



PR4 Entrega redes

Redes y aplicaciones Internet (Universitat Oberta de Catalunya)



Escanea para abrir en Studocu

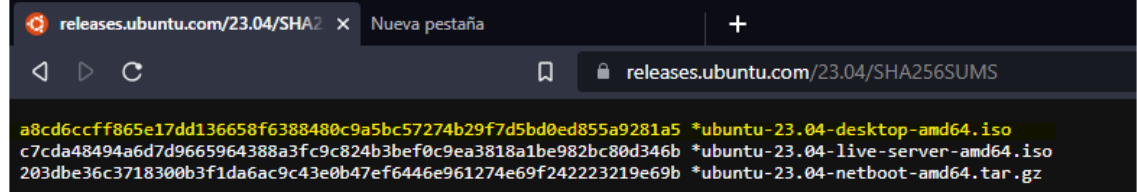
Contents

Ejercicio 1	2
Ejercicio 2	3
Ejercicio 3. Generar claves	5
Ejercicio 4. Enviar clave pública	7
Ejercicio 5. Importar claves públicas	7
Ejercicio 6. Desenscriptar el fichero que os han enviado	8
Ejercicio 7. Otras opciones de GPG	9
Ejercicio 8. Configuración de HTTPS.....	10
Ejercicio 9	12
Ejercicio 10	13

Ejercicio 1

- a) Descarga algún fichero que en la web tenga su SHA disponible. Mediante el comando adecuado en Ubuntu comprueba que el SHA del fichero descargado coincide con el que aparece en la página. Incluye una captura de pantalla donde se vea que coinciden.

```
[05/06/2023] gmartinez98@10:14:02 >sudo sha256sum ubuntu-23.04-desktop-amd64.iso
a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5  ubuntu-23.04-desktop-amd64.iso
```



Como se puede observar coincide.

- b) Busca información sobre las diferentes funciones que forman la familia SHA. ¿Cuál es la longitud del resumen que produce cada una?

Las diferentes funciones son:

- SHA-0: Salida de resumen de 160 bits.
- SHA-1: Salida de resumen de 160 bits.
- SHA-2:
 - SHA-224: Salida de resumen de 224 bits
 - SHA-256: Salida de resumen de 256 bits
 - SHA-384: Salida de resumen de 384 bits
 - SHA-512: Salida de resumen de 512 bits
- SHA-3:
 - SHA3-224: Salida de resumen de 224 bits
 - SHA3-256: Salida de resumen de 256 bits
 - SHA3-384: Salida de resumen de 384 bits
 - SHA3-512: Salida de resumen de 512 bits

- c) ¿Qué problema hay asociado al uso de SHA-0? Y SHA-1?

Tanto SHA-0 como SHA-1 tienen una vulnerabilidad creada por colisiones, donde es posible encontrar dos mensajes diferentes que produzcan el mismo hash, esto puede afectar a que es fácil de falsificar documentos.

En resumen, ni SHA-0 ni SHA-1 son seguros.

- d) Crea un fichero de texto y calcula un hash con alguna función de la familia SHA-2. Modifica el fichero y comprueba que el resumen no coincide (captura de pantalla)

```
[05/06/2023] gmartinez98@10:30:14 >cat doc.txt
Me llamo Guillem y mi usuario es gmartinez98

[05/06/2023] gmartinez98@10:30:17 >shasum doc.txt
8535bf53ad253da6172b6a619819905a4b9c6eb4 doc.txt

[05/06/2023] gmartinez98@10:31:01 >cat doc.txt
Me llamo Guillem y mi usuario es gmartinez98 modificado

[05/06/2023] gmartinez98@10:31:03 >shasum doc.txt
84a03a0d95e3742aaddeefd4583ceaa1c023b900 doc.txt
```

- e) ¿Es posible a partir del hash de un texto obtener el texto original?
No es posible dado que la función hash es resistente a colisiones.

Ejercicio 2

- a) En la criptografía simétrica toda la seguridad recae en la clave, por lo que ésta debe ser muy difícil de romper. Cita dos características que hacen que una clave sea difícil de romper.

Una de las características es la longitud dado que cuantos mas bits tenga la información en la clave, el número de combinaciones que debería probar el atacante aumenta significativamente.

Por otro lado, tendríamos el tiempo de vida de una clave, cuanto más tiempo de vida tenga una clave más insegura se vuelve, por lo que tendríamos que ir actualizándola con cierta periodicidad.

- b) Cita tres algoritmos de cifrado simétrico y busca en la red qué longitud de clave tienen. AES:

- AES-128: Longitud de clave 128 bits.
- AES-192: Longitud de clave 192 bits.
- AES-256: Longitud de clave 256 bits.

