

Tarea 1 Redes -SMTP IMAP

Jeffrey Ricardo Alfaro Fonseca 2015069856

Introducción:

- SMTP-Server: Se crea un mail server el cual implementa el protocolo SMTP, este servidor se deberá de implementar con la ayuda de la biblioteca Twisted para python. Este servidor permitirá el envío y recepción de correos electrónicos utilizando dicho protocolo(locales). Además este servidor permitirá la comunicación utilizando una capa segura como la de TLS. Para ello se implementó con la herramienta openssl. Además permite el envío de mensajes con estándar MIME.
- SMTP-client: El cliente recibirá una lista de correos electrónicos con nombre y correo entre comillas separados por filas en un CSV. El mensaje se le pasará como parámetro. Y permitirá el pasaje de parámetros para completar la comunicación y el mensaje
- Servidor de IMAP: El servidor de IMAP brinda servicio a un usuario, con el objetivo de que el usuario pueda acceder y leer su correo

electrónico desde cualquier cliente que permita el uso de dicho protocolo. Para ello se autenticará en el servidor, descargará los correos, y se mostrarán en el cliente de correo como thunderbird. El acceso a este servicio también deberá de ser configurado para que soporte una capa de cifrado como TLS.

Herramientas usadas

- Lenguaje de Programación : python 2.7
- SO: Kali Linux (linux)
- Dillinger: docu
- Api o Biblioteca de Twisted
- Entre otras bibliotecas
- openssl

Estructuras de datos usadas y funciones:

- ServerOptions: esta clase se usa para manejo de argumentos en la consola
- LocalMessageDelivery ,MaildirMessage,LocalSMTPFactory: LocalMessage se usa para validar las entradas de los dominios de los correos en este caso solo se validan los correos de los remitentes y no de los destinatarios, el Maildir obtiene el mensaje y toda su estructura y la guarda en una carpeta con el nombre del correo del remitente y el SMTPFactory es el

encargado de crear el protocolo SMTP

- `setBodyMsg`: es el que tiene el template del mensaje para incluir los nombres de los destinatarios en el cliente smtp
- `sendmail`: envia el mensaje con su respectivo cuerpo encabezado to y from
- `IMAPUserAccount`: Evalua que existe la carpeta del usuario y todos sus mensajes
- `MaildirMessage`: el que maneja el contenido de los mensajes
- `IMAPFactory`: crea el protocolo IMAP
- `IMAPServerProtocol`: contruye el protocolo imap (el paso de mensajes entre cliente y servidor)
- `IMAPMailbox`: classe que evalua el paso de los mensajes que se cumpla con el protocolo si no se cumple lo rechaza
- Se usa openssl para generar los certificados para usar el protocolo TLS

Instrucciones para ejecutar el programa

1. Priemro para almacenar los mensajes y para cargarlos se ocupa un lugar donde hacerlo: podemos crea un directorio para eso o simplemente pasarlo como parametro en el [smtpservidor.py](#)
2. Para correr el [smtpserver.py](#), (Todo se corre en terminal de linux)

```
# python smtpserver.py -d <dominio que va a aceptar el se  
rver> -s <path del almacenamiento> -p <puerto>
```

```
# puede usar la opcion: python smtpserver.py --help
$ python smtpserver.py -d izthero.com -p 2500 -s /tmp/mail/
```

- Para generar el certificado

```
# se usa password como ejemplo
$ openssl genrsa -aes256 -passout pass:password -out server.key 2048
$ openssl req -new -key server.key -passin pass:password -out server.csr
$ openssl x509 -req -passin pass:password -days 1024 -in server.csr -signkey server.key -out server.crt
$ openssl rsa -in server.key -out server_no_pass.key -passin pass:password
$ mv server_no_pass.key server.key
$ cat server.crt server.key > server.pem
```

