TP2 Investigación SMTP

Integrantes:

- Valentina CARERA
- Kiara Micaela KOO

Materia: Administración de Sistemas y Redes

Profesor: Ignacio García

Fecha de entrega: 02-05-2025

Introducción

¿Qué es SMTP?

SMTP (Simple Mail Transfer Protocol) es un protocolo de red utilizado para la transmisión de correos electrónicos entre servidores de correo electrónico. Este protocolo fue diseñado para facilitar la entrega confiable y eficiente de mensajes de correo electrónico a través de redes TCP/IP. A lo largo del tiempo, SMTP ha evolucionado mediante la incorporación de extensiones y mejoras, consolidándose en el RFC 5321, publicado en 2008, que es el estándar actual en uso.

Historia

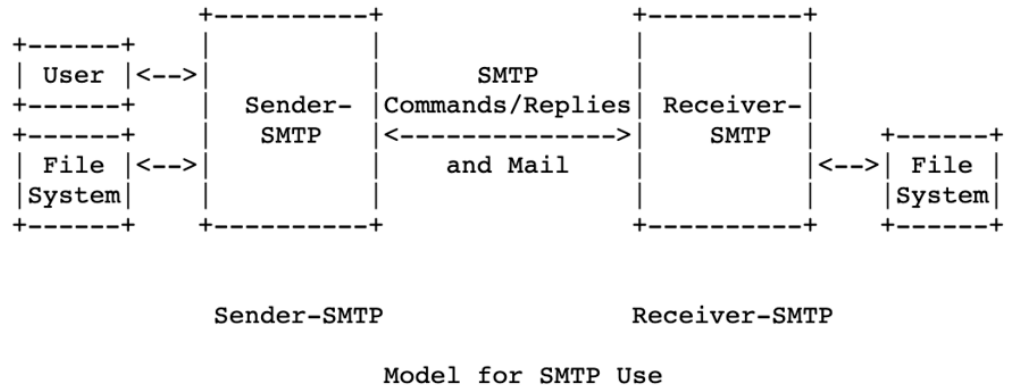
El protocolo **SMTP**(Simple Mail Transfer Protocol) fue introducido en 1982 por Jonathan B. Postel, una figura de renombre en el ámbito de la informática e ingeniería en redes. Este fue quien publicó **RFC 821**, que definía el estándar original de SMTP con el objetivo de simplificar el proceso de intercambio de emails, proporcionando un método seguro y confiable de transmisión de mensajes de correo electrónico entre diferentes sistemas conectados a través de redes.
SMTP surgió en un contexto donde las redes informáticas comenzaban a expandirse y se necesitaba un método confiable para transmitir mensajes electrónicos entre diferentes sistemas. Su diseño simple y efectivo facilitó su adopción en entornos académicos, científicos y gubernamentales.
A medida que el internet iba creciendo en la década de 1990, el servicio de mail se volvió la forma de comunicación principal, el uso de SMTP se volvió esencial.
Sin embargo, la versión original del protocolo presentaba importantes limitaciones de seguridad, ya que no incluía mecanismos de autenticación ni de cifrado. Esto lo hizo vulnerable al spam y phishing. Para solucionar este problema, se desarrollaron extensiones y mejoras a su versión original:

- En 1995, se introdujo **ESMTP**(Extended SMTP) con RFC 1869, incorporando más funcionalidades al protocolo, entre ellas encriptación y transferencia de archivos más grandes.
- También en 1995, se introdujo **TLS**(Transport Layer Security) para asegurar las conexiones de SMTP, evitando la intercepción de emails durante su transferencia.

A lo largo de los años, el protocolo fue evolucionando para adaptarse a nuevas necesidades. En 2001, el **RFC 821** fue reemplazado por el **RFC 2821**, y en 2008 por el **RFC 5321**, que es el estándar actual.

Modelo

SMTP utiliza un modelo cliente-servidor. El cliente SMTP inicia la conexión TCP al servidor SMTP y luego intercambian comandos y respuestas en formato texto.