TWOFISH:

- Se pueden encontrar diferentes longitudes de clave de 128bits, 192bits y 256 bits.

SERPENT:

- Serpent permite diferentes longitudes de clave de 128bits, 192bits, 256bits.

- c) Utiliza el comando *gpg* para averiguar qué algoritmos de cifrado simétrico soporta.

```
gmartinez98@10:45:45 >
[05/06/2023] gmartinez98@10:45:47 >gpg --version
gpg (GnuPG) 2.2.27
libgcrypt 1.9.4
Copyright (C) 2021 Free Software Foundation, Inc.
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/gmartinez98/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

- d) Utiliza *man gpg* para indicar con qué opción se puede cifrar de forma simétrica. Crea un fichero de texto con algún contenido y cifralo. Intenta verlo con *cat* o *more*.

```
--symmetric
-c      Encrypt with a symmetric cipher using a passphrase. The default symmetric cipher
used is AES-128, but may be chosen with the --cipher-algo option. This command may
be combined with --sign (for a signed and symmetrically encrypted message), --en-
crypt (for a message that may be decrypted via a secret key or a passphrase), or
--sign and --encrypt together (for a signed message that may be decrypted via a se-
cret key or a passphrase). gpg caches the passphrase used for symmetric encryption
so that a decrypt operation may not require that the user needs to enter the
passphrase. The option --no-symkey-cache can be used to disable this feature.
```

```
gmartinez98@10:49:49 >cat doc.txt
Me llamo Guillem y mi usuario es gmartinez98 modificado

[05/06/2023] gmartinez98@10:49:56 >gpg -c doc.txt
```

```
gmartinez98@10:49:49 >cat doc.txt
Me llamo Guillem y mi usuario es gmartinez98 modificado

[05/06/2023] gmartinez98@10:49:56 >cat doc.txt.gpg
♦          ♦♦
          ♦j♦♦vj♦♦M♦♦O♦♦R♦♦♦♦le♦):u♦]♦♦,♦(q♦B♦\♦♦♦♦
♦t♦♦BP♦2♦U♦♦ ~♦8♦=♦H♦m8 P♦♦a♦♦♦♦`X;Z♦♦♦[05/06/2023] gmartinez98@10:50:18 >
```

- e) ¿Con qué opción puedes descifrar el fichero? Descifralo.

```
--decrypt
-d      Decrypt the file given on the command line (or STDIN if no file is specified) and
write it to STDOUT (or the file specified with --output). If the decrypted file is
signed, the signature is also verified. This command differs from the default oper-
ation, as it never writes to the filename which is included in the file and it re-
jects files that don't begin with an encrypted message.
```

```
[05/06/2023] gmartinez98@10:52:04 >cat doc.txt.gpg
♦          ♦♦
          ♦j♦♦vj♦♦M♦♦O♦♦R♦♦♦♦le♦):u♦]♦♦,♦(q♦B♦\♦♦♦♦
♦t♦♦BP♦2♦U♦♦ ~♦8♦=♦H♦m8 P♦♦a♦♦♦♦`X;Z♦♦♦[05/06/2023] gmartinez98@10:52:17 >
[05/06/2023] gmartinez98@10:52:19 >gpg -d doc.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
Me llamo Guillem y mi usuario es gmartinez98 modificado
```

- f) ¿Qué algoritmo por defecto usa GPG? Utiliza la opción adecuada para cifrar un fichero de texto con el algoritmo de cifrado CAMELLIA256.

Como podemos ver utiliza por defecto AES256

```
[05/06/2023] gmartinez98@10:56:14 >gpg --cipher-algo CAMELLIA256 -c doc.txt
```

```
gmartinez98@10:56:09 >gpg -d doc.txt.gpg
gpg: CAMELLIA256.CFB encrypted data
gpg: encrypted with 1 passphrase
Me llamo Guillem y mi usuario es gmartinez98 modificado
```

Ejercicio 3. Generar claves

- a) Comprueba mediante el comando gpg si tienes alguna clave pública/privada

```
[05/06/2023] gmartinez98@10:57:39 >gpg --list-public-keys
[05/06/2023] gmartinez98@10:57:43 >gpg --list-key
```

- b) Genera un par de claves con el comando gpg.

