

# Reading Email Headers

## All About Email Headers

### Introduction

This document is intended to provide a comprehensive introduction to the behavior of email headers. It is primarily intended to help victims of unsolicited email ("email spam") attempting to determine the real source of the (generally forged) email that plagues them; it should also help in attempts to understand any other forged email. It may also be beneficial to readers interested in a general-purpose introduction to mail transfer on the Internet.

Although the document intentionally avoids "how-to-forge" discussions, some of the information contained in it might be turned to that purpose by a sufficiently determined mind. The author explicitly does not endorse malicious or deceptive falsification of email, of course, and any use for such purposes of the information contained in this document is contrary to its purpose.

Because of the nature of the examples in this document, there are several fictitious domain names with associated IP (Internet Protocol) addresses. The chance that some of these domain names may be used at some future time is, inevitably, nonzero. Similarly, all IP addresses used in the examples are unidentified at this writing, but they will undoubtedly be assigned someday. Naturally, nothing in this document is intended to reflect in any way on future users of these domain names or IP addresses.

### Where Email Comes From

This section consists of a brief analysis of the life of a piece of email. This background material is important for understanding what the headers are telling you.

Superficially, it appears that email is passed directly from the sender's machine to the recipient's. Normally, this isn't true; a typical piece of email passes through at least four computers during its lifetime.

This happens because most organizations have a dedicated machine to handle mail, called a "mail server"; it's normally not the same machine that users are looking at when they read their mail. In the common case of an ISP whose users dial in from their home computers, the "client" computer is the user's home machine, and the "server" is some machine that belongs to the ISP. When a user sends mail, she normally composes the message on her own computer, then sends it off to her ISP's mail server. At this point her computer is finished with the job, but the mail server still has to deliver the message. It does this by finding the recipient's mail server, talking to that server and delivering the message. It then sits on that second mail server until the recipient comes along to read his mail, when he retrieves it onto his own computer, normally deleting it from the mail server in the process.

Illustration.

Consider a couple of fictitious users, <rth@bieberdorf.edu> and <tmh@immense-isp.com>. tmh is a dialup user of Immense ISP, Inc., using a mail program called Loris Mail (which, by the way, is also fictitious); rth is a faculty member at the Bieberdorf Institute, with a workstation on his desk which is

networked with the Institute's other computers.

If rth wants to send a letter to tmh, he composes it at his workstation (which is called, let's say, alpha.bieberdorf.edu); the composed text is passed from there to the mail server, mail.bieberdorf.edu. (This is the last rth sees of it; further processing is handled by machines with no intervention from him.) The mail server, seeing that it has a message for someone at immense-isp.com, contacts its mail server---called, perhaps, mailhost.immense-isp.com---and delivers the mail to it. Now the message is stored on mailhost.immense-isp.com until tmh dials in from his home computer and checks his mail; at that time, the mail server delivers any waiting mail, including the letter from rth, to it.

Illustration.

During all this processing, headers will be added to the message three times: At composition time, by whatever email program rth is using; when that program hands control off to mail.bieberdorf.edu; and at the transfer from Bieberdorf to Immense. (Normally, the dialup node that retrieves the message doesn't add any headers.) Let's watch the evolution of these headers.

As generated by rth's mailer and handed off to mail.bieberdorf.edu:

**From: rth@bieberdorf.edu (R.T. Hood)**  
**To: tmh@immense-isp.com**  
**Date: Tue, Mar 18 1997 14:36:14 PST**  
**X-Mailer: Loris v2.32**  
**Subject: Lunch today?**

As they are when mail.bieberdorf.edu transmits the message to mailhost.immense-isp.com:

**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?**

As they are when mailhost.immense-isp.com finishes processing the message and stores it for tmh to retrieve:

**Received: from mail.bieberdorf.edu (mail.bieberdorf.edu [124.211.3.78]) by mailhost.immense-isp.com (8.8.5/8.7.2) with ESMTP id LAA20869 for ; 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?**

This last set of headers is the one that tmh sees on the letter when he downloads and reads his mail. Here's a line-by-line analysis of these headers and exactly what each one means.

**Received: from mail.bieberdorf.edu**

This piece of mail was received from a machine calling itself mail.bieberdorf.edu...

**(mail.bieberdorf.edu [124.211.3.78])**

...which is really named mail.bieberdorf.edu (i.e., it identified itself correctly---see Section Whatever for more on this) and has the IP address 124.211.3.78.

**by mailhost.immense-isp.com (8.8.5/8.7.2)**

The machine that did the receiving was mailhost.immense-isp.com; it's running a mail program called sendmail, version 8.8.5/8.7.2 (don't worry about what the version numbers mean unless you already know).

