INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

E-MAIL PROTOCOLS: VULNERABILITIES, ATTACKS AND DEFENSE MECHANISMS

KAMINI NALAVADE, B. B. MESHRAM

1. Research Scholar, Computer Department, VJTI Matunga, Mumbai, India.
2. Professor & Head, Computer Department, VJTI Matunga, Mumbai, India

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

E-mail System,

Vulnerabilities,

Intrusion,

Defense

Corresponding Author

Ms. Kamini Nalavade

Abstract

A protocol is about a standard method used at each end of a communication channel, in order to properly transmit information. In order to deal with your email you must use a mail client to access a mail server. The mail client and mail server can exchange information with each other using a variety of protocols. Electronic mail (e-mail) is one of the most popular network services nowadays. Most e-mail systems that send mail over the Internet use simple mail transfer protocol (SMTP) to send messages from one server to another. The messages can then be retrieved with an e-mail client using either post office protocol (POP) or Internet message access protocol (IMAP). SMTP is also generally used to send messages from a mail client to a mail server in "host based" (or Unix-based) mail systems, where a simple mailbox utility might be on the same system [or via Network File System (NFS) provided by Novell for access without POP or IMAP. In this paper we describe the vulnerability, attacks and defense mechanisms for important e-mail protocols.

Introduction

Basically, a protocol is about a standard method used at each end of a communication channel, in order to properly transmit information. In order to deal with your email you must use a mail client to access a mail server. The mail client and mail server can exchange information with each other using a variety of protocols.

An E-mail message consists of e-mail Body and e-mail Header. The Body is text which can also include multimedia elements in Hyper Text Markup Language (HTML) and attachments encoded in Multi-Purpose Internet Mail Extensions (MIME) [01]. The Header is a structured set of fields that include ‘From’, ‘To’, ‘Subject’, ‘Date’, ‘CC’, ‘BCC’, ‘Return-To’, etc. Headers are included in the message by the sender or by a component of the e-mail system. TCP/IP e-mail address consists of username and domain name separated by @ sign e.g. aliace@a.com. Ray Tomlinson [02] first initiated the use of @ sign to separate username from the domain name. An e-mail communication between a sender ‘Alice’ having e-mail address ‘alice@a.com’ and recipient ‘Bob’ having e-mail address ‘bob@b.com’ is shown in figure 1. ‘Alice’ composes an e-mail message on her computer called client for ‘Bob’ and sends it to her sending server ‘smtp.a.org’ using Simple Mail Transfer Protocol (SMTP) [03, 4]. Sending server performs a lookup for the mail exchange record of receiving server ‘b.org’ through DNS protocol on DNS server ‘dns.b.org’. The DNS server responds with the highest priority mail exchange server ‘mx.b.org’ for the domain ‘b.org’. Sending server establishes SMTP connection with the receiving server and delivers the e-mail message to the mailbox of ‘Bob’ on the receiving server. ‘Bob’ downloads the message from his mailbox on receiving server to local mailbox on his client computer using POP3 or IMAP protocols. Optionally, ‘Bob’ can also read the message stored in his server mailbox without downloading it to the local mailbox by using some Webmail program. This model of electronic communication involves a number of hardware and software components that communicate with each other using some protocols especially SMTP protocol. SMTP protocol has evolved as a complex system since its inception. Its commands have been augmented by inclusion of various extensions which may or may not be adopted by every SMTP client and server.

E-mail communication is insecure. E-mails can be read and modified as they are passed through the Internet as clear-text. Therefore, two basic needs have emerged: -Confidentiality The e-mail can only be read by the intended recipient. This is ensured using encryption.

-Authentication The e-mail has been written by particular person and has not been altered on its way over the Internet.

The remaining paper is organized as follows: Section 2 describes vulnerabilities and attacks about SMTP protocol. Section 3 describes PGP protocol, its flaws and attacks. Section 4 summarizes our study of E-mail protocols followed by conclusion.

