

How to Interpret Email Headers

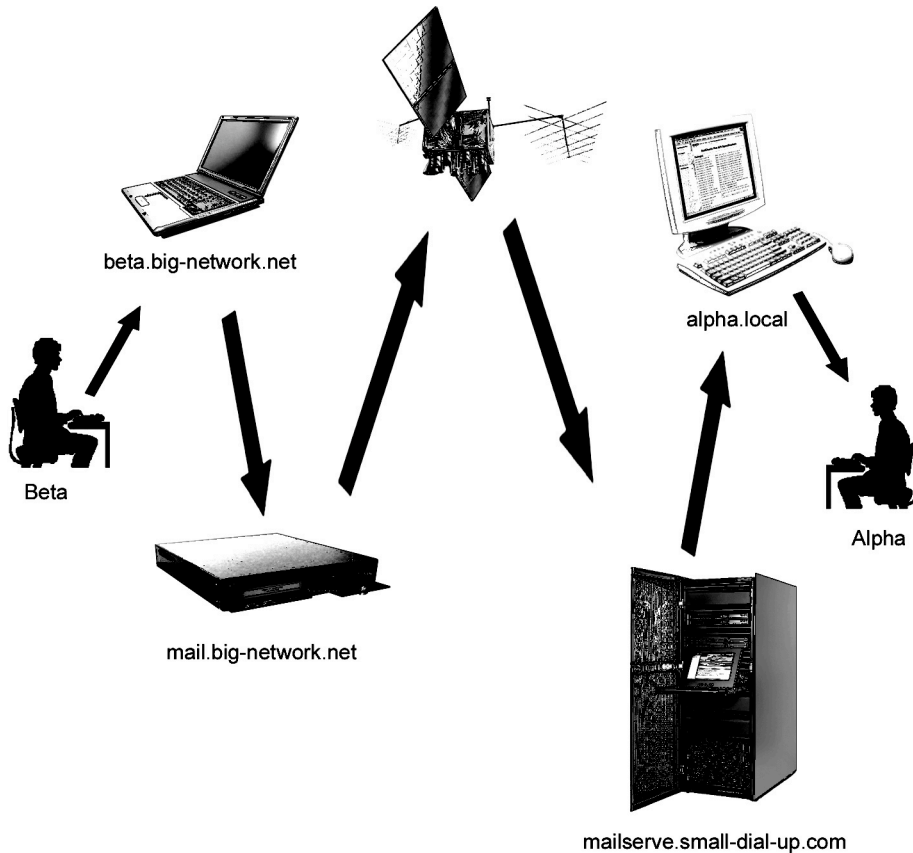
By: G.E. Investigations, LLC & Team Majestic Designs, LLC

Some of you may ask, What is an email header? Basically, an email header is the part of an email that comes before the body of the letter and contains information about the email including the senders email and date sent among other things. The header of an email is the return address and route label of an email. But wouldn't it be nice if you could tell what server your email comes from? Or perhaps what servers your email had to go through before it came to you? Say hello to the Extended Header. An extended header includes much more than the subject, sender, recipient, and date and time. The extended header is a documentation of the life of the email.

Here is an overview of the e-mail process. This background material is important to understand how the headers are generated. Email does not transfer from the sender's computer directly to the recipients usually. Generally, a single message passes through at-least 4 computers during its delivery. This is because the message is created on your machine, then it is sent out to your ISP's email server which will then route it. After that, it gets sent to the recipients mails server, which is where sits until he/she downloads his/her e-mail. This is the common life of an e-mail. We will go into abnormal situations later.

I will create a situation here for you to relate to. In order to do this I will make some fictitious users to create a situation. We will use (alpha@small-dial-up.com) and (beta@big-network.net). Alpha is a home dial up user and Beta is on a big network.

When Beta wants to email a message to Alpha, he will create the message on his computer (called beta.big-network.net). The message will then be sent from his computer to his mail server (mail.big-network.net). After this point, Beta has no control of the message and it will be processed and sent by other computers out of his control. The mail server then sees that a message has arrived for a user at small-dial-up.com and contacts its mail server (mailserve.small-dial-up.com) and delivers the message. Then the e-mail is stored on mailserve.small-dial-up.com until Alpha dials into his ISP and downloads his email onto his computer. See Example 1.



(Example 1)

During this processing, headers will be added three times. First at composition by Beta's mail program. The next set of headers are added when the mail program sends it to mail.big-network.net and then again at the transfer to mailserve.small-dial-up.com. Normally, dial-up servers do not add any headers. Now you will see a step by step evolution of the headers.