**with ESMTP id LAA20869**

The receiving machine assigned the ID number LAA20869 to the message. (This is used internally by the machine---it's something an administrator would need to know to look up the message in the machine's log files, but it's not usually meaningful to anyone else.)

**for <tmh@immense-isp.com>;**

The message was addressed to tmh@immense-isp.com. Note that this header is *not* related to the To: line (see Section Whatever).

**Tue, 18 Mar 1997 14:39:24 -0800 (PST)**

This mail transfer happened on Tuesday, March 18, 1997, at 14:39:24 (2:39:24 in the afternoon) Pacific Standard Time (which is 8 hours behind Greenwich Mean Time; hence the "-0800").

**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)**

This line documents the mail handoff from alpha.bieberdorf.edu (rth's workstation) to mail.bieberdorf.edu; this handoff happened at 14:36:17 Pacific Standard Time. The sending machine called itself alpha.bieberdorf.edu; it really is called alpha.bieberdorf.edu, and its IP address is 124.211.3.11. Bieberdorf's mail server is running sendmail version 8.8.5, and it assigned the ID number 004A21 to this letter for internal processing.

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

The mail was sent by rth@bieberdorf.edu, who gives his real name as R.T. Hood.

**To: tmh@immense-isp.com**

The letter is addressed to tmh@immense-isp.com.

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

The message was composed at 14:36:14 Pacific Standard Time on Tuesday, March 18, 1997.

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

The message has been given this number (by mail.bieberdorf.edu) to identify it. This ID is different from the SMTP and ESMTP ID numbers in the Received: headers because it is attached to this message for life; the other IDs are only associated with specific mail transactions at specific machines, so that one machine's ID number means nothing to another machine. Sometimes (as in this example) the Message-ID has the sender's email address embedded in it; more often it has no intelligible meaning of its own.

**X-Mailer: Loris v2.32**

The message was sent using a program called Loris, version 2.32.

**Subject: Lunch today?**

Self-explanatory.

## **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.xyzzzy.gov" (**HELO frobozz.xyzzzy.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. The picture above might be modified to look like this:

Illustration.

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 ; 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 beiberdorf.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.

## List of Common Headers

- **Apparently-To:** Messages with many recipients sometimes have a long list of headers of the form "Apparently-To: rth@bieberdorf.edu" (one line per recipient). These headers are unusual in legitimate mail; they are normally a sign of a mailing list, and in recent times mailing lists have generally used software sophisticated enough not to generate a giant pile of headers.
- **Bcc:** (stands for "Blind Carbon Copy") If you see this header on incoming mail, something is wrong. It's used like Cc: (see below), but does *not* appear in the headers. The idea is to be able to send copies of email to persons who might not want to receive replies or to appear in the headers. Blind carbon copies are popular with spammers, since it confuses many inexperienced users to get email that doesn't appear to be addressed to them.
- **Cc:** (stands for "Carbon Copy", which is meaningful if you remember typewriters) This header is sort of an extension of "To:"; it specifies additional recipients. The difference between "To:" and "Cc:" is essentially connotative; some mailers also deal with them differently in generating replies.
- **Comments:** This is a nonstandard, free-form header field. It's most commonly seen in the form "Comments: Authenticated sender is <rth@bieberdorf.edu>". A header like this is added by some mailers (notably the popular freeware program Pegasus) to identify the sender; however, it is often added by hand (with false information) by spammers as well. Treat with caution.
- **Content-Transfer-Encoding:** This header relates to MIME, a standard way of enclosing non-text content in email. It has no direct relevance to the delivery of mail, but it affects how MIME-compliant mail programs interpret the content of the message.
- **Content-Type:** Another MIME header, telling MIME-compliant mail programs what type of content to expect in the message.
- **Date:** This header does exactly what you'd expect: It specifies a date, normally the date the message was composed and sent. If this header is omitted by the sender's computer, it might conceivably be added by a mail server or even by some other machine along the route. It shouldn't be treated as gospel truth; forgeries aside, there are an awful lot of computers in the world with their clocks set wrong.
- **Errors-To:** Specifies an address for mailer-generated errors, like "no such user" bounce messages, to go to (instead of the sender's address). This is not a particularly common header, as the sender usually wants to receive any errors at the sending address, which is what most (essentially all) mail server software does by default.
- **From** (without colon) This is the "envelope From" discussed above.
- **From:** (with colon) This is the "message From:" discussed above.
- **Message-Id:** (also Message-id: or Message-ID:) The Message-Id is a more-or-less unique identifier assigned to each message, usually by the first mailserver it encounters. Conventionally, it is of the form "gibberish@bieberdorf.edu", where the "gibberish" part could be absolutely anything and the second part is the name of the machine that assigned the ID. Sometimes, but not often, the "gibberish" includes the sender's username. Any email in which the message ID is malformed (e.g., an empty string or no @ sign), or in which the site in the message ID isn't the real site of origin, is probably a forgery.
- **In-Reply-To:** A Usenet header that occasionally appears in mail, the In-Reply-To: header gives the message ID of some previous message which is being replied to. It is unusual for this header to appear except in email directly related to Usenet; spammers have been known to use it, probably in an attempt to evade filtration programs.
- **Mime-Version:** (also MIME-Version:) Yet another MIME header, this one just specifying the version of the MIME protocol that was used by the sender. Like the other MIME headers, this one is usually eminently ignorable; most modern mail programs will do the right thing with it.
- **Newsgroups:** This header only appears in email that is connected with Usenet---either email