Model for SMTP Use

El usuario ingresa el mensaje en el sender (cliente), el cual establece una conexión TCP con el reciever (servidor) del destino. A partir de ahí, SMTP pasa líneas de texto de un lado a otro. Todo flujo SMTP se basa en líneas delimitadas por CRLF. El protocolo está limitado originalmente a caracteres ASCII y cada línea de texto puede tener como máximo 1000 caracteres, aunque el uso de extensiones ESMTP puede aumentar estos límites. En la actualidad, los clientes de correo (MUA) suelen conectarse a un agente de envío (MSA) sobre el puerto 587 con TLS y autenticación, en lugar del puerto 25. El MSA entrega entonces el correo al Mail Transfer Agent (MTA) interno que lo encaminara, usando DNS (registro MX) para localizar el servidor de destino.

Cómo Funciona

El protocolo SMTP sigue un modelo cliente-servidor. A grandes rasgos, una sesión SMTP típica sigue estos pasos numerados:

- **Inicio de conexión:** El cliente abre una conexión TCP al puerto 25 (o 587) del servidor SMTP de destino y espera el saludo inicial. El servidor SMTP puede devolver un mensaje “220 Service ready” o “421 Service not available”.
- **Identificación del cliente (HELO/EHLO):** Una vez recibido el saludo, el cliente envía EHLO (o HELO si el servidor no reconoce EHLO) en donde se identifica y abre la sesión. El servidor responde con un código “250 OK”. Este intercambio inicial confirma que ambos extremos están en estado inicial.
- **Ingreso de remitente (MAIL FROM):** El cliente envía MAIL FROM con la dirección del remitente para indicar el remitente del correo. El servidor debe responder 250 OK si acepta el remitente.
- **Ingreso de destinatarios (RCPT TO):** Luego, el cliente envía uno o varios comandos RCPT TO (uno por cada destinatario) para añadir destinatarios al correo. El servidor responde 250 OK si puede aceptar ese destinatario, o un código de error en caso contrario.

- **Inicio de datos (DATA):** Una vez aceptados todos los destinatarios, el cliente emite el comando DATA. El servidor entonces debe responder con un código 354. A partir de ese momento, el cliente transmite el contenido del mensaje en líneas de texto.
- **Fin de datos:** El cliente finaliza el envío de datos colocando en una línea aislada solo un punto (.) que indica el fin del mensaje. Al recibir esta señal, el servidor procesa el correo completo. Si todo está bien y se envía correo correctamente, el servidor envía un 250 OK de confirmación o un código de error 5xx en caso de algún error.
- **Finalización de sesión (QUIT):** Después de terminar, el cliente debe envíar el comando QUIT. El servidor responde con 221 y cierra la conexión.

Con este flujo, SMTP asegura la interacción básica de envío de correo. El cliente debe respetar la secuencia iniciando con un EHLO/HELO y finalizando con un QUIT. Cualquier comando enviado fuera de orden provocará errores de tipo 503 Bad sequence of commands.

Estructura del mensaje

Para que un correo electrónico pueda ser enviado correctamente a través de SMTP, debe seguir un formato bien definido que separa el mensaje en dos secciones: las cabeceras y el cuerpo. Esta estructura no es opcional, es una regla que todo cliente de correo debe respetar al comunicarse con un servidor SMTP.
Después de que el cliente envía el comando DATA, el servidor queda a la espera del contenido del mensaje. El cliente debe enviar primero la sección de cabeceras, luego una línea en blanco, y por último el cuerpo del mensaje.
Las cabeceras se componen de líneas con el formato “Nombre-del-campo: valor”. Algunos de los campos más habituales son From: (remitente), To: (destinatarios), Cc: (Carbon Copy), y Subject: (asunto). Un ejemplo típico sería:

From: Kiara Koo <kkoo@alumno.huergo.edu.ar>
To: Valentina Carera <vcarera@alumno.huergo.edu.ar>
Subject: Informe SMTP

Cada uno de estos campos debe seguir un formato específico, por ejemplo, en la forma de las direcciones de correo o el formato de fecha. Además, ninguna línea puede superar los 1000 caracteres.
Tras la línea en blanco que separa las cabeceras del resto, se encuentra el cuerpo del mensaje, que es simplemente texto plano que SMTP transmite sin interpretar.
Para finalizar el envío, el cliente debe enviar una línea que contenga únicamente un punto (.). Al recibirla, el servidor considera completo el mensaje, lo almacena y responde con una confirmación, típicamente un 250 OK.

