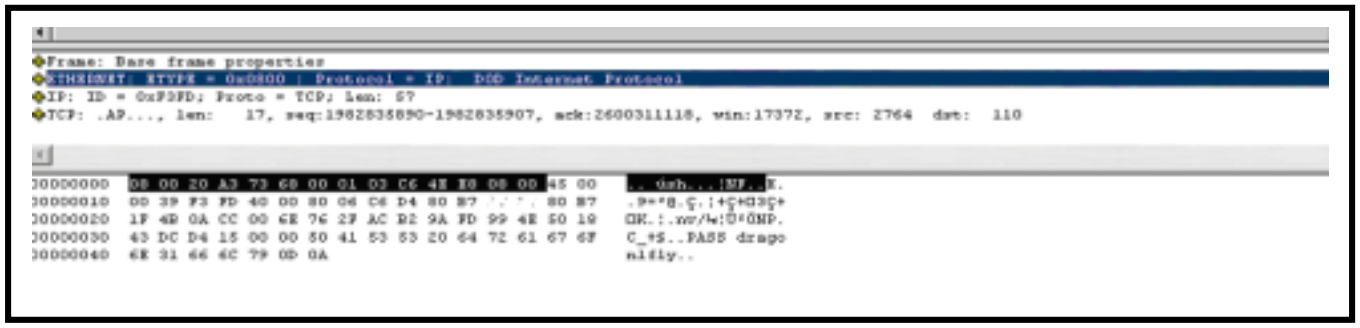
Securing POP Mail on Windows Clients

The Problem:

POP mail, which is often the mail server for Eudora and Outlook clients, historically uses an insecure protocol. Protocol analyzers allow folks, good and bad, to see network traffic. Using the output of a protocol analyzer see if you can see the problem with a typical POP transaction below.



As you can hopefully see, the password is directly viewable. This password may also be (and typically is) a Unix account password, which could lead directly to an account compromise on the mail server.

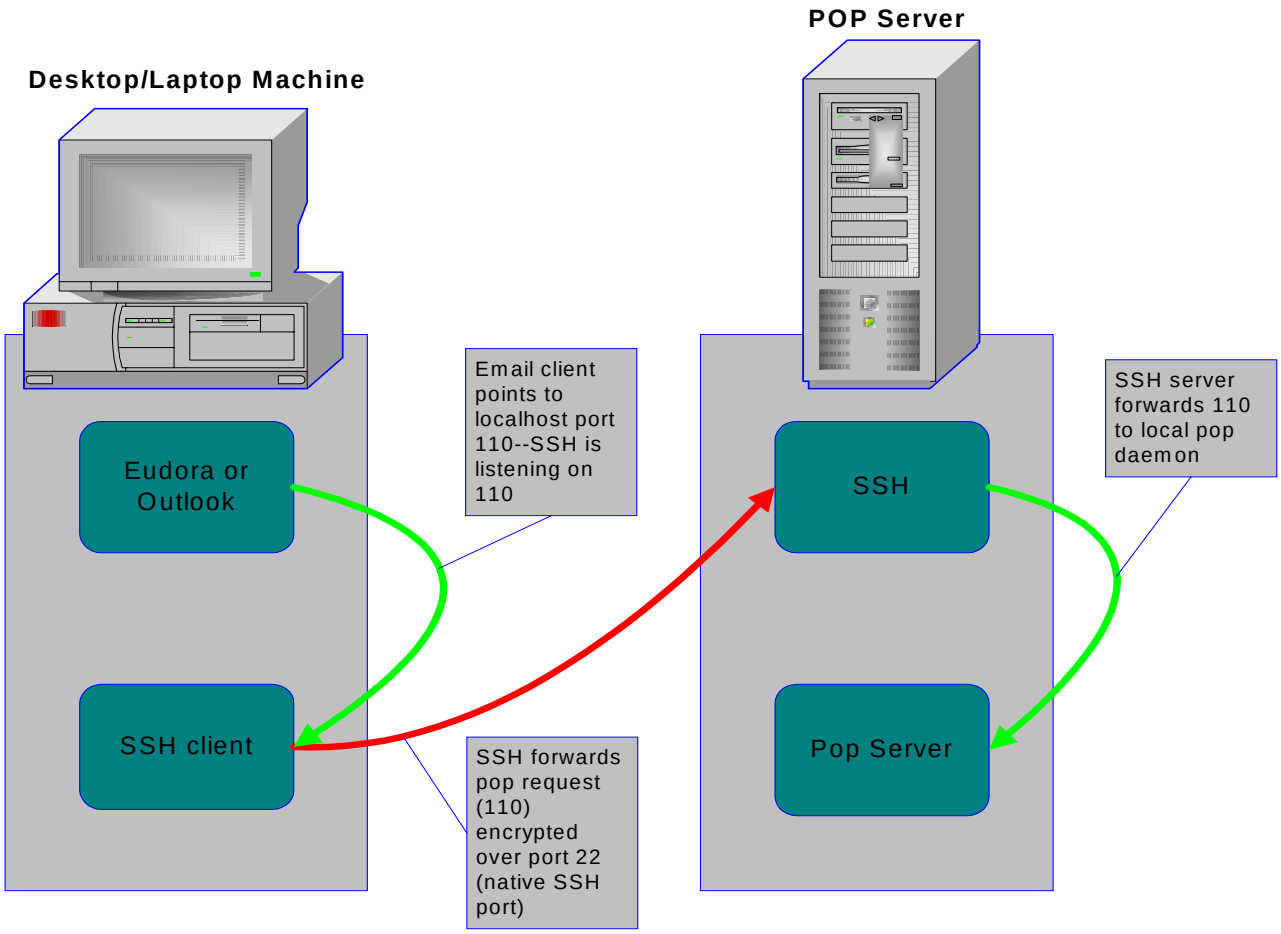
A Solution:

Use the SSH windows application, which is widely available as freeware and already ubiquitously deployed, to secure the POP protocol. The primary objective is to hide the password. Secondarily, all mail messages traveling from the mail server to the client are secured through encryption. An obvious caveat here is that mail directed to you will probably have already traveled across the Internet, or across your local network, to the mail server in the clear. To guaranty your mail content privacy other solutions such as PGP and/or PKI technologies should be employed.

A brief technical description of the solution follows. If you are not interested or you are uncomfortable with the term TCP port skip to the *How to* section.

Instead of having the Eudora/Outlook client point to the mail server at TCP port 110 the client is redirected to localhost (back to your machine) still on port 110. SSH will be configured to listen for 110 locally and to forward to port 110 on the mail server. The forward will be an SSH tunnel. The SSH tunnel relies on the native SSH security features of authentication, integrity, and confidentiality.

The flow of data for POP through an SSH tunnel follows:

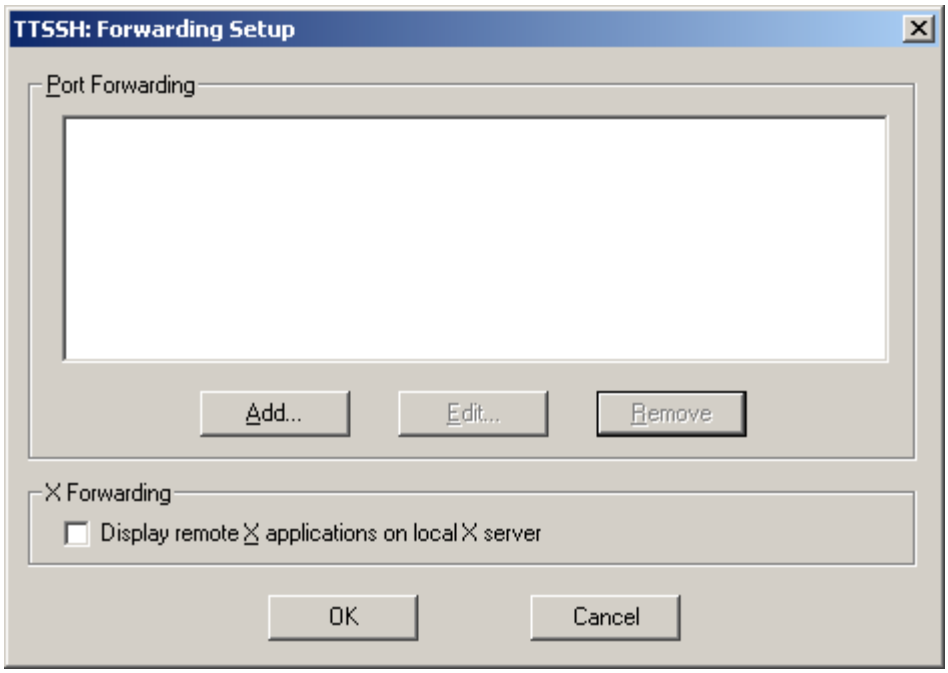


How to:

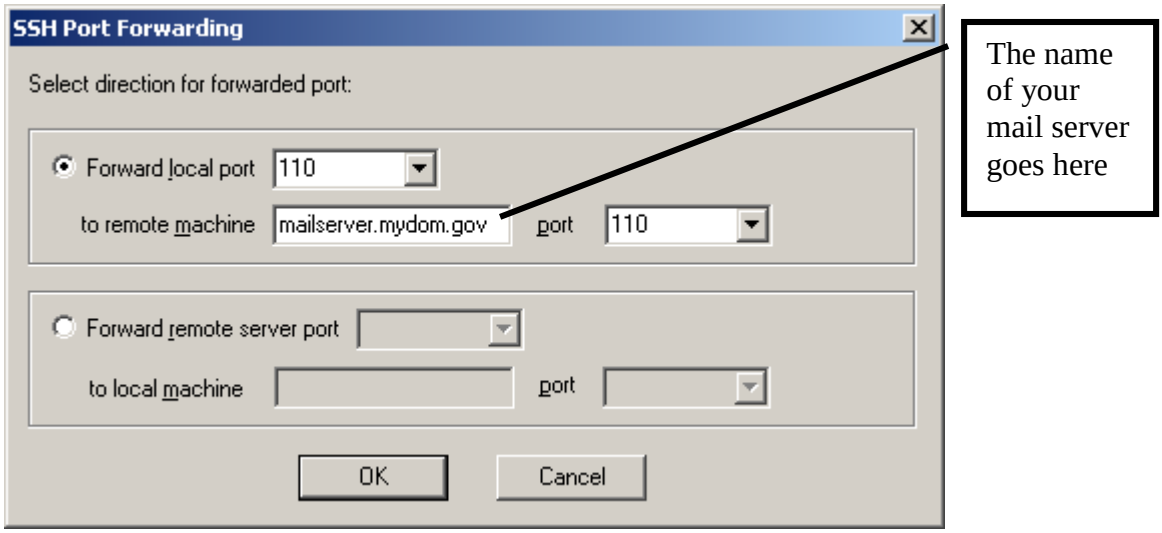
1. Ensure that you can logon to the mail server using the SSH client. Please consult other references on using SSH if you have problems with this step. Typically SSH clients will allow you to save a profile with your target server name and account name.

2. Set up SSH port forwarding as so (this example uses the Tera Term terminal application with the TTSSH extension):

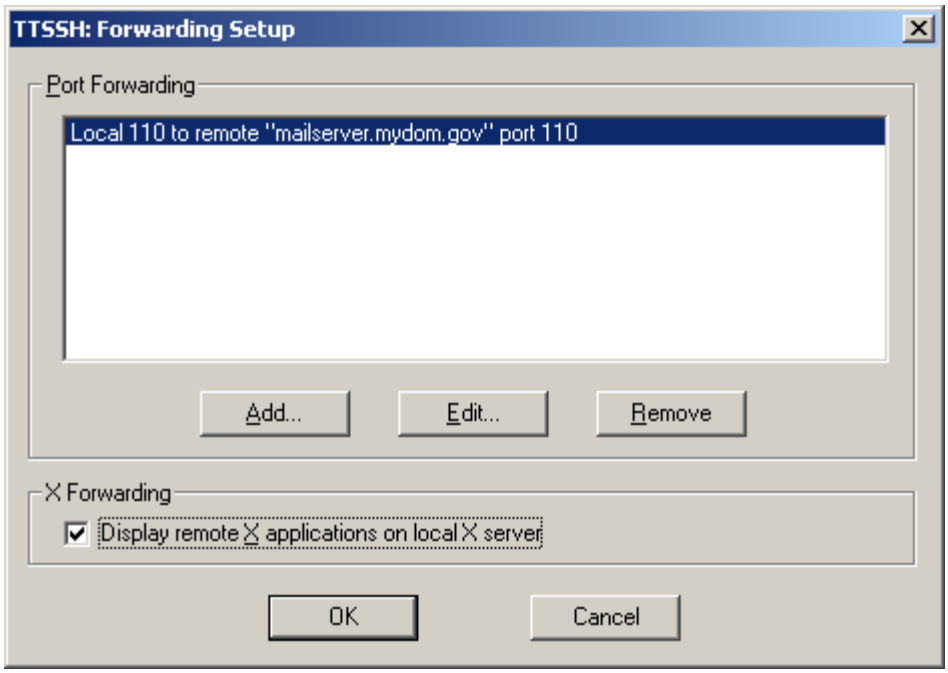
    a. On the Menu bar—Setup/SSH Forwarding the following dialog is available:



b. Add the forwarding as so:



c. Your configuration should then look something like this:

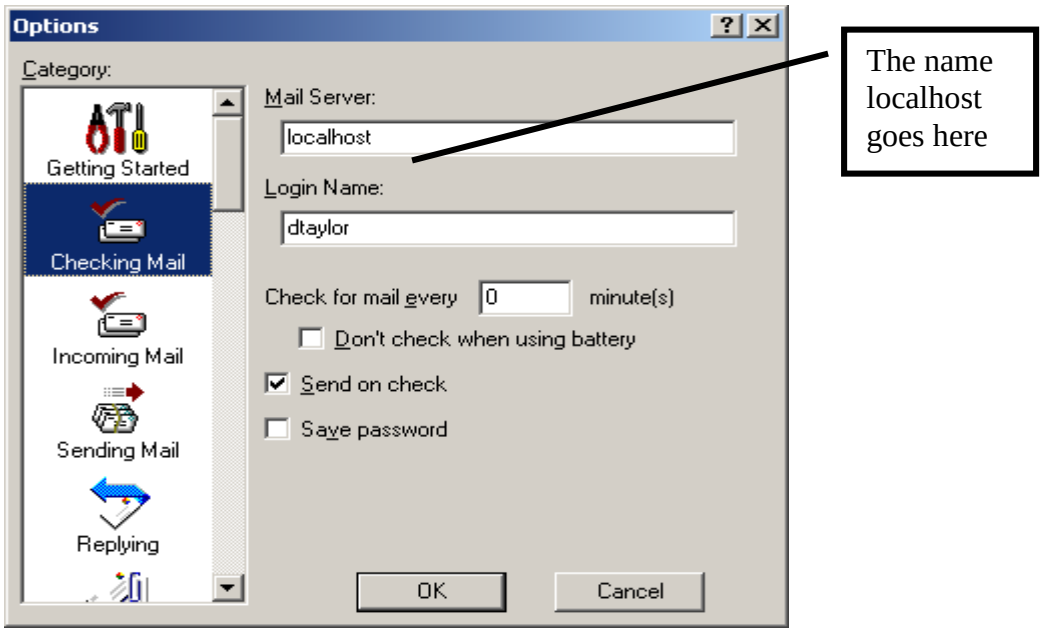


d. For security reasons you should edit your teraterm.ini file to make sure variable "LocalForwardingIdentityCheck" is set to 1. If this variable would be set to 0 then a malicious user could route through your machine to reach the target mail server.

3. Change your mail application to use localhost as the POP server.

For Eudora:

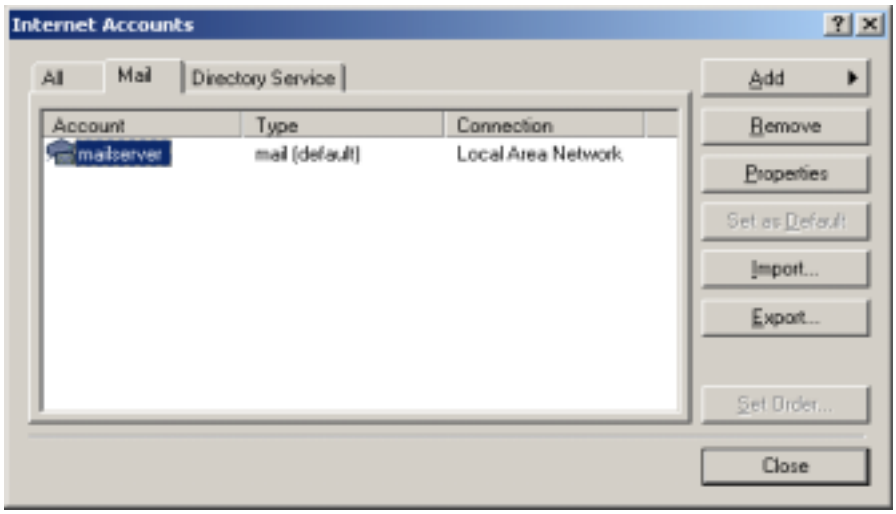
a. Set your mail server to localhost—from the menu bar, under tools/options