```
[05/06/2023] gmartinez98@10:58:39 >gpg --full-generate-key

[05/06/2023] gmartinez98@11:00:21 >gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
Requested keysize is 3072 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Guillem Martinez Paredes
Email address: gmartinez98@uoc.edu
Comment: Clave de pruebas
You selected this USER-ID:
    "Guillem Martinez Paredes (Clave de pruebas) <gmartinez98@uoc.edu>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? _

^[[BWe need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
^[[Agpg: key 3EDA453C5CDBB3DF marked as ultimately trusted
gpg: directory '/home/gmartinez98/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/gmartinez98/.gnupg/openpgp-revocs.d/09AB2A2AA2289145FD011F413EDA453C5CDBB3DF.rev'
public and secret key created and signed.

pub   rsa3072 2023-06-05 [SC]
      09AB2A2AA2289145FD011F413EDA453C5CDBB3DF
uid     Guillem Martinez Paredes (Claves de pruebas) <gmartinez98@uoc.edu>
sub   rsa3072 2023-06-05 [E]
```

c) Comprueba que se han generado.

```
gmartinez98@11:02:26 >gpg --list-public-keys  
/home/gmartinez98/.gnupg/pubring.kbx  
-----  
pub   rsa3072 2023-06-05 [SC]  
      09AB2A2AA2289145FD011F413EDA453C5CDBB3DF  
uid    [ultimate] Guillem Martinez Paredes (Claves de pruebas) <gmartinez98@uoc.edu>  
sub   rsa3072 2023-06-05 [E]  
  
[05/06/2023] gmartinez98@01:10:26 >gpg --list-keys  
/home/gmartinez98/.gnupg/pubring.kbx  
-----  
pub   rsa3072 2023-06-05 [SC]  
      09AB2A2AA2289145FD011F413EDA453C5CDBB3DF  
uid    [ultimate] Guillem Martinez Paredes (Claves de pruebas) <gmartinez98@uoc.edu>  
sub   rsa3072 2023-06-05 [E]  
  
[05/06/2023] gmartinez98@01:10:28 >
```

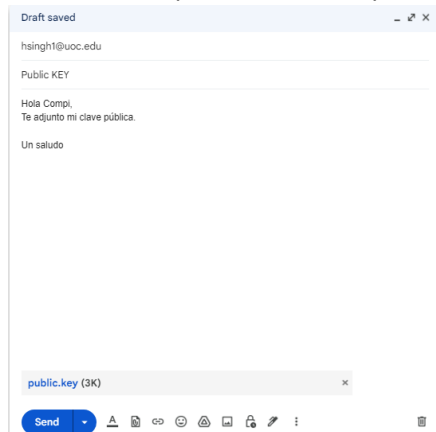
Ejercicio 4. Enviar clave pública

Ahora vas a enviar la clave pública a otro compañero por correo-e. Lo más frecuente es subir la clave pública a un servidor de claves (<https://www.rediris.es/keyserver/index.html.es>) pero nosotros lo haremos así para simplificar: Ponte de acuerdo con algún compañero del aula a través del foro.

- a) Exporta tu clave pública a un fichero.

```
gmartinez98@gmartinez98:/mnt/shared$ gpg --export -a gmartinez98 > public.key_
```

- b) Envía la clave pública a tu compañero por correo.



Ejercicio 5. Importar claves públicas

- a) Importa la clave pública de tu compañero.

```
gmartinez98@gmartinez98:/mnt/shared$ gpg --import hsingh_public.key
gpg: key FD23316318EB30BE: public key "Harmandeep Singh (Test 1) <hsingh1@uoc.edu>" imported
gpg: Total number processed: 1
gpg:         imported: 1
```

- b) Comprueba que se ha importado la clave a tu anillo de claves.

```
gmartinez98@gmartinez98:/mnt/shared$ gpg --list-key
/home/gmartinez98/.gnupg/pubring.kbx
-----
pub   rsa3072 2023-06-06 [SC]
      EED83268F296A1580DD853EC7DA907DC6B5CC40F
uid           [ultimate] Guillem Martinez Paredes (Clave de pruebas) <gmartinez98@uoc.edu>
sub   rsa3072 2023-06-06 [E]

