International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Secure Email Software using e-SMTP

Harshwardhan .P.Shitole [1], Prof.S.Y.Divekar [2]

[1] Student,Dept. of Computer Engineering, AISSMS Polytechnic, Maharashtra, India
[2]Professor, Dept. of Computer Engineering, AISSMS Polytechnic, Maharashtra, India

---

Abstract - Simple Mail Transfer Protocol is being used since 1980 to send and receive Emails.
To send and receive plaintext messages is the main purpose of this protocol . So, the initial design of this protocol did not address any security considerations. However, nowadays, we are using this protocol to send confidential messages. This brings in many security issues to message transfer.
The main weakness in this protocol is that it does not address any security considerations (such as Confidentiality, Integrity, Authentication, Authorization). We have proposed a new Email transfer model in order to bring in those security considerations into message transfer. This model has a central server to authenticate and authorize Email traffic between Email users. Moreover, this model provides confidentiality and integrity to email messages with the help of encryption and hash.

Key Words: SMTP(Simple mail Transfer Protocol) ,E-SMTP(Extended SMTP),IMAP (Internet Message Access Protocol) ,POP3(Post Office Protocol 3),Encryption, Hash value.

1.INTRODUCTION

The default protocol that is used to send and receive E-mail is Simple Mail Transfer Protocol (SMTP) . Electronic Mail (E-Mail) in the Internet. It was firstly introduced in Request for Comments (RFC) 780 by Sluizer and Postel in 1980. To improve this protocol many RFC versions have been released. Nevertheless, the current accepted version is RFC 5321 which was written by Klensin in October 2008.

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

1. **MAIL command:**
It is used to establish the return address, also called return-path, reverse-path, bounce address, envelope sender.

2. **RCPT command**:
It is used 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.

3. **DATA command**:
It is used to signal the beginning of the message text. 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.

1.1 Limitations of SMTP Protocol:

The most significant security weakness in the protocol is to send messages in plaintext. This allows a hacker to read messages by listening to the traffic. Figure 1 shows a network traffic analyzer, such as Wireshark , is able to capture an Email message which is sent as a plaintext.

One other weakness is that, until transfer, messages are kept as a plaintext files in the Email server. Those who can log in to the server can access these files.

To address those security issues and to improve the functionality of the protocol, extensions have been proposed by various studies. Extended SMTP (ESMTP) is a well-known example for extension which was introduced by RFC 1869 in 1995.

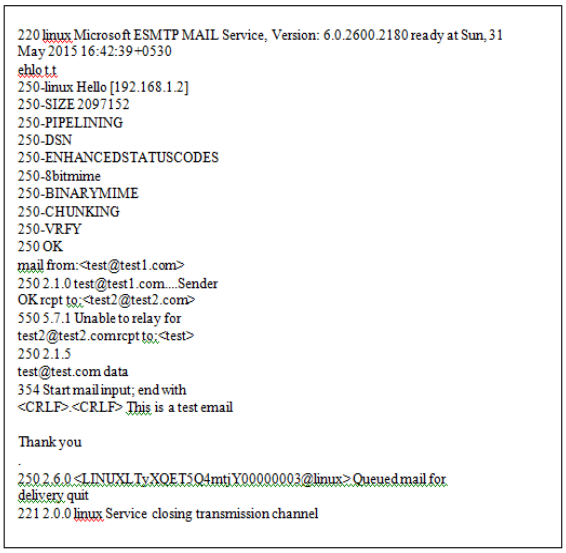

FIGURE 1: PLAIN TEXT TRAFFIC SENT ON NETWORK

By default, SMTP does not authenticate the sender before it accepts Emails. An attacker can use this weakness to spoof sender's Email address. Furthermore, anyone can send any Email to any person without prior approval from the receiver. Various authentication schemes have been introduced to address those weaknesses and to improve the authentication in messages transfer.

2. ESMTP Protocol:

Extended SMTP (ESMTP), sometimes referred to as Enhanced SMTP, is a definition of protocol extensions to the Simple Mail Transfer Protocol standard. The extension format was defined in November 1995 in IETF publication RFC 1869 which established a general structure for all existing and future extensions.

ESMTP defines consistent and manageable means by which ESMTP clients and servers can be identified and servers can indicate supported extensions.

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.

2.1 Features of ESMTP 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 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.