Created by Beta's mail program and sent to mail.big-network.net:

From: beta@big-network.net (Beta)
To: alpha@small-dial-up.com
Date: Wed, Sep 14 2005 13:58:47 MST
X-Mailer: OutMailer v3.5
Subject: How are you?

Then, when mail.big-network.net sends the message to mailserve.small-dial-up.com:

Received: from beta.big-network.net (beta.big-network.net [123.123.1.23]) by mail.big-network.net (1.2.3) id 001A23; Wed, Sep 14 2005 13:58:49 MST
From: beta@big-network.net

To: alpha@small-dial-up.com
Date: Wed, Sep 14 2005 13:58:47 MST
Message-Id: <beta091705135847-000000123@mail.big-network.net>
X-Mailer: OutMailer v3.5
Subject: How are you?

Followed by the when mailserve.small-dial-up.com finishes processing the message and stores it to receive:

Received: from mail.big-network.net (mail.big-network.net [123.123.1.23]) by mailserve.small-dial-up.com (2.3.2) id 001A23 with ESMTP id LAA100968 for <alpha@small-dial-up.com>; Wed, Sep 14 2005 13:58:56 MST
Received: from beta.big-network.net (beta.big-network.net [123.123.1.23]) by mail.big-network.net (1.2.3) id 001A23; Wed, Sep 14 2005 13:58:49 MST
From: beta@big-network.net (Beta)
To: alpha@small-dial-up.com
Date: Wed, Sep 14 2005 13:58:47 MST
Message-Id: <beta091705135847-000000123@mail.big-network.net>
X-Mailer: OutMailer v3.5
Subject: How are you?

The last set of headers will be the one that the recipient sees. In this case, Alpha will dial in and download his mail into his mail program.

Here is a line-by-line analysis of each header and what they mean.

Received: from mail.big-network.net

This message came from a computer supposedly named mail.big-network.net

(mail.big-network.net [123.123.1.23])

The servers real name is mail.big-network.net. Meaning that it identified itself correctly and it's IP address is 123.123.1.23

by mailserve.small-dial-up.com (2.3.2)

The receiving computer was mailserve.small-dial-up.com; It's running a mail program called mailsend version 2.3.2.

with ESMTP id LAA100968

The receiving machine assigned the message the id number of LAA100968. This is an ID for system administrators to find a message in a system log.

for <alpha@small-dial-up.com>;

The message was addressed to alpha@small-dial-up.com. Note that this header is not related to the "To:" header.

Wed, Sep 14 2005 13:58:56 MST

The e-mail was transferred on Wednesday, September 14 2005 at 13:58:56 Mountain Standard Time.

Received: from beta.big-network.net (beta.big-network.net [123.123.1.23]) by mail.big-network.net (1.2.3) id 001A23; Wed, Sep 14 2005 13:58:49 MST

This header documents the transfer from beta.big-network.net (Beta's computer) to mail.big-network.net and the time it occurred. The sending machine called itself beta.big-network.net and was actually named beta.big-network.net (meaning that it was true) with the IP address 123.123.1.23. It also says, that Beta's mail server is running mailsend version 1.2.3 assigning the message the ID 001A23 for internal logs.

From: beta@big-network.net (Beta)

The e-mail was sent by beta@big-network.net, who gives the real name of Beta.

To: alpha@small-dial-up.com

The letter is addressed to alpha@small-dial-up.com.

Date: Wed, Sep 14 2005 13:58:47 MST

The message was created on Wednesday, September 14 2005 at 13:58:37 Mountain Standard Time.

Message-Id: <beta091705135847-000000123@mail.big-network.net>

The e-mail has been assigned this ID number by mail.big-network.net to identify it. This ID is unique from the SMPT and ESMTP ID numbers in the "Received:" headers because it is assigned to this message forever. The other IDs are only associated to specific mail transfers from specific machines giving those ID's no meaning to any other system. Sometimes the Message-ID has the senders e-mail address embedded in it but often has no understandable meaning.

X-Mailer: OutMailer v3.5

The message was sent using a program called OutMailer, version 3.5

Subject: How are you?

Obviously, the subject of the message as entered by the sender.

A quick note, all the servers, e-mails, and any specific information has been changed to protect the privacy of the real users.

About Mail Protocols

This section will educate you on the matter of spoofed emails. This section has been adapted from "StopSpam.org's Reading Email Headers by Ken Lucke". It is the leading paper on the subject of interpreting email headers.