pub   rsa3072 2023-06-03 [SC]
      63758DD5091FF152BC738D28FD23316318EB30BE
uid           [ unknown] Harmandeep Singh (Test 1) <hsingh1@uoc.edu>
sub   rsa3072 2023-06-03 [E]
```

- c) Crea un fichero que se llame *fichero_rai_ejer_5.txt* y encriptalo con la clave que acabas de importar.

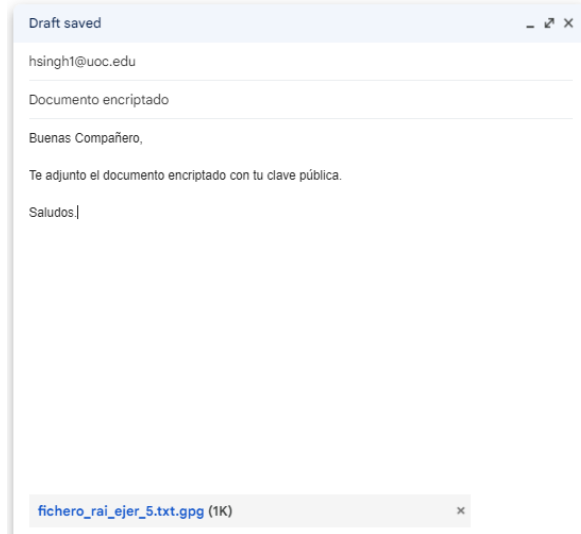
```
gmartinez98@gmartinez98:/mnt/shared$ gpg --encrypt --recipient hsingh1@uoc.edu fichero_rai_ejer_5.txt
gpg: 977C11FA6FA7422E: There is no assurance this key belongs to the named user

sub   rsa3072/977C11FA6FA7422E 2023-06-03 Harmandeep Singh (Test 1) <hsingh1@uoc.edu>
      Primary key fingerprint: 6375 8DD5 091F F152 BC73  8D28 FD23 3163 18EB 30BE
      Subkey fingerprint: 674A A483 E1F6 83CD 2B2F  1FC5 977C 11FA 6FA7 422E

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```


- d) Envía por correo-e el fichero a tu compañero encriptado con su clave pública.



Ejercicio 6. Desencriptar el fichero que os han enviado

- a) Comprueba que no puedes ver el contenido del fichero que te han enviado

```
gmartinez98@martinez98:/mnt/shared$ cat hsingh_fichero_rai_ejer_5.txt.gpg
***$gX
      NXK6uH***q**z] *x0**8*,***|**P**G**[J**>***o **/u**R**
                                q*vo**u**D*_**N8**7w**w***' '[**
*3v*I***j***w*n*}**V**b|*ÜR**<{*p+Swo*I**uc*t0*5
JK*Q**_Q*L*C*2*!>p*(**3t***E**a*****W*1**Ue'5*I:J*j*+=.9**$o>*c<PD(■@êi:Ñ*~K÷éc*ó@i!²! oàHGfUf+
;Nds~%â³iMçñc7dô"cuûââ5pG[+iAVô7BæcbU+>`êwþ17#ê0/ÆJx0·²;|1ô:0«iAO
U-i- To||A≥sYiê6âæR0%./"%×ÛL:bJ7+%%$70%#
UIGêbS0ûêUVJx#â)0·b+B`ôÊIµbâV***e***$êWÍYpd`Hx²dF-Ü±±ÜWô±L||r|i+e≥98@±||r|i+e≥98:/4|/s||red$ _
```

- b) Desencripta el fichero

```
gmartinez98@martinez98:/mnt/shared$ gpg -d hsingh_fichero_rai_ejer_5.txt.gpg
gpg: gpg (options): 'gpg' (filename)
gmartinez98@martinez98:/mnt/shared$ gpg --output hsingh_fichero_rai_ejer_5.txt -d hsingh_fichero_rai_ejer_5.txt.gpg
gpg: encrypted with 3072-bit RSA key, ID CA1BB71C24FC6758, created 2023-06-06
"Guillem Martínez Paredes (Clave de pruebas) <gmartinez98@uoc.edu>"
gmartinez98@martinez98:/mnt/shared$ cat fichero_rai_ejer_5.txt
gmartinez98@martinez98:/mnt/shared$ cat fichero_rai_ejer_5.txt
gmartinez98@martinez98:/mnt/shared$ cat hsingh_fichero_rai_ejer_5.txt
Encriptado con la clave del alumno Guillem Martínez Paredes
```

- c) Comprueba que ahora puedes ver el contenido del fichero que te han enviado encriptado con tu clave pública.

```
gmartinez98@martinez98:/mnt/shared$ gpg --output hsingh_fichero_rai_ejer_5.txt -d hsingh_fichero_rai_ejer_5.txt.gpg
gpg: encrypted with 3072-bit RSA key, ID CA1BB71C24FC6758, created 2023-06-06
"Guillem Martínez Paredes (Clave de pruebas) <gmartinez98@uoc.edu>"
gmartinez98@martinez98:/mnt/shared$ cat fichero_rai_ejer_5.txt
gmartinez98@martinez98:/mnt/shared$ cat fichero_rai_ejer_5.txt
gmartinez98@martinez98:/mnt/shared$ cat hsingh_fichero_rai_ejer_5.txt
Encriptado con la clave del alumno Guillem Martínez Paredes
```