I. SMTP Vulnerabilities, Attacks and Defense

The SMTP (Simple Mail Transfer Protocol) protocol is used by the Mail Transfer Agent (MTA) to deliver your eMail to the recipient's mail server. The SMTP protocol can only be used to send emails, not to receive them. Depending on your network / ISP settings, you may only be able to use the SMTP protocol under certain conditions. SMTP is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. It is an application layer protocol. Under SMTP, a client SMTP process opens a TCP connection to a server SMTP process on a remote host and attempts to send mail across the connection. The server SMTP listens for a TCP connection on a specific port (25), and the client SMTP process initiates a connection on that port (Cisco SMTP, 2005). When the TCP connection is successful, the two processes execute a simple request–response dialogue, defined by the SMTP protocol (see RFC 821 for details), in which the client process transmits the mail addresses of the originator and the recipient(s) for a message. When the server process accepts these mail addresses, the client process transmits the e-mail instant message. The message must contain a message header and message text (“body”) formatted in accordance with RFC 822. Mail that arrives via SMTP is forwarded to a remote server, or it is delivered to mailboxes on the local server. POP3 or IMAP allow users download mail that is stored on the local server. The header includes a number of key words and values that define the sending date, sender’s address, where replies should go, and some other information. The header is a list of lines, of the form (SMTP, 2005): field-name: field-value Fields begin in column 1: Lines beginning with white space characters (SPACE or TAB) are continuation lines, which are unfolded to create a single line for each field in the canonical representation. Strings enclosed in ASCII quotation marks indicate single tokens within which special characters such as the colon are not significant. Many important field values (such as those for the "To" and "From" fields) are "mailboxes."

Vulnerabilities in SMTP

1. Not only does SMTP not have any real security mechanism, the original model of SMTP communication is entirely designed around the idea of "cooperation" and "trust" between servers. Since most SMTP servers would be asked to handle a certain number of intermediate transfers, each server was required to accept mail from any originator to be delivered to any destination. The basic assumption in this model is that SMTP servers would all be "well-behaved", and not abuse the system by flooding intermediate servers with lots of mail to be delivered, or sending bogus messages to cause problems. This all changed as the Internet exploded in popularity in the 1990s. Con artists, hackers, and disreputable salespeople all discovered that e-mail could be used for "free" delivery of messages simply by submitting them to an SMTP server for delivery. The result was overloaded servers, primarily due to the sending of large quantities of unwanted e-mail, which Internet users commonly call spam. [12]

2. The processes of retrieving e-mail from servers and managing data communication through the Internet are vulnerable to various attacks. A review of vulnerabilities can be found in "Vulnerability Tutorials" (2005) released by the Saint Corporation. The Common Vulnerabilities and Exposures (CVE) organization provides a list of standardized names for SMTP vulnerabilities and other information security exposures. All CVE references (CVE entries and CAN candidates) cited in this text can be found at the CVE Web site, provided in the references (CVE, 2005). Summaries of major SMTP vulnerability problems are given in Table 4

3. Sendmail contains a buffer overflow in code that parses e-mail addresses (CAN-2003-0161). When processing email messages, sendmail creates tokens from address elements (user, host, domain). The code that performs this function (prescan() in parseaddr.c) contains logic to check that the tokens are not malformed or overly long. In certain cases, a variable in prescan() is set to the special control value –1, which may alter the program logic to skip the length checks. Using an e-mail message with a specially crafted address containing 0xFF, an

attacker could case the length checks to be skipped and overwrite the saved instruction pointer on the stack. A remote attacker could execute arbitrary code or cause a denial of service on a vulnerable system. Upgraded versions of sendmail should be used for protection. Another remote buffer overflow in sendmail was reported (CAN-2002-1337). This vulnerability may allow remote attackers to gain root privileges of the sendmail daemon. A properly patched sendmail server (version 8.12.8) will drop invalid headers, thus preventing downstream servers from receiving them.

4. The SMTP that is used by a mail server to send, receive, or route e-mail across a network requires the MAIL FROM (sender) address and the RCPT TO (recipient) address to be specified. Normally, either the sender or the recipient address is in the server's domain. Some SMTP servers accept any sender or recipient address without checking whether at least one of them is in the server's domain. On such servers, it is possible to supply a fake sender address and an arbitrary recipient address, which greatly facilitates the spread of spam. Even SMTP servers, which generally do not allow relaying, do allow it if the session originates from a host in the server's domain or from a host from which relaying is explicitly permitted. If the scan is performed from such a host, a false alarm may result. To resolve this issue, UNIX mail servers should be upgraded to the latest version of Sendmail, which does not allow relaying by default (Antirelay Parse, 2005).