Mail Protocols

This section is a little more technical than the others, and focuses on the details of how mail gets from one point to another. You don't need to understand every word, but familiarity with this subject can do a lot to clarify what's happening in strange situations. Since email spammers often intentionally create such strange situations (partly to confuse their victims), the ability to understand those situations can be quite helpful.

To communicate over a network, computers often use "points of entry" called **ports**; you might think of a port as a channel through which a computer can listen to communications from the network. To listen to many communications at once, a computer needs to have multiple ports; to distinguish them, they're generally numbered. On systems connected to the Internet (or any systems using the same protocols for email), port 25 is of particular importance for the present discussion; that's the port that's used to transmit and receive mail.

Normal Behavior

Let's return to the example of the last section, and specifically to the point where mail.bieberdorf.edu communicates with mailhost.immense-isp.com. What really happens here is that mail.bieberdorf.edu *opens a connection to port 25* of mailhost.immense-isp.com, and sends the mail through that connection, along with some administrative data. The commands it uses to do this, and the responses issued by the receiving system, are more or less human-readable; they're commands in a rudimentary language called **SMTP**, for Simple Mail Transfer Protocol. Someone eavesdropping on the "conversation" between the machines would see something like the following transcript (the commands issued by mail.bieberdorf.edu are in boldface):

```
220 mailhost.immense-isp.com ESMTP Sendmail 8.8.5/1.4/8.7.2/1.13; Tue, Mar 18 1997 14:38:58 -0800 (PST)
```

```
HELO mail.bieberdorf.edu
```

```
250 mailhost.immense-isp.com Hello mail.bieberdorf.edu [124.211.3.78],
```

pleased to meet you

MAIL FROM: rth@bieberdorf.edu

250 rth@bieberdorf.edu... Sender ok

RCPT TO: tmh@immense-isp.com

250 tmh@immense-isp.com... Recipient ok

DATA

354 Enter mail, end with "." on a line by itself

Received: from alpha.bieberdorf.edu (alpha.bieberdorf.edu [124.211.3.11]) by mail.bieberdorf.edu (8.8.5) id 004A21; Tue, Mar 18 1997 14:36:17 -0800 (PST)

From: rth@bieberdorf.edu (R.T. Hood)

To: tmh@immense-isp.com

Date: Tue, Mar 18 1997 14:36:14 PST

Message-Id: <rth031897143614-00000298@mail.bieberdorf.edu>

X-Mailer: Loris v2.32

Subject: Lunch today?

Do you have time to meet for lunch?

--rth

.

250 LAA20869 Message accepted for delivery

QUIT

221 mailhost.immense-isp.com closing connection

This whole transaction depends on five commands which constitute the core of SMTP (there are a few others, but they're peripheral to the actual process of passing mail from one place to another): HELO, MAIL FROM, RCPT TO, DATA, and QUIT.

HELO identifies the sending machine; "**HELO mail.bieberdorf.edu**" should be read as "Hello, I'm mail.bieberdorf.edu". The sender can lie; nothing, in principle, prevents mail.bieberdorf.edu from saying "Hello, I'm frobozz.xyzyzy.gov" (**HELO frobozz.xyzyzy.gov**) or even "Hello, I'm a misconfigured computer" (**HELO a misconfigured computer**). However, in most circumstances, the receiver has some tools with which to discover this and find out the sending machine's real identity.

MAIL FROM initiates mail processing; it means "I have mail to deliver from so-and-so". The address given turns into the so-called "envelope From" (see Section Whatever); it need not be the same as the sender's own address! This apparent security hole is inevitable (after all, the receiving machine doesn't know anything about who has what username on the sending machine), and in certain circumstances it turns out to be a useful feature.

RCPT TO is dual to MAIL FROM; it specifies the intended recipient of the mail. One piece of mail can be sent to multiple recipients simply by including multiple RCPT TO commands (see the section on mail relaying, which explains how this feature is sometimes abused on insecure systems). The given address turns into the

so-called "envelope To" (see Section Whatever); it actually determines who the mail will be delivered to, *regardless of what the To: line in the message says*.

DATA starts the actual mail entry. Everything entered after a DATA command is considered part of the message; there are no restrictions on its form. Lines at the beginning of the message (before the first blank line) that start with a single word and a colon are considered to be headers by most mail programs. A line consisting only of a period terminates the message.

QUIT terminates the connection.