copies of Usenet postings, or email replies to postings. In the first case, it specifies the newsgroup(s) to which the message was posted; in the second, it specifies the newsgroup(s) in which the message being replied to was posted. The semantics of this header are the subject of a low-intensity holy war, which effectively assures that both sets of semantics will be used indiscriminately for the foreseeable future.

- **Organization:** A completely free-form header that normally contains the name of the organization through which the sender of the message has net access. The sender can generally control this header, and silly entries like "Royal Society for Putting Things on Top of Other Things" are commonplace.
- **Priority:** An essentially free-form header that assigns a priority to the mail. Most software ignores it. It is often used by spammers, usually in the form "Priority: urgent" (or something similar), in an attempt to get their messages read.
- **Received:** Discussed in detail above.
- **References:** The References: header is rare in email except for copies of Usenet postings. Its use on Usenet is to identify the "upstream" posts to which a message is a response; when it appears in email, it's usually just a copy of a Usenet header. It may also appear in email responses to Usenet postings, giving the message ID of the post being responded to as well as the references from that post.
- **Reply-To:** Specifies an address for replies to go to. Though this header has many legitimate uses (perhaps your software mangles your From: address and you want replies to go to a correct address), it is also widely used by spammers to deflect criticism. Occasionally a naive spammer will actually solicit responses by email and use the Reply-To: header to collect them, but more often the Reply-To: address in junk email is either invalid or an innocent victim.
- **Sender:** This header is unusual in email (X-Sender: is usually used instead), but appears occasionally, especially in copies of Usenet posts. It should identify the sender; in the case of Usenet posts, it is a more reliable identifier than the From: line.
- **Subject:** A completely free-form field specified by the sender, intended, of course, to describe the subject of the message.
- **To:** The "message To:" described above. Note that the To: header need not contain the recipient's address!
- **X-headers** is the generic term for headers starting with a capital X and a hyphen. The convention is that X-headers are nonstandard and provided for information only, and that, conversely, any nonstandard informative header should be given a name starting with "X-". This convention is frequently violated.
- **X-Confirm-Reading-To:** This header requests an automated confirmation notice when the message is received or read. It is typically ignored; presumably some software acts on it.
- **X-Distribution:** In response to problems with spammers using his software, the author of Pegasus Mail added this header. Any message sent with Pegasus to a sufficiently large number of recipients has a header added that says "X-Distribution: bulk". It is explicitly intended as something for recipients to filter against.
- **X-Errors-To:** Like Errors-To:, this header specifies an address for errors to be sent to. It is probably less widely obeyed.
- **X-Mailer:** (also X-mailer:) A freeform header field intended for the mail software used by the sender to identify itself (as advertising or whatever). Since much junk email is sent with mailers invented for the purpose, this field can provide much useful fodder for filters.
- **X-PMFLAGS:** This is a header added by Pegasus Mail; its semantics are nonobvious. It appears in any message sent with Pegasus, so it doesn't obviously convey any information to the recipient that isn't covered by the X-Mailer: header.
- **X-Priority:** Another priority field, used notably by Eudora to assign a priority (which appears as

a graphical notation on the message).

- **X-Sender:** The usual email analogue to the Sender: header in Usenet news, this header purportedly identifies the sender with greater reliability than the From: header. In fact, it is nearly as easy to forge, and should therefore be viewed with the same sort of suspicion as the From: header.
- **X-UIDL:** This is a unique identifier used by the POP protocol for retrieving mail from a server. It is normally added between the recipient's mail server and the recipient's actual mail software; if mail arrives at the mail server with an X-UIDL: header, it is probably junk (there's no conceivable use for such a header, but for some unknown reason many spammers add one).