Comandos

SMTP cuenta con distintos comandos que son usados para que el cliente se pueda comunicar con el servidor. Estos comandos textuales forman el núcleo de SMTP/ESMTP.

- **HELO/EHLO (HELLO/Extended HELLO):** Este es el comando de saludo inicial donde el cliente se identifica con el servidor mediante su dominio o dirección literal. EHLO solo es aceptado por servidores que tienen soporte de extensiones ESMTP, en caso de no soportarlo el servidor responde con un error y el cliente debe de usar HELO. Luego de que se mande el comando HELO/EHLO, el servidor responderá con 250 OK indicando que el cliente y el servidor están listos para la transacción.
    - Sintaxis
        - EHLO <dominio>
        - HELO <dominio>

- **MAIL (MAIL FROM):** Este comando especifica la dirección del remitente (reverse-path) y el inicio de una nueva transacción. Una vez enviada, el servidor elimina cualquier transacción pendiente y borra los buffers previos. Si el remitente es válido, el sistema devuelve 250 OK, de lo contrario se enviará un error. Se pueden agregar parámetros opcionales asociados con algunas extensiones, en caso de que no se reconozcan, se ignorarán o rechazarán con un código de error.
    - Sintaxis
        - MAIL FROM:<reverse-path> [SP <parameters>]

- **RECIPIENT (RCPT TO):** Con este comando, se agrega el destinatario del correo. Este comando se puede repetir varias veces por cada destinatario que se quiera agregar. El argumento de este comando es la dirección del destinatario (foward-path). Luego el servidor se encarga de validar cada destinatario devolviendo 250 OK en caso de que sea válido, y un código de error en caso de que no. En este comando también se pueden agregar parámetros opcionales asociados con algunas extensiones.
    - Sintaxis
        - RCPT TO:<forward-path> [SP <parameters>]

- **DATA:** Este comando inicia la transferencia del contenido del correo (cabeceras y cuerpo). Tras enviarlo, el servidor responde con 354 indicando que el cliente debe enviar los datos del mensaje en líneas de texto. El final de los datos se indica con una línea que contiene solo un punto (.). Al recibirlo, el servidor procesa y entrega el mensaje. Si todo es correcto responde con 250 OK, en caso contrario envía un código de error. No puede haber aceptaciones parciales: o se acepta todo el mensaje o se rechaza.
    - Sintaxis
        - DATA

- **RESET (RSET):** Este comando cancela la transacción actual. El servidor debe eliminar toda la información del remitente/destinatario y datos almacenados, dejando los buffers vacíos y poniendo la sesión en su estado inicial. El comando puede ser usado en cualquier momento de la transacción sin cerrar la conexión y el servidor responderá con 250 OK. Si el comando es usado después del HELO/EHLO, antes de empezar o antes de QUIT, este actuará como el comando NOOP (que será explicado más adelante). Este comando nunca cierra la conexión.
    - Sintaxis
        - RSET

- **VERIFY (VRFY):** Este comando le pide al servidor que verifique que cierto username o mailbox son válidos. el servidor puede devolver información de la cuenta, si es que esta existe, o un error. Este comando no afecta a ninguno de los buffer de la transacción. Por cuestiones de seguridad muchos servidores deshabilitan VRFY o responden siempre con errores genéricos.
    - Sintaxis
        - VRFY <usuario>

- **EXPAND (EXPN):** Este comando pide al servidor que expanda un alias o lista de correo, devolviendo los miembros del listado en líneas separadas. No modifica los buffers de la transacción. Como VRFY, suele estar deshabilitado para evitar revelar listas de correo.
    - Sintaxis
        - EXPN <lista>

- **NOOP:** Este comando no realiza ninguna acción. Su único propósito es verificar que el servidor esté activo. Una vez enviado, el servidor responderá con 250 OK. Se puede usar en cualquier momento.
    - Sintaxis
        - NOOP

- **HELP:** Este comando solicita información de ayuda. Sin argumentos, el servidor envía ayuda genérica. Con argumento, el servidor envía información específica de un comando. No afecta a buffers y puede usarse en cualquier momento. El servidor debe responder con 250 OK y la información.
    - Sintaxis
        - HELP [<comando>]