SMTP is fully defined in RFC 821. Copies of the RFCs are widely available on the Web; this one is well worth reading, as it sheds much light on the intricacies of mail processing.

Unusual Scenarios

The scenario above is a little bit oversimplified. The biggest assumption is that the mail servers of the two organizations involved have free access to one another. This was almost always true in the early days of the Internet, and it's still sometimes the case today, but as security has become a greater concern, and as organizations and networks have gotten bigger, sometimes requiring many separate mail servers, it's become more and more unusual.

Firewalls

Many, perhaps most, organizations with computers on the Internet are protected by some kind of *firewall*. A firewall is just a computer whose primary job is to act as a gatekeeper between an organization's own machines and the great unwashed world of the net (so that, for instance, crackers can't easily connect to a piece of IBM's corporate network and start stealing corporate secrets). From the standpoint of another computer trying to deliver mail to a system behind a firewall, what this means is that you can't talk directly to the system; you have to talk to the firewall.

No surprises here; this just introduces another "hop" in the journey of a piece of email, with the firewall acting as just another machine that passes mail.

If immense-isp.com had a firewall in place, here's what the headers from our sample piece of email might look like. Notice the first Received: line. (I'm assuming that the firewall machine is named firewall.immense-isp.com; in fact, giving a machine a name like "firewall" is tantamount to inviting every teenage cracker-wannabe in the world to try to break in, so firewalls usually have perfectly ordinary, innocuous names.)

**Received: from firewall.immense-isp.com (firewall.immense-isp.com
[121.214.13.129]) by mailhost.immense-isp.com (8.8.5/8.7.2) with ESMTP id**

LAA20869 for <tmh@immense-isp.com>; Tue, 18 Mar 1997 14:40:11 -0800 (PST)

Received: from mail.bieberdorf.edu (mail.bieberdorf.edu [124.211.3.78]) by firewall.immense-isp.com (8.8.3/8.7.1) with ESMTP id LAA20869

for<tmh@immense-isp.com>; Tue, 18 Mar 1997 14:39:24 -0800 (PST)

Received: from alpha.bieberdorf.edu (alpha.bieberdorf.edu [124.211.3.11]) by mail.bieberdorf.edu (8.8.5) id 004A21; Tue, Mar 18 1997 14:36:17 -0800 (PST)

From: rth@bieberdorf.edu (R.T. Hood)

To: tmh@immense-isp.com

Date: Tue, Mar 18 1997 14:36:14 PST

Message-Id: <rth031897143614-00000298@mail.bieberdorf.edu>

X-Mailer: Loris v2.32

Subject: Lunch today?

In similar fashion, if all outgoing mail from bieberdorf.edu were routed through a firewall, there would be another Received: line inserted by that firewall machine. By the same token, there might be machines involved that aren't strictly firewalls, but simply common points for routing---perhaps immense-isp.com maintains machines in many physical locations, with several separate mailservers, and uses a single machine (called, say, mailgate.immense-isp.com) to decide which server incoming mail should be routed to. Hence the following set of headers is a little extreme, but not implausible:

Received: from mailgate.immense-isp.com (mailgate.immense-isp.com [121.214.11.102]) by mailhost3.immense-isp.com (8.8.5/8.7.2) with ESMTP id LAA30141 for <tmh@immense-isp.com>; Tue, 18 Mar 1997 14:41:08 -0800 (PST)

Received: from firewall.immense-isp.com (firewall.immense-isp.com [121.214.13.129]) by mailgate.immense-isp.com (8.8.5/8.7.2) with ESMTP id LAA20869 for <tmh@immense-isp.com>; Tue, 18 Mar 1997 14:40:11 -0800 (PST)

Received: from firewall.bieberdorf.edu (firewall.bieberdorf.edu [124.211.4.13]) by firewall.immense-isp.com (8.8.3/8.7.1) with ESMTP id LAA28874 for <tmh@immense-isp.com>; Tue, 18 Mar 1997 14:39:34 -0800 (PST)

Received: from mail.bieberdorf.edu (mail.bieberdorf.edu [124.211.3.78]) by firewall.bieberdorf.edu (8.8.5) with ESMTP id LAA61271; Tue, 18 Mar 1997 14:39:08 -0800 (PST)

Received: from alpha.bieberdorf.edu (alpha.bieberdorf.edu [124.211.3.11]) by mail.bieberdorf.edu (8.8.5) id 004A21; Tue, Mar 18 1997 14:36:17 -0800 (PST)

From: rth@bieberdorf.edu (R.T. Hood)

To: tmh@immense-isp.com

Date: Tue, Mar 18 1997 14:36:14 PST

Message-Id: <rth031897143614-00000298@mail.bieberdorf.edu>

X-Mailer: Loris v2.32

Subject: Lunch today?

