邮件

25端口（SMTP）： 25端口为SMTP（Simple Mail Transfer Protocol，简单邮件传输协议）服务所开放的，是用于发送邮件。 如今绝大多数邮件服务器都使用该协议。当你给别人发送邮件时，你的机器的某个动态端口（大于1024）就会与邮件服务器的25号端口建立一个连接，你发送的邮件就会通过这个连接传送到邮件服务器上，保存起来。

109端口（POP2）：109端口是为POP2（Post Office Protocol Version 2，邮局协议2）服务开放的，是用于接收邮件的。

110端口（POP3）：110端口是为POP3（Post Office Protocol Version 3，邮局协议3）服务开放的，是用于接收邮件的。

143端口（IMAP）：143端口是为IMAP（INTERNET MESSAGE ACCESS PROTOCOL）服务开放的，是用于接收邮件的。 目前POP3使用的比POP2广得多，POP2几乎被淘汰，也有某些服务器同时支持POP2和POP3协议。客户端可以使用POP3协议来访问服务端的邮件服务，如今ISP的绝大多数邮件服务器都是使用POP3协议（极少用POP2协议）。在使用邮件客户端程序的时候，会要求输入POP3服务器地址，默认情况下使用的就是110端口。当你用邮件客户端（比如、Thunderbird、foxmail、MS Outlook Express以及各类邮件精灵）登录时，你的机器就会自动用机器的某一个动态端口（大于1024）连接邮件服务器的110端口，服务器就把别人给你发的邮件（之前保存在邮件服务器上），发送到你机器，这样你就可以看到你客户端工具上的收件箱里的新邮件了。 IMAP协议，和POP3协议一样是用来接收邮件的，但是它有它的特别和新颖之处，它是面向用户的，它和POP3协议的主要区别是：用户可以不用把所有的邮件内容全部下载，而是只下载邮件标题和发件人等基本信息，用户可以由标题等基本信息，去决定是否下载邮件全文，用户可以通过客户端的浏览器直接对服务器上的邮件进行操作（比如：打开阅读全文、丢进垃圾箱、永久删除、整理到某文件夹下、归档、）。再简单来说就是：浏览器用的IMAP协议（143端口）来为你接收邮件以及让你很方便的操作服务器上的邮件。邮件客户端用的POP3协议（110端口）来为你接收邮件的全部信息和全文内容保存到你的本地机器成为一个副本，你对邮件客户端上的副本邮件的任何操作都是在副本上，不干涉邮件服务器上为你保存的邮件原本。 上面介绍的SMTP协议、POP2协议、POP3协议、IMAP协议都是不安全的协议。因考虑到网络安全的因素，下面给你介绍基于SSL（Secure Sockets Layer 安全套接层）协议的安全的邮件收发协议。你的邮件在传输过程中可能被网络黑客截取邮件内容，如果你的邮件机密性非常强，不想被收件人以外的任何人和任何黑客截取，或者是涉及国家机密安全的，等等。那么你的邮件就不该使用上述的三种协议进行收发。 若你采用SMTP协议发邮件，那么你发出的邮件从你的机器传到服务器的过程中，可能被黑客截取从而泄露。若你采用POP2或者POP3协议收取邮件，那么你的邮件从服务器传至你当前机器的过程可能被黑客截取从而泄露。

465端口（SMTPS）：465端口是为SMTPS（SMTP-over-SSL）协议服务开放的，这是SMTP协议基于SSL安全协议之上的一种变种协议，它继承了SSL安全协议的非对称加密的高度安全可靠性，可防止邮件泄露。SMTPS和SMTP协议一样，也是用来发送邮件的，只是更安全些，防止邮件被黑客截取泄露，还可实现邮件发送者抗抵赖功能。防止发送者发送之后删除已发邮件，拒不承认发送过这样一份邮件。

995端口（POP3S）：995端口是为POP3S（POP3-over-SSL）协议服务开放的，这是POP3协议基于SSL安全协议之上的一种变种协议，它继承了SSL安全协议的非对称加密的高度安全可靠性，可防止邮件泄露。POP3S和POP3协议一样，也是用来接收邮件的，只是更安全些，防止邮件被黑客截取泄露，还可实现邮件接收方抗抵赖功能。防止收件者收件之后删除已收邮件，拒不承认收到过这样一封邮件。

993端口（IMAPS）：993端口是为IMAPS（IMAP-over-SSL）协议服务开放的，这是IMAP协议基于SSL安全协议之上的一种变种协议，它继承了SSL安全协议的非对称加密的高度安全可靠性，可防止邮件泄露。IMAPS和IMAP协议一样，也是用来接收邮件的，只是更安全些，防止邮件被黑客截取泄露，还可实现邮件接收方抗抵赖功能。防止收件者收件之后删除已收邮件，拒不承认收到过这样一封邮件。

