Life after mail();

Marcus Bointon

PHP London '08

Life after mail();

- This is a follow-up to my previous email presentation
- Now you know how to get mail out of PHP
- What happens next?

Coverage

- Bounces
- Mailing lists
- Unsubscribes
- Suppression
- Spam reports
- Deliverability
- Security
- The Law

Bounces

- Something has gone wrong when trying to deliver a message, and it is returned to you
- Either from your own local MTA, or from the remote end of a connection
- If you send email on any kind of scale, you *have* to deal with bounces properly
- Black & white "hard" and "soft" bounces are largely a marketing myth
- Handling bounces is easy in theory
  - *REALLY HARD* in practice – one of the least pleasant things you'll ever write!

Local bounces

- What you'll see most often
- Something wrong immediately:
  - DNS, network problem, remote rejection at send time
- Usually happen very quickly
  - ...or very slowly, in the case of delivery deferrals
- Still get sent to the return path
- Your mail server shown as the bounce source
- Bounce format depends on your server
  - ...which means they are very consistent

Remote Bounces

- Bounces generated by ultimate or an intermediate mail server
- Uses format of that mail server
- Mostly predictable formats, but just like mail filters, wide open to misconfiguration and misimplementation...
  - Microsoft Exchange sometimes sends bounces that don't actually contain the address that bounced!
  - You'll probably get some in Chinese, just for fun
- Use VERP for a little sanity.

Receiving Bounces

- Return path is an email address, so it arrives just like any other message
- Only difference is that return path is “<>”
- Two ways of dealing with them:
  - Drop them into a mailbox for later batch processing
  - Route them directly to a script for immediate processing

Batch processing

- Bounces delivered to a regular mailbox
- Use a PHP POP3 or IMAP client to open the mailbox
  - If on local server, look in mbox or maildirs for speed
  - ezcMailMboxTransport can do that
- Get new bounce message
- Parse message, classify & store results
- Remove from server if successful

Immediate processing

- Event based, no polling required
- May be resource intensive at busy times
- Map incoming address to a script
  - How exactly you do this depends on your mail server
  - qmail `.qmail-bounces-default`:
  - `|/var/scripts/mybouncehandler.php`
- Parse bounce message, classify & store results
- Unhandled bounces can be forwarded to a mailbox or stuffed in your DB for later analysis

How to parse a bounce?

- This is the hard part
- Good news: there are standards
- Bad news: nobody sticks to them
  - And you thought HTML standards compliance was bad...
- Some bits do work reliably:
  - RFC2822
  - MIME

Bounce Flavours

- RFC3464
  - `multipart/report; report-type=delivery-status`
  - Containing a `message/delivery-status` part
- qmail qsbmf
  - Only ever permanent
- Microsoft Exchange DSN
  - Consistent, really easy to parse
  - ...but often completely useless – needs VERP
- exim

RFC 3464 Standard Bounce

```
Content-Type: message/delivery-status

Reporting-MTA: dns; mail.example.com
Received-From-MTA: DNS; mail.example.com
Arrival-Date: Fri, 29 Feb 2008 14:41:20 -0000

Final-Recipient: RFC822; joe@example.com
Action: failed
Status: 5.1.1
Remote-MTA: DNS; mail.example.com
Last-Attempt-Date: Fri, 29 Feb 2008 14:38:30 -0000
```

- Example of a nice clean RFC3464 DSN
- Easy to parse – RFC1893 status is easily extracted

Microsoft Bounce™

```
Content-Type: text/plain; charset="iso-8859-1"

Your message

  To:      joe@example.com
  Subject: Newsletter
  Sent:    Wed, 30 Jul 2007 13:02:03 +0100

did not reach the following recipient(s):

joe@example.com on Wed, 30 Jul 2007 13:02:04 +0100
    The recipient name is not recognized
   The MTS-ID of the original message is:
c=gb;a=bt;p=bt;l=ABCMSXC013487301202PAKDFVCZ
    MSEXCH:IMS:EXCHANGE:BTMOBILE:ABCDUXC01 0 (000C05A6) Unknown Recipient
```

- Typical Microsoft Exchange bounce
- Easy to parse, but often relies on string matching
- Many Exchange plugin filters fail to return proper status – 5.0.0 response is useless

Useless bounce

```
This message was created automatically by mail delivery software (Exim).

A message that you sent could not be delivered to one or more of its
recipients. This is a permanent error. The following address(es) failed:

  joe@example.com
    retry time not reached for any host after a long failure period
```

- Not MIME, simplifies things to an extent
- Can you figure out what that error message means?
- Now figure it out in Finnish/Korean/Swahili...