The history of the message can be reconstructed by reading the Received: headers from bottom to top; it went from alpha.bieberdorf.edu to mail.bieberdorf.edu to firewall.bieberdorf.edu to firewall.immense-isp.com to mailgate.immense-isp.com to mailhost3.immense-isp.com, where it waits for tmh to come along and read it.

Relaying

Here are some possible headers from a message that had a very different "life cycle" than anything described so far:

```
Received: from unwilling.intermediary.com (unwilling.intermediary.com
[98.134.11.32]) by mail.bieberdorf.edu (8.8.5) id 004B32 for
<rth@bieberdorf.edu>; Wed, Jul 30 1997 16:39:50 -0800 (PST)
Received: from turmeric.com ([104.128.23.115]) by
unwilling.intermediary.com (8.6.5/8.5.8) with SMTP id LAA12741; Wed, Jul
30 1997 19:36:28 -0500 (EST)
From: Anonymous Spammer <junkmail@turmeric.com>
To: (recipient list suppressed)
Message-Id: <w45qxz23-34ls5@unwilling.intermediary.com>
X-Mailer: Massive Annoyance
Subject: WANT TO MAKE ALOT OF MONEY???
```

A variety of things in this header might clue the reader in to the fact that this is a piece of electronic junk mail, but the thing to focus on here is the Received: lines. This message originated at turmeric.com, was passed from there to unwilling.intermediary.com, and from there to its final destination at mail.bieberdorf.edu. All well and good; but how did unwilling.intermediary.com get there, since it has nothing to do with either the sender or the recipient?

Understanding the answer requires some knowledge of SMTP. In essence, turmeric.com simply connected to the SMTP port at unwilling.intermediary.com and told it "Send this message to rth@bieberdorf.edu". It did this, probably, in the most direct manner imaginable, by saying **RCPT TO: rth@bieberdorf.edu**. At that point, unwilling.intermediary.com took over processing the message; since it had been told to send it to a user at some other domain (bieberdorf.edu), it went out and found the mail server for that domain and handed off its mail in the usual manner. This process is known as *mail relaying*.

Historically, there are good reasons for allowing relaying; on much of the net until about the late 1980s, machines rarely sent mail by talking directly to each other. Rather, they worked out a route for a message to travel, and sent it step by step along that route. It was a cumbersome system (especially since the sender often had to work out the route by hand!) By way of analogy, imagine sending a letter from San Francisco to New York, and having to address the envelope thus:

San Francisco, Sacramento, Reno, Salt Lake City, Rock Springs, Laramie, North Platte, Lincoln, Omaha, Des Moines, Cedar Rapids, Dubuque, Rockford, Chicago, Gary, Elkhart, Fort Wayne, Toledo, Cleveland, Erie, Elmira, Williamsport, Newark, New York City, Greenwich Village, #12 Desolation Row, Apt. #35, R.A. Zimmermann

It's clear why this is a useful addressing model if you're a postal worker---the post office in Gary, Indiana only has to be able to communicate with the adjacent offices in Chicago and Elkhart, rather than having to devote its resources to figuring out how to get something to New York. (It's also clear why this isn't a good idea from the standpoint of the letter-writer, and why email is no longer commonly routed this way!) This is exactly how email was sent; so it was important that one machine be able to give another instructions that said "I have email for rth@bieberdorf.edu, to be sent from you to turmeric.com to galangal.org to asafoetida.com to bieberdorf.edu". Hence relaying.

In modern times, however, relaying is usually used by unethical advertisers as a technique for concealing the source of their messages, deflecting complaints to the (innocent) relay site rather than to the spammers' own ISPs. (It also offloads the work of processing addresses and contacting recipients from the spammers' machines to those of an uninvolved third party; it's widely felt that relaying, especially large-scale relaying, constitutes theft of service for that reason.) The essential point here is to realize that the content of the message was formulated at the sending point---turmeric.com in the example above; the intermediate link, unwilling.intermediary.com, is involved only as an unwilling intermediary. They have no control over the sender, much as the Gary post office has no real influence over someone writing letters in San Francisco. (They do, however, have the power to turn off relaying at their site!)