8BITMIME — 8 bit data transmission, RFC 6152
ATRN — Authenticated TURN for On-Demand Mail Relay, RFC 2645
AUTH — Authenticated SMTP, RFC 4954
CHUNKING — Chunking, RFC 3030
DSN — Delivery status notification, RFC 3461 (See Variable envelope return path)
ETRN — Extended version of remote message queue starting command TURN, RFC 1985
HELP — Supply helpful information, RFC 821
PIPELINING — Command pipelining , RFC 2920
SIZE — Message size declaration, RFC 1870
STARTTLS — Transport Layer Security, RFC 3207 (2002)
SMTPUTF8 — Allow UTF-8 encoding in mailbox names and header fields, RFC 6531
UTF8SMTP — Allow UTF-8 encoding in mailbox names and header fields, RFC 5336 (deprecated)

The most important among these are:

8BITMIME:

In March 2011, 8BITMIME was published as RFC 6152 corresponding to the then new STD 71.The 8BITMIME extension was standardized in 1994.The transparent exchange of e-mail messages(containing octets outside the seven-bit ASCII character set)is facilitated. For transmission of non-ASCII text, each of these workarounds inflates the required amount of data. The sending of 8-bit characters is allowed by some non-ESMTP servers, however it is risky to send such data whose 8-bit capabilities are unknown to a server. Mail user agents employed several techniques to cope with the seven-bit limitation (prior to the availability of 8BITMIME implementations) such as binary-to-text encodings (including ones provided by MIME) and UTF-7.

SMTP-AUTH:

RFC 4954 has defined the SMTP-AUTH extension. The SMTP-AUTH extension provides an access control mechanism. In SMTP-AUTH extension, when relaying mails, one mail server is allowed to indicate to another that the sender has been authenticated. During the process of sending mails, the client effectively logs into the mail server, with the use of an authentication step available in it.

Servers supporting SMTP-AUTH can be configured to require clients to use this extension, ensuring the true identity of the sender.

In general this requires the recipient server to trust the sending server. This aspect of SMTP-AUTH is rarely used on the Internet.

While denying relay service to unauthorized users (such as spammers) SMTP-AUTH can be used to allow legitimate users to relay mail.
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 AUTH user is authorized for.
It does not necessarily guarantee the authenticity of either the SMTP envelope sender or the RFC 2822 "From:" header.

SMTPUTF8:

This extension was added in 2012 by RFC 6531.The UTF-8 encoding is allowed in SMTPUTF8 extension in mailbox names and header fields. This provides the capability for sending email to internationalized addresses such as Pelé@example.com, δοκιμή@παράδειγμα.δοκιμή, and 测试@测试.测试 . But this protocol is not yet widely supported.

2.2 Retrieving E-mails using E-SMTP:

All modern e-mail clients and servers support the two most prevalent standard protocols for email retrieval: IMAP, which along with the earlier POP3 (Post Office Protocol).Incoming e-mail messages are sent to an e-mail server that stores messages in the recipient's e-mail box. The user retrieves the messages with an e-mail client that uses one of a number of e-mail retrieval protocols. While some clients and servers preferentially use vendor-specific, proprietary protocols, for retrieving e-mail.

2.2.1 IMAP Protocol:

RFC 3501 has defined IMAP. For permitting complete management of an email box by multiple email clients, the IMAP Protocol was designed. Therefore until the user explicitly deletes messages, these messages are left by clients on the server as it is. The IMAP (Internet Message Access Protocol) is an Internet standard protocol used to retrieve email messages over a TCP/IP connection from a mail server. An IMAP server listens on port number 143(for normal applications). The port number 993 is assigned to IMAP over SSL (IMAPS).
The Internet Message Access Protocol is an Application Layer Internet protocol .It allows an e-mail client to access e-mail on a remote mail server. The current version is defined by RFC 3501. Many webmail service providers such as Gmail, Outlook.com and Yahoo! Mail also provide support for either IMAP or POP3.
Access is offered by IMAP to the mail storage. Clients may store local copies of the messages, which are considered to be a temporary cache.
E-mail clients using IMAP generally leave messages on the server until the user explicitly deletes them. This and other characteristics of IMAP operation allow multiple clients to manage the same mailbox. E-mail clients mostly support IMAP in addition to POP (Post Office Protocol) to fetch email messages.

2.2.2 POP3 Protocol:

POP version 3 (POP3) is the version most commonly used all over the world.A POP3 server listens on well-known port number 110 (for normal applications) for service requests. Using the TCP port number 995, after protocol initiation, encrypted communication for POP3 is either requested using the STLS command(only if supported by POP3S) which connects to the server using TLS (Transport Layer Security) or SSL(Secure Sockets Layer).
POP3 supports download-and-delete from a remote mailbox ("maildrop" in POP3 terminology). Although most POP3 clients have an option to leave mail on server after download, they generally connect, retrieve all messages, store them on the client system, and delete them from the server (the IMAP (Internet Message Access Protocol) normally leaves all messages on the server).
When a POP3 session opens the maildrop, all the available messages to the client are fixed, and are identified by message-number local to that session or optionally, by a unique identifier assigned to the message by the POP server. This unique identifier is permanent and unique to the maildrop and allows a client to access the same message in different POP sessions.The message-number of the mail retrieved is marked for deletion. The mail marked for deletion is removed from the maildrop, when the client exits the session.

