WIKIPEDIA
The Free Encyclopedia

Simple Mail Transfer Protocol
简单邮件传输协议

The **Simple Mail Transfer Protocol** (**SMTP**) is an Internet standard communication protocol for electronic mail transmission. Mail servers and other message transfer agents use SMTP to send and receive mail messages. User-level email clients typically use SMTP only for sending messages to a mail server for relaying, and typically submit outgoing email to the mail server on port 587 or 465 per RFC 8314 (https://datatracker.ietf.org/doc/html/rfc8314). For retrieving messages, IMAP (which replaced the older POP3) is standard, but proprietary servers also often implement proprietary protocols, e.g., Exchange ActiveSync.

简单邮件传输协议（SMTP）是用于电子邮件传输的因特网标准通信协议。邮件服务器和其他邮件传输代理使用 SMTP 发送和接收邮件。用户级电子邮件客户端通常仅使用 SMTP 将邮件发送到邮件服务器进行中继，并且通常根据 RFC 8314 通过端口 587 或 465 将传出电子邮件提交到邮件服务器。对于检索邮件，IMAP（取代了旧的POP3）是标准的，但专有服务器也经常实现专有协议，例如Exchange ActiveSync。

SMTP's origins began in 1980, building on concepts implemented on the ARPANET since 1971. It has been updated, modified and extended multiple times. The protocol version in common use today has extensible structure with various extensions for authentication, encryption, binary data transfer, and internationalized email addresses. SMTP servers commonly use the Transmission Control Protocol on port number 25 (for plaintext) and 587 (for encrypted communications).

SMTP的起源始于1980年，建立在自1971年以来在阿帕网实施的概念之上。它已被多次更新，修改和扩展。目前常用的协议版本具有可扩展的结构，具有用于身份验证、加密、二进制数据传输和国际化电子邮件地址的各种扩展。SMTP 服务器通常在端口号 25（用于明文）和 587（用于加密通信）上使用传输控制协议。

History 历史

Predecessors to SMTP SMTP 的前身

Various forms of one-to-one electronic messaging were used in the 1960s. Users communicated using systems developed for specific mainframe computers. As more computers were interconnected, especially in the U.S. Government's ARPANET, standards were developed to permit exchange of messages between different operating systems.

1960年代使用了各种形式的一对一电子信息。用户使用为特定大型计算机开发的系统进行通信。随着越来越多的计算机相互连接，特别是在美国政府的阿帕网中，标准被开发出来，允许在不同操作系统之间交换消息。

Mail on the ARPANET traces its roots to 1971: the Mail Box Protocol, which was not implemented,[1] but is discussed in RFC 196 (https://datatracker.ietf.org/doc/html/rfc196); and the SNDMSG program, which Ray Tomlinson of BBN adapted that year to send messages across

two computers on the ARPANET.[2][3][4] A further proposal for a Mail Protocol was made in RFC 524 in June 1973,[5] which was not implemented.[6]

阿帕网上的邮件可以追溯到1971年：邮箱协议，该协议未实现，[1] 但在RFC 196中进行了讨论;以及SNDMSG程序，BBN的Ray Tomlinson在那一年进行了改编，以在ARPANET上的两台计算机上发送消息。[2] [3] [4] 1973 年 6 月的 RFC 524 中提出了邮件协议的进一步建议，[5] 但未实施。[6]

The use of the File Transfer Protocol (FTP) for "network mail" on the ARPANET was proposed in RFC 469 in March 1973.[7] Through RFC 561, RFC 680, RFC 724, and finally RFC 733 in November 1977, a standardized framework for "electronic mail" using FTP mail servers on was developed.[8]

在1973年3月的RFC 469中提出了在ARPANET上使用文件传输协议（FTP）作为“网络邮件”。[7] 通过 RFC 561、RFC 680、RFC 724 以及最终在 1977 年 11 月的 RFC 733，开发了使用 FTP 邮件服务器的“电子邮件”标准化框架。[8]

SMTP grew out of these standards developed during the 1970s.

SMTP源于1970年代制定的这些标准。

Original SMTP 原始短信通信

In 1980, Jon Postel and Suzanne Sluizer published RFC 772 (https://datatracker.ietf.org/doc/html/rfc772) which proposed the Mail Transfer Protocol as a replacement for the use of the FTP for mail. RFC 780 (https://datatracker.ietf.org/doc/html/rfc780) of May 1981 removed all references to FTP and allocated port 57 for TCP and UDP, an allocation that has since been removed by IANA. In November 1981, Postel published RFC 788 (https://datatracker.ietf.org/doc/html/rfc788) "Simple Mail Transfer Protocol".

1980年，Jon Postel和Suzanne Sluizer发布了RFC 772，提出了邮件传输协议作为使用FTP进行邮件的替代品。1981 年 5 月的 RFC 780 删除了对 FTP 的所有引用，并为 TCP 和 UDP 分配了端口 57，此后 IANA 删除了该分配。1981年11月，Postel发布了RFC 788“简单邮件传输协议”。

The SMTP standard was developed around the same time as Usenet, a one-to-many communication network with some similarities.

SMTP标准与Usenet同时开发，Usenet是一个具有一些相似之处的一对多通信网络。

SMTP became widely used in the early 1980s. At the time, it was a complement to the Unix to Unix Copy Program (UUCP), which was better suited for handling email transfers between machines that were intermittently connected. SMTP, on the other hand, works best when both the sending and receiving machines are connected to the network all the time. Both used a store and forward mechanism and are examples of push technology. Though Usenet's newsgroups were still propagated with UUCP between servers,[9] UUCP as a mail transport has virtually disappeared[10] along with the "bang paths" it used as message routing headers.[11]

SMTP在1980年代初被广泛使用。当时，它是Unix到Unix复制程序（UUCP）的补充，UUCP更适合处理间歇性连接的机器之间的电子邮件传输。另一方面，当发送和接收机器始终连接到网络时，SMTP 效果最佳。两者都使用了存储和转发机制，并且是推送技术的示例。尽管Usenet的新闻组仍然在服务器之间使用UUCP传播，[9] 但UUCP作为邮件传输实际上已经与它用作邮件路由标头的“爆炸路径”[10] 一起消失了。[11]

Sendmail, released with 4.1cBSD in 1983, was one of the first mail transfer agents to implement SMTP.[12] Over time, as BSD Unix became the most popular operating system on the Internet, Sendmail became the most common MTA (mail transfer agent).[13]

Sendmail 于 1983 年随 4.1cBSD 一起发布，是最早实施 SMTP 的邮件传输代理之一。[12] 随着时间的推移，随着BSD Unix成为互联网上最流行的操作系统，Sendmail成为最常见的MTA（邮件传输代理）。[13]

The original SMTP protocol supported only unauthenticated unencrypted 7-bit ASCII text communications, susceptible to trivial man-in-the-middle attack, spoofing, and spamming, and requiring any binary data to be encoded to readable text before transmission. Due to absence of a proper authentication mechanism, by design every SMTP server was an open mail relay. The Internet Mail Consortium (IMC) reported that 55% of mail servers were open relays in 1998,[14] but less than 1% in 2002.[15] Because of spam concerns most email providers blocklist open relays,[16] making original SMTP essentially impractical for general use on the Internet.

原始 SMTP 协议仅支持未经身份验证的未加密 7 位 ASCII 文本通信，容易受到微不足道的中间人攻击、欺骗和垃圾邮件，并且要求在传输之前将任何二进制数据编码为可读文本。由于缺乏适当的身份验证机制，根据设计，每个SMTP服务器都是开放邮件中继。互联网邮件联盟（IMC）报告说，1998年55%的邮件服务器是开放中继，[14] 但2002年不到1%。[15] 由于垃圾邮件问题，大多数电子邮件提供商将开放中继列入阻止名单，[16] 这使得原始SMTP在互联网上普遍使用基本上是不切实际的。

Modern SMTP 现代短信通信

In November 1995, RFC 1869 (https://datatracker.ietf.org/doc/html/rfc1869) defined Extended Simple Mail Transfer Protocol (ESMTP), which established a general structure for all existing and future extensions which aimed to add-in the features missing from the original SMTP. ESMTP defines consistent and manageable means by which ESMTP clients and servers can be identified and servers can indicate supported extensions.

1995 年 11 月，RFC 1869 定义了扩展简单邮件传输协议 （ESMTP），该协议为所有现有和未来的扩展建立了一个通用结构，旨在添加原始 SMTP 中缺少的功能。ESMTP 定义了一致且可管理的方法，通过这些方法可以识别 ESMTP 客户端和服务器，服务器可以指示支持的扩展。

Message submission (RFC 2476 (https://datatracker.ietf.org/doc/html/rfc2476)) and SMTP-AUTH (RFC 2554 (https://datatracker.ietf.org/doc/html/rfc2554)) were introduced in 1998 and 1999, both describing new trends in email delivery. Originally, SMTP servers were typically internal to an organization, receiving mail for the organization *from the outside*, and relaying messages from the organization *to the outside*. But as time went on, SMTP servers (mail transfer agents), in practice, were expanding their roles to become message submission agents for Mail user agents, some of which were now relaying mail *from the outside* of an organization. (e.g. a company executive wishes to send email while on a trip using the corporate SMTP server.) This issue, a consequence of the rapid expansion and popularity of the World Wide Web, meant that SMTP had to include specific rules and methods for relaying mail and authenticating users to prevent abuses such as relaying of unsolicited email (spam). Work on message submission (RFC 2476 (https://datatracker.ietf.org/doc/html/rfc2476)) was originally started because popular mail servers would often rewrite mail in an attempt to fix problems in it, for example, adding a domain name to an unqualified address. This behavior is helpful when the message being fixed is an initial submission, but dangerous and harmful when the message originated elsewhere and is being relayed. Cleanly separating mail into submission and relay was seen as a way to permit and encourage rewriting submissions while prohibiting rewriting relay. As spam became more prevalent, it was also seen as a way to provide authorization for mail being sent out from an organization, as well as traceability. This separation of relay and submission quickly became a foundation for modern email security practices.