One more thing to notice in the sample headers: The Message-Id: line was filled in, not by the sending machine (turmeric.com), but by the relayer (unwilling.intermediary.com). This is a common feature of relayed mail; it just reflects the fact that the sending machine didn't supply a Message-Id.

Envelope Headers

The section on SMTP, above, alluded to a distinction between "message" and "envelope" headers. This distinction and some of its consequences are detailed here.

Briefly, the "envelope" headers are actually generated by the machine that receives a message, rather than by the sender. By this definition, Received: headers are envelope headers; however, the term usually refers to the "envelope From" and "envelope To" only.

The envelope From header is the header derived from the information in a MAIL FROM command. For instance, if a sending machine says **MAIL FROM: ginger@turmeric.com**, the receiving machine will generate an envelope From header

that looks like this:

From ginger@turmeric.com

Notice the absence of the colon---"From", not "From:". Frequently, envelope headers don't have colons after them; this convention is not universal, but it is common enough to pay attention to.

Symmetrically, the envelope To is derived from a RCPT TO command. If the sender says **RCPT TO: tmh@bieberdorf.edu**, then the envelope To is tmh@bieberdorf.edu. There often isn't an actual header containing this information; sometimes it's embedded in the Received: headers.

An important consequence of the existence of envelope information is that **the message From: and To: headers are meaningless**. The contents of the From: header are provided by the sender; and so, counterintuitively, are the contents of the To: header. Mail is routed **only** based on the envelope To, never based on the message To: header.

To see this in action, consider an SMTP transaction like this:

HELO galangal.org

250 mail.bieberdorf.edu Hello turmeric.com [104.128.23.115], pleased to meet you

MAIL FROM: forged-address@galangal.org

250 forged-address@galangal.org... Sender ok

RCPT TO: tmh@bieberdorf.edu

250 tmh@bieberdorf.edu... Recipient OK

DATA

354 Enter mail, end with "." on a line by itself

From: another-forged-address@lemongrass.org

To: (your address suppressed for stealth mailing and annoyance)

.

250 OAA08757 Message accepted for delivery

Here are the corresponding headers (excerpted for clarity):

From forged-address@galangal.org

Received: from galangal.org ([104.128.23.115]) by mail.bieberdorf.edu (8.8.5) for <tmh@bieberdorf.edu>...

From: another-forged-address@lemongrass.org

To: (your address suppressed for stealth mailing and annoyance)

Notice that the contents of the envelope From, the message From:, and the message To: are all dictated by the sender, and have no bearing whatsoever on reality! This example illustrates why the From, From:, and To: headers can **never** be

trusted in mail that might be forged; they're simply too easy to falsify.

The Importance of Received: Headers

We've seen already, in the examples above, that the Received: headers provide a detailed log of a message's history, and so make it possible to draw some conclusions about the origin of a piece of email even when other headers have been forged. This section explores some details associated with these singularly important headers, and in particular how to circumvent common forgery techniques.

Unquestionably, the single most valuable forgery protection in the Received: headers is the information logged by the receiving host from the sender. Recall that the sender can lie about its identity (by putting garbage in its HELO command to the receiver); fortunately, modern mail transfer programs are able to detect such false information and correct it.

If, for instance, the machine turmeric.com, whose IP address is 104.128.23.115, sends a message to mail.bieberdorf.edu, but falsely says **HELO galangal.org**, the resultant Received: line might start like this:

Received: from galangal.org ([104.128.23.115]) by mail.bieberdorf.edu (8.8.5)...

(The rest of the line is omitted for clarity.) Notice that, although the bieberdorf.edu machine doesn't explicitly announce that galangal.org wasn't really the sending machine, it does record the correct IP address of the sender. If someone receiving the mail had reason to think that galangal.org appeared in the headers through the work of a forger, they could look up the IP address 104.128.23.115 (with a tool like the UNIX program nslookup) and find that that address in fact belonged to turmeric.com (not galangal.org). In other words, logging the IP address of the sending machine provides enough information to confirm a suspected forgery.

Many modern mail programs actually automate this process, looking up the name of the sending machine on their own. (The lookup process is called **reverse DNS** (for Domain Name Service)---"reverse" because it reverses the usual process of translating a name to an address for routing purposes.) If mail.bieberdorf.edu were using software that did this, the Received: line would start something like this:

Received: from galangal.org (turmeric.com [104.128.23.115]) by mail.bieberdorf.edu...