- **QUIT:** Este comando indica la terminación de la sesión SMTP. El servidor responde 221 OK y cierra la conexión. Las transacciones no finalizadas se abortan. El cliente debe enviar QUIT antes de cerrar la conexión. Puede usarse en cualquier momento.
    - Sintaxis
        - QUIT

Respuestas del servidor

Las respuestas del servidor SMTP siguen el formato <código> <texto>. Los códigos más relevantes son:

- **220:** saludo inicial (“SMTP service ready”). Se envía al abrir conexión TCP.
- **250:** acción completada correctamente. Se usa tras comandos como MAIL, RCPT, EHLO (indica éxito y lista de extensiones), RSET, NOOP.
- **354:** inicio de entrada de datos. Se envía como respuesta al comando DATA, indicando que el servidor está listo para recibir el cuerpo del mensaje, y se termina con (.).
- **221:** cierre de la sesión. Se envía en respuesta a QUIT e indica que el servidor cerrará la conexión.
- **421:** servicio no disponible (temporal). El servidor cierra la conexión o informa de error.
- **450/451/452:** errores temporales (por ejemplo buzón ocupado(ocupado o bloqueado), error de procesamiento, falta de espacio).
- **500/501/502/504:** errores de sintaxis o comando no implementado o no reconocido. Se usan si el comando no es válido en ese momento o no se soporta.
- **550/551/552/553/554:** errores permanentes (p.ej. buzón no encontrado, espacio insuficiente, mailbox no disponible(no encontrado o rechazado), error grave). El comando ha fallado definitivamente.
- **251/252:** errores relacionados al usuario(error de verificación de usuario, aunque lo envía igual).
- **455/555:** error de parámetros (parámetros no reconocidos o no implementados).

Comparación con imap pop3

POP3

POP3 (Post Office Protocol versión 3) es un protocolo de comunicación que permite la descarga y eliminación de mensajes. Al tener una cuenta como POP3, se descargan los mensajes de la bandeja de entrada del servidor de correo, se descargan en el ordenador local y se eliminan del servidor. Este protocolo sirve también para acceder a tu bandeja de entrada sin necesidad de acceso conexión de internet. Existe una versión moderna de este protocolo que te permite mantener una copia de los mensajes en el servidor si lo especificas en la configuración.

IMAP

IMAP(Internet Message Access Protocol) es un protocolo de acceso a mensajes electrónicos almacenados en un servidor. Al tener una cuenta como IMAP, los mensajes se almacenan físicamente en un servidor remoto, lo cual consume espacio en tu hosting, pero a su vez asegura una copia de seguridad de los mismos. Por otro lado, se permite el acceso al correo electrónico desde cualquier dispositivo que tenga

conexión a internet. Todos los cambios realizados en el buzón se sincronizarán en varios dispositivos y los mensajes sólo se eliminarán del servidor si el usuario elimina el correo electrónico.

Relación con SMTP

Por un lado, se encuentran los protocolos entrantes, IMAP y POP3. Estos protocolos de acceso, tienen como objetivo la recuperación de correos electrónicos.
Por otro lado, se encuentra el protocolo saliente SMTP. Este es quien maneja el envío de email desde una cuenta de email.
Gracias al protocolo saliente, un usuario de email, desde un gestor de correo, puede enviar emails, y puede recibirlos gracias a los protocolos entrantes.

ESMTP

El Protocolo Simple de Transferencia de Correo Extendido (ESMTP) fue introducido en 1995 mediante el RFC 1869 con el objetivo de superar las limitaciones del SMTP original. ESMTP permite la incorporación de extensiones que añaden funcionalidades como autenticación, cifrado y soporte para mensajes de mayor tamaño. Una de las principales diferencias es el uso del comando EHLO en lugar de HELO, lo que permite al cliente identificar las capacidades del servidor y utilizar las extensiones disponibles. Si el servidor no soporta EHLO, se puede revertir al uso de HELO, manteniendo la compatibilidad con versiones anteriores.

Bibliography

- https://datatracker.ietf.org/doc/html/rfc5321
- https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo
- https://es.mailpro.com/definicion/smtp
- https://mysmtp.com/blog/2024/09/09/the-history-of-the-smtp-protocol/
- https://support.microsoft.com/es-es/office/-cu%C3%A1l-es-la-diferencia-entre-pop-e-imap-85c0e47f-931d-4035-b409-af3318b194a8