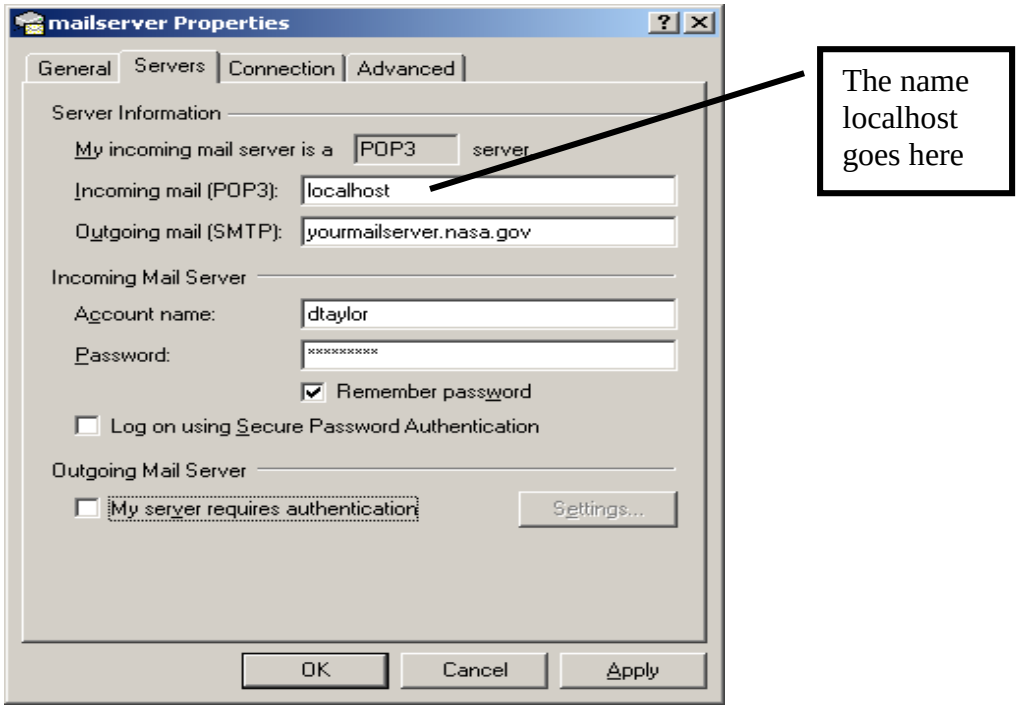
b. Save the file, then start or restart your Eudora application.

For Outlook:

a. Select Tools then Accounts from the Menu Bar which will bring up the Internet Accounts dialog (below)



b. Select Properties (above) and the Servers tab (below), and then change the server name from your POP server to the name “localhost”.



4. Operational requirements: You will need to keep an SSH terminal session logged on to your mailserver to receive mail. This SSH session will keep the POP mail tunnel active between your machine and the mailserver.

Conclusion:

There are no perfect security solutions. However, avoiding clear text passwords transmission is probably the single largest security improvement that can be made.

Using SSH port forwarding as an authenticated and encrypted tunnel as described in this paper for obtaining your POP email is highly recommended as a prudent security measure.

There are many further references to Secure Shell, SSH tunneling, and SSH tunneling for POP on the Web. Please consult them for further reference. Some selected references follow.

References:

1. SSH
   - http://www.openssh.org/
2. Tera Term (version used in example: 2.3)
   - http://hp.vector.co.jp/authors/VA002416/teraterm.html
3. TTSSH – an SSH extension to Tera Term (version used in example: 1.5.4)
   - http://www.zip.com.au/~roca/ttssh.html