or the first MTA server name. Right side of the first dot is other domain information. Once domain name is found then domain's point of contact and other domain registration details can be found with readily available tools (Nelson, Philips, Enfinger, & Steuart, 2008, p484). Some of such web based tools are www.arin.net, www.internic.com, www.freeality.com and a command line tool is 'whois' (Mulligan, 1999, p32). Once the domain administrator is identified then forensic analysers can get her/his help to track the source with message-id.

$t: Date and time

Time is a critical factor in forensic investigation. The time part of message-id provides when the message was handed over for delivery. This time information will help to solve some of the problem stated below.

Dynamic IP addressing

In order to conserve IP address space most ISPs provide dynamic IP addresses. During investigation if IP address of the sender is found to be dynamic then the time information will help to search in the billing server who used this particular IP address at the specified time. This will help to identify the email sender. Billing servers contain session information such as period of login and allocated IP address for billing purposes. If sender used company's SMTP server then both SMTP log and DHCP log must be collected for analysis (Al-Zarouni, 2004).

Remember the time retrieved from message-id is UTC. So it is important to find out actual zone time. This can be done in several ways. Country of origin of email can be found from domain name as discussed in section $j. Once the country is known then time difference can be determined from several timing servers. Even this will help to verify the originating date and time of the email. The calculated time will help forensic analysers to check whether the source MTA is in sync with any standard time reference or not.

Email-server log file

Sendmail records all SMTP communication between servers in mail.log file. This log file contains date, host, process-id, queue-id and the log information. This log file maintains queue-id as a unique-id to distinguish each record (Costales et al., 2007, p517). By analysing the log file of source MTA with either message-id or queue-id, the expected record can be found. Time stamp and hostname/IP address found in the record can be verified with suspected email header so as to confirm the sender. Below figure shows a typical sendmail log file.

```
Aug 13 20:18:54 ubuntu-jeos sm-mta[9746]: m7DCGa9j009632: to=[redacted]@student.ecu.edu.au, delay=00:01:15, xdelay=00:00:00, mailer=esmtp, pri=120050,
relay=mailhost.ecu.edu.au. [139.230.225.12], dsn=2.0.0, stat=Sent (CJC70872 Message accepted for delivery)
Aug 13 20:22:09 ubuntu-jeos sm-mta[9495]: m7DC6Hs8009489: to=[redacted]@openduck.com, delay=00:15:28, xdelay=00:15:12, mailer=esmtp, pri=120033,
relay=aspmx2.googlemail.com. [209.85.135.27], dsn=4.0.0, stat=Deferred: Connection timed out with aspmx2.googlemail.com.
Aug 13 20:24:41 ubuntu-jeos sm-mta[9512]: m7DC8BfZ009507: to=[redacted]@kronicd.net, delay=00:16:11, xdelay=00:16:01, mailer=esmtp, pri=120014,
relay=aspmx5.googlemail.com. [66.249.83.27], dsn=4.0.0, stat=Deferred: Connection timed out with aspmx5.googlemail.com.
Aug 13 20:26:12 ubuntu-jeos sm-mta[9817]: m7DCKVem009817: Syntax error in parameters scanning "to"
Aug 13 20:28:53 ubuntu-jeos sm-mta[9843]: m7DCMQm0009843: sl00016c8168a2.ads.ecu.edu.au [10.77.11.59] did not issue MAIL/EXPN/VRFY/ETRN during co
Aug 13 20:30:42 ubuntu-jeos sm-mta[9817]: m7DCKVem009817: from=[redacted]@student.ecu.edu.au, size=136, class=0, nrcpts=1,
msgid=<200808131227.m7DCKVem009817@ubuntu-jeos.u.suk>, proto=SMTP, daemon=MTA-v4, relay=sl00016c815c1e.ads.ecu.edu.au [10.77.11.74]
Aug 13 20:30:42 ubuntu-jeos sm-mta[10140]: m7DCKVem009817: to=[redacted]@student.ecu.edu.au, delay=00:03:39, xdelay=00:00:00, mailer=esmtp, pri=120
```

Message-ID Queue-ID

Figure4: Typical sendmail log file

The source MTA may be maintained by ISP or it may belong to a company. Forensic analysers will require legal authorisation to access the log files.

