

Electronic mail

Aka e-mail (or email according to Knuth)

Karst Koymans / Jeroen van der Ham

Informatics Institute
University of Amsterdam

Tuesday, September 25, 2011

Table of Contents

Email history

Basic concepts

Message Agents in detail

Email security

Message format

Message transfer

Message Store

History of email (1)

- ▶ 1971 Tomlinson's first email (e-mail?)
 - ▶ Introduces the use of the @-symbol
 - ▶ First based on CPYNET/SNDMSG, later piggybacked on FTP over ARPANET
- ▶ 1979 UUCP-based email
 - ▶ introduces the bang (!)
 - ▶ not based on TCP/IP (or NCP)

History of email (2)

- ▶ 1982 SMTP (Simple Mail Transfer Protocol) specified
- ▶ 1983 sendmail released (4.1c BSD)
- ▶ (late) 1983 DNS specified
- ▶ 1984 DNS toplevel domains specified
- ▶ sendmail knows about
 - ▶ @ (ARPANET)
 - ▶ ! (UUCP)
 - ▶ : (BerkNet)

Survey October 2001 (Dan Bernstein)

- ▶ 401 UNIX (Sendmail)
- ▶ 176 Windows (Exchange/IIS)
- ▶ 167 UNIX (qmail)
- ▶ 57 Windows (Ipswitch IMail)

- ▶ 23 UNIX (smap)
- ▶ 15 UNIX (IBM Postfix, formerly VMailer)
- ▶ 14 UNIX (Exim)

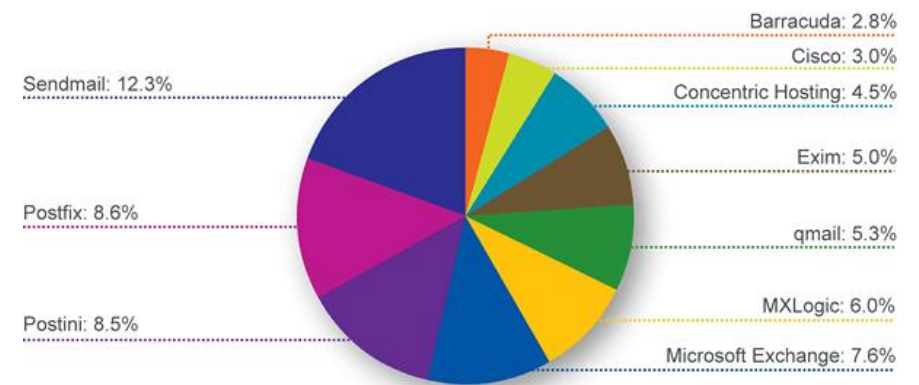
Survey May 2003 (Thomas Pircher)

- ▶ 19169 (35.59%) Sendmail
- ▶ 4537 (8.42%) qmail
- ▶ 4104 (7.62%) Postfix
- ▶ 2812 (5.22%) Microsoft (Exchange/IIS)
- ▶ 2464 (4.57%) Exim

Survey October 2004 (SNE)

- ▶ 20492 (35.3%) Sendmail
- ▶ 12172 (21.0%) Microsoft (Exchange/IIS)
- ▶ 6836 (11.8%) Exim
- ▶ 4008 (6.9%) iMail
- ▶ 3669 (6.3%) qmail
- ▶ 3172 (5.5%) Postfix

Survey 2006/2007 (MailChannels)



Source: O'Reilly SysAdmin

Email concepts

- ▶ Message transfer (RFC 5321)
- ▶ Message format (RFC 5322)
- ▶ Message agents (RFC 5598)
- ▶ Message stores (RFC 5598)

Message Agents

Agent acronym	Agent use
MUA	Message User Agent ¹
MTA	Message Transfer Agent
MDA	Message Delivery Agent
MSA	Message Submission Agent
MAA ²	Message Access Agent
MRA ²	Message Retrieval Agent

¹According to RFC 5598, called **Mail** User Agent in RFC 5321

²Not standardised in RFC5598: "Internet Mail Architecture"

Mail User Agent

- ▶ Interface for the email user
 - ▶ Reads and composes messages
 - ▶ Thunderbird, Outlook, mutt, pine, mh ...
- ▶ Often uses SMTP to send mail (→ MSA)
- ▶ Often uses IMAP/POP3 to get mail (← MAA)
- ▶ May have direct access to message store
 - ▶ Trend is to not have direct access