邮件提交（RFC 2476）和SMTP-AUTH（RFC 2554）在1998年和1999年被引入，两者都描述了电子邮件传递的新趋势。最初，SMTP 服务器通常是组织内部的，从外部接收组织的邮件，并将来自组织的邮件中继到外部。但随着时间的推移，SMTP服务器（邮件传输代理）在实践中正在扩展其

角色，成为邮件用户代理的邮件提交代理，其中一些现在正在中继来自组织外部的邮件。（例如，公司高管希望在旅行时使用公司SMTP服务器发送电子邮件。这个问题是万维网迅速扩展和普及的结果，这意味着SMTP必须包括中继邮件和对用户进行身份验证的特定规则和方法，以防止滥用，例如中继未经请求的电子邮件（垃圾邮件）。邮件提交工作（RFC 2476）最初是因为流行的邮件服务器经常重写邮件以试图解决其中的问题，例如，将域名添加到非限定地址。当要修复的消息是初始提交时，此行为很有用，但当消息源自其他位置并正在中继时，此行为是危险和有害的。将邮件干净地分为提交和中继被视为允许和鼓励重写提交同时禁止重写中继的一种方式。随着垃圾邮件变得越来越普遍，它也被视为为从组织发送的邮件提供授权以及可追溯性的一种方式。这种中继和提交的分离很快成为现代电子邮件安全实践的基础。

As this protocol started out purely ASCII text-based, it did not deal well with binary files, or characters in many non-English languages. Standards such as Multipurpose Internet Mail Extensions (MIME) were developed to encode binary files for transfer through SMTP. Mail transfer agents (MTAs) developed after Sendmail also tended to be implemented 8-bit clean, so that the alternate "just send eight" strategy could be used to transmit arbitrary text data (in any 8-bit ASCII-like character encoding) via SMTP. Mojibake was still a problem due to differing character set mappings between vendors, although the email addresses themselves still allowed only ASCII. 8-bit-clean MTAs today tend to support the 8BITMIME extension, permitting some binary files to be transmitted almost as easily as plain text (limits on line length and permitted octet values still apply, so that MIME encoding is needed for most non-text data and some text formats). In 2012, the SMTPUTF8 extension was created to support UTF-8 text, allowing international content and addresses in non-Latin scripts like Cyrillic or Chinese.

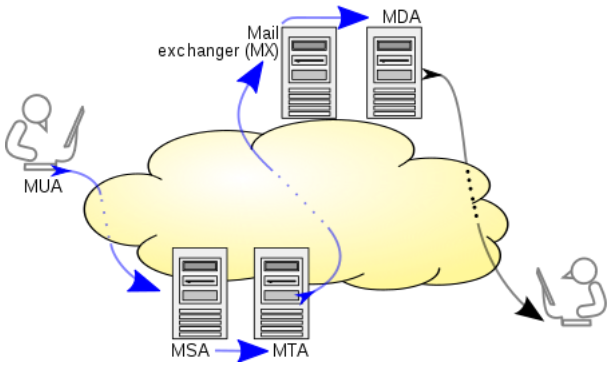
由于该协议最初纯粹是基于ASCII文本的，因此它不能很好地处理二进制文件或许多非英语语言中的字符。诸如多用途互联网邮件扩展（MIME）之类的标准被开发用于对二进制文件进行编码，以便通过SMTP传输。在Sendmail之后开发的邮件传输代理（MTA）也倾向于实现8位干净，因此可以使用替代的“只需发送8”策略通过SMTP传输任意文本数据（在任何8位类似ASCII的字符编码中）。由于供应商之间的字符集映射不同，Mojibake仍然是一个问题，尽管电子邮件地址本身仍然只允许ASCII。今天的8位清洁MTA倾向于支持8BITMIME扩展，允许一些二进制文件几乎像纯文本一样容易地传输（行长和允许的八位字节值的限制仍然适用，因此大多数非文本数据和某些文本格式都需要MIME编码）。2012 年，该 SMTPUTF8 扩展被创建以支持 UTF-8 文本，允许使用非拉丁字母（如西里尔文或中文）的国际内容和地址。

Many people contributed to the core SMTP specifications, among them Jon Postel, Eric Allman, Dave Crocker, Ned Freed, Randall Gellens, John Klensin, and Keith Moore.

许多人为核心SMTP规范做出了贡献，其中包括Jon Postel，Eric Allman，Dave Crocker，Ned Freed，Randall Gellens，John Klensin和Keith Moore。

Mail processing model 邮件处理模型

Email is submitted by a mail client (mail user agent, MUA) to a mail server (mail submission agent, MSA) using SMTP on TCP port 587. Most mailbox providers still allow submission on traditional port 25. The MSA delivers the mail to its mail transfer agent (mail transfer agent, MTA). Often, these two agents are instances of the same software launched with different options on the same machine. Local processing can be done either on a single machine, or split among multiple machines; mail agent processes on one machine can share files, but if processing is on multiple machines, they transfer messages between each other using SMTP, where each machine is configured to use the next machine



Blue arrows depict implementation of SMTP variations

蓝色箭头描述 SMTP 变体的实现

as a smart host. Each process is an MTA (an SMTP server) in its own right.

电子邮件由邮件客户端（邮件用户代理，MUA）使用 TCP 端口 587 上的 SMTP 提交到邮件服务器（邮件提交代理，MSA）。大多数邮箱提供商仍允许在传统端口 25 上进行提交。MSA 将邮件传递到其邮件传输代理（邮件传输代理，MTA）。通常，这两个代理是在同一台计算机上使用不同选项启动的同一软件的实例。本地处理可以在单台机器上完成，也可以在多台机器之间拆分;一台计算机上的邮件代理进程可以共享文件，但如果在多台计算机上处理，则它们使用 SMTP 在彼此之间传输邮件，其中每台计算机都配置为将下一台计算机用作智能主机。每个进程本身就是一个MTA（SMTP服务器）。

The boundary MTA uses DNS to look up the MX (mail exchanger) record for the recipient's domain (the part of the email address on the right of @). The MX record contains the name of the target MTA. Based on the target host and other factors, the sending MTA selects a recipient server and connects to it to complete the mail exchange.

边界 MTA 使用 DNS 查找收件人域（右侧 @ 电子邮件地址的一部分）的 MX（邮件交换器）记录。MX 记录包含目标 MTA 的名称。根据目标主机和其他因素，发送 MTA 选择收件人服务器并连接到该服务器以完成邮件交换。

Message transfer can occur in a single connection between two MTAs, or in a series of hops through intermediary systems. A receiving SMTP server may be the ultimate destination, an intermediate "relay" (that is, it stores and forwards the message) or a "gateway" (that is, it may forward the message using some protocol other than SMTP). Per RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) section 2.1, each hop is a formal handoff of responsibility for the message, whereby the receiving server must either deliver the message or properly report the failure to do so.

邮件传输可以在两个 MTA 之间的单个连接中进行，也可以通过中间系统在一系列跃点中进行。接收 SMTP 服务器可以是最终目的地、中间“中继”（即存储和转发邮件）或“网关”（即，它可能使用 SMTP 以外的某种协议转发邮件）。根据 RFC 5321 第 2.1 节，每个跃点都是消息责任的正式移交，接收服务器必须传递消息或正确报告失败。

Once the final hop accepts the incoming message, it hands it to a mail delivery agent (MDA) for local delivery. An MDA saves messages in the relevant mailbox format. As with sending, this reception can be done using one or multiple computers, but in the diagram above the MDA is depicted as one box near the mail exchanger box. An MDA may deliver messages directly to storage, or forward them over a network using SMTP or other protocol such as Local Mail Transfer Protocol (LMTP), a derivative of SMTP designed for this purpose.

最后一个跃点接受传入邮件后，它会将其传递给邮件传递代理 （MDA） 进行本地传递。MDA 以相关邮箱格式保存邮件。与发送一样，可以使用一台或多台计算机完成此接收，但在上图中，MDA被描述为邮件交换器框附近的一个框。MDA 可以将邮件直接传递到存储，或使用 SMTP 或其他协议（如本地邮件传输协议 （LMTP））通过网络转发邮件，该协议是为此目的设计的 SMTP 的衍生产品。

Once delivered to the local mail server, the mail is stored for batch retrieval by authenticated mail clients (MUAs). Mail is retrieved by end-user applications, called email clients, using Internet Message Access Protocol (IMAP), a protocol that both facilitates access to mail and manages stored mail, or the Post Office Protocol (POP) which typically uses the traditional mbox mail file format or a proprietary system such as Microsoft Exchange/Outlook or Lotus Notes/Domino. Webmail clients may use either method, but the retrieval protocol is often not a formal standard.

