The Regular Emailing Protocols with Drawbacks

For the day-today usage we use email protocol like SMTP, POP3, IMAP4 etc. Here, we like to discuss the protocols (specially the SMTP and POP3) and show some practical scenario where it will demonstrate how insecure and unreliable the protocols are.

1.1 Simple Mail Transfer Protocol (SMTP)

The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently. Within the Internet, email is delivered by having the source machine establish a TCP connection to port 25 of the destination machine. Listening to this port is an email daemon that speaks SMTP. SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. SMTP uses RFC (Request For comment) 821, 822.

1.1.1 Commands

The mail commands and replies have a rigid syntax. The mail commands, replies and host names are not case sensitive. Replies have numeric code. Commands and replies are composed of characters from the ASCII character set. When the transport service provides an 8-bit byte (octet) transmission channel, each 7-bit character is transmitted right justified in an octet with the high order bit cleared to zero. [1]

The first command from the client is HELO. This command is used to identify the sender-SMTP to the receiver-SMTP. The argument field contains the host name of the sender-SMTP. The receiver-SMTP identifies itself to the sender-SMTP in the connection greeting reply, and in the response to this command. This command and an OK reply to it confirm that both the sender-SMTP and the receiver-SMTP are in the initial state, that is, there is no transaction in progress and all state tables and buffers are cleared.

1.1.2 A Sample Session between SMTP Client C and Server(S)

```
   -----------------------Connection opening--------------------------

                 S: 220  web.bdmail.net Simple Mail Transfer Service Ready
                 C: HELO  msrahman.bdmail.net
                 S: 250  web.bdmail.net

    ----------------------- SMTP procedure-------------------------------

                 C: MAIL FROM:<msrahman@bdmail.net>
                 S: 250 OK

                C: RCPT TO:<shaila2002bd@yahoo.com>
                S: 250 OK

                C: DATA
                S: 354 Start mail input; end with <CRLF>.<CRLF>

                C: Testing SMTP protocol……
                C: ...etc. etc. etc.
                C: <CRLF>.<CRLF>
                S: 250 OK

 ----------------------------- Connection closing--------------------------

                C: QUIT
                S: 221 web.bdmail.net Service closing transmission channel

------------------------------End session--------------------------------------
```

1.1.3 Problem Associates with SMTP Protocol

There are few problems in SMTP protocols. These problems are discussed below.

- **Message length problem:** Some older implementation can not handle messages exceeding 64KB.

- **Timeout problem:** If the client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection.

- **Infinite mail storms:** In rare case, infinite mail storms can be triggered. For example, if host one holds mailing list A and host two holds mailing list B and each list contains an entry for the other one, then any message sent either list will generate a never-ending amount of email traffic.

1.2 Extended SMTP Protocol (ESMTP)

To overcome above problems ESMTP protocols are defined in RFC 1425.Clients send an “EHLO” message initially instead of “HELO”. If this is rejected then the server is a regular SMTP server and the client should proceed in the usual way. If the EHLO is accepted, then new commands and parameters are accepted.

1.3 Post Office Protocol (POP3)

POP3 protocol is used for fetching mail from a remote mailbox and store it in the user’s local machine to be read later. It is defined in RFC 1939 (Version 3). The default port of POP3 server is 110.It has similarities with SMTP protocol.

1.3.1 Commands

Commands in the POP3 protocol consists case-insensitive keyword. Keywords and arguments consist of printable ASCII characters. Keywords are three or four characters long. Each argument may be up to 40 characters long. Responses in the POP3 consist of a status indicator and a keyword. Responses may be up to 512 characters long, including the terminating CRLF. In POP3 commands and responses, all message-numbers and message sizes are expressed in base-10 (i.e., decimal). There are currently two status indicators: positive ("+OK") and negative ("-ERR"). Servers MUST send the "+OK" and "-ERR" in upper case.

A POP3 session progresses through a number of states during its lifetime. The states are discussed below.

- **Authorization State:** Once the TCP connection has been opened and the POP3 server has sent the greeting, the session enters the AUTHORIZATION state. In this state, the client must identify itself to the POP3 server. In this state USER, PASS, QUIT commands are used.

