COMP312-09A Communications and Systems Software

Lecture 3 – Internet Email

Pages 588 to 611 – Tanenbaum 4th Ed.

Matthew Luckie

mluckie@cs.waikato.ac.nz

THE UNIVERSITY OF WAIKATO

In the beginning

- Internet email used file transfer protocols to shift messages
  - e.g. TFTP mail mode (though now deprecated)
- Messages had no internal structure, except for the first line which specified the recipient's address
- Problems:
  - Sending messages to more than one person was inconvenient
  - No delivery failure notification – no internal structure means the sender's address could not automatically be determined to tell them a piece of mail could not be delivered
  - Text only – no images or formatting
  - Poor protocol support for client applications

Overview

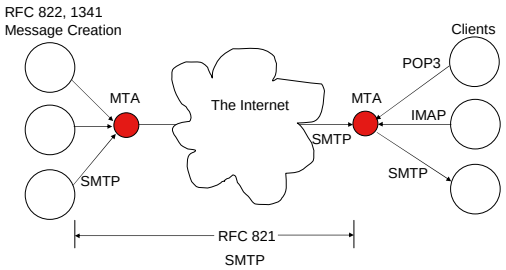
- RFC 821 (SMTP) 1982 (RFC 2821)
- RFC 822 (Message Format) 1982 (RFC 2822)
- RFC 1341 (MIME) 1992
- RFC 1939 (POP3) 1996
- RFC 2060 (IMAP4rev1) 1996

- History
- How Internet Email works
- Issues

Overview

- Specified protocols use TCP
  - Reliable, byte-stream, connection-oriented
  - Properties useful to sending a piece of mail
- Specified protocols and messages sent are ASCII text-based
  - Including binary attachments like pictures and movies
- Protocols that are text-based are easier to debug – you only need to use telnet and can interact with your keyboard

Overview



Internet Email

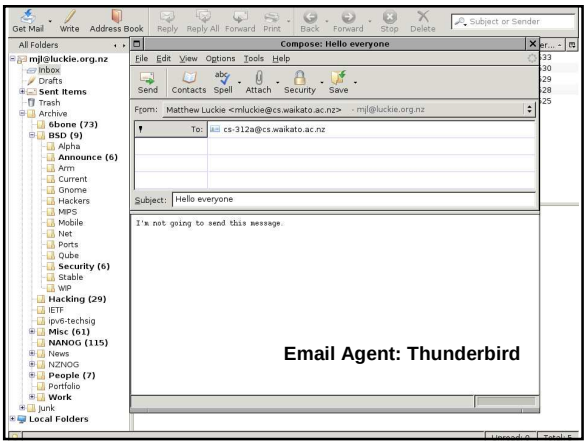
- Two groups of systems
- User Agents (Email Readers)
  - Mozilla Thunderbird
  - Microsoft Outlook
  - Mutt, pine, …
- Message Transfer Agents (MTA)
  - Move email towards to the intended destination
  - Run continuously in the background
  - Exim, Sendmail, qmail, …

Internet email

- User Agent (Email Reader)
  - Composition – create message, using text editor or other
  - Initial Transfer – speaks enough SMTP to send mail to local MTA for it to handle
  - Displaying – parse message, extract and display envelope
  - Mailbox organisation – create email folders
  - Disposition – deleting and moving messages to folders
- Message Transfer Agents (MTAs)
  - Transfer – send message closer to its destination
  - Reporting – relay message failure (if it occurs) to sender

Alternatives

- Proprietary systems (Microsoft Exchange, Novell Groupwise, others)
  - Still exist, though they implement Internet email protocols to reach recipients outside of the system
- Open systems (CCITT X.400)
  - Too complicated compared to RFC 821/822



Envelopes, Headers, Message Body

- Envelope
  - Sender
  - Recipient for this piece of mail
- Header
  - From:
  - When message was sent
  - To:, CC:
  - Subject
  - MTAs visited so far
  - Message IDs, etc.
- Message Body
  - Text of message, attachments

Used by MTA

Displayed by User Agent

Message Body:

Message-ID: <49B033B8.2030907@cs.waikato.ac.nz>
Date: Fri, 06 Mar 2009 09:19:04 +1300
From: Matthew Luckie <mluckie@cs.waikato.ac.nz>
User-Agent: Thunderbird 2.0.0.19 (X11/20090113)
MIME-Version: 1.0
To: mjl@luckie.org.nz
Subject: SMTP demonstration
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

Hello COMP312,

Enjoy your weekend.