In-Reply-To header

In-reply-to header holds message-id of original message to which it is replying to. Also this may include comma separated several message-ids, as a reply to several emails (Costales et al., 2007, p1158). Checking this header will help to find other suspicious emails.

```
In-Reply-To: <847.193925.780455@hostA.com>, <1021169802.330@HostB.co.th>,
        <200106020731.BAA20313@HostC.br.ca>
```

Figure5: An in-reply-to header with a few message-ids

References header

In case of threaded emails, continuous correspondence between parties, the reference header holds all message-ids from the first email to the last email (Loshin, 2000, p 92). Supposing the message-id of the interested email is spoofed the other message-ids will help to trace the email source. Figure 7 shows reference header with two message-ids.

Masquerade options

Sendmail and even some of the other MTAs like Microsoft exchange server support an option called masquerade. This option is used to hide the local host behind a local domain name or central email server. It usually rewrites sender address field with local domain name. Then any outgoing email will not have FQDN or $j. It might be tricky when tracing email with source address domain name. But this option does not affect FQDN of message-id (Costales et al., 2007, p600). In most of the situations analysing a message-id will help to directly locate the local host or the mail server which handled the initial delivery process. Figure 6 shows an email header with masqueraded source address.

```
Return-Path: <abcxyz@unsw.edu.au>

Received: from smtp.unsw.edu.au ([127.0.0.1])
        by localhost (snarl.comms.unsw.edu.au [127.0.0.1]) (amavisd-new, port 10025)
        with ESMTP id j2nhwsBjmdj7 for <satheesaan@slt.com.lk>;
        Mon, 28 May 2007 12:52:09 +1000 (EST)
Received: from central12-eng.eng.unsw.edu.au (central12-eng.eng.unsw.EDU.AU [129.94.131.112])
        by smtp.unsw.edu.au (8.13.6/8.13.6) with ESMTP id l4S2q9Gm010024;
        Mon, 28 May 2007 12:52:09 +1000 (EST)

Date: Mon, 28 May 2007 12:53:57 +1000                    Local host or Local mail server

Message-ID: <D9307185226FDB4FA388C58AE2A347859D34C7@central12-eng.eng.unsw.edu.au>

In-Reply-To: <C280794D.BDAC%eng.faculty@unsw.edu.au>

References: <200705250328.l4P3ShY5024597@smtp.slt.com.lk>   Reference header with multiple
msgid
        <C280794D.BDAC%eng.faculty@unsw.edu.au>

From:  abcxyz@unsw.edu.au   Masquerade with local domain name
To:    <sssssss@slt.com.lk>
```

Figure 6: Reference and masquerade header

In the above figure 'From:' field has local domain name 'unsw.edu.au' but 'Message-id field' and 'Received field' have the local email server that handled the first delivery process. If investigators can provide the email server details that handled the email delivery process, it will help to speed up the process. Consequently it will reduce the burden of log file analysis.

Intermediate SMTP-ID

As emails go through intermediate MTAs (hops) each MTA insert their unique-id (SMTP-ID) on the email header. If it is necessary to analyse intermediate server log file this unique-id is important. Knowledge about

intermediate smtp-id also will help to identify any fake smtp-id. Intermediate sendmail servers create a queue id as stated in section $i and use it as smtp-id (Costales et al., 2007, p826). Intermediate mail servers stamp starts with 'Received:' header. Figure6 shows an email that is routed through three sendmail MTAs. Each MTA insert their stamp with smtp-id. This smtp-id can be used in log file analysis.

```
Return-Path: <satheesaan@slt.com.lk>
Authentication-Results: mta379.mail.mud.yahoo.com from=slt.com.lk; domainkeys=neutral (no sig)
Received: from 203.115.19.199 (EHLO xmail.slt.com.lk) (203.115.19.199) by
          mta379.mail.mud.yahoo.com with SMTP; Fri, 23 Jun 2006 02:24:38 -0700
Received: from smtp.slt.com.lk (smtp.slt.com.lk [172.25.1.100]) by xmail.slt.com.lk (8.12.11/8.12.11) with
          ESMTP id k5N9LmIO004855; Fri, 23 Jun 2006 14:51:48 +0530
Received: from slt.com.lk (pop.slt.com.lk [172.25.1.101]) by smtp.slt.com.lk      (8.12.10/8.12.10) with
          ESMTP id k5NKrSuA013912; Fri, 23 Jun 2006 14:53:28 -0600 (GMT)
Received: from slt.com.lk (slt.com.lk [127.0.0.1]) by slt.com.lk (8.13.1/8.13.1) with ESMTP
          id k5N8rfWP021228; Fri, 23 Jun 2006 14:53:51 +0600
From:  <satheesaan@slt.com.lk>
To: camalan@celetronix.com.uk,
Subject: Fw: you've got to see this Date: Fri, 23 Jun 2006 15:53:41 +0700
Message-Id: <20060623085301.M4685@slt.com.lk>
In-Reply-To: <000701c6969f$78ff94e0$374019ac@RAIN>
```