5. Encapsulated SMTP Address Vulnerability

The security vulnerability in Microsoft Exchange Server 5.5 (CVE, 2002, No. 0054) could allow an attacker to perform mail relaying via an Exchange server that is configured to act as a gateway for other Exchange sites, using the Internet Messaging Service. The vulnerability lies in the way that site-to-site relaying is performed viaSMTP. TheSMTPservice in Microsoft Windows 2000 and Internet Mail Connector in Exchange Server 5.5 does not properly handle responses to NTLM authentication, which allows remote attackers to perform mail relaying via an SMTP AUTH command using null session credentials. Encapsulated SMTP addresses could be used to send mail to any e-mail address. The method of configuring the Exchange Internet Mail Service (IMS) (called Internet Mail Connector in prior versions of Exchange), is vulnerable to the attack. The IMS service provides encapsulated addresses, when used as a Site Connector, and uses a special form of addressing called “encapsulated SMTP,” which is used to encapsulate various message types into SMTP addresses. The Exchange supports three kinds of Site Connectors: an X.400 connector, the Exchange Site Connector, and the Exchange Internet Mail Service. A malicious user could address e-mails using this format and route mail through an Exchange Server, even if mail relaying has been disabled. Any customer who has configured an IMS on an Internet-connected Exchange Server should consider installing the patch (“Patch Availability,” 2005) that eliminates the vulnerability.

Attacks in SMTP

It is actually very easy to “impersonate” an SMTP server. You can use the Telnet Protocol to connect directly to an SMTP server on port 25. SMTP commands are all sent as text, and so are SMTP replies, so you can have a “conversation” with a server and even manually perform a mail transaction. This is useful for debugging, but also makes abuse of a wide open SMTP server trivially easy. Since spammers often don't want to be identified, they employ spoofing techniques to make it more difficult to identify them, which makes matters even more difficult.

Originally (see RFC 821), e-mail servers (configured for SMTP relay) did not verify the claimed sender identity and would simply pass the mail on with whatever return address was specified. Bulk mailers have taken advantage of this to send huge volumes of mail with bogus return addresses. This results in slowing down servers.

1. **Spoofing**

On the Internet, mail is usually delivered directly from the sending host to the receiving host. This inherent “open” design of SMTP allows a host computer, which needs to deliver a message to another computer(s), to make a connection (or multiple connections) to some other SMTP server and ask that server to relay the message(s) on its behalf. Gateways can be used to bridge firewalls. By denying access to a sending machine with a firewall, many companies and ISPs have been blocking the receipt

of unwanted mail from known sources. The “blocked” senders of junk mail may attempt to deliver it through another computer by requesting the computer to route that mail for them. Senders of unsolicited e-mail can also use this method to hide their real identity by manipulating the headers in the message and then sending the message through client's system for delivery to its final destination.

This "spoofing" action gives the appearance that the message originated from the relaying server. When

a bulk mailer chooses a client's computer to deliver unsolicited mail to thousands of other people (known as "spamming"), the client's system immediately becomes busy delivering messages that did not originate with the client's users. The SMTP server may protect the client's system against this type of abuse in two ways. First, the server allows administrators to configure the system to accept only mail originating from local users or destined for local users. Second, the server administrator can define systems from which the client never wants to receive mail. It blocks mail from known sources of spam mail ("Setting SMTP Security," 2005).

2. Bounce Attack

In the case of anonymous file transfer protocol (FTP) services, the attacker can instruct the FTP server to send a file to the SMTP service being attacked on the victim's system (see "FTP Security Considerations, RFC 2577). Using the FTP server to connect to the service on the attacked computer

makes it difficult to track down the attacker (Campbellet al., 2003). Particularly, a client - attacker can upload a file that contains SMTP commands to an FTP server. Then, using an appropriate PORT command, the client instructs the attacked server to open a connection to a third computer's SMTP port 25 and transfer the uploadedfile containing SMTP commands to the third computer. This action may allow the client-attacker to forge mail on the third computer without making a direct connection.