传递到本地邮件服务器后，邮件将被存储起来，以供经过身份验证的邮件客户端 （MUA） 进行批量检索。邮件由最终用户应用程序（称为电子邮件客户端）检索，使用 Internet 邮件访问协议（IMAP）（一种既便于访问邮件又管理存储邮件的协议）或邮局协议 （POP），该协议通常使用传统的 mbox 邮件文件格式或专有系统，如 Microsoft Exchange/Outlook 或 Lotus Notes/Domino。Webmail客户端可以使用任一方法，但检索协议通常不是正式标准。

SMTP defines message *transport*, not the message *content*. Thus, it defines the mail *envelope* and its parameters, such as the envelope sender, but not the header (except *trace information*) nor the body of the message itself. STD 10 and RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) define SMTP (the envelope), while STD 11 and RFC 5322 (https://datatracker.ietf.org/doc/html/rfc5322) define the message (header and body), formally referred to as the Internet Message Format.

SMTP 定义邮件传输，而不是邮件内容。因此，它定义邮件信封及其参数，例如信封发件人，但不定义邮件头（跟踪信息除外）或邮件本身的正文。STD 10 和 RFC 5321 定义 SMTP（信封），而 STD 11 和 RFC 5322 定义邮件（标头和正文），正式称为 Internet 邮件格式。

Protocol overview 协议概述

SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An *SMTP session* consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A session may include zero or more SMTP transactions. An *SMTP transaction* consists of three command/reply sequences:

SMTP 是一种面向连接、基于文本的协议，其中邮件发件人通过发出命令字符串并通过可靠的有序数据流通道（通常是传输控制协议 （TCP） 连接）提供必要的数据来与邮件接收方进行通信。SMTP 会话由 SMTP 客户端（发起代理、发送方或发送方）发出的命令和来自 SMTP 服务器（侦听代理或接收方）的相应响应组成，以便打开会话并交换会话参数。一个会话可以包含零个或多个 SMTP 事务。SMTP 事务由三个命令/回复序列组成：

1. **MAIL** command, to establish the return address, also called return-path,[17] reverse-path,[18] bounce address, mfrom, or envelope sender.

   MAIL 命令，用于建立寄信人地址，也称为回信路径、反向路径、退回地址、 [17] [18] mfrom 或信封发件人。

2. **RCPT** command, to establish a recipient of the message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.

   RCPT 命令，用于建立邮件的收件人。此命令可以发出多次，每个收件人一个。这些地址也是信封的一部分。

3. **DATA** to signal the beginning of the *message text*; the content of the message, as opposed to its envelope. It consists of a *message header* and a *message body* separated by an empty line. DATA is actually a group of commands, and the server replies twice: once to the *DATA command* itself, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.

   DATA 表示消息文本的开头;邮件的内容，而不是其信封。它由邮件头和由空行分隔的邮件正文组成。DATA 实际上是一组命令，服务器回复两次：一次是 DATA 命令本身，确认它已准备好接收文本，第二次是在数据结束序列之后，接受或拒绝整个消息。

Besides the intermediate reply for DATA, each server's reply can be either positive (2xx reply codes) or negative. Negative replies can be permanent (5xx codes) or transient (4xx codes). A **reject** is a permanent failure and the client should send a bounce message to the server it received

it from. A **drop** is a positive response followed by message discard rather than delivery.

除了 DATA 的中间回复外，每个服务器的回复可以是肯定的（2xx 回复代码）或否定的。否定回复可以是永久性（5xx 代码）或暂时性（4xx 代码）。拒绝是永久性故障，客户端应向其接收退回邮件的服务器发送退回邮件。丢弃是肯定响应，然后是丢弃消息而不是传递。

The initiating host, the SMTP client, can be either an end-user's email client, functionally identified as a mail user agent (MUA), or a relay server's mail transfer agent (MTA), that is an SMTP server acting as an SMTP client, in the relevant session, in order to relay mail. Fully capable SMTP servers maintain queues of messages for retrying message transmissions that resulted in transient failures.

发起主机 SMTP 客户端可以是最终用户的电子邮件客户端（在功能上标识为邮件用户代理（MUA），也可以是中继服务器的邮件传输代理 （MTA），即在相关会话中充当 SMTP 客户端的 SMTP 服务器，以便中继邮件。功能齐全的 SMTP 服务器维护邮件队列，以便重试导致暂时性故障的邮件传输。

A MUA knows the *outgoing mail* SMTP server from its configuration. A relay server typically determines which server to connect to by looking up the MX (Mail eXchange) DNS resource record for each recipient's domain name. If no MX record is found, a conformant relaying server (not all are) instead looks up the A record. Relay servers can also be configured to use a smart host. A relay server initiates a TCP connection to the server on the "well-known port" for SMTP: port 25, or for connecting to an MSA, port 587. The main difference between an MTA and an MSA is that connecting to an MSA requires SMTP Authentication.

MUA 从其配置中知道传出邮件 SMTP 服务器。中继服务器通常通过查找每个收件人域名的 MX（邮件交换）DNS 资源记录来确定要连接到哪个服务器。如果未找到 MX 记录，则一致的中继服务器（并非所有中继服务器）将改为查找 A 记录。中继服务器也可以配置为使用智能主机。中继服务器在 SMTP（端口 25）的“已知端口”上启动与服务器的 TCP 连接，或者用于连接到 MSA（端口 587）。MTA 和 MSA 之间的主要区别在于连接到 MSA 需要 SMTP 身份验证。

SMTP vs mail retrieval SMTP与邮件检索

SMTP is a delivery protocol only. In normal use, mail is "pushed" to a destination mail server (or next-hop mail server) as it arrives. Mail is routed based on the destination server, not the individual user(s) to which it is addressed. Other protocols, such as the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP) are specifically designed for use by individual users retrieving messages and managing mail boxes. To permit an intermittently-connected mail server to *pull* messages from a remote server on demand, SMTP has a feature to initiate mail queue processing on a remote server (see Remote Message Queue Starting below). POP and IMAP are unsuitable protocols for relaying mail by intermittently-connected machines; they are designed to operate after final delivery, when information critical to the correct operation of mail relay (the

"mail envelope") has been removed.

SMTP 仅是一种传递协议。在正常使用中，邮件在到达时被“推送”到目标邮件服务器（或下一跃点邮件服务器）。邮件的路由基于目标服务器，而不是其寻址到的单个用户。其他协议（如邮局协议（POP） 和 Internet 邮件访问协议 （IMAP））专门设计用于个人用户检索邮件和管理邮箱。为了允许间歇性连接的邮件服务器按需从远程服务器拉取邮件，SMTP 具有在远程服务器上启动邮件队列处理的功能（请参阅下面的远程邮件队列开始）。POP 和 IMAP 不适合通过间歇性连接的机器中继邮件;它们被设计为在最终交付后运行，当对邮件中继的正确操作至关重要的信息（“邮件信封”）已被删除时。

Remote Message Queue Starting

远程消息队列启动

Remote Message Queue Starting enables a remote host to start processing of the mail queue on a server so it may receive messages destined to it by sending a corresponding command. The original `TURN` command was deemed insecure and was extended in RFC 1985 (https://datatracker.ietf.org/doc/html/rfc1985) with the `ETRN` command which operates more securely using an authentication method based on Domain Name System information.[19]

远程邮件队列启动使远程主机能够开始处理服务器上的邮件队列，以便它可以通过发送相应的命令来接收发往它的邮件。原始 `TURN` 命令被认为是不安全的，并在 RFC 1985 中进行了扩展， `ETRN` 该命令使用基于域名系统信息的身份验证方法更安全地运行。 [19]

Outgoing mail SMTP server

发送邮件 **SMTP** 服务器

An email client needs to know the IP address of its initial SMTP server and this has to be given as part of its configuration (usually given as a DNS name). This server will deliver outgoing messages on behalf of the user.

电子邮件客户端需要知道其初始SMTP服务器的IP地址，并且必须将其作为其配置的一部分（通常作为DNS名称提供）。此服务器将代表用户传递传出邮件。

Outgoing mail server access restrictions

发送邮件服务器访问限制

Server administrators need to impose some control on which clients can use the server. This enables them to deal with abuse, for example spam. Two solutions have been in common use:

服务器管理员需要对哪些客户端可以使用服务器施加一些控制。这使他们能够处理滥用行为，例如垃圾邮件。两种解决方案已普遍使用：

- In the past, many systems imposed usage restrictions by the *location* of the client, only permitting usage by clients whose IP address is one that the server administrators control. Usage from any other client IP address is disallowed.

  过去，许多系统按客户端位置施加使用限制，仅允许其 IP 地址是服务器管理员控制的客户端使用。不允许从任何其他客户端 IP 地址使用。

- Modern SMTP servers typically offer an alternative system that requires authentication of clients by credentials before allowing access.

现代 SMTP 服务器通常提供另一种系统，该系统要求在允许访问之前通过凭据对客户端进行身份验证。

Restricting access by location

按位置限制访问

Under this system, an ISP's SMTP server will not allow access by users who are outside the ISP's network. More precisely, the server may only allow access to users with an IP address provided by the ISP, which is equivalent to requiring that they are connected to the Internet using that same ISP. A mobile user may often be on a network other than that of their normal ISP, and will then find that sending email fails because the configured SMTP server choice is no longer accessible.