Figure7: Intermediate SMTP-ID

The pattern of smtp-id and reverse mapping of smtp-id proves that smtp-id is a queue-id.

FAKE MESSAGE-ID

Just like spoofing other header fields of email, spoofing message-id is also possible. By observing a few email headers where the first MTA is sendmail, it is possible to make a message-id that look legitimate.

E.g.: 200808131227.m7DCKVem009817@Here.US.EDU

LHS of the dot is simply date and time and RHS of the dot contains 14 characters, first 8 characters are a combination of numbers and English alphabets and other 6 are just numbers.

So before using message-id for forensic analysis the message-id must be verified for its validity.

Message-id verification

Knowledge of sendmail message-id construction will help to verify the message-id. With the help of mapping table, $t part can be verified with first 5 characters of $i. The sequence number and process id are dynamically created characters so verifying them is difficult.

Spam identification

Spam mail filters check for empty message-id or illegal message-id pattern only. The message-id is an optional field and it also can be spoofed. So message-id cannot be a reliable spam indicator(Allman, Assmann, & Shapiro, n.d).
Spam mail senders harvest email addresses through several ways one such method is scanning USENET articles (Costale & Flynt, 2005). If any email received from known source is suspected to be sent by spammers, the suspected email can be verified by comparing message-id of the email against known good email message-id from the same source. However, checking message-id is not a consistent spam checking method because a good spammer can create same pattern of message-id.

ISSUES RELATED TO MESSAGE-IDS

No Standard algorithm

RFC2822 standard states every email should have a unique identifier and provides syntax of message-id and some suggestion to create unique identifier. However, it does not define how it should be generated. Email software developers use their own algorithm to generate message-ids. Forensic analyser or relevant technical advisor must be well informed on the different vendor message ids as he/she might come across different types of message-ids.

This drawback makes it difficult to make a tool for checking validity of message-ids. Sendmail checks message-id header, if it is blank it will insert a new message-id otherwise it will not alter the available message-id (Costales et al., 2007, p834). This vulnerability aids the successful transmission of emails with spoofed message-ids. Spoofed message-id will compromise forensic analysis results.

Open source and closed source

In case of open source softwares it is possible to find out the construction mechanism of message-ids but it will be difficult to determine the message-id construction mechanism in closed source softwares.

Identifying Source MTA

There are several MTAs in use. In order to select the suitable analysis procedure investigator must know the source MTA. If the source MTA is known it will help to verify the message-id against fake ids. Sendmail will not generate new message-id if the email already has a message-id. Some MUA also generate message-ids (Costales et al., 2007, p1159). Even the first MTA is a sendmail; the message-id might not be sendmail compatible. In this case first smtp-id will help log analysis. This area needs the special attention of researchers.

Versions

The message–id algorithm of sendmail has already changed thrice (Costales et al., 2007, p387). Therefore for analysis, continuous research and updates on message-ids is important. Determining the version of the sendmail is also necessary before start of message-id analysis.

Host time

MTA host time must be synchronized with reliable time reference. Since forensic investigation is time sensitive, if there is any difference it time it may invalidate the case in court or it may be very difficult to prove in court. There are some tools, such as NTP, STIME and GPS clock, that can be used to synchronise the host time (Al-Zarouni, 2004). Incorrect timing and time setting may cause message-id collision in the specific host itself.