Defense Mechanisms for SMTP

The S/MIME is a security scheme for the MIME protocol. It was developed by RSA Security and is an alternative to the pretty good privacy (PGP) encryption and digital signature scheme that uses public-key cryptography. The S/MIME scheme was standardized by IETF. According to "Report of the IAB Security ArchitectureWorkshop" (RFC 2316), the designated security mechanism for adding secured sections to MIME-encapsulated e-mail is security/multipart, as described in "Security

Multiparts for MIME: Multipart/Signed and Multipart/Encrypted" (RFC 1847).

The S/MIME is widely used by large companies that need to standardize e-mail security for both interorganization and intraorganization mail exchange (Internet Engineering Task Force [IETF] SMIME, 2005). It requires establishing a public-key infrastructure either in-house or by using any of the public certificate authorities (Sheldon, 2001).

Antispoofing measures are under active development. Mail Abuse Prevention System (MAPS) and Open Relay Behavior-Modification System (ORBS) provide testing, reporting and cataloging of e-mail servers configured for SMTP relay. These organizations maintain real-time blackhole lists (RBL) of mail servers with problematic histories. For protection and security purposes, companies may configure theirSMTPservers and other e-mail service systems in such manner that any mail coming from RBLblacklisted mail servers is automatically rejected (Campbell, 2003). Other initiatives for restricting the sender address spoofing include SPF, Hotmail domain cookies, and Microsoft's caller ID. An e-mail message typically transports through a set of SMTP servers (including the sender's and receiver's servers) before reaching the destination host. Along this pass, messages get "stamped" by the intermediate SMTP servers. The stamps release tracking information that can be identified in the mail headers. Mismatches between the IP addresses and the domain names in the header could unveil the real source of spam mail. The real domain names that correspond to the indicated IP addresses can be found out by executing a reverse DNS lookup. Modern mail programs have incorporated this functionality, which generates a Received: header line that includes the identity of the attacker

Mail Encryption

SMTP is not a secure protocol. Messages sent over the Internet are not secure unless some form of encryption is implemented. S/MIME is a widely used Internet e-mail standard. This and some other security topics (PGP, transport layer security [TSL], host-to-host encryption) are discussed in other chapters

Restricting Access to an Outgoing Mail Server

The access to an outgoing mail server can be restricted by verifying that the computer is on the ISP's local network. When the user dials the modem and connects to the ISP, his computer is given an IP address that identifies him as being a part of that network. If the user has two ISPs and dials up to one and then connects to the other's mail server, it may prevent him or her from relaying mail because the

Bastille Hardening System

The Bastille Hardening System (Bastille Project, 2005) has been designed to "harden" or "tighten" UNIX-based operating systems. It currently supports the Red HatEnterprise 3, Debian, Mandrake, SuSE, and TurboLinux Linux distributions along with HP-UX and Mac OS X. The Bastille Linux Hardening software [Version 2.1.2 is available from the Source Forge Web site (Bastille Linux Project, 2005)] enhances the security of a Linux box by configuring daemons, system settings, and firewalling. Written in Perl, the Bastille Linux intends to improve Linux-based computer security. Among others, it has a revised sendmail module dedicated to secure holes that were discovered previously (see Table 4). A review of other service modules (Remote Access, Pluggable Authentification, DNS, Apache, FTP, SecureInetd, File Permission, Patch Download, and Firewall Configuration IPChains) can be found in Raynal (2000).

II. Pretty Good Privacy (PGP) Vulnerabilities, Attacks and defense

Pretty Good Privacy (PGP) is a popular program used to encrypt and decrypt e-mail over the Internet. It can also be used to send an encrypted digital signature that lets the receiver verify the sender's identity and know that the message was not changed en route. Available both as freeware and in a low-cost commercial version, PGP is the most widely used privacy-ensuring program by individuals and is also used by many corporations. Developed by Philip R. Zimmermann in 1991, PGP has become a de facto standard for e-mail security. PGP can also be used to encrypt files being stored so that they are unreadable by other users or intruders.

The most commonly used aspect of PGP is the signing and encryption of email or files. "Signing" a document is a way of verifying the integrity of the original work. The method is as follows:

1. Make a digest or "hash" of the file or email. A hash is an algorithm that produces (theoretically) a unique output (the hash) from a given input (the message).