在此系统下，ISP 的 SMTP 服务器将不允许 ISP 网络外部的用户访问。更准确地说，服务器可能只允许具有ISP提供的IP地址的用户访问，这相当于要求他们使用相同的ISP连接到Internet。移动用户可能经常在其普通 ISP 的网络之外使用，然后会发现发送电子邮件失败，因为配置的 SMTP 服务器选择不再可访问。

This system has several variations. For example, an organisation's SMTP server may only provide service to users on the same network, enforcing this by firewalling to block access by users on the wider Internet. Or the server may perform range checks on the client's IP address. These methods were typically used by corporations and institutions such as universities which provided an SMTP server for outbound mail only for use internally within the organisation. However, most of these bodies now use client authentication methods, as described below.

该系统有多种变体。例如，组织的SMTP服务器可能只向同一网络上的用户提供服务，通过防火墙来阻止用户在更广泛的Internet上的访问来强制实施这一点。或者，服务器可能会对客户端的 IP 地址执行范围检查。这些方法通常由公司和机构（如大学）使用，这些公司和机构为出站邮件提供SMTP服务器，仅供组织内部使用。但是，这些机构中的大多数现在都使用客户端身份验证方法，如下所述。

Where a user is mobile, and may use different ISPs to connect to the internet, this kind of usage restriction is onerous, and altering the configured outbound email SMTP server address is impractical. It is highly desirable to be able to use email client configuration information that does not need to change.

如果用户是移动用户，并且可能使用不同的 ISP 连接到互联网，则这种使用限制非常繁琐，并且更改配置的出站电子邮件 SMTP 服务器地址是不切实际的。非常希望能够使用不需要更改的电子邮件客户端配置信息。

Client authentication 客户端身份验证

Modern SMTP servers typically require authentication of clients by credentials before allowing access, rather than restricting access by location as described earlier. This more flexible system is friendly to mobile users and allows them to have a fixed choice of configured outbound SMTP server. SMTP Authentication, often abbreviated SMTP AUTH, is an extension of the SMTP in order to log in using an authentication mechanism.

现代 SMTP 服务器通常要求在允许访问之前通过凭据对客户端进行身份验证，而不是如前所述按位置限制访问。这个更灵活的系统对移动用户很友好，并允许他们有固定的配置出站 SMTP 服务器选择。SMTP 身份验证，通常缩写为 SMTP AUTH，是 SMTP 的扩展，以便使用身份验证机制登录。

Ports 港口

Communication between mail servers generally uses the standard TCP port 25 designated for SMTP.

邮件服务器之间的通信通常使用为 SMTP 指定的标准 TCP 端口 25。

Mail *clients* however generally don't use this, instead using specific "submission" ports. Mail services generally accept email submission from clients on one of:

然而，邮件客户端通常不使用它，而是使用特定的“提交”端口。邮件服务通常接受客户通过以下任一方式提交的电子邮件：

- 587 (Submission), as formalized in RFC 6409 (https://datatracker.ietf.org/doc/html/rfc6409) (previously RFC 2476 (https://datatracker.ietf.org/doc/html/rfc2476))

  587（提交），在 RFC 6409（以前为 RFC 2476）中正式化

- 465 This port was deprecated after RFC 2487 (https://datatracker.ietf.org/doc/html/rfc2487), until the issue of RFC 8314 (https://datatracker.ietf.org/doc/html/rfc8314).

  465 此端口在 RFC 2487 之后被弃用，直到 RFC 8314 发布。

Port 2525 and others may be used by some individual providers, but have never been officially supported.

端口 2525 和其他端口可能由某些单独的提供程序使用，但从未得到官方支持。

Many Internet service providers now block all outgoing port 25 traffic from their customers. Mainly as an anti-spam measure,[20] but also to cure for the higher cost they have when leaving it open, perhaps by charging more from the few customers that require it open.

许多互联网服务提供商现在阻止来自其客户的所有传出端口 25 流量。主要作为一种反垃圾邮件措施，但也是为了解决他们在保持打开状态时更高的成本，[20] 也许是通过向需要打开它的少数客户收取更多费用。

SMTP transport example SMTP 传输示例

A typical example of sending a message via SMTP to two mailboxes (*alice* and *theboss*) located in the same mail domain (*example.com*) is reproduced in the following session exchange. (In this example, the conversation parts are prefixed with *S:* and *C:*, for *server* and *client*, respectively; these labels are not part of the exchange.)

通过 SMTP 将邮件发送到位于同一邮件域（example.com）中的两个邮箱（alice 和 theboss）的典型示例在以下会话交换中重现。（在此示例中，服务器和客户端的对话部分分别以 S： 和 C： 为前缀;这些标签不是交换的一部分。

After the message sender (SMTP client) establishes a reliable communications channel to the message receiver (SMTP server), the session is opened with a greeting by the server, usually containing its fully qualified domain name (FQDN), in this case *smtp.example.com*. The client initiates its dialog by responding with a `HELO` command identifying itself in the command's parameter with its FQDN (or an address literal if none is available).[21]

在邮件发送方（SMTP 客户端）与邮件接收方（SMTP 服务器）建立可靠的通信通道后，会话将打开，服务器会发出问候语，通常包含其完全限定域名（FQDN），在本例中为 smtp.example.com。客户端通过响应命令来启动其对话，该命令在 HELO 命令的参数中标识自身及其 FQDN（如果没有可用的地址文本）。[21]

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

The client notifies the receiver of the originating email address of the message in a `MAIL FROM` command. This is also the return or bounce address in case the message cannot be delivered. In this example the email message is sent to two mailboxes on the same SMTP server: one for each recipient listed in the `To:` and `Cc:` header fields. The corresponding SMTP command is `RCPT TO`. Each successful reception and execution of a command is acknowledged by the server with a result code and response message (e.g., `250 Ok`).

客户端在 MAIL FROM 命令中通知收件人邮件的原始电子邮件地址。这也是邮件无法传递时的返回或退回地址。在此示例中，电子邮件发送到同一 SMTP 服务器上的两个邮箱： To: 和 Cc: 标头字段中列出的每个收件人一个邮箱。相应的 SMTP 命令是 RCPT TO 。每次成功接收和执行命令都会由服务器确认，并带有结果代码和响应消息（例如， 250 Ok ）。

The transmission of the body of the mail message is initiated with a `DATA` command after which it is transmitted verbatim line by line and is terminated with an end-of-data sequence. This sequence consists of a new-line (`<CR><LF>`), a single full stop (`.`), followed by another new-line (`<CR><LF>`). Since a message body can contain a line with just a period as part of the text, the client sends *two* periods every time a line starts with a period; correspondingly, the server replaces every sequence of two periods at the beginning of a line with a single one. Such escaping method is called *dot-stuffing*.

邮件消息正文的传输通过命令 DATA 启动，之后逐行逐行传输，并以数据结束序列终止。这个序列由一个换行符 （ <CR><LF> ）、一个句号（ . ）组成，后跟另一个换行符 （ <CR><LF> ）。由于消息正文可以包含仅包含一个句点作为文本一部分的行，因此每次行以句点开头时，客户端都会发送两个句点;相应地，服务器将行首的两个句点的每个序列替换为单个序列。这种转义方法称为填点。

The server's positive reply to the end-of-data, as exemplified, implies that the server has taken the responsibility of delivering the message. A message can be doubled if there is a communication failure at this time, e.g. due to a power shortage: Until the sender has received that `250 Ok` reply, it must assume the message was not delivered. On the other hand, after the receiver has decided to accept the message, it must assume the message has been delivered to it. Thus, during this time span, both agents have active copies of the message that they will try to deliver.[22] The probability that a communication failure occurs exactly at this step is directly proportional to the amount of

filtering that the server performs on the message body, most often for anti-spam purposes. The limiting timeout is specified to be 10 minutes.[23]

例如，服务器对数据末尾的肯定回复意味着服务器已承担传递消息的责任。如果此时出现通信故障（例如由于电源不足），则消息可以加倍：在发件人收到该 250 Ok 回复之前，它必须假定消息未送达。另一方面，在接收方决定接受消息后，它必须假定消息已传递给它。因此，在这段时间内，两个代理都有它们将尝试传递的消息的活动副本。[22] 在此步骤中恰好发生通信故障的概率与服务器对邮件正文执行的筛选量成正比，通常用于反垃圾邮件目的。限制超时指定为 10 分钟。[23]

The `QUIT` command ends the session. If the email has other recipients located elsewhere, the client would `QUIT` and connect to an appropriate SMTP server for subsequent recipients after the current destination(s) had been queued. The information that the client sends in the `HELO` and `MAIL FROM` commands are added (not seen in example code) as additional header fields to the message by the receiving server. It adds a `Received` and `Return-Path` header field, respectively.

该命令将 QUIT 结束会话。如果电子邮件的其他收件人位于其他位置，则客户端将在 QUIT 当前目标排队后连接到相应的 SMTP 服务器以供后续收件人使用。客户端在 and HELO MAIL FROM 命令中发送的信息由接收服务器添加（在示例代码中未显示）作为附加标头字段添加到消息中。它分别添加一个 Received and Return-Path 标头字段。

Some clients are implemented to close the connection after the message is accepted (`250 Ok: queued as 12345`), so the last two lines may actually be omitted. This causes an error on the server when trying to send the `221 Bye` reply.