Message Transfer Agent

- ▶ Transfers email across the Internet
 - ▶ Uses SMTP as transfer protocol
 - ▶ sendmail, Postfix, qmail, Exim
- ▶ Often also operates as a Message Submission Agent
- ▶ Makes use of MX records to transfer email

Message Delivery Agent

- ▶ Delivers email into the message store (MS)
 - ▶ mail, mail.local, rmail, procmail
 - ▶ May do filtering, SPAM and virus checking, ...
 - ▶ Has knowledge about mailbox formats
 - ▶ Can use
 - ▶ Global file space (for example `/var/mail/mbox`)
 - ▶ User specific file space (for example `$HOME/mbox`)
 - ▶ Database (often not directly accessible)

Message Submission Agent (1)

- ▶ See RFC 4409
- ▶ Injects message into the mail system
- ▶ Sanitizes message content
 - ▶ Envelope domains must be FQDN's
- ▶ Often combined with MTA
 - ▶ Should bind to its own port (587), if possible

Message Submission Agent (2)

- ▶ Can operate locally
 - ▶ sendmail (no daemon mode)
 - ▶ postdrop
 - ▶ without SMTP or with piped SMTP
- ▶ May be an MTA-frontend
 - ▶ smapd

Message Access Agent

- ▶ Can get message out of Message Store (MS)
- ▶ Offers services to access mail to MUA (or MRA)
 - ▶ POP3 (Post Office Protocol)
 - ▶ IMAP (Internet Message Access Protocol, version 4)

Message Retrieval Agent

- ▶ Program that uses a MAA to collect mail
- ▶ Possibly reinjects mail into the mail system
 - ▶ fetchmail
 - ▶ SMTP TURN (**Insecure!**)
 - ▶ SMTP ETRN (More secure variant, RFC 1985)
 - ▶ SMTP ATRN³ (Authenticated variant, RFC 2645)

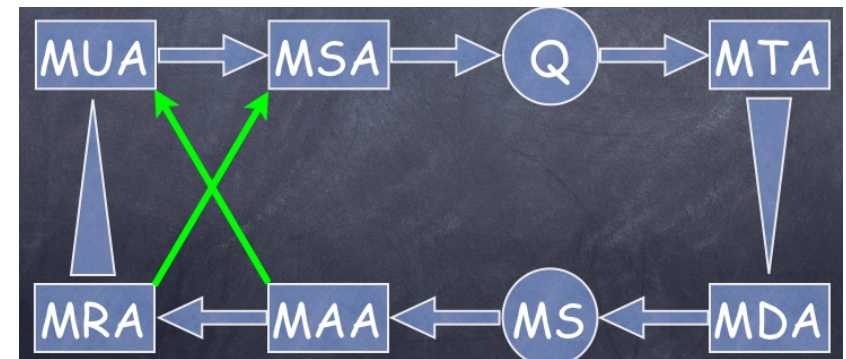
³ODMR (On-Demand Mail Relay) for clients with dynamic IP addresses

Securing Email (1)

- ▶ Use secure protocols
 - ▶ imap (port 143) → imaps (port 993)
 - ▶ pop3 (port 110) → pop3s (port 995)
 - ▶ smtp (port 25) → smtps, ssmtp (port 465)⁴

⁴Not registered as such with IANA

Message Agent Relationships



Securing Email (2)

- ▶ SMTP improvements
 - ▶ Authenticated SMTP
 - ▶ AUTH extension (RFC 4954)
 - ▶ Based on SASL (RFC 4422)
 - ▶ STARTTLS extension (RFC 3207)
 - ▶ Replaces (s)smtp(s)
- ▶ These mechanisms are often used on mail submission via port 587

Securing Email (3)

Use MUA-based encryption and authentication

- ▶ PGP (Pretty Good Privacy)
 - ▶ Inline or PGP/MIME
 - ▶ GPG (GNU Privacy Guard)
 - ▶ Uses a **web** of trust
- ▶ S/MIME
 - ▶ Uses a **hierarchy** of trust (PKI)
- ▶ MIME
 - ▶ Multipurpose Internet Mail Extensions
 - ▶ Different character sets, binary attachments, multiple parts, internationalised headers

Message format

- ▶ RFC 5322
- ▶ Headers, empty line, body
- ▶ Only 7-bit US-ASCII (1-127) allowed
 - ▶ MIME extends this to possibly 8-bit
- ▶ Lines are delimited by <CR><LF>
- ▶ Lines should be no longer than 78 characters