Here the forgery is crystal-clear; this line effectively says "turmeric.com, whose address is 104.128.23.115, reported its name as galangal.org". Needless to say, information like this is extremely helpful in identifying and tracking forged email! (For this very reason, spammers try to avoid using relaying machines that report reverse DNS information. Sometimes they even find machines that don't do the kind of IP logging described in the previous paragraph---though there aren't very many of those

around on the net any more.)

Another trick used by forgers of email, this one increasingly common, is to add spurious Received: headers before sending the offending mail. This means that the hypothetical email sent from turmeric.com might have Received: lines that looked something like this:

**Received: from galangal.org ([104.128.23.115]) by mail.bieberdorf.edu (8.8.5)...
Received: from nowhere by fictitious-site (8.8.3/8.7.2)...
Received: No Information Here, Go Away!**

Obviously, the last two lines are complete nonsense, written by the sender and attached to the message before it was sent.

Since the sender has no control over the message once it leaves turmeric.com, and Received: headers are always added at the top, the forged lines have to appear at the bottom of the list. This means that someone reading the lines from top to bottom, tracing the history of the message, can safely throw out anything after the first forged line; even if the Received: lines after that point look plausible, they're guaranteed to be forgeries.

Of course, the sender doesn't have to use obvious garbage; a really devious forger could create a plausible list of Received: lines like this:

**Received: from galangal.org ([104.128.23.115]) by mail.bieberdorf.edu (8.8.5)...
Received: from lemongrass.org by galangal.org (8.7.3/8.5.1)...
Received: from graprao.com by lemongrass.org (8.6.4)...**

Here the only dead giveaway is the inaccurate IP address for galangal.org in the very first Received: line. The forgery would be still harder to detect if the forger had written in correct IP addresses for lemongrass.org and graprao.com, but the IP mismatch in the first line would still reveal that the message had been forged and "injected" into the network at the site 104.128.23.115 (i.e., turmeric.com). However, most header forgeries are considerably less sophisticated, and the extra Received: lines are obvious garbage.

This is the end of the Mail Protocol section. This section featured part of Reading Email Headers by Ken Lucke.

The full paper can be downloaded at <http://www.stopspam.org/email/headers.html>

Listing of Common Headers

Apparently-To: Email with a long list of recipients will often have this header, with one recipient per line. This header is not common for legitimate email and are normally a sign of a mailing list. Although more recent mailing list programs do not generate a long and unnecessary list of headers like this.

Bcc: This stands for Blind Carbon Copy. This header should NOT appear in a received messages header. This header is like Cc: (see below), but appears invisible. This header for sending a message to several recipients without their emails appearing in the header. Blind Carbon Copies are popular with spammer and scammers since it can confuse inexperienced users who get mail that appears not to belong to them.

Cc: This means Carbon Copy. This header allows the sender to add multiple recipients similar to the "To:" header. The difference is really only in the name, although some mail programs process differently for replies.

Comments: This is a non-standard header field. It is often seen in the form of "Comments: Authenticated sender is <user@theirdomain.com>." This header is usually added by mail programs (especially Pegasus) to identify the sender. But beware, it is also added by hand with false information by many scammers as well.

Content-Transfer-Encoding: This header is related to MIME. It affects how MIME compliant programs interpret multimedia files.

Content-Type: A MIME header telling programs what to expect in the email. (i.e. picture, movie, program, etc)

Date: This is the time the message was sent or composed. It could also be the time it was sent from a mail server. It is also possible to forge this, but besides that, it may be inaccurate as many computers across the world have their clocks set wrong.

Errors-To: This header indicates an address for mail-generated errors to go to instead of the sender. This is not a common header because most senders want to know if an error occurred.

From (without colon) This is the envelope From discussed above.

From: This is the header telling you who sent this email. This section of the header is created when the email is sent from the sender, so it is possible that this is forged.

Message-Id: This is the Id for your message. It is for keeping track of your email by the mail program or even the mail server. It is usually assigned by the first mail server, and will follow the format of blahblahblah@server.com. The "blahblahblah" can be anything, from random numbers and rarely the senders username. If the Message-Id

header has been mailformed (empty header or no @ symbol) or the site in the message ID isn't the real site of origin, it is probably a forgery.

In-Reply-To: This is a Usenet header that sometimes appears in email. It gives the Message-Id of some previous message which is being replied to. It is unusual for this header to appear in email unless the mail is directly related to a Usenet. Spammers have used this to try to get passed filtration systems.

Mime-Version: (also MIME-Version) This MIME header specifies the version of the MIME protocol that was used by the sender. This mail header is not extremely important, as most modern mail programs auto detect MIME.