某些客户端被实现为在接受消息（ 250 Ok: queued as 12345 ）后关闭连接，因此实际上可以省略最后两行。这会导致在尝试发送 221 Bye 答复时服务器上出错。

SMTP Extensions SMTP 扩展

Extension discovery mechanism

扩展发现机制

Clients learn a server's supported options by using the `EHLO` greeting, as exemplified below, instead of the original `HELO`. Clients fall back to `HELO` only if the server does not support `EHLO` greeting.[24]

客户端通过使用问候语（如下例所示）而不是原始 EHLO HELO 问候语来学习服务器支持的选项。仅当服务器不支持 EHLO 问候语时，客户端才会回退到。 HELO [24]

Modern clients may use the ESMTP extension keyword `SIZE` to query the server for the maximum message size that will be accepted. Older clients and servers may try to transfer excessively sized messages that will be rejected after consuming network resources, including connect time to network links that is paid by the minute.[25]

现代客户端可以使用 ESMTP 扩展关键字 SIZE 向服务器查询将接受的最大邮件大小。较旧的客户端和服务器可能会尝试传输过大的邮件，这些邮件将在消耗网络资源后被拒绝，包括按分钟付费的网络连接时间。[25]

Users can manually determine in advance the maximum size accepted by ESMTP servers. The client replaces the `HELO` command with the `EHLO` command.

用户可以手动提前确定 ESMTP 服务器接受的最大大小。客户端将命令 HELO 替换为 EHLO 命令。

```
S: 220 smtp2.example.com ESMTP Postfix
C: EHLO bob.example.org
```

```
S: 250-smtp2.example.com Hello bob.example.org [192.0.2.201]
S: 250-SIZE 14680064
S: 250-PIPELINING
S: 250 HELP
```

Thus *smtp2.example.com* declares that it can accept a fixed maximum message size no larger than 14,680,064 octets (8-bit bytes).

因此，smtp2.example.com 声明它可以接受不大于 14,680,064 个八位字节（8 位字节）的固定最大消息大小。

In the simplest case, an ESMTP server declares a maximum `SIZE` immediately after receiving an `EHLO`. According to RFC 1870 (https://datatracker.ietf.org/doc/html/rfc1870), however, the numeric parameter to the `SIZE` extension in the `EHLO` response is optional. Clients may instead, when issuing a `MAIL FROM` command, include a numeric estimate of the size of the message they are transferring, so that the server can refuse receipt of overly-large messages.

在最简单的情况下，ESMTP 服务器 SIZE 在收到 EHLO .但是，根据 RFC 1870， EHLO 响应中 SIZE 扩展的数字参数是可选的。相反，客户端在发出 MAIL FROM 命令时，可以包括它们正在传输的消息大小的数字估计值，以便服务器可以拒绝接收过大的消息。

Binary data transfer

Original SMTP supports only a single body of ASCII text, therefore any binary data needs to be encoded as text into that body of the message before transfer, and then decoded by the recipient. Binary-to-text encodings, such as uuencode and BinHex were typically used.

The 8BITMIME command was developed to address this. It was standardized in 1994 as RFC 1652 (https://datatracker.ietf.org/doc/html/rfc1652)[26] It facilitates the transparent exchange of e-mail messages containing octets outside the seven-bit ASCII character set by encoding them as MIME content parts, typically encoded with Base64.

Mail delivery mechanism extensions

On-Demand Mail Relay

On-Demand Mail Relay (**ODMR**) is an SMTP extension standardized in RFC 2645 (https://datatracker.ietf.org/doc/html/rfc2645) that allows an intermittently-connected SMTP server to receive email queued for it when it is connected.

Internationalization extension

Original SMTP supports email addresses composed of ASCII characters only, which is inconvenient for users whose native script is not Latin based, or who use diacritic not in the ASCII character set. This limitation was alleviated via extensions enabling UTF-8 in address names. RFC 5336 (https://datatracker.ietf.org/doc/html/rfc5336) introduced experimental[25] `UTF8SMTP` command and later was superseded by RFC 6531 (https://datatracker.ietf.org/doc/html/rfc6531) that introduced `SMTPUTF8` command. These extensions provide support for multi-byte and non-ASCII characters in email addresses, such as those with diacritics and other language characters such as Greek and Chinese.[27]

Current support is limited, but there is strong interest in broad adoption of RFC 6531 (https://datatracker.ietf.org/doc/html/rfc6531) and the related RFCs in countries like China that have a large user base where Latin (ASCII) is a foreign script.

Extensions

Like SMTP, ESMTP is a protocol used to transport Internet mail. It is used as both an inter-server transport protocol and (with restricted behavior enforced) a mail submission protocol.

The main identification feature for ESMTP clients is to open a transmission with the command `EHLO` (Extended HELLO), rather than `HELO` (Hello, the original RFC 821 (https://datatracker.ietf.org/doc/html/rfc821) standard). A server will respond with success (code 250), failure (code 550) or error (code 500, 501, 502, 504, or 421), depending on its configuration. An ESMTP server returns the code 250 OK in a multi-line reply with its domain and a list of keywords to indicate supported extensions. A RFC 821 compliant server returns error code 500, allowing ESMTP clients to try either `HELO` or `QUIT`.

Each service extension is defined in an approved format in subsequent RFCs and registered with the Internet Assigned Numbers Authority (IANA). The first definitions were the RFC 821 optional services: `SEND`, `SOML` (Send or Mail), `SAML` (Send and Mail), `EXPN`, `HELP`, and `TURN`. The format of additional SMTP verbs was set and for new parameters in `MAIL` and `RCPT`.

Some relatively common keywords (not all of them corresponding to commands) used today are:

- `8BITMIME` – 8 bit data transmission, RFC 6152 (https://datatracker.ietf.org/doc/html/rfc6152)
- `ATRN` – Authenticated `TURN` for On-Demand Mail Relay, RFC 2645 (https://datatracker.ietf.org/doc/html/rfc2645)
- `AUTH` – Authenticated SMTP, RFC 4954 (https://datatracker.ietf.org/doc/html/rfc4954)
- `CHUNKING` – Chunking, RFC 3030 (https://datatracker.ietf.org/doc/html/rfc3030)
- `DSN` – Delivery status notification, RFC 3461 (https://datatracker.ietf.org/doc/html/rfc3461) (See Variable envelope return path)
- `ETRN` – Extended version of remote message queue starting command `TURN`, RFC 1985 (https://datatracker.ietf.org/doc/html/rfc1985)
- `HELP` – Supply helpful information, RFC 821 (https://datatracker.ietf.org/doc/html/rfc821)
- `PIPELINING` – Command pipelining, RFC 2920 (https://datatracker.ietf.org/doc/html/rfc2920)
- `SIZE` – Message size declaration, RFC 1870 (https://datatracker.ietf.org/doc/html/rfc1870)
- `STARTTLS` – Transport Layer Security, RFC 3207 (https://datatracker.ietf.org/doc/html/rfc3207) (2002)
- `SMTPUTF8` – Allow UTF-8 encoding in mailbox names and header fields, RFC 6531 (https://datatracker.ietf.org/doc/html/rfc6531)
- `UTF8SMTP` – Allow UTF-8 encoding in mailbox names and header fields, RFC 5336 (https://datatracker.ietf.org/doc/html/rfc5336) (deprecated[28])

The ESMTP format was restated in RFC 2821 (https://datatracker.ietf.org/doc/html/rfc2821) (superseding RFC 821) and updated to the latest definition in RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) in 2008. Support for the `EHLO` command in servers became mandatory, and `HELO` designated a required fallback.

Non-standard, unregistered, service extensions can be used by bilateral agreement, these services are indicated by an `EHLO` message keyword starting with "X", and with any additional parameters or verbs similarly marked.

SMTP commands are case-insensitive. They are presented here in capitalized form for emphasis only. An SMTP server that requires a specific capitalization method is a violation of the standard.

8BITMIME

At least the following servers advertise the 8BITMIME extension:

- Apache James (since 2.3.0a1)[29]
- Citadel (since 7.30)
- Courier Mail Server
- Gmail[30]
- IceWarp
- IIS SMTP Service
- Kerio Connect
- Lotus Domino
- Microsoft Exchange Server (as of Exchange Server 2000)
- Novell GroupWise
- OpenSMTPD
- Oracle Communications Messaging Server
- Postfix
- Sendmail (since 6.57)

The following servers can be configured to advertise 8BITMIME, but do not perform conversion of 8-bit data to 7-bit when connecting to non-8BITMIME relays:

- Exim and qmail do not translate eight-bit messages to seven-bit when making an attempt to relay 8-bit data to non-8BITMIME peers, as is required by the RFC.[31] This does not cause problems in practice, since virtually all modern mail relays are 8-bit clean.[32]
- Microsoft Exchange Server 2003 advertises 8BITMIME by default, but relaying to a non-8BITMIME peer results in a bounce. This is allowed by RFC 6152 section 3 (http://tools.ietf.org/html/rfc6152#section-3).

SMTP-AUTH

The SMTP-AUTH extension provides an access control mechanism. It consists of an authentication step through which the client effectively logs into the mail server during the process of sending mail. Servers that support SMTP-AUTH can usually be configured to require clients to use this extension, ensuring the true identity of the sender is known. The SMTP-AUTH extension is defined in RFC 4954 (https://datatracker.ietf.org/doc/html/rfc4954).