2. Add the hash to the end of the message.

When someone wants to verify that the message has not been modified, they run the hash algorithm on the message and compare it to the hash at the end of the message. If the signatures match, the message has not been altered. PGP uses a variation of the public key system. In this system, each user has a publicly known encryption key and a private key known only to that user. Encryption takes place using public key and decryption takes place using private key. Since encrypting an entire message can be time-consuming, PGP uses a faster encryption algorithm to encrypt the message and then uses the public key to encrypt the shorter key that was used to encrypt the entire message. Both the encrypted message and the short key are sent to the receiver who first uses the receiver's private key to decrypt the short key and then uses that key to decrypt the message. Sender starts with invocation of PGP program on computer.

Step1: PGP hashes the plain text message using MD5 algorithm.

Step2.: PGP then Encrypts the resulting hash using Sender's private key.

Step3: The encrypted hash and original message are concatenated in one single message call it P1.

Step4: This concatenated message is now compressed using ZIP program call it P1.Z.

Step6: Next PGP asks Sender to enter some random input. Both the content and typing speed are used to generate a 128 bit IDEA message key K_M Session key.

Step6. This K_M Session key is now used to encrypt P1.Z with IDEA in cipher feedback mode.

Step7: K_M is also encrypted using receiver's public key.

Step 8: The encrypted key and encrypted P1.Z are concatenated and converted to base64.

Receiver reverses the base64 encoding and decrypts the IDEA key using his private key. He can now decrypt the message using IDEA key to get P1.Z. After decompression , the plain text and encrypted hash are recovered. The hash is decrypted using senders public key. If the plaintext hash and MD% computation of plain text matches, then it can be concluded that the message is intact.

1. PGP is largely based on asymmetric encryption. The strength of the asymmetric key used is crucial to the secure use of PGP. if an attacker is able to break the PGP asymmetric key all encrypted documents or messages of the past, present and future may be compromised. Therefore it is very important that the public key algorithm PGP users select is proven to be strong, secure and immune to cryptanalysis.

2. PGP gives remote attackers the ability to use encrypted files to force a vulnerable machine to run arbitrary code. The flaw lies in the way that the Pretty Good Privacy Corporate Desktop 7.1.1 application handles encrypted files. In many instances, the application fails to check the length of the filename. As a result, PGP will crash if the user tries to encrypt or decrypt a document with an overly long filename. An attacker could exploit this fairly easily. Once he creates a filename of the specified length, he would then simply encrypt the file with the public key of the targeted user and send the file[2]. When the user tries to decrypt the document, the filename would overflow the memory buffer set up for it and execute whatever code the attacker has included. In some cases, this condition may also reveal the users passphrase, as PGP crashes after the file is decrypted but before the passphrase is overwritten in memory.

3. If an attacker has access to the pass phrase and secret key file they can read the compromised user's encrypted messages and make signatures using that user's private key5.

4. In a public key cryptosystem, the public keys of users

should be distributed so all have the components necessary to securely communicate and exchange information with each other. A crucial component in this system is the fact that users must be able to trust that a public key really belongs to whom it appears to belong to.

5. As the PGP source code freely available, any attacker with enough time, skill and motivation could potentially develop a "trojaned" version of the software. This trojaned version of PGP may be widely circulated, and may claim to be from a reliable or well-known source. A trojaned version may also be introduced to the target user's computer by other tactics.

6. PGP Desktop versions 10.0.3 and earlier, as well as the upcoming 10.1 release, are vulnerable to a "piggy-back" attack. The PGP Desktop user interface incorrectly displays messages or files with unsigned data as signed. This includes unsigned data (possibly malicious) inserted into previously signed messages and files. A user will not be able to distinguish the legitimate signed part from the malicious unsigned parts. This

vulnerability compromises the ability to use digital signatures to verify the integrity of files and E-mail.

Attacks on PGP

PGP is a hybrid cryptosystem. It is made up of 4 cryptographic elements: It contains a symmetric cipher (IDEA), an asymmetric cipher (RSA), a one-way hash (MD5), and a random number generator (Which is two-headed, actually: it samples entropy from the user and then uses that to seed a PRNG). Each is subject to a different form of attack.