- **Transaction State:** After successful identification of the client, the server acquires resources associated with the client's mail drop, and the session enters the TRANSACTION state. In this state, the client requests actions on the part of the POP3 server. In this state DELE, RETR, LIST, RSET, NOOP are used.

- **Update State:** When the client has issued the QUIT command, the session enters the UPDATE state. In this state, the POP3 server releases any resources acquired during the TRANSACTION state and says goodbye. The TCP connection is then closed.

Authentication is done by using the USER and PASS commands together. An alternate method of authentication is required which provides for both origin authentication and replay protection, but which does not involve sending a password in the clear over the network.

A POP3 server may have an inactivity auto logout timer having at least 10 minutes' duration. The receipt of any command from the client during that interval should suffice to reset the auto logout timer. When the timer expires, the session does not enter in to the UPDATE state--the server should close the TCP connection without removing any messages or sending any response to the client.

1.3.2 A Sample Session between POP3 Client(C) and Server(S)

```
S: +OK POP3 server ready
C: USER srahman2

S: +OK Password required for srahman2
C: PASS tania
S:  +OK srahman2 has a 1 message (958 octets)

C: STAT
S: + OK 1 958

C: LIST
S: +OK 1 message (958 octets)

C:RETR 1
```

```
  S:+OK 958 octets

 C: TOP 1,10
 S: +OK

C:RETR 1
S:+OK 958 octets

C: DELE 1
S: +OK message 1 is deleted
.

C:QUIT
S:+OK POP3 server signing off
```

1.3.3 Problem Associates with POP3 Protocol

In POP3 protocol security considerations is essential. The reasons are as follows.

- Servers that answer -ERR to the USER command are giving potential attackers clues about which names are valid.

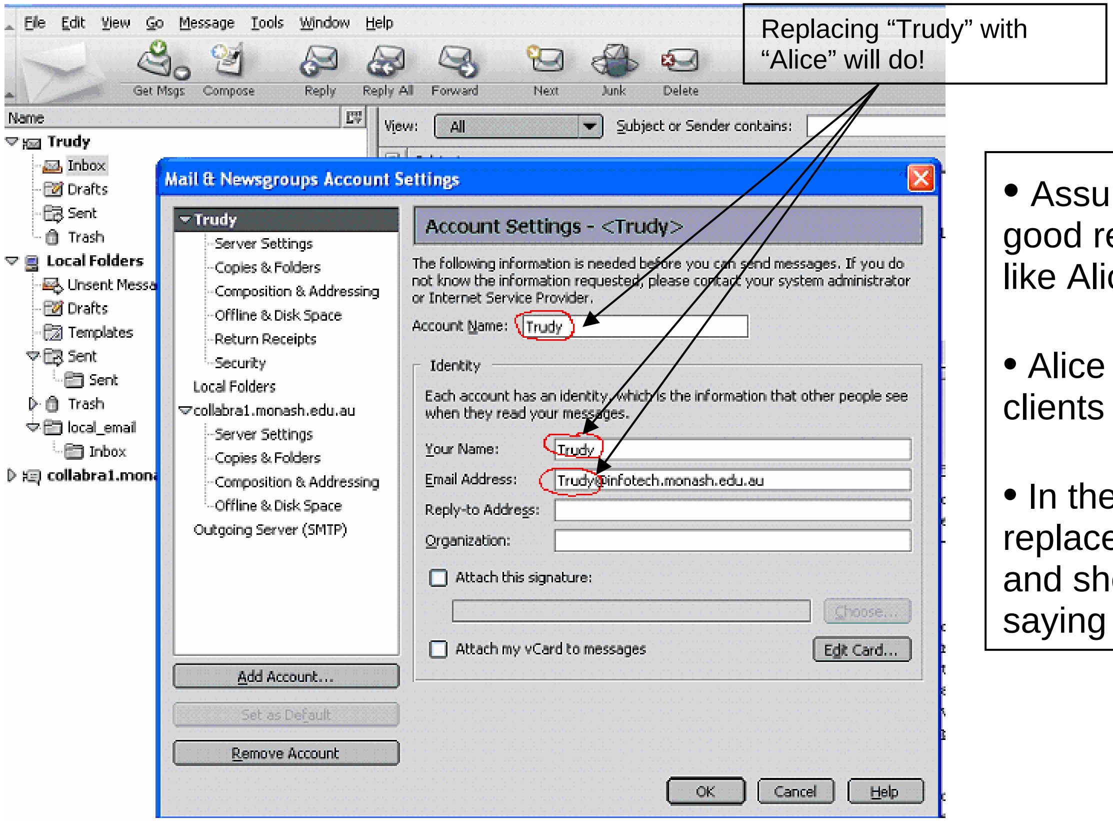
- Use of the PASS command sends passwords in the clear over the network.