SMTP-AUTH can be used to allow legitimate users to relay mail while denying relay service to unauthorized users, such as spammers. It does not necessarily guarantee the authenticity of either the SMTP envelope sender or the RFC 2822 (https://datatracker.ietf.org/doc/html/rfc2822) "From:" header. For example, spoofing, in which one sender masquerades as someone else, is still possible with SMTP-AUTH unless the server is configured to limit message from-addresses to addresses this AUTHed user is authorized for.

The SMTP-AUTH extension also allows one mail server to indicate to another that the sender has been authenticated when relaying mail. In general this requires the recipient server to trust the sending server, meaning that this aspect of SMTP-AUTH is rarely used on the Internet.

SMTPUTF8

Supporting servers include:

- Postfix (version 3.0 and later)[33]
- Momentum (versions 4.1[34] and 3.6.5, and later)
- Sendmail (experimental support in 8.17.1)
- Exim (experimental as of the 4.86 release, quite mature in 4.96)
- CommuniGate Pro as of version 6.2.2[35]
- Courier-MTA as of version 1.0[36]
- Halon as of version 4.0[37]
- Microsoft Exchange Server as of protocol revision 14.0[38]
- Haraka and other servers.[39]
- Oracle Communications Messaging Server as of release 8.0.2.[40]

Security extensions

Mail delivery can occur both over plain text and encrypted connections, however the communicating parties might not know in advance of other party's ability to use secure channel.

STARTTLS or "Opportunistic TLS"

The STARTTLS extensions enables supporting SMTP servers to notify connecting clients that it supports TLS encrypted communication and offers the opportunity for clients to upgrade their connection by sending the STARTTLS command. Servers supporting the extension do not inherently gain any security benefits from its implementation on its own, as upgrading to a TLS encrypted session is dependent on the connecting client deciding to exercise this option, hence the term *opportunistic* TLS.

STARTTLS is effective only against passive observation attacks, since the STARTTLS negotiation happens in plain text and an active attacker can trivially remove STARTTLS commands. This type of man-in-the-middle attack is sometimes referred to as STRIPTLS, where the encryption negotiation information sent from one end never reaches the other. In this scenario both parties take the invalid or unexpected responses as indication that the other does not properly support STARTTLS, defaulting to traditional plain-text mail transfer.[41] Note that STARTTLS is also defined for IMAP and POP3 in other RFCs, but these protocols serve different purposes: SMTP is used for communication between message transfer agents, while IMAP and POP3 are for end clients and message transfer agents.

In 2014 the Electronic Frontier Foundation began "STARTTLS Everywhere" project that, similarly to "HTTPS Everywhere" list, allowed relying parties to discover others supporting secure communication without prior communication. The project stopped accepting submissions on 29 April 2021, and EFF recommended switching to DANE and MTA-STS for discovering information on peers' TLS support.[42]

RFC 8314 (https://datatracker.ietf.org/doc/html/rfc8314) officially declared plain text obsolete and recommend always using TLS for mail submission and access, adding ports with implicit TLS.

SMTP MTA Strict Transport Security

A newer 2018 RFC 8461 (https://datatracker.ietf.org/doc/html/rfc8461) called "SMTP MTA Strict Transport Security (MTA-STS)" aims to address the problem of active adversary by defining a protocol for mail servers to declare their ability to use secure channels in specific files on the server and specific DNS TXT records. The relying party would regularly check existence of such record, and cache it for the amount of time specified in the record and never communicate over insecure channels until record expires.[41] Note that MTA-STS records apply only to SMTP traffic between mail servers while communications between a user's client and the mail server are protected by Transport Layer Security with SMTP/MSA, IMAP, POP3, or HTTPS in combination with an organizational or technical policy. Essentially, MTA-STS is a means to extend such a policy to third parties.

In April 2019 Google Mail announced support for MTA-STS.[43]

SMTP TLS Reporting

Protocols designed to securely deliver messages can fail due to misconfigurations or deliberate active interference, leading to undelivered messages or delivery over unencrypted or unauthenticated channels. RFC 8460 (https://datatracker.ietf.org/doc/html/rfc8460) "SMTP TLS Reporting" describes a reporting mechanism and format for sharing statistics and specific information about potential failures with recipient domains. Recipient domains can then use this information to both detect potential attacks and diagnose unintentional misconfigurations.

In April 2019 Google Mail announced support for SMTP TLS Reporting.[43]

Spoofing and spamming

The original design of SMTP had no facility to authenticate senders, or check that servers were authorized to send on their behalf, with the result that email spoofing is possible, and commonly used in email spam and phishing.

Occasional proposals are made to modify SMTP extensively or replace it completely. One example of this is Internet Mail 2000, but neither it, nor any other has made much headway in the face of the network effect of the huge installed base of classic SMTP.

Instead, mail servers now use a range of techniques, such as stricter enforcement of standards such as RFC 5322 (https://datatracker.ietf.org/doc/html/rfc5322),[44][45] DomainKeys Identified Mail, Sender Policy Framework and DMARC, DNSBLs and greylisting to reject or quarantine suspicious emails.[46]

Implementations

Related requests for comments

- RFC 1123 (https://datatracker.ietf.org/doc/html/rfc1123) – Requirements for Internet Hosts—Application and Support (STD 3)

- RFC 1870 (https://datatracker.ietf.org/doc/html/rfc1870) – SMTP Service Extension for Message Size Declaration (obsoletes: RFC 1653 (https://datatracker.ietf.org/doc/html/rfc1653))
- RFC 2505 (https://datatracker.ietf.org/doc/html/rfc2505) – Anti-Spam Recommendations for SMTP MTAs (BCP 30)
- RFC 2821 (https://datatracker.ietf.org/doc/html/rfc2821) – Simple Mail Transfer Protocol
- RFC 2920 (https://datatracker.ietf.org/doc/html/rfc2920) – SMTP Service Extension for Command Pipelining (STD 60)
- RFC 3030 (https://datatracker.ietf.org/doc/html/rfc3030) – SMTP Service Extensions for Transmission of Large and Binary MIME Messages
- RFC 3207 (https://datatracker.ietf.org/doc/html/rfc3207) – SMTP Service Extension for Secure SMTP over Transport Layer Security (obsoletes RFC 2487 (https://datatracker.ietf.org/doc/html/rfc2487))
- RFC 3461 (https://datatracker.ietf.org/doc/html/rfc3461) – SMTP Service Extension for Delivery Status Notifications (obsoletes RFC 1891 (https://datatracker.ietf.org/doc/html/rfc1891))
- RFC 3463 (https://datatracker.ietf.org/doc/html/rfc3463) – Enhanced Status Codes for SMTP (obsoletes RFC 1893 (https://datatracker.ietf.org/doc/html/rfc1893), updated by RFC 5248 (https://datatracker.ietf.org/doc/html/rfc5248))
- RFC 3464 (https://datatracker.ietf.org/doc/html/rfc3464) – An Extensible Message Format for Delivery Status Notifications (obsoletes RFC 1894 (https://datatracker.ietf.org/doc/html/rfc1894))
- RFC 3798 (https://datatracker.ietf.org/doc/html/rfc3798) – Message Disposition Notification (updates RFC 3461 (https://datatracker.ietf.org/doc/html/rfc3461))
- RFC 3834 (https://datatracker.ietf.org/doc/html/rfc3834) – Recommendations for Automatic Responses to Electronic Mail
- RFC 3974 (https://datatracker.ietf.org/doc/html/rfc3974) – SMTP Operational Experience in Mixed IPv4/v6 Environments
- RFC 4952 (https://datatracker.ietf.org/doc/html/rfc4952) – Overview and Framework for Internationalized Email (updated by RFC 5336 (https://datatracker.ietf.org/doc/html/rfc5336))
- RFC 4954 (https://datatracker.ietf.org/doc/html/rfc4954) – SMTP Service Extension for Authentication (obsoletes RFC 2554 (https://datatracker.ietf.org/doc/html/rfc2554), updates RFC 3463 (https://datatracker.ietf.org/doc/html/rfc3463), updated by RFC 5248 (https://datatracker.ietf.org/doc/html/rfc5248))
- RFC 5068 (https://datatracker.ietf.org/doc/html/rfc5068) – Email Submission Operations: Access and Accountability Requirements (BCP 134)
- RFC 5248 (https://datatracker.ietf.org/doc/html/rfc5248) – A Registry for SMTP Enhanced Mail System Status Codes (BCP 138) (updates RFC 3463 (https://datatracker.ietf.org/doc/html/rfc3463))
- RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) – The Simple Mail Transfer Protocol (obsoletes RFC 821 (https://datatracker.ietf.org/doc/html/rfc821) aka STD 10, RFC 974 (https://datatracker.ietf.org/doc/html/rfc974), RFC 1869 (https://datatracker.ietf.org/doc/html/rfc1869), RFC 2821 (https://datatracker.ietf.org/doc/html/rfc2821), updates RFC 1123 (https://datatracker.ietf.org/doc/html/rfc1123))
- RFC 5322 (https://datatracker.ietf.org/doc/html/rfc5322) – Internet Message Format (obsoletes RFC 822 (https://datatracker.ietf.org/doc/html/rfc822) aka STD 11, and RFC 2822 (https://datatracker.ietf.org/doc/html/rfc2822))
- RFC 5504 (https://datatracker.ietf.org/doc/html/rfc5504) – Downgrading Mechanism for Email Address Internationalization
- RFC 6409 (https://datatracker.ietf.org/doc/html/rfc6409) – Message Submission for Mail (STD 72) (obsoletes RFC 4409 (https://datatracker.ietf.org/doc/html/rfc4409), RFC 2476 (https://datatracker.ietf.org/doc/html/rfc2476))
- RFC 6522 (https://datatracker.ietf.org/doc/html/rfc6522) – The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages (obsoletes RFC 3462 (https://datatr