3. Para correr el `smtpclient.py` (Nota: el servidor `smtpserver.py` debe estar corriendo)

```
# python smtpclient.py -c <archivo.csv> -e <correo del From:> -n <'Nombre del From:'> -h <host_servidor> -p <puerto_servidor> -s <"Subject"> -m "archivo con el cuerpo del mensaje"
# python smtpclient.py --help
$ python smtpclient.py -c mailTo.csv -e jeffmy@izthero.com -n 'Jeffrey Ricardo Alfaro Fonseca' -h localhost -p 2500
```

```
-s "Message Test" -m bodymessage
```

- Ejemplo de un .csv

```
nombreTest1,Test1@hotmail.com
nombreTest2,Test2@gmail.com
nombreTest3,Test3@gmail.com
nombreTest4,Test4@example.com
nombreTest5,Test5@izthero.com
nombreTest6,Test6@localhost.com
```

- En la carpeta MIMEATTACH se encuentra otro cliente que maneja mensajes MIME para adjuntar un archivo la diferencia de este es que tiene otro argumento [-a <nombre archivo para adjuntar>]

```
# python smtpclient.py -c <archivo.csv> -e <correo del From:
m:> -n <'Nombre del From:'> -h <host_servidor> -p <puerto_
servidor> -s <"Subject"> -m "archivo con el cuerpo del men
saje" -a <nombre archivo para adjuntar>
# python smtpclient.py --help
$ python smtpclient.py -c mailTo.csv -e jeffmy@izthero.com
-n 'Jeffrey Ricardo Alfaro Fonseca' -h localhost -p 2500
-s "Message Test" -m bodymessage -a bodymessage
```

4. Para correr el [imapserver.py](#), primero hay que crear una archivo de texto llamado 'passwords.txt' dentro del lugar donde se almacenan los mensajes, este archivo contiene el acceso de

control de passwords y users de cada usuario al que le aya llegado el correo ahi debemos manualmente meter los los correos y passwords en formato "email:password"

```
# Ejemplo en el caso que los el mail storage sea /tmp/mail/  
/  
$ cd /tmp/mail/  
$ nano passwords.txt
```

- Ingresamos y Guardamos

```
Test6@localhost.com:12345
```

- Para correr el `imapserver.py`

```
# python imapserver.py -s <lugar donde se encuentra el storage> -p <puerto>  
$ python imapserver.py -s /tmp/mail/ -p 1430
```

Actividades realizadas:

19 - 22 de Febrero: investigar y leer la docu de twisted, manejo de classes, como funciona crear protocolos en twisted, como funciona el reactor para correr el api de twisted en python con un promedio de 4.75 horas por dia(4 dias)

24 - 26 Febrero: Crear El SMTPserver y el cliente: se implementa la funcionalidad de los argumentos las clases para obtener los mensajes , validar dominios y enviar mensajes promedio de 7 horas

por dia(3 dias)

27 - 28 Febrero: Se implementa el servidor imap y se adiere el protocolo TLS al servidoresmtp y cliente: 10.5 horas por dia(2 dias)

1 de Marzo: Se establece el protocolo TLS al imapserver se termina la docu establecida para la tarea

Estado del programa

- No se implementa nntp
- No se envia mensajes del servidor smtpserver a otros servidores, solamente local
- Se implementa el attachment de archivos pero por separado al cliente aunque son lo mismo solo que uno tiene un argumento mas que el otro en el paso de parametros para indicar el archivo a adjuntar
- Problema con thunderbird que si se autentica al servidor imap pero no carga los mensajes

7. Conclusiones y Recomendaciones del proyecto: Debe mostrar las lecciones aprendidas y recomendaciones para un posible futuro estudiante el presente curso.

Referencias que se usaron principalmente

- [Templates] <https://docs.python.org/2.4/lib/node109.html>

- [Twisted] <https://twistedmatrix.com/documents/8.1.0/api/>
- [Twisted] https://zodml.org/sites/default/files/Twisted_Network_Programming.pdf
- [Twisted] <https://media.readthedocs.org/pdf/twisted/latest/twisted.pdf>