Bounce Parsing

- PEAR's Mail_MIME class is a wonderful thing
  - Also Zend_Mime_Message::createFromMessage and ezcMailParser
- Dismantle the message, often a big MIME structure, for example:
  - `multipart/mixed > message/rfc822 > multipart/mixed > text/plain`
- Look for an RFC1891 Delivery Status Notification (DSN) part
  - MIME type: `message/delivery-status`

Analysing the DSN

- Decide what kind of bounce it really is
- One key thing to look out for: RFC1893 response codes
  - `5.1.1` – Unknown user (the only real “hard” bounce)
  - `5.0.0` – “I am an inept mail server”
  - `5.7.1` – Spam filter
- Despite all the standards and structured data, sadly you will still end up doing regular expression matching on foreign language strings
- The world could really do with a good rule-based bounce analysis library

Reducing bounce rates

- Format your messages properly (don’t use mail()!)
- Don’t send to addresses you know are bad
- Address them correctly – use VERP
- Use SPF / SenderID / DomainKeys
  - For your own domains, and any that you send for
  - Refuse to send with an SPF fail on from address
- Don't send to group mailboxes – sales@, info@ etc
- Don’t send to people you don't have permission for – don’t do “send to a friend” – it's very easy to accidentally create a spam gateway

Mailing List Subscriptions

- A bunch of people want to receive stuff from you on a regular basis
- Not 2-way lists that would be better dealt with by mailman/ezmlm
- Make sure you get appropriate permission:
  - Opt-out – User doesn't know about it
    - Not polite or legal in some places
  - Opt-in – User asks, no action required
  - Confirmed opt-in – User asks, is notified, no action required
  - Double opt-in – User asks, is notified, action required

Unsubscribes

- Do them immediately
  - None of this "will be processed within 24 hours" rubbish
  - CAN-SPAM says 10 days to process request
    - Links/addresses must remain active for 30
- Web page link
  - Always include it – generally a better user experience
  - Pre-populate their details, it's only polite if you know them
- A reply address
  - Handle similarly to an immediately processed bounce
- Add to your suppression lists

Mailing list headers

- RFC2369 – www.list-unsubscribe.com
- `list-unsubscribe`:
  - <http://www.host.com/list.cgi?cmd=unsub&lst=list>,<mailto:list-request@host.com?subject=unsubscribe>
  - No reason not to use VERP style addressing here too
- `list-help`:
  - <http://www.host.com/list/>
- `list-subscribe`:
  - <mailto:list-on@host.com>

Suppression lists

- A list of addresses you don't want to send mail to
- Add to it:
  - When someone unsubscribes (so they don't get resubscribed by client list uploads)
  - When an address bounces consistently
  - When you receive a spam report
  - When you're given a suppression list
- Some commercial lists available, e.g. DMA
- Important for you reputation, and hence deliverability

Spam reports

- Some big ISPs have spam reporting programmes
  - AOL, MSN Hotmail / Windows Live, Google Mail
    - MS JMRP is apparently fictional, never responds
- Send you a message whenever someone reports a message you sent (by IP) as spam
- Unsubscribe the address from everything
- Route straight to your suppression list
- AOL the best at it – easy to get onto, parseable notifications, works perfectly
- Users no help

Deliverability

- It's all very well sending stuff out, but did it actually get there?
- Total sent - bounces - opens - clicks - spam reports
    - May still leave 90% unaccounted for!
- You can seed the list with your own deliverability test addresses, one or two at each major ISP...
    - But a commercial service will probably do a much more comprehensive job, e.g. ReturnPath
- Sender reputation most significant factor
    - Use trusted sender programmes like GoodMail for critical *transactional* messages

Security

- Don't make promises you can't keep
  - The limiting factor is the recipient's own security
- Anywhere you accept, display or transfer personal info, use SSL
  - That's paranoid, but SSL is cheap and easy now
  - Not yet UK law, but it's being considered
  - PHP classes support it for SMTP
  - Same certificates for web and email
- Even if someone can see their own data, it should be impossible for them to see anyone else's
  - Hash everything – DON'T USE INTEGER IDs!

The Law

- IANAL
- UK Data Protection Act 1998
  - `http://www.opsi.gov.uk/acts/acts1998/ukpga_19980029_en_1`
  - Register, it's cheap and easy
  - Not very compatible with keeping data in USA
  - Pretty loose compared with other implementations of EC Directive 95/46/EC
- US CAN-SPAM Act 2003
  - Doesn't really impose any major restrictions!
  - Centred around opt-out, falsifying headers

Data Protection Principles

1. Personal data shall be processed fairly and lawfully and, in particular, shall not be processed unless (a) at least one of the conditions in Schedule 2 is met, and (b) in the case of sensitive personal data, at least one of the conditions in Schedule 3 is also met.

2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.

3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.

4. Personal data shall be accurate and, where necessary, kept up to date.

5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.

6. Personal data shall be processed in accordance with the rights of data subjects under this Act.

7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.

8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.

The Law

- Law in other EC countries can be much more severe
- Germany, France, Spain particularly strong
- Generally:
  - Do unto others...
    - It’s in your best interests
  - Good articles on Wikipedia, out-law.com
  - If in doubt, consult a lawyer

Life after mail() – Summary

- Bounces
- Mailing lists
- Unsubscribes
- Suppression
- Spam reports
- Deliverability
- Security
- The Law
- Q&A