3. LITERATURE REVIEW

Confidentially, integrity, authentication and authorization are major security goals in SMTP. Following section explains some studies that have been done to improve security in those areas.

SSL based connection between sender and receiver
Author: D. Mooloo and T. P. Fowdur .

This proposed model used to send a confirmation of the Email. In that, the sender has to send a confirmation of the Email sent to the receiver via this SSL session. Emails will be marked as a SPAM, if the confirmation has not been received by the recipient. One of the limitations in this model is that, it is only applicable within a given IP subnet. To overcome this limitation they have proposed another solution . On that approach, to make the application globally available, they combined SSL with HTTPS protocol.

Identity based Email sender authentication method
Author:Hameed, Kloht, and Fu.

This Model was used to verify Email sender before accepting Emails. In their model, a trusted authority is used to generate required keys and parameters. The verification

process has to complete before the MAIL FROM: command takes place. That process gives a benefit to the recipient to reject the connection before the contents received. Li and Kim proposed a hierarchical anti-Spam framework which includes text classification, image processing and Optical Character Recognition.

Password Based Email Protocol
Author: Zhang and H. Chen.

In this approach, they use Signcryption to prevent the sender server's forgery attack.Non repudiation of an Email can be achieved by digitally signing the Email. However, applying both digital signature and encryption to the Email message can degrade the performance of the Email system. Because, the Email first has to sign by using a signing algorithm and then has to encrypt by using an encryption algorithm. Similarly, at the receiving end, the Email has to first decrypt by using the decryption algorithm and then has to validate the signature by using the signing algorithm.

Signcryption
Author:Yuliang Zheng.

Introduced a new scheme named as Signcryption to overcome the performance issues. In this scheme, message signing and encryption are done in a logically single step. Later, this scheme has been improved by various studies.
Further, to prevent spoofing in Email systems, security measures such as Sender Policy Framework (SPF) , Sender ID Framework and Domain Keys Identified Mail (DKIM) are introduced. However, it is still possible to spoof Emails as those techniques are not strong enough to stand against the new spoofing techniques.
There are several encryption methods that have been introduced to preserve confidentiality and integrity of Email messages.

Encryption scheme with Data compression
Author: Jain and Gosavi .

Jain and Gosavi introduced an encryption scheme with data compression. In this scheme, they first compress an Email message according to a given code book. After that, the Email will encrypt with a shared secret key which is based on receiver's ID. Further, the recipient's ID can be any publicly available information about the recipient, such as recipient Email address.

Identity based encryption
Author: Chen, et al.

Chen, et al. also proposed an identity based encryption scheme to encrypt an Email messages. On that scheme, they used a proxy server to encrypt and decrypt Email messages. This proxy server uses receiver's ID to encrypt an Email message before it sends the Email. At the receiving end, that proxy server uses receiver's private key to decrypt the Email. By having such method of encryption, they eliminated encryption settings from Email client applications. The reason for this change is that, as the authors say, the end users are not aware of encryption settings in Email systems also they are reluctant to do those configurations. Bai, et al. too support this statement. They said, the users believe that they have nothing to hide so they do not need encryption.

4. PROPOSED MODEL

Our proposed model addresses the above mentioned security considerations. This model comprises of Gmail SMTP server to authenticated and authorized users. Further, this provides confidentiality and integrity to messages by having an encryption mechanism.
Below Figure3.1 illustrates the architecture of proposed secure Email transfer model.

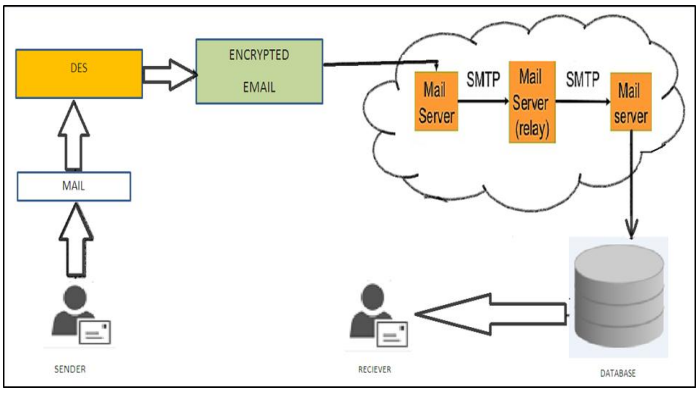

Figure 2: Proposed Architecture

All traffic between Gmail servers and Client should be encrypted by using the encryption algorithm . This mitigates the man in the middle attack on key transfer process.