Newsgroups: This header is exclusively for email that is connected to Usenets, either email copies of Usenet posts or email replies to posts. In email copies, it specifies the newsgroup(s) to which the message was posted. For replies, it specifies the post the message is being replied to.

Organization: This the organization that the sender is part of. (i.e. Microsoft). It is possible that this is forged, as it is made by the sender.

Priority: A free-form header that assigns a priority to mail. Most mail programs do not use it. Spammers use it sometimes to make you read their messages.

Received: This header tells you where your email has been and is probably the most important header. There are usually multiple received headers, each one documenting the transfer of the email to each server until it reaches you. The most recent activity is at the top, and the oldest at the bottom of the list.

References: This header is uncommon in email other than for copies of Usenet posts. It is used to identify posts that are replies sent in email form. It is usually just a copy of the header of the Usenet. It can also appear in Usenet posts, giving the message ID of the post being responded to as well as references to the post.

Reply-To: This header specifies an address for replies to go to. This header can be used for legitimate purposes, such as showing your email address more clearly for responses. But it is also used by spammers to avoid getting responses. This is usually done in a few ways, either to collect all the angry responses into a "junk box", to send all the replies to a nonexistent mail box, or to an unaware victim.

Sender: This header is uncommon in email (X-Sender: is generally used instead) but appears especially in copies of Usenet posts. It is a more reliable identifier than the From: line.

Subject: The subject of the letter as entered by the sender.

To: This header tells you the recipient of the email (usually you). But, it is free form and

thus may not have an email entered at all.

X-headers These headers that start with a capital X and then a dash are added to headers as often as extra information. It is also the standard in which all non standard headers are to be entered as, but this is commonly violated.

X-Antivirus-Scanner: This tells you that the email has been checked for viruses, usually by your mail server. This by no means makes it safe, you should still use anti virus software at all times.

X-Confirm-Reading-To: This header requests an automated confirmation notice when the message is received or read. It is usually ignored, likely some program uses it.

X-Distribution: This header was added by the creator of Pegasus to help stop spam. When a user sends an email to a large amount of people the program automatically adds this header. It is explicitly intended for recipients to filter against.

X-Errors-To: This header is similar to Errors-To:, although it is less used.

X-Mailer: (also X-mailer:) This header tells you what mail client or software the sender is using. It should also tell you the version number of the software. (for example MailSend Version 3) This header can be useful for blocking spam, since specific mail programs were made for spamming.

X-PMFLAGS: This is a Pegasus mailer specific header. It essentially identifies the message as being sent from Pegasus Mail.

X-Priority: This is a priority field used most commonly by Eudora.

X-Sender: Similar to the Sender: header in Usenet posts. This header supposedly identifies the sender more reliably than the From: header. In reality, it is almost as easy to forge this tag, so it should still be treated with the same suspicion.

X-UIDL: This is a special identifier used by the POP protocol for getting mail from a server. It is usually added in-between the recipient's mail server and the recipient's actual mail software. If mail arrives with this header, it is most likely junk mail because there is no real use for this header, but spammers often add it on.

Recommended Reading & Links

Reading Email Headers By Ken Lucke

<http://www.stopspam.org/email/headers.html>

This is a great paper written on the subject of reading email headers. It was written with the intention to stop spammers or at least tell if a message is spam, but is also useful for learning how to figure out the true sender of an email. This paper covers most of what we have in this paper, and was an invaluable resource to us.

Request for Comments 2822 - SMTP Email Headers By P. Resnick, Editor

<http://www.rfc.net/rfc2822.html>

This is the technical documentation for SMTP header and how they should be implemented. Highly technical, but worth the read if you want to know everything about email. This replaces RFC 822.

Request for Comments 2821 - SMTP

By: J. Klensin, Editor

<http://www.rfc.net/rfc2821.html>

This is the technical documentation for SMTP. It is an invaluable insight into the process of SMTP. Again, highly technical, but necessary to read for complete knowledge of email. This replaces RFC 821.

<http://www.rfc.net/>

This the Request for Comments main page. Documentations on several topics can be found here.

<http://www.stopspam.org/>

A good anti spam website, which well written papers.

Bibliography

Reading Email Headers

By Ken Lucke

<http://www.stopspam.org/email/headers.html>

Request for Comments 2822 - SMTP Email Headers

By P. Resnick, Editor

<http://www.rfc.net/rfc2822.html>

Request for Comments 2821 - SMTP

By: J. Klensin, Editor

<http://www.rfc.net/rfc2821.html>