Message (specified in the ABNF formalism)

- ▶ message = (fields / obs-fields) [CRLF body]
- ▶ body = $^{*}(*998\text{text CRLF})^{*998\text{text}}$
- ▶ CRLF = %d13.10
- ▶ Mathematical isomorphism
 - ▶ $\text{text}^{*+} \cong (\text{text} \cup \{\text{CRLF}\})^{*}$
- ▶ CRLF is delimiter or separator, not terminator

Header format

- ▶ <Field name>:<Field body>
- ▶ <Field name>
 - ▶ printable US-ASCII (33-126)
 - ▶ except ":" (58)
- ▶ <Field body>
 - ▶ US-ASCII (1-127) except CR(13) and LF(10)...
 - ▶ ... but also (un)folding is allowed

Some important headers

- ▶ From:<originator mailbox>
- ▶ Sender:<sender mailbox>
- ▶ To:<recipient mailbox>
- ▶ Message-Id:<unique message identification>
- ▶ Received:<registration of message transfer>

Normal (E)SMTP session

- ▶ “EHLO” (greeting, option negotiation)
- ▶ “MAIL FROM:” (envelope sender)
- ▶ “RCPT TO:” (envelope recipient)
- ▶ “DATA” (content, ended by <CRLF>.<CRLF>)
- ▶ “QUIT” (goodbye)

Message transfer

- ▶ SMTP (RFC 5321)
 - ▶ Uses Network Virtual Terminal (NVT) presentation layer from the TELNET RFC 854
 - ▶ Net-ASCII might be replaced in the future by Net-Unicode, see RFC 5198
- ▶ Mail objects
 - ▶ content (in “message format”)
 - ▶ envelope (SMTP parameters)

Message store

- ▶ In database
 - ▶ Only accessible via IMAP, POP3
- ▶ In flat files
 - ▶ Also accessible via direct access
 - ▶ Enables “grepping” the message store

Mbox format

- ▶ Ordinary file with
 - ▶ multiple messages
 - ▶ separated by "From_" at start of line
 - ▶ has extra blank line at end of message
 - ▶ quotes "From_" to ">From_"
 - ▶ and ">From_" to ">>From_"...
 - ▶ a first characteristic line
 - ▶ "From_<envelopesender>_<date>_<optionalinfo>"

MMDF format

- ▶ Variant of mbox format
 - ▶ Uses ^A ^A ^A ^A as separator
 - ▶ Optionally has the mbox "From_" information

MH format

- ▶ Mailbox is a directory
- ▶ Every message is a file with a numeric name
- ▶ Used by mh, nmh, xnmh MUA's

Maildir format

- ▶ Mailbox is again a directory
- ▶ Subdirectories tmp, new, cur
- ▶ Arriving mail: tmp/<time>.<pid>.<host>
- ▶ No mailbox locking needed
- ▶ Works reliably over NFS

Mailbox locking

- ▶ Uses flock, lockf, fcntl system calls
- ▶ Does not always work reliably over NFS
- ▶ Needed if delivery agents and/or access agents operate on the same file (mailbox)

Cyrus MDA/MAA at OS3

- ▶ IMAP server with support for local delivery through LMTP
- ▶ LMTP: Local Mail Transfer Protocol
 - ▶ Similar to ESMTP
 - ▶ Uses LHLO in stead of EHLO
 - ▶ Reports separate status results for every "RCPT TO:"
- ▶ Uses (improved) Maildir format as message store
- ▶ Does not support direct access to mail files

OS3 Mail Infrastructure (1)

- ▶ Separate incoming and outgoing mail services
- ▶ Incoming: smtp.os3.nl
 - ▶ Listens on port 25 to the world for mail destined for os3.nl
 - ▶ Includes SPAM checking
 - ▶ Delivers local mail via LMTP to imap.os3.nl (Cyrus)
 - ▶ Forwards outbound aliases to mail.serv.os3.nl
- ▶ Outgoing: mail.serv.os3.nl
 - ▶ Listens on port 25 to the internal network for mail destined for the world (including os3.nl)
 - ▶ Forwards local mail to smtp.os3.nl

OS3 Mail Infrastructure (2)

- ▶ Enable relaying for authenticated users
 - ▶ Listen on port 587 as a mail submission agent
 - ▶ First enforce STARTTLS
 - ▶ Use username/password authentication inside the protected connection