Matthew.

Creating the Envelope

- The Mail User Agent will create the envelope based on the From:, To:, CC: addresses specified in the email
- MTAs can (and do) change the envelope addresses as the message is forwarded
  - The header addresses should not change

SMTP

- Very simple protocol
- Connect to SMTP service on specified port (25)
- Say hello to the service
  - HELO / EHLO
- Say you have mail to send (check it is OK to send)
  - MAIL FROM: <sender>
- Say who the message is going to (check it is OK to send)
  - RCTP TO: <recipient>
- Then send the message. End of message is signalled with a dot on a line all by itself
  - DATA
- QUIT.

Initial SMTP exchange 1/2

```
S:  220 University of Waikato, School of Computing and Mathematics
C:  EHLO sorcerer.cs.waikato.ac.nz
S:  250-zombie.scms.waikato.ac.nz Hello sorcerer.cs.waikato.ac.nz
S:  250-SIZE 12582912
S:  250-ETRN
S:  250-EXPN
S:  250-PIPELINING
S:  250-STARTTLS
S:  250 HELP
C:  MAIL FROM:<mluckie@cs.waikato.ac.nz> SIZE=404   } Envelope
S:  250 OK
C:  RCPT TO:<mjl@luckie.org.nz>
S:  250 Accepted
```


2/2

```
C:  DATA
S:  354 Enter message, ending with "." on a line by itself
C:  Message-ID: <49B033B8.2030907@cs.waikato.ac.nz>
C:  Date: Fri, 06 Mar 2009 09:19:04 +1300
C:  From: Matthew Luckie <mluckie@cs.waikato.ac.nz>
C:  User-Agent: Thunderbird 2.0.0.19 (X11/20090113)
C:  MIME-Version: 1.0
C:  To: mjl@luckie.org.nz
C:  Subject: SMTP demonstration
C:  Content-Type: text/plain; charset=ISO-8859-1; format=flowed
C:  Content-Transfer-Encoding: 7bit
C:
C:  Hello COMP312,
C:
C:  Enjoy your weekend.
C:
C:  Matthew.
C:  .
S:  250 OK id=1LfK24-0005JX-VH
C:  QUIT
S:  221 zombie.scms.waikato.ac.nz closing connection
```


Initial SMTP exchange

- Message was to mjl@luckie.org.nz
- However, my Mail User Agent sent it to zombie.scms.waikato.ac.nz
- What next?
  - zombie.scms.waikato.ac.nz has to forward the message closer to the recipient
  - It looks up in DNS the MX (mail exhanger) address for the domain luckie.org.nz

```
[mluckie@sorcerer mjl]$ host -t mx luckie.org.nz
luckie.org.nz mail is handled by 10 zuul.ihug.co.nz
```

  - Sends the message to zuul.ihug.co.nz using SMTP exhange

DNS MX Records

```
[mluckie@sorcerer apps]$ host -t mx gmail.com
gmail.com mail is handled by 40 alt4.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 5 gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt1.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 20 alt2.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 30 alt3.gmail-smtp-in.l.google.com.
```

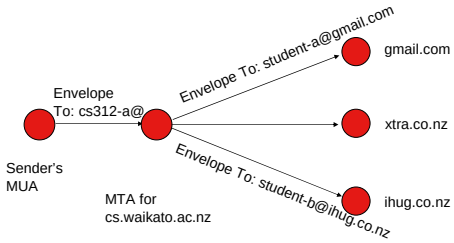
- MTA tries sending to the MX with the highest priority first (in this case 5)
- If it cannot send the message, it can then try the next highest priority (10)
- And so on

