

# Comprobar una conexión SMTP autenticada en el shell con SSL o TLS

## Introducción

Muchas veces el soporte técnico o nosotros, tenemos que comprobar si los datos que tenemos y si el servidor **SMTP remoto**, están operando correctamente. No es necesario hacer como hacen algunos un cambalache creando una cuenta en el programa de correo electrónico, sino que como casi siempre podemos acudir a nuestra shell para realizar las pruebas pertinentes.

## Comprobar la autenticación SMTP y la conexión SSL usando la línea de comandos o shell.

### Autenticación SMTP

La autenticación (autenticación) es el mecanismo por el cual un usuario se identifica a sí mismo en un **servicio** de un servidor. En este caso el servicio es el **correo electrónico saliente** o **SMTP** y es necesario para que podamos **enviar** correo electrónico.

### Preparación, prueba y verificación

Para hacer la prueba es necesario tener instalado el paquete **openssl** de nuestro ordenador.

### Crear la cadena de autenticación para una login basado en PLAIN

Generalmente los servidores de correo electrónico, usan como medio de autenticación uno denominado **\*\*PLAIN\*\***, que consiste en pasar un texto plano (ASCII) que contiene el par **\*\*usuario + contraseña\*\***

Antes de realizar la prueba debemos obtener la cadena de caracteres ASCII que contiene el par usuario\_smtp + contraseña.

## Usando Bash

En el momento de escribir esto, el tip que tenía en mi entrada original [Cómo comprobar la autenticación SMTP SMTP Auth y la conexión con StartTLS en el shell](#) me da error. La verdad es que he comprobado si había un error en mi escritura, y revisado con otros colegas. Así que he optado por no hacer el comando en una línea sino dividirlo en dos que si me funciona

```
$ echo -ne usuario@servidor.smtp.com | base64
emFiYml4QQDlbnRyYWwuY2FzdHJpcy5jb20=
$ echo -ne 4Mmr8Hop3FsmQvKtb8Ei | base64
NEltcjVUb3BhRnNtUXZldGI4RWk=
```

Ahora conectamos vía openssl

```
h y p son variables de entorno para poder trabajar más fácilmente
```

El puerto deberá ser el apropiado a la conexión, en este caso **startssl**

```
$ h=servidor.smtp.com
$ p=455
$ openssl s_client -connect $h:$p -starttls smtp
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R3
verify return:1
depth=0 CN = servidor.smtp.com
verify return:1
---
Certificate chain
 0 s:CN = servidor.smtp.com
...
    Extended master secret: no
    Max Early Data: 0
...
read R BLOCK
```

Esto ya nos indica que el servidor está activo, escuchando en el puerto solicitado, y admitiendo la conexión vía startssl

Ahora podemos usar `EHL0 there` para obtener los comandos disponibles

```
EHL0 there
250-servidor.smtp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
```

O pasar directamente a la autenticación

```
AUTH LOGIN
334 VXNlcm5hbWU6
emFiYml4Q0QlbnRyYWwuY2FzdHJpcy5jb20=
334 UGFzc3dvcmQ6
NEltcjVUb3BhRnNtUXZldGI4RWk=
235 2.7.0 Authentication successful
```

## Usando Perl

Si el usuario contiene la @ esta deberá escaparse con la barra invertida (\) de otra manera perl interpretará un arreglo (array) en lugar de una cadena (string)

Con Perl no tengo problemas para hacer lo mismo pero en lugar de usar AUTH LOGIN usar **AUTH PLAIN** usando la única cadena codificada del par usuario y contraseña

```
$ perl -MMIME::Base64 -e 'print encode_base64("\000usuario\@servidor.remoto.tld\000PaSsW0rD")'
AHphYmJpeEBjZW50cmFsLkTgqW3RyaXMuY29tADRNbXI1VG9wM0ZzbVF2S3Ri0EVp
$ h=servidor.smtp.com
$ p=455
$ openssl s_client -connect $h:$p -starttls smtp
...
---
```

```
read R BLOCK
```

```
AUTH PLAIN AHphYmJpeEBjZW50cmFsLkTgqW3RyaXMuY29tADRNbXI1VG9wM0ZzbVF2S3Ri0EVp
```

```
235 2.7.0 Authentication successful
```

# Enlaces relacionados

- [Verificar versiones y ciphers soportados de TLS, SSL](#)
- [Test SMTP with telnet or openssl](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

---

Revision #1

Created 1 September 2021 07:27:48 by Abkrim

Updated 1 September 2021 08:28:20 by Abkrim