Spoofed message-ids

Spoofing email message-ids is possible and it will compromise the forensic analysis. If message-id is spoofed with an earlier valid email message-id then this will change the direction of the investigation. This will create unnecessary problems and delay in the investigation. Figure8 shows an email header with spoofed message-id.

```
Return Path: <dhjmifpo@msn.com>
Received: from 200.94.239.104 (HELO 216.136.129.5) (200.94.239.104) by
mta136.mail.sc5.yahoo.com with SMTP; Sun, 27 Jun 2004 17:14:03
Received: from 162.134.15.76 by 200.94.239.104; Sun, 27 Jun 2004 20:11:02 -
Message-ID: <P[20
```

Fighure8: An email header with spoofed message-id from my inbox

Headers without message-id

Some emails, especially drafted for illegal activity or spam, do not have message-ids in their headers. In such circumstances message-id forensic is not applicable. Below figure shows a successfully delivered webmail without message-id.

```
Return-Path: <good@mta463.mail.mud.yahoo.com>
Authentication-Results:  mta463.mail.mud.yahoo.com from=yahoo.com;
                domainkeys=neutral
 Received: from 122.44.118.105 (HELO fpyd.net) (122.44.118.105) by
         mta463.mail.mud.yahoo.com with SMTP; Fri, 05 Sep 2008 16:04:41 -0700
From: <ptheghost@yahoo.com>
To: <ptheghost@yahoo.com>
Subject: Hurry.. Buy US based medications here !..save your money ! MIME-Version: 1.0 Content-Type:
multipart/mixed;boundary= "----=_NextPart_000_00CA_A0C54188.80C58DC2" Content-Length: 690
```

Figure9: Fake header without message-id

In the above header both 'From' and 'To' header fields have same address. Thus it is confirmed that the email is a fake. Also it does not have a message-id.

International cooperation

Message-ID based forensics analysis needs log file analysis. In some occasions the source server might be located in another country. To handle this type of situation investigator needs cooperation from that foreign country to carry out the analysis successfully.

CONCLUSION:

This discussion reveals that email message-id plays an important role in email forensic analysis. The global unique feature of message-id helps to distinguish each email and so help in forensic analysis. Knowledge of message-id construction part will help to identify spoofed emails, source host, email log file analysis and time details. This paper also discussed the ways to determine fake message-ids. Beyond some of the identified weaknesses in message-id, the information that is carried by the message-id is highly important in tracing the email source.

This study is carried out only on sendmail message-id. However this area needs more study on other message-ids that are created by different email software. The key factor in message-id analysis is that the source email software must be known to the investigator in order to apply suitable methods during analysis.

REFERENCES:

Al-Zarouni, M. (2004). *Tracing E-mail Headers*. Retrieved 02-Sep-2008, from http://scissec.scis.ecu.edu.au/publications/forensics04/Al-Zarouni.pdf

Allman, E., Assmann, C., & Shapiro, G. N. *Sendmail Installation and operation Guide*. Retrieved 03-Sep-2008, from http://www.sendmail.org/doc/sendmail-current/doc/op/op.pdf`

Costales, B., & Flynt, M. (2005). *Sendmail Milters AGuide for Fighting Spam*. NJ: Addison-Wesley.

Costales, B., Janse, G., Abmann, C., & Shapiro, G. N. (2007). *Sendmail* (4th ed.). Sebastopol: O'Reilly.

Hunt, C. (2001). *Linux Sendmail Administration*. Retrieved 04-Sep-2008, from http://library.ecu.edu.au/

Klensin, J. (2001). *RFC2821:Simple Mail Transfer Protocol*. Retrieved 01-Sep-2008, from http://www.ietf.org/rfc/rfc2821.txt

Loshin, P. (2000). *Essential Email Standards: RFCs and Protocols Made Practical*. NY: Wiley.

Mulligan, G. (1999). *Removing the Spam: Email Processing and Filtering*. Reading: Addison-Wesley.

Nelson, B., Philips, A., Enfinger, F., & Steuart, C. (2008). *Guide to Computer Forensics and Investigations* (3rd ed.). Boston: THOMSON COURSE TECHNOLOGY.

Resnick, P. (2001). *RFC2822: Internet Message Format*. Retrieved 01-Sep-2008, from http://www.ietf.org/rfc/rfc2822.txt

Sendmail [Computer Software]. (2007). *Sendmail* (Version V8.14.2): Sendmail Inc.

COPYRIGHT