A detour: Internet spam

- Spam email exists because it is hard to prevent
- From address, in both the envelope and headers, is trivial to forge because there is no acceptable way to prevent it
- Forging the sender's address makes it hard to trace and halt a spammer

A detour: mailing lists

Header To: cs312-a@cs.waikato.ac.nz



Message Bodies

- Text based.
- To send binary attachments, non-english text characters, a new method was needed
  - MIME: Multipurpose Internet Mail Extensions
  - Encoding and encapsulation method for parts of a message
  - Completely backwards compatible with existing protocols
  - Essentially, take 3 bytes and encode them in 4 printable characters

MIME

```
--_----------=_123613920462690
Content-Disposition: inline; filename="up.gif"
Content-Id: <up>
Content-Transfer-Encoding: base64
Content-Type: image/gif; name="up.gif"

R0lGODdhCwAKAIAAAAHAAP///ywAAAAACwAKAAACE4wNpwi50eKK9NCr8Kxb
9vZoQQEAOw==

--_----------=_123613920462690
Content-Disposition: inline; filename="down.gif"
Content-Id: <down>
Content-Transfer-Encoding: base64
Content-Type: image/gif; name="down.gif"

R0lGODdhCwAKAIAAAP8AAP///ywAAAAACwAKAACFIyBaBup3ByMaTrArjwa
3kmBB1QAADs
```

Final Delivery

- So far, been concerned with SMTP
- Ignored the fact that user agents do not accept SMTP connections
- Problem to solve:
  - How do we deliver a message to the recipient?
  - What if they are offline at the moment?
  - Solutions:
    - POP3
    - IMAP4
    - Webmail

POP3

- Mail is held on a system until the recipient comes online
- When the recipient starts their User Agent, it connects to the system and asks for mail held using the POP3 protocol
- Mail is expunged from the mailbox when the recipient has a copy and has issued the delete command
  - Is possible to leave mail on the server with POP3, but in practice this is not done
  - Too complicated to keep local mail user agent's mailbox in synchronisation with that on the server with POP3
  - System administrator would perhaps prefer that your mail was moved off their system so they have more disk space free

POP3 example

```
S:  +OK POP3 server ready
C:  USER mluckie
S:  +OK
C:  PASS <password>
S:  +OK login successful
C:  LIST
S:  1 404          <-------  Message size 404 bytes
S:  .
C:  RETR 1
S:  (sends message 1)
C:  DELE 1
C:  QUIT
S:  +OK bye
```

IMAP

- Client/Server protocol
- Mail user agent remains connected to email server
  - Polls periodically to check for new email
  - Can copy (cache) messages downloaded locally
- Supports mailbox folders to organise email
- User can access their mailbox from any device anywhere that speaks IMAP protocol
- Well supported by mail user agents

Email Privacy

- SMTP, POP3, and IMAP all support security extensions to hide content of TCP connection
- TLS: Transport Layer Security
  - Security system used in HTTPS
  - See Section 8.9.3 for more information
- Everything after the initial greeting can be encrypted so anyone snooping on the connection will be unable to read your email as it is transmitted
- No guarantee that all MTAs support TLS, so cannot guarantee message privacy as it is relayed through the Internet

Email Privacy

- Application-layer alternatives exist
  - PGP (pretty good privacy)
  - Microsoft have a method as well
- Requires an involved key-exchange process to obtain privacy, and mail user agents don't support the methods well

Spam defences

- DNS based blackhole lists
  - Ask a centralised server if an MTA trying to send you email has recently sent spam
- URL blacklists
  - If the message has a URL embedded in it, is the URL one used for selling things advertised with spam?
- Bayesian analysis
  - Does the message's content look like spam? Or does it look like ordinary email
- Sender Policy Framework
  - DNS record that says which IP addresses are permitted to relay messages from a particular domain. Defense against address spoofing.

Conclusion

- Internet email has become the predominant email system because it is simple to implement
  - i.e. SMTP does one thing fairly well.
- Extensions (particularly those required for mail user agents) can be added without breaking existing mail systems
- Spam a problem