1. Pass-phrase and Private Key File Compromises

One of the most common ways for an attacker to compromise the security of PGP on a system is to obtain another user's pass-phrase and/or private key file. If an attacker has access to the pass phrase and secret key file they can read the compromised user's encrypted messages and make signatures using that user's private key. This weakness is not just specific to PGP; this is a typical weakness found in most password/pass-phrase authentication cryptosystems. Users may select easily guessed pass-phrases or may store their private key in a location where someone with malicious intent may access it. An attacker may also use a tool or utility that will try to obtain a user's pass phrase from the local workstation. Brute force or dictionary attack utilities such as PGPCrack or PGPpass are designed to crack PGP encrypted files. The attacker may also use a keyboard logger utility that can capture the keystrokes of an unsuspecting user and save it to a designated location where they can then retrieve the user's revealed plaintext pass-phrase.

2. Public Key Tampering

In a public key cryptosystem, the public keys of users should be distributed so all have the components necessary to securely communicate and exchange information with each other. A crucial component in this system is the fact that users must be able to trust that a public key really belongs to whom it appears to belong to. This particular vulnerability is quite important to be aware of, many novices and even IT professionals may fall victim to trusting the tampered public key of an impostor. Once a user encrypts a document or message with the tampered public key and sends it, the impostor will be able to decrypt and read the contents. The user may have just unintentionally disclosed sensitive personal or proprietary company information.

3. Operating System Attacks

Many users are not aware that in a lot of cases, when a file is deleted only the file allocation information changes and the file contents still reside on the disk until that space is overwritten by another file. If PGP is used to encrypt a file, and the original document is then deleted, the user may think that the deleted file is completely gone forever. When in actuality one of many disk recovery tools can be deployed on the local machine to recover and resurrect several of that user's previously deleted files. If a potential attacker has access to the deleted disk blocks before they have been reallocated, they may be able to recover the user's unencrypted original document.

4. Trojan Horse Attacks

Trojan horse attacks pose one of the most serious threats to application security today. A Trojan horse is a seemingly harmless program that contains malicious code, which may infect a users PGP application or their operating system subverting the security of both. An attacker could use this code to capture a user's plaintext messages, or to obtain the user's pass-phrase or private key. The wide scale use of the internet as a software delivery mechanism has increased the chances a user downloading and installing a rogue or modified copy of PGP software. Because the PGP source code freely available, any attacker with enough time, skill and motivation could potentially develop a "trojaned" version of the software. This trojaned version of PGP may be widely circulated, and may claim to be from a reliable or well-known source. A trojaned version may also be introduced to the target user's computer by other tactics. The attacker may replace the legitimate commercial or open source copy of PGP already installed on a user's computer with a rogue copy when the user isnot at their workstation. The attacker may also conduct a "Man-in-the-middle attack" hijacking a user's download session from a legitimate download site and substituting the download with the attacker's altered PGP copy. These altered versions may appear to behave like PGP in many respects, but they may be crippled in some manner, for instance the software may not check signatures properly, allowing counterfeit certificates to be accepted. The random number generation routine may be modified to produce predictable results, the cryptographic routines involved may be weakened, and the program may even encrypt the infected user's messages with an additional key giving the attacker access to all files encrypted using the altered version.

Recent versions of PGP allow users to select a preferred symmetric algorithm that will be used when pople encrypt messages for them. The options available to users include: IDEA, CAST, Triple DES and in newer releases the AES and Twofish algorithms. All of the algorithms are considered strong and adequately secure enough to use with confidence. All Diffie-Hellman/ElGamal private keys are encrypted to the CAST, and will therefore be the symmetric cipher. Entrust Technologies' CAST symmetric encryption algorithm, is one of the fastest and most secure algorithms available. This version of CAST known as CAST5, or CAST-128 has a block-size of 64-bits and a key length of 128-bits. Research conducted for this paper has shown that CAST is resistant to most known cryptanalysis techniques and that there really is no known way of breaking CAST short of brute force

III. Conclusion

Simple Mail Transfer Protocol is the primary and most deployed protocol for e-mail communication. It is being continuously revised by the inclusion of new commands, security mechanisms, message formats and efficiency procedures. Most SMTP servers are implemented using some library that needs to be constantly upgraded to take advantage of improvements in SMTP. The study of SMTP involves use of utilities permitting issuance of individual SMTP commands and studying of their responses directly. Add-on e-mail security protocols use encryption, PKI based cryptographic techniques, IP address verification and DNS based domain validation for providing security against spoofing and other e-mail threats. H In this paper we tried to portray some of the major vulnerabilities of SMTP and PGP and attacks because of them. However, no protocol independently provides all required security features. Further, domains that are not compatible with security protocols continue to pose security threats by allowing transmission of spoofed e-mails that are not detected by receiving domains using security protocols.