E-SMTP majorly focuses on Confidentiality, Integrity, Authentication and Authorization.

4.1 General sending and receiving mail procedure:

1) Client1 writes email.
2) Client1 sends email(Click send), the email is delivered to sender's G-mail SMTP server .
3) Sender's G-mail SMTP server looks up the domain ,and finds out that Receiver's SMTP server is the correct server.
4) Sender's SMTP server sends the email to Receiver's SMTP server.
5) SMTP server 2 receives the email, and put it up in INBOX folder (windows SMTP) .
6) With an IMAP server, the IMAP server checks the INBOX folder for emails, then move it to the correct mailbox.
7) Receiver connects to the IMAP server, to download email, and the email gets loaded onto the Receiver's PC.
8) Receiver reads the email sent by the Sender.

4.2 The Authentication and Authorization Process:

The sending Email server should authenticate itself at the receiving Email server before it starts sending Emails. This process ensures that the sending Email server is a valid authentic server.
Hence this is achieved by logging on to the Gmail SMTP server using your Gmail username and password.

4.3 Encryption Methodology and Key Management:

The DES(Data Encryption Standard) Encryption Algorithm is used in this system.
When the E-mail is to be sent to the Recipient,it is first gets Encrypted and the DES Symmetric Key and Random number is generated.
At the time of sending E-mail by the server,it gets again encrypted by using MD5 Encryption Algorithm.
The message(message in MD5) gets Decrypted by the Reciever's SMTP Server.
This Random Key is being sent to the Receiver on his/her Account is used to Decrypt message(messasge in DES).
At the time of decrypting the message, the Receiver passes this message to the decryptor and the message is being decrypted.

4.4 Checking Integrity of the Message:

The System provides hash tool to check the Integrity of message.Hash is one of the best way to check integrity of message.
The Receiver calculates the hash of the message using the Phoenix Hash Tool available into the built system.

5.CONCLUSION AND FUTURE WORKS :

SMTP protocol is still having security weaknesses as the studies that have been done to improve this protocol which only addresses security in Application level. The model that we proposed improves security in core SMTP. In this model we introduce authentication, authorization, confidentiality and integrity into Email system. By having these security considerations, we can improve the security in message transfer.

As future works, we need to design a proper revocation policy to maintain a valid list of Email servers. By having such policy we can maintain a good valid PAD. Further, we are planning to embed public key of CCS with Email server installation software.

REFERENCES

[1] S. Sluizer and J. Postel, "Mail Transfer Protocol,"*DOI 1017487RFC0780*, 1980.

[2] Y. Zheng, "Signcryption and its applications in efficient public key solutions," in *Information Security*, vol. 1396, E. Okamoto, G. Davida, and
Mambo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 291–312.

[3] "Wireshark · Go Deep." [Online]. Available: https://www.wireshark.org/. [Accessed: 23-Mar-2015].

[4] T. P. Fowdur and L. Veerasoo, "An email application with active spoof monitoring and control," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 2016, pp. 1–6.

[5] H. Tanta-ngai, T. Abou-Assaleh, S. Jiampojamarn, and N. Cercone, "Secure Mail Transfer Protocol (SecMTP)."

[6] Y. K. Jain and P. B. Gosavi, "Email Security Using Encrption and Compression," in *2008 International Conference on Computational Intelligence for Modelling Control Automation*, 2008, pp. 136–139.

[7] W. Bai, D. Kim, N. Moses, Y. Qian, P. G. Kelly, and M. Mazurek, "#x0022;Most of us trust our email provider #x0022;: Balancing security and usability in encrypted email," *IEEE Internet Comput.*, vol. PP, no. 99, pp. 1–1, 2017.

[8] J. Zhang and H. Chen, "An improved password-based authenticated email protocol," in *2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, 2010, pp. 545–549.

[9] D. Mooloo and T. P. Fowdur, "An SSL-based client-oriented anti-spoofing email application," in *AFRICON, 2013*, 2013, pp. 1–5.

[10] S. Hameed, T. Kloht, and X. Fu, "Identity based email sender authentication for spam mitigation," in *2013 Eighth International Conference on Digital Information Management (ICDIM)*, 2013, pp. 14–19.