

perl lightning talk
oscon 2007

qpsmtpd



smtp daemon that does what you want!
(except having an easy-to-pronounce name)

Brief History

- I wanted some feature in `qmail-smtpd`
- First version released in 2001 – in production since before then
- v0.10 with the modular features a year later
- Used for all inbound mail at `perl.org`, `apache.org`, `cpan.org` and many others (corporations small and large)
- **MIT licensed!** (makes all other licenses look cumbersome and restrictive!)



Major Feature: Plug-ability



- Minimal feature set in the “core”
- New Perl programmers make their first plugin in an afternoon
- Experienced ones in minutes!

My first plugin! (Be friendly to “human users”)

```
sub hook_quit {
  my $qp = shift->qp;

  # if she talks EHL0 she is probably too
  # sophisticated to enjoy the fun, so skip it.
  return (DECLINED) if ($qp->connection->hello || '') eq "ehlo";

  my $fortune = '/usr/games/fortune';
  return DECLINED unless -e $fortune;

  my @fortune = ` $fortune -s `;
  @fortune = map { chop; s/^/ \/ /; $_ } @fortune;
  $qp->respond(221, $qp->config('me')
    . " closing connection.", @fortune);

  return DONE;
}
```



Check basic headers



```
sub hook_data_post {
    my ($self, $transaction) = @_ ;

    return (DENY, "You have to send some data first")
        if $transaction->data_size == 0;

    return (DENY, "Mail with no From header not accepted here")
        unless $transaction->header->get('From');

    my $date = $transaction->header->get('Date');

    return (DENY, "Mail with no Date header not accepted here")
        unless $date;

    # check that the date header is "within a few days of now"

    return (DECLINED);
}
```

Enforce “server says hello first” protocol

(and reject spam and viruses)

```
use IO::Select;
```

```
sub register {  
    my ($self, $qp) = @_;  
    $self->register_hook('connect',  
        'connect_handler');  
}
```

```
sub connect_handler {  
    my ($self, $transaction) = @_;  
    my $in = new IO::Select;  
    $in->add(\*STDIN) || return DECLINED;  
    if ($in->can_read(1)) {  
        $self->log(LOGDEBUG, "remote host talked before we said hello");  
        return (DENYSOFT, "Don't be rude and talk before I say hello!");  
    }  
    $self->log(LOGINFO, "remote host said nothing spontaneous, proceeding");  
    return DECLINED;  
}
```





Save incoming mail to DB

```
sub hook_queue {
    my ($self, $transaction) = @_ ;
    my $dbh = DBI->connect_cached(...);

    $dbh->do(q[insert into mails
                (msg_id,header,body)
                values (?,?)] , undef,
            $transaction->header->get('Message-Id'),
            $transaction->header->as_string,
            $transaction->body_as_string
        );

    return (OK, "Queued! $msg_id");
}
```

More Plugins!

greylisting

spamassassin

clamd

dnsbl

authentication

tls (transport level
security = ssl)

check OS of connecting
client

domainkeys

and many more



So flexible!

Wait, there's more!

qpsmtpd is a “smtp processing engine”

“**transports**” plugs **into** qpsmtpd plugs
into **plugins**

Transports

- Start a process per connection (CGI model, the original ‘qpsmtpd’ program)



- Start the daemon and fork on demand (‘qpsmtpd-forkserver’)
- Start the daemon and prefork (like Apache!) (‘qpsmtpd-prefork’)
- Apache::Qpsmtpd (run under mod_perl!)
- Event based, handle 1000s of concurrent connections. Hello spammers, we can cope! (‘qpsmtpd-async’)



Thank you!

<http://smtpd.developer.com/>
ask@developer.com

www.yellowbot.com