acker.ietf.org/doc/html/rfc3462), and in turn RFC 1892 (https://datatracker.ietf.org/doc/html/rfc1892))
- RFC 6531 (https://datatracker.ietf.org/doc/html/rfc6531) – SMTP Extension for Internationalized Email Addresses (updates RFC 2821 (https://datatracker.ietf.org/doc/html/rfc2821), RFC 2822 (https://datatracker.ietf.org/doc/html/rfc2822), RFC 4952 (https://datatracker.ietf.org/doc/html/rfc4952), and RFC 5336 (https://datatracker.ietf.org/doc/html/rfc5336))
- RFC 8314 (https://datatracker.ietf.org/doc/html/rfc8314) – Cleartext Considered Obsolete: Use of Transport Layer Security (TLS) for Email Submission and Access

See also

- Bounce address
- CRAM-MD5 (a SASL mechanism for ESMTPA) RFC 2195 (https://datatracker.ietf.org/doc/html/rfc2195)
- Email
  - Email encryption
- DKIM
- Ident
- List of mail server software
- List of SMTP server return codes
- POP before SMTP / SMTP after POP
- Internet Message Access Protocol Binary Content Extension RFC 3516 (https://datatracker.ietf.org/doc/html/rfc3516)
- Sender Policy Framework (SPF)
- Simple Authentication and Security Layer (SASL) RFC 4422 (https://datatracker.ietf.org/doc/html/rfc4422)
- SMTP Authentication
- Variable envelope return path
- Comparison of email clients for information about SMTP support

Notes

1. The History of Electronic Mail (http://www.multicians.org/thvv/mail-history.html) Archived (https://web.archive.org/web/20171202025034/http://www.multicians.org/thvv/mail-history.html) December 2, 2017, at the Wayback Machine, *Tom Van Vleck*: "It is not clear this protocol was ever implemented"
2. *The First Network Email* (https://openmap.bbn.com/~tomlinso/ray/firstemailframe.html), Ray Tomlinson, BBN
3. Picture of "The First Email Computer (https://openmap.bbn.com/~tomlinso/ray/ka10.html)" by Dan Murphy, a PDP-10
4. Dan Murphy's TENEX and TOPS-20 Papers (http://www.opost.com/dlm/tenex/) Archived (https://web.archive.org/web/20071118204016/http://www.opost.com/dlm/tenex/) November 18, 2007, at the Wayback Machine
5. RFC 524 (https://datatracker.ietf.org/doc/html/rfc524) – A Proposed Mail Protocol
6. Crocker, David H. (December 1977). "Framework and Functions of the "MS" Personal Message System" (https://www.rand.org/content/dam/rand/pubs/reports/2007/R2134.pdf) (PDF). *The RAND Corporation*. Archived (https://web.archive.org/web/20220513083616/https://www.rand.org/content/dam/rand/pubs/reports/2007/R2134.pdf) (PDF) from the original on May 13, 2022. Retrieved April 17, 2022.
7. RFC 469 (https://datatracker.ietf.org/doc/html/rfc469) – Network Mail Meeting Summary

8. RFC 733, 21 November 1977, Standard for the Format of ARPA Network Text Message
9. "Tldp.org" (http://tldp.org/HOWTO/Usenet-News-HOWTO/x64.html). Archived (https://web.archive.org/web/20070817090558/http://tldp.org/HOWTO/Usenet-News-HOWTO/x64.html) from the original on August 17, 2007. Retrieved August 25, 2007.
10. Barber, Stan O. (December 19, 2000). "draft-barber-uucp-project-conclusion-05 – The Conclusion of the UUCP Mapping Project" (https://tools.ietf.org/html/draft-barber-uucp-project-conclusion-05). Archived (https://web.archive.org/web/20071013094756/http://tools.ietf.org/html/draft-barber-uucp-project-conclusion-05) from the original on October 13, 2007. Retrieved August 25, 2007.
11. The article about sender rewriting contains technical background info about the early SMTP history and source routing before RFC 1123 (https://datatracker.ietf.org/doc/html/rfc1123).
12. Eric Allman (1983), *Sendmail – An Internetwork Mail Router* (https://docs.freebsd.org/44doc/smm/09.sendmail/paper.pdf) (PDF), BSD UNIX documentation set, Berkeley: University of California, archived (https://web.archive.org/web/20130520171455/http://docs.freebsd.org/44doc/smm/09.sendmail/paper.pdf) (PDF) from the original on May 20, 2013, retrieved June 29, 2012
13. Craig Partridge (2008), *The Technical Development of Internet Email* (https://web.archive.org/web/20110512165437/http://www.ir.bbn.com/~craig/email.pdf) (PDF), IEEE Annals of the History of Computing, vol. 30, IEEE Computer Society, pp. 3–29, doi:10.1109/MAHC.2008.32 (https://doi.org/10.1109%2FMAHC.2008.32), S2CID 206442868 (https://api.semanticscholar.org/CorpusID:206442868), archived from the original (http://www.ir.bbn.com/~craig/email.pdf) (PDF) on May 12, 2011
14. Paul Hoffman (February 1, 1998). "Allowing Relaying in SMTP: A Survey" (http://www.imc.org/imcr-006.html). Internet Mail Consortium. Archived (https://web.archive.org/web/20160305105916/https://www.imc.org/imcr-006.html) from the original on March 5, 2016. Retrieved May 30, 2010.
15. Paul Hoffman (August 2002). "Allowing Relaying in SMTP: A Series of Surveys" (https://web.archive.org/web/20070118121843/http://www.imc.org/ube-relay.html). Internet Mail Consortium. Archived from the original (http://www.imc.org/ube-relay.html) on January 18, 2007. Retrieved May 30, 2010.
16. "In Unix, what is an open mail relay? - Knowledge Base" (https://web.archive.org/web/20070617083024/http://kb.iu.edu/data/aivh.html). June 17, 2007. Archived from the original (http://kb.iu.edu/data/aivh.html) on June 17, 2007. Retrieved March 15, 2021.
17. "The MAIL, RCPT, and DATA verbs" (http://cr.yp.to/smtp/mail.html) Archived (https://web.archive.org/web/20140222014813/http://cr.yp.to/smtp/mail.html) February 22, 2014, at the Wayback Machine, [D. J. Bernstein]
18. RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) Section-7.2
19. Systems, Message. "Message Systems Introduces Latest Version Of Momentum With New API-Driven Capabilities" (https://www.prnewswire.com/news-releases/message-systems-introduces-latest-version-of-momentum-with-new-api-driven-capabilities-277568381.html). *www.prnewswire.com* (Press release). Archived (https://web.archive.org/web/20200719173812/https://www.prnewswire.com/news-releases/message-systems-introduces-latest-version-of-momentum-with-new-api-driven-capabilities-277568381.html) from the original on July 19, 2020. Retrieved July 19, 2020.
20. Cara Garretson (2005). "ISPs Pitch In to Stop Spam" (http://www.pcworld.com/article/116843/article.html). *PC World*. Archived (https://web.archive.org/web/20150828005734/http://www.pcworld.com/article/116843/article.html) from the original on August 28, 2015. Retrieved January 18, 2016. "Last month, the Anti-Spam Technical Alliance, formed last year by Yahoo, America Online, EarthLink, and Microsoft, issued a list of antispam recommendations that includes filtering Port 25."
21. RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321), *Simple Mail Transfer Protocol*, J. Klensin, The Internet Society (October 2008)