- Use of the RETR and TOP commands sends mail in the clear over the network.

<u>Practical demonstration to show the drawbacks of the SMTP, below:</u>

Drawbacks of the SMTP Protocol

- SMTP protocol does not check who is sending email, checks whether sender's and recipient's email addresses exist or not.
- Unauthenticated sender: Change the name of the sender in your email program and test.



Replacing "Trudy" with "Alice" will do!

Account Settings - <Trudy>

The following information is needed before you can send messages. If you do not know the information requested, please contact your system administrator or Internet Service Provider.

Account Name: Trudy

Identity

Each account has an identity, which is the information that other people see when they read your messages.

Your Name: Trudy

Email Address: Trudy@infotech.monash.edu.au

Reply-to Address:

Organization:

□ Attach this signature:

Choose...

□ Attach my vCard to messages

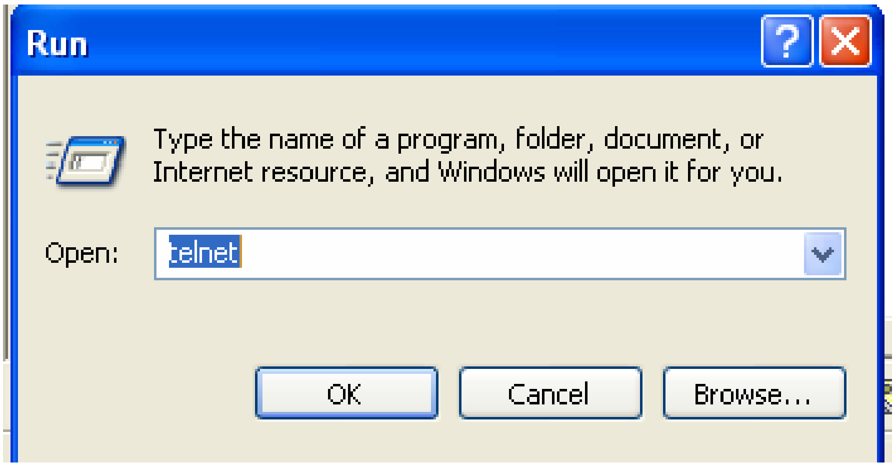
Edit Card...

- Assume, Bob and Alice have good relation. Trudy does not like Alice and the relation.

- Alice and Trudy are email clients of the same server.

- In the email software, Trudy replaces her name with "Alice" and she sends an email to Bob saying "I hate you, Bob …"

To have an in-depth understanding of the protocol you can test it direct using telnet. Telnet is a program can be found in you PC and can be run through the Run window as shown below.



Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: telnet

Figure 1: Running telnet.

Then press ok.

To study how to use telnet type ‘help’ and then enter.

Now, you need to know the address of the SMTP server of you email account. For example in Monash a server can be found ‘smtp.monash.edu.au’ having port number 25.

To connect to the server to need to insert the following command:

```
Open smtp.monash.edu.au 25
```

Be careful about spacing.

The reply would be

```
220 …..
```

Which (the number 220) implies that it is positive that you are connected with the mail server. Consult RFC 821 and 822 for the meaning of the coding in details.