网易免费邮箱相关服务器信息：

邮件服务器 | 服务器地址 | 端口

POP3服务器 | pop.163.com | 110

SMTP服务器 | smtp.163.com | 25

IMAP服务器 | imap.163.com | 143

现在大部分的邮箱都支持POP3和SMTP，就连一直雪藏着不免费开放这项功能的网易年前也开放了，利用SMTP/POP3设置我们可以方便的利用客户端（例如OUTLOOK、FOXMAIL、WINDOES LIVE MAIL）发送邮件。在这里我把一部分常用邮箱的SMTP/POP3的服务器地址和端口整理一下，供大家以后使用。

备注：利用邮箱的SMTP/POP3功能时请先确认您的邮箱支持此项功能，并且你已经在邮箱中开启了这项功能。

QQ邮箱：

接收邮件服务器： pop.sina.com

接收端口： 110

发送邮件服务器： smtp.sina.com

发送端口： 25

GMAIL:

接收邮件服务器： pop.gmail.com

接收端口： 995

发送邮件服务器： smtp.gmail.com

发送端口： 465

IMAP接收邮件服务器： imap.gmail.com

接收端口： 993

SINA邮箱：

接收邮件服务器： pop.sina.com

接收端口： 110

发送邮件服务器： smtp.sina.com

发送端口： 25

126邮箱：

接收邮件服务器： pop.126.com

接收端口： 110

发送邮件服务器： smtp.126.com

发送端口： 25

163邮箱：

接收邮件服务器： pop.163.com

接收端口： 110

发送邮件服务器： smtp.163.com

发送端口： 25

YEAH邮箱：

接收邮件服务器： pop.yeah.net

接收端口： 110

发送邮件服务器： smtp. yeah.net

发送端口： 25

HOTMAIL：

接收邮件服务器： pop3.live.com

接收端口： 995

发送邮件服务器： smtp.live.com

发送端口： 25

实现一个简单的SMTP发送库

最近在优化“语音邮箱”业务时，发现采用python-smtplib库虽然能满足需要，但是还是有很多不足，因此决定自己重新写一个库，当然仅仅实现发送email（包含附件）即可。实现一个完整的SMTP服务可能比较复杂，而实现一个单纯的、高性能的SMTP客户端发送库却比较容易。

与SIP、HTTP等协议类似，SMTP也采用文本方式定义消息格式，而附件等二进制内容，则采用base64转码成文本，这样就极为方便开发和调试。以下是SMTP最基础的几个RFC文档：

RFC5321： SMTP

RFC5322： Internet message format

RFC4954： SMTP Service Extension for AuthenticationRFC3207： SMTP Service Extension for Secure SMTP over Transport Layer Security

RFC2035： Multipurpose Internet Mail Extensions

我们结合SMTP的流程，来简要说明SMTP协议中的各个关键点。

一、标准端口

SMTP基于TCP连接，SMTP服务器默认打开的端口是25、587或者465。通常是采用25端口，后期设计加密SMTP（即SMTPS）时采用了465端口。而IANA在新业务端口规划时占用了465端口，因此将587端口定义为TLS加密的SMTP端口。

目前各email服务商通常都会同时打开25和587端口，有些为了和以前老客户端程序兼容，也会打开465端口。

这里面其实可能存在一个误解，似乎25端口就是不加密的SMTP，而587、465就是加密的SMTP。实际上不是这样，现代SMTP服务器一般都已经支持STARTTLS（即RFC3207），因此在25端口也可以启动SSL/TLS加密，这取决于客户端是否具备SSL/TLS加密、解密能力，这点在后续流程中会详细描述。

二、基础流程

2.1 建立与SMTP服务器的连接

建立TCP连接，连接到服务器的25端口或者587端口。如果是25端口，则是建立普通TCP连接即可。如果是587端口，通常可以直接建立SSL/TLS连接。需要注意的是，有些SMTP服务器的587端口也是要求普通TCP连接，后续STARTTLS流程中再启动SSL/TLS加密。这种情况下，587端口的流程与25端口的流程一致。

我们的示例流程中，采用QQMail的SMTP服务器做参考，因此我们首先向“smtp.qq.com”服务器的25端口建立连接。连接成功后，服务器应返回220消息，告诉客户端连接成功：

