**LAB 22**

**E-mail - SMTP**

1. In this lab we will “manually” talk with the Howest SMTP server, the same way your email client does behind the scenes. What is the SMTP protocol used for: retrieving or sending e-mail?

Sending an e-mail

1. For unsecured SMTP, you can login to an SMTP server using the ***telnet*** command. For a secured SMTP server, we’ll use the ***openssl*** command on your Linux VM. What is the default port for a secured SMTP server?

Port 587

1. Start your linux VM. It will be most convenient to start an SSH session from your laptop host to your VM (for easy copy/pasting), rather than typing in the linux terminal within VMware. (And this is a recommendation in general, actually.) As a reminder, following steps are to be executed

* Login to your Debian locally and find out its IP address.
* Open on your laptop host a command prompt / PowerShell / Terminal (Mac) and ssh to your VM using **ssh <user>@<ip-address>**

Hint: if you can’t ssh, verify if you can ping. If you can’t ping, maybe you’ll need to disable/re-enable your virtual VMnet8 adapter.

* Start a second ssh session to your VM as well, as you’ll need a second CLI in the next steps.

1. The Howest secured SMTP server is hosted in Microsoft’s Office365 cloud. In one of your SSH sessions with your VM, use following command to connect to the SMTP server:

***openssl s\_client -starttls smtp -crlf -connect smtp.office365.com:587 --quiet***

* Note the used well-known port number and note that the “STARTTLS” option is used in the command for authenticated SMTP (cfr slides)
* You should see the server sending its server certificate to you (scroll up to see it).
* Then scroll down to the last line of info you received. That line starts with a **status code**. What is that status code and what does that code mean? Use <https://tools.ietf.org/html/rfc5321#section-4.2.3> (i.e. the official RFC describing SMTP) to find out the meaning of that status code.

200 – service ready

***Note: If you are idle for too long (a few minutes or so), you’ll be kicked out and you’ll get a status code saying so. Anytime during the lab when you would encounter a timeout, you’ll have to re-execute all previous steps.***

1. Now, be polite 😊 and greet the server with the command: ***EHLO***

Note: this is not a typo, the old command was HELO (with one L indeed, as one uses commands of only 4 letters), but the newer command is EHLO (which stands for Extended Hello).

The server will greet you as well and you’ll get again the same status codes with the mentioning of some supported features.

1. As this is a secured SMTP server, you’ll need to login. Type the command:

***AUTH LOGIN***

to tell the server you want to login.

* You should get this line: 334 VXNlcm5hbWU6 . This is a server challenge encoded in Base64. In your second SSH session, use the ***base64*** command in linux to decode that string to know what it means.

You can decode an encoded string this way:

***echo -n encodedstring | base64 -d***

1. You will have to provide your Howest mail address, encoded in base64 to login. In your second SSH session, use this command to encode your mail address:

***echo -n firstname.lastname@student.howest.be | base64***

(using your own mail address of course).

Copy that output into your SMTP session.

1. You should then receive a 334 UGFzc3dvcmQ6 . What does this mean?

This is ‘PASSWORD’, the reply-code encoded in BASE64

1. Now you need to provide your Howest password, encoded using base64. You could use the base64 command in the same way as you’ve encoded your mail address, but then it would be visible to anyone sitting next to you (or worse: to your lecturers 😲). Besides, it would also be recorded in the shell history. Therefore, better is to use a trick by saving your password interactively in a variable $PASS and hide characters while typing. That password can then be piped to the base64 command.

* This is the way how to do it : ***read -s PASS && echo -n $PASS | base64*** . You can then type your password interactively (password not shown while typing) and terminate with ‘Enter’.
* The output is your base64-encoded password. Copy that into your SMTP session.
* You should finally have a 235 2.7.0 Authentication successful

1. Now specify you want to send mail from your own mail address, with the command: ***mail from:firstname.lastname@student.howest.be***

(using your own mail address of course)

1. Now specify where you want to send the mail to, with the command:

***rcpt to:someone@something.com***

(e.g. your private mail address)

1. Then, specify you’ll start the content of the mail, with the command:

***data***

1. Now specify an e-mail subject of your choice by preceding it with the string:

“Subject:”, e.g.

Subject:my smtp test

1. Then end the message by typing a dot (.) on an empty line (that’s why the server said “end with <CRLF>.<CRLF>”)
2. That’s it! You’ve just sent an e-mail completely via CLI by talking with the SMTP server itself directly, without using any e-mail client software!