Now, say your email address is, john@student.monash.edu.au and you want to send a fake email which will show that it is coming from lisa@student.monash.edu.au and will be received bob@student.monash.edu.au. Please don’t do this in practice to confuse person, it should be a limited practice to understand how unreliable the protocol is, and the intension of this document to let you have a feeling of having a secured emailing, like PGP. Also, the moral it to protect yourself from fake emails.

Next comment should be

```
MAIL FROM:<lisa@student.monash.edu.au>
```

Comment: Be careful about the exact spacing and error correction using 'backspace or del'. There might be hidden character for using those the command may need to enter again for an unsuccessful reporting.

Reply should be: 250 Ok

Next command:

```
RCPT TO:<bob@student.monash.edu.au>
```

Reply should be: 250 Ok

Next command:

```
DATA
```

That is you want message (data) for bob.

Reply should be: End data with <CR><LF>.<CR><LF>

The meaning of the reply is, when want to finish putting message;
Press enter, then press a '.', and then again press enter.

Putting some data:

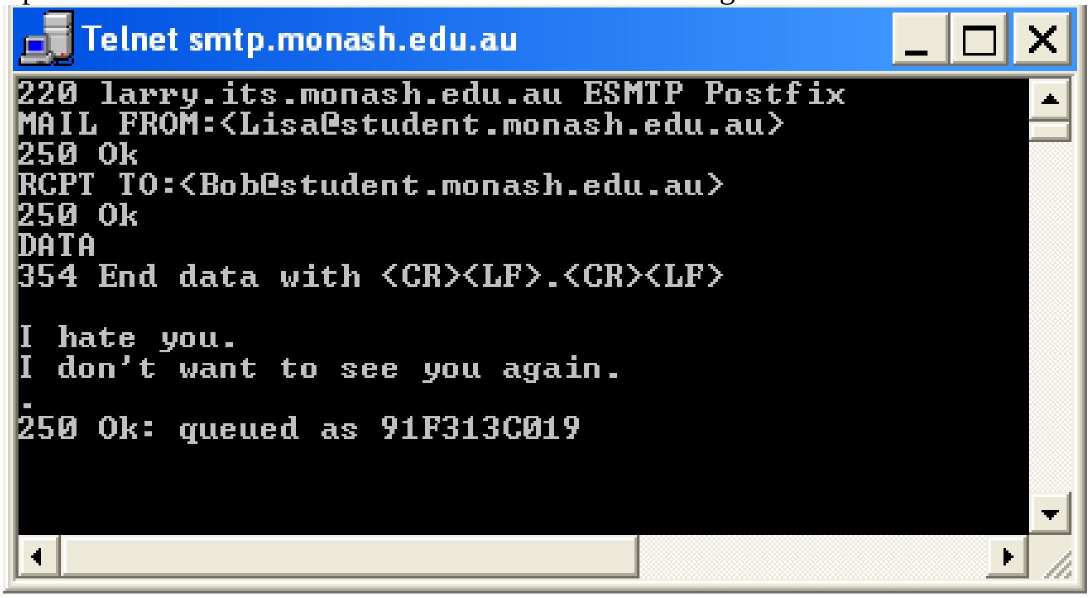
```
I hate you.
I don't like to see you again.
```

Next command: Then to make an end of the message press <enter> then a dot '.', and then <enter>.

You will get a massage number returned; the server assigns a number for the message. To quit, you need the next command:

```
Quit
```

A practical scenario of the server communication is in Figure 2.



```
220 larry.its.monash.edu.au ESMTP Postfix
MAIL FROM:<Lisa@student.monash.edu.au>
250 Ok
RCPT TO:<Bob@student.monash.edu.au>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>

I hate you.
I don't want to see you again.
.
250 Ok: queued as 91F313C019
```

Figure 2: A practical scenario, talking to SMTP server using telnet.

For you information, the SMTP protocol is worse that you think. Even if you put some garbage address say, hsgds8q@ajhdjdsasd.wqer instead of Lisa@student.monash.edu.au, it will work, though it will say “250 Ok”.
So, can you imagine the level of drawback of the SMTP protocol in case of proving authenticity?