22. RFC 1047 (https://datatracker.ietf.org/doc/html/rfc1047)
23. Klensin, John C. (October 2008). "rfc5321#section-4.5.3.2.6" (https://tools.ietf.org/html/rfc5321#section-4.5.3.2.6). Archived (https://web.archive.org/web/20150116021100/http://tools.ietf.org/html/rfc5321#section-4.5.3.2.6) from the original on January 16, 2015. Retrieved June 7, 2010.
24. John Klensin; Ned Freed; Marshall T. Rose; Einar A. Stefferud; Dave Crocker (November 1995). *SMTP Service Extensions* (https://datatracker.ietf.org/doc/html/rfc1869). IETF. doi:10.17487/RFC1869 (https://doi.org/10.17487%2FRFC1869). RFC 1869 (https://datatracker.ietf.org/doc/html/rfc1869).
25. "MAIL Parameters" (https://www.iana.org/assignments/mail-parameters/mail-parameters.txt). IANA. February 14, 2020. Archived (https://web.archive.org/web/20190528161544/https://www.iana.org/assignments/mail-parameters/mail-parameters.txt) from the original on May 28, 2019. Retrieved May 28, 2019.
26. Which was obsoleted in 2011 by RFC 6152 (https://datatracker.ietf.org/doc/html/rfc6152) corresponding to the then new STD 71
27. Jiankang Yao (December 19, 2014). "Chinese email address" (http://www.ietf.org/mail-archive/web/ima/current/msg05395.html). *EAI* (Mailing list). IETF. Archived (https://web.archive.org/web/20151002170118/http://www.ietf.org/mail-archive/web/ima/current/msg05395.html) from the original on October 2, 2015. Retrieved May 24, 2016.
28. "SMTP Service Extension Parameters" (https://www.iana.org/assignments/mail-parameters/mail-parameters.txt). IANA. Archived (https://web.archive.org/web/20190528161544/https://www.iana.org/assignments/mail-parameters/mail-parameters.txt) from the original on May 28, 2019. Retrieved November 5, 2013.
29. James Server - ChangeLog (http://james.apache.org/server/2.3.0/changelog.html) Archived (https://web.archive.org/web/20200220075532/http://james.apache.org/server/2.3.0/changelog.html) February 20, 2020, at the Wayback Machine. James.apache.org. Retrieved on 2013-07-17.
30. 8BITMIME service advertised in response to EHLO on gmail-smtp-in.l.google.com port 25, checked 23 November 2011
31. Qmail bugs and wishlist (https://archive.today/20120630212728/http://home.pages.de/~mandree/qmail-bugs.html). Home.pages.de. Retrieved on 2013-07-17.
32. The 8BITMIME extension (http://cr.yp.to/smtp/8bitmime.html) Archived (https://web.archive.org/web/20110607045931/http://cr.yp.to/smtp/8bitmime.html) June 7, 2011, at the Wayback Machine. Cr.yp.to. Retrieved on 2013-07-17.
33. "*Postfix SMTPUTF8 support is enabled by default*" (http://www.postfix.org/SMTPUTF8_README.html) Archived (https://web.archive.org/web/20200807234345/http://www.postfix.org/SMTPUTF8_README.html) August 7, 2020, at the Wayback Machine, February 8, 2015, postfix.org
34. "Message Systems Introduces Latest Version Of Momentum With New API-Driven Capabilities" (http://www.prnewswire.com/news-releases/message-systems-introduces-latest-version-of-momentum-with-new-api-driven-capabilities-277568381.html) (Press release). Archived (https://web.archive.org/web/20200915234830/https://www.prnewswire.com/news-releases/message-systems-introduces-latest-version-of-momentum-with-new-api-driven-capabilities-277568381.html) from the original on September 15, 2020. Retrieved September 17, 2020.
35. "Version 6.2 Revision History" (https://communigate.com/Communigatepro/History62.html#6.2.2). *CommuniGate.com*. Archived (https://web.archive.org/web/20201029221905/https://www.communigate.com/Communigatepro/History62.html#6.2.2) from the original on October 29, 2020. Retrieved September 17, 2020.
36. Sam Varshavchik (September 18, 2018). "New releases of Courier packages" (https://sourceforge.net/p/courier/mailman/message/36417878/). *courier-announce* (Mailing list). Archived (https://web.archive.org/web/20210817181032/https://sourceforge.net/p/courier/mailman/message/36417878/) from the original on August 17, 2021. Retrieved September 17, 2020.

37. "Halon MTA changelog" (https://github.com/halon/changelog). *GitHub*. November 9, 2021. Archived (https://web.archive.org/web/20200918190844/https://github.com/halon/changelog) from the original on September 18, 2020. Retrieved September 17, 2020. "v4.0: New SMTPUTF8 support" Updated for new versions
38. "MS-OXSMTP: Simple Mail Transfer Protocol (SMTP) Extensions" (https://interoperability.blob.core.windows.net/files/MS-OXSMTP/%5bMS-OXSMTP%5d-180724.docx). July 24, 2018. Archived (https://web.archive.org/web/20210816051106/https://interoperability.blob.core.windows.net/files/MS-OXSMTP/%5bMS-OXSMTP%5d-180724.docx) from the original on August 16, 2021. Retrieved September 17, 2020.
39. "EAI Readiness in TLDs" (https://uasg.tech/wp-content/uploads/2019/02/UASG021D-EN-EAI-Readiness-in-TLDs.pdf) (PDF). February 12, 2019. Archived (https://web.archive.org/web/20210124205507/https://uasg.tech/wp-content/uploads/2019/02/UASG021D-EN-EAI-Readiness-in-TLDs.pdf) (PDF) from the original on January 24, 2021. Retrieved September 17, 2020.
40. "Communications Messaging Server Release Notes" (https://docs.oracle.com/communications/E72263_01/doc.802/e72267/msvrn.htm#BAJGIBHB). *oracle.com*. October 2017. Archived (https://web.archive.org/web/20201124100920/https://docs.oracle.com/communications/E72263_01/doc.802/e72267/msvrn.htm#BAJGIBHB) from the original on November 24, 2020. Retrieved September 17, 2020.
41. "Introducing MTA Strict Transport Security (MTA-STS) | Hardenize Blog" (https://www.hardenize.com/blog/mta-sts). *www.hardenize.com*. Archived (https://web.archive.org/web/20190425063147/https://www.hardenize.com/blog/mta-sts) from the original on April 25, 2019. Retrieved April 25, 2019.
42. "STARTTLS Everywhere" (https://starttls-everywhere.org/). EFF. Archived (https://web.archive.org/web/20190809085808/https://www.starttls-everywhere.org/) from the original on August 9, 2019. Retrieved December 4, 2021.
43. Cimpanu, Catalin. "Gmail becomes first major email provider to support MTA-STS and TLS Reporting" (https://www.zdnet.com/article/gmail-becomes-first-major-email-provider-to-support-mta-sts-and-tls-reporting/). *ZDNet*. Archived (https://web.archive.org/web/20190429022852/https://www.zdnet.com/article/gmail-becomes-first-major-email-provider-to-support-mta-sts-and-tls-reporting/) from the original on April 29, 2019. Retrieved April 25, 2019.
44. "Message Non Compliant with RFC 5322" (https://support.google.com/mail/?p=RfcMessageNonCompliant). Archived (https://web.archive.org/web/20230117175715/https://support.google.com/mail/community?hl=en&gpf=) from the original on January 17, 2023. Retrieved January 20, 2021.
45. "Message could not be delivered. Please ensure the message is RFC 5322 compliant" (https://answers.microsoft.com/en-us/outlook_com/forum/oemail-ocompose/message-could-not-be-delivered-please-ensure-the/87a52762-7d08-467e-85a2-120721c2dd8e?auth=1). Archived (https://web.archive.org/web/20210128115816/https://answers.microsoft.com/en-us/outlook_com/forum/oemail-ocompose/message-could-not-be-delivered-please-ensure-the/87a52762-7d08-467e-85a2-120721c2dd8e?auth=1) from the original on January 28, 2021. Retrieved January 20, 2021.
46. "Why are the emails sent to Microsoft Account rejected for policy reasons?" (https://answers.microsoft.com/en-us/outlook_com/forum/oemail-osend/why-are-the-emails-sent-to-microsoft-account/b64e3e4a-0d93-40c8-8e28-4be849012f9c). Archived (https://web.archive.org/web/20210214021030/https://answers.microsoft.com/en-us/outlook_com/forum/oemail-osend/why-are-the-emails-sent-to-microsoft-account/b64e3e4a-0d93-40c8-8e28-4be849012f9c) from the original on February 14, 2021. Retrieved January 20, 2021.

References

- Hughes, L (1998). *Internet E-mail: Protocols, Standards and Implementation*. Artech House Publishers. ISBN 978-0-89006-939-4.
- Hunt, C (2003). *sendmail Cookbook*. O'Reilly Media. ISBN 978-0-596-00471-2.

- Johnson, K (2000). *Internet Email Protocols: A Developer's Guide*. Addison-Wesley Professional. ISBN 978-0-201-43288-6.
- Loshin, P (1999). *Essential Email Standards: RFCs and Protocols Made Practical*. John Wiley & Sons. ISBN 978-0-471-34597-8.
- Rhoton, J (1999). *Programmer's Guide to Internet Mail: SMTP, POP, IMAP, and LDAP*. Elsevier. ISBN 978-1-55558-212-8.
- Wood, D (1999). *Programming Internet Mail* (https://archive.org/details/livesofcaptivere00petz). O'Reilly. ISBN 978-1-56592-479-6.

External links

- RFC 1869 (https://datatracker.ietf.org/doc/html/rfc1869) SMTP Service Extensions
- RFC 5321 (https://datatracker.ietf.org/doc/html/rfc5321) Simple Mail Transfer Protocol
- RFC 4954 (https://datatracker.ietf.org/doc/html/rfc4954) SMTP Service Extension for Authentication (obsoletes RFC 2554 (https://datatracker.ietf.org/doc/html/rfc2554))
- RFC 3848 (https://datatracker.ietf.org/doc/html/rfc3848) SMTP and LMTP Transmission Types Registration (with ESMTPA)
- RFC 6409 (https://datatracker.ietf.org/doc/html/rfc6409) Message Submission for Mail (obsoletes RFC 4409 (https://datatracker.ietf.org/doc/html/rfc4409), which obsoletes RFC 2476 (https://datatracker.ietf.org/doc/html/rfc2476))