References

1. Marks Jacobson and Stevan Myers, “A phishing and countermeasure: understanding the increasing problem of electronic identity theft”, Adobe E-book, ISBN 978-0-470-080609-4 Dec 2006

2. Resnic P.Ed. “Internet Message Format”, IETF RFC 2822, Apr2001

3. Klensin "Simple Mail Transfer protocol", IETF RFC 2821

4. lastimil Klíma 1 and Tomáš Rosa, "Attack on Private Signature Keys of the OpenPGP format", PGPTM programs and other applications compatible with OpenPGP, Dept. of Computers, FEE, CTU in Prague 22nd of March, 2001

5. ttp://www.eweek.com/c/a/Desktops-and-Notebooks/PGP-Vulnerability-Opens-Door-to-Remote-Attacks/

6. "Attacks on PGP: A Users Perspective" SANS Institute InfoSec Reading Room

7. Kahil Jallad, Jonathan Katz, Jena J. Lee, and Bruce Schneier, "Implementation of Chosen-Ciphertext Attacks

8. against PGP and GnuPG" Eric R. Verheul, "Pretty Good Piggy-backing Parsing vulnerabilities in PGP Desktop", Digital Security group, Radboud University Nijmegen Security & Technology group, PwC

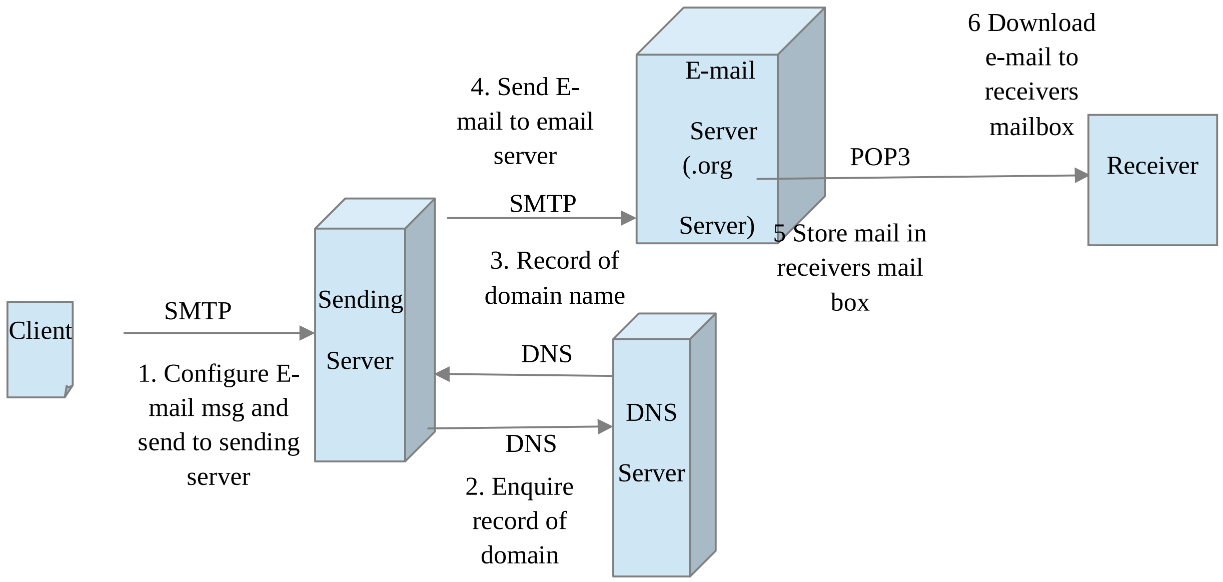
6 Download e-mail to receivers mailbox
E-mail Server (.org Server)
4. Send E-mail to email server
SMTP
POP3
Receiver
5 Store mail in receivers mail box
3. Record of domain name
Sending Server
Client
SMTP
1. Configure E-mail msg and send to sending server
DNS
DNS
DNS Server
2. Enquire record of domain

Figure 1 SMTP Mail System Architecture