220 smtp.qq.com Esmtp QQ Mail Server

2.2 初次握手

SMTP客户端发送EHLO命令给服务器，进行初次握手协商：

EHLO 192.168.1.101

“EHLO”实际是ESMTP（Extended SMTP）的指令，意思是要进行用户名和密码的鉴权。远古时代的SMTP不需要鉴权的情况下，可以直接发送HELO命令进行握手。目前的SMTP服务器基本都是支持ESMTP，因此如果不刻意区分的话，后续的SMTP实际就是指ESMTP。

SMTP服务器器收到上述消息后，返回本服务器的一些能力信息给客户端：

250-smtp.qq.com

250-PIPELINING

250-SIZE 73400320

250-STARTTLS

250-AUTH LOGIN PLAIN

250-AUTH=LOGIN

250-MAILCOMPRESS

250 8BITMIME

其中最关键的就是STARTTLS能力，也就是该SMTP服务器具备加密连接能力。通常都建议采用加密连接以确保数据安全，毕竟互联网上总有一些黑暗势力在窥探您的信息。

如果客户端不具备SSL/TLS能力，则直接跳到“账户鉴权”流程即可。此时就不能采用PLAIN鉴权模式，必须采用MD5等加密鉴权模式。

2.3 加密连接

既然服务器明确了SSL/TLS能力，因此客户端只要启动SSL/TLS加密TCP连接即可。客户端发送STARTTLS命令，通知服务器准备加密：

STARTTLS

服务器返回响应消息，通知客户端可以开始加密：

220 Ready to start TLS

接下来就是在TCP连接上建立SSL/TLS加密连接，后续通信以后都基于SSL/TLS连接。

请注意，QQMail服务器目前似乎只支持SSLv2、SSLv3加密，不支持TLSv1.0，TLSv1.1和TLSv1.2加密，其他的SMTP服务器（例如outlook、gmail等）则支持全部加密方式。

2.4 第二次握手

加密连接建立后，再次发送EHLO命令进行握手：

EHLO 192.168.1.101

同样，服务器返回相应能力进行响应。此时，通常不再包含STARTTLS指令了：

250-smtp.qq.com

250-PIPELINING

250-SIZE 73400320

250-AUTH LOGIN PLAIN

250-AUTH=LOGIN

250-MAILCOMPRESS

250 8BITMIME

2.5 账户鉴权

鉴权有多种方式，其中PLAIN方式是要求所有SMTP服务器都必须具备的。在未加密的情况下，采用PLAIN方式无疑会泄露账户密码信息。而在已经加密的SSL/TLS连接里，则不存在这个问题，因此我们采用了最简单的PLAIN方式传递账户密码信息进行鉴权。

PLAIN方式很简单，采用”\0user name\0password”进行编码，然后base64转化成文本格式即可。

客户端发送以下鉴权命令：

AUTH PLAIN my\_demo\_base64\_string\_here

服务器鉴权成功，返回响应消息：

235 Authentication successful

2.6 提交邮件地址信息

鉴权成功后，客户端可以开始告诉服务器邮件地址信息，例如发送者、接收者等。此类消息不一一累述，请直接参考以下消息流：

MAIL From:<Iloveu@foxmail.com>

250 Ok

RCPT To:<Iloveu2@msn.com>

250 Ok

2.7 准备递交邮件内容

客户端发送DATA命令，告诉服务器准备提交内容：

DATA

服务器返回响应消息，通知客户端可以开始发送数据：

354 End data with <CR><LF>.<CR><LF>

由于后续数据都采用文本编码，因此标准规定用“\r\n.\r\n”标识数据发送结束，也就是SMTP服务器在上述响应消息中指示的“<CR><LF>.<CR><LF>”。

2.8 发送邮件内容

邮件采用文本格式，附件采用base64编码，可以分多次发送，以下为一个简单示例：

From: <Ilovu@foxmail.com>

To: <Iloveu2@msn.com>

Subject: You have a new voice mail: From 301 Day 2015-12-14 Time 10.29.7

MIME-Version: 1.0

Content-type: multipart/mixed; boundary="==491D4D07305F440E009A0125=="

--==491D4D07305F440E009A0125==

Content-Type: application/octet-stream

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename="From 301 Day 2015-12-14 Time 10.29.7.wav"

UklGRpKaAABXQVZFZm10IBIAAAAGAAEAQB8AAEAfA......

--==491D4D07305F440E009A0125==--

.

对于有附件的情况，应在Content-Type中指明boundary参数，例如本例中的“==491D4D07305F440E009A0125==”。而附件内容起始处用“–boundary参数”标识，附件结尾用“–boundary参数–”标识。

发送完数据后，服务器响应消息：

250 Ok: queued as

2.9 结束

发送完邮件后，客户端友好地向服务器告别：

QUIT

服务器也友好地告别：

221 Bye

最后关闭TCP连接，完成整个流程。