Ejercicio 7. Otras opciones de GPG

- a) Indica qué comando usarías para borrar una clave de tu anillo de claves

Según el manual utilizaría uno de los siguientes parámetros:

```
--delete-keys name
  Remove key from the public keyring. In batch mode either --yes is required or the
  key must be specified by fingerprint. This is a safeguard against accidental dele-
  tion of multiple keys. If the exclamation mark syntax is used with the fingerprint
  of a subkey only that subkey is deleted; if the exclamation mark is used with the
  fingerprint of the primary key the entire public key is deleted.

--delete-secret-keys name
  Remove key from the secret keyring. In batch mode the key must be specified by fin-
  gerprint. The option --yes can be used to advise gpg-agent not to request a con-
  firmation. This extra pre-caution is done because gpg can't be sure that the se-
  cret key (as controlled by gpg-agent) is only used for the given OpenPGP public
  key. If the exclamation mark syntax is used with the fingerprint of a subkey only
  the secret part of that subkey is deleted; if the exclamation mark is used with the
  fingerprint of the primary key only the secret part of the primary key is deleted.

--delete-secret-and-public-key name
  Same as --delete-key, but if a secret key exists, it will be removed first. In
  batch mode the key must be specified by fingerprint. The option --yes can be used
  to advise gpg-agent not to request a confirmation.
```

- b) Indica cómo firmarías un mensaje, sin encriptarlo.

Para firmar un mensaje sin encriptarlo utilizaríamos el siguiente comando:

`gpg --detach-sign -u ID_clave -o documento.txt.sig documento.txt`

```
gmartinez98@gmartinez98:/mnt/shared$ gpg --detach-sign -u gmartinez98 -o fichero_rai_ejer_5.txt.sig
fichero_rai_ejer_5.txt
```

Ejercicio 8. Configuración de HTTPS

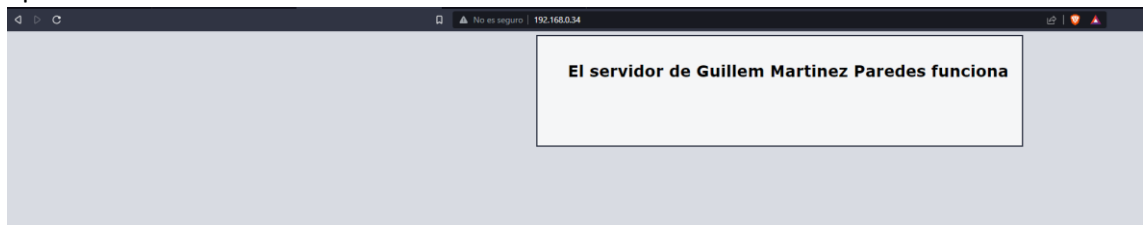
Apache ya trae un certificado autofirmado que podemos usar en nuestros sitios web, así como un fichero de configuración para SSL llamado default-ssl.

- a) Instala Apache mediante apt y comprueba que funciona por HTTP.

```
gmartinez98@01:12:56 >sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 ya está en su versión más reciente (2.4.52-1ubuntu4.5).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 57 no actualizados.
[05/06/2023] gmartinez98@01:13:00 >sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
```



- b) Cambia la página por defecto para que muestre “El servidor de <tu nombre y apellidos> funciona”



- c) Localiza en el fichero `/etc/apache2/sites-available/default-ssl` las directivas que tienen que ver con SSL y explica su significado

```
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

SSLEngine habilita SSL/TLS

SSL CertificateFile: Documento de certificado.

SSL CertificateKeyFile: Documento de claves generadas al crear la CSR.

- d) Habilita HTTPs tecleando:

- `sudo a2ensite default-ssl`

```
gmartinez98@gmartinez98:/var/www/html$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
gmartinez98@gmartinez98:/var/www/html$ systemctl reload apache2.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Authenticating as: Guillem (gmartinez98)
Password:
==== AUTHENTICATION COMPLETE ====
```

- e) Comprueba mediante la orden `netstat -ntl` que HTTPs está habilitado

```
gmartinez98@gmartinez98:/var/www/html$ netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53:53         0.0.0.0:*               LISTEN
tcp6       0      0 :::443                  :::*                     LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
tcp6       0      0 :::80                    :::*                     LISTEN
```

Ejercicio 9

Conéctate a tu servidor utilizando la siguiente orden:

- `openssl s_client -connect 127.0.0.1:443`

```
gmartinez98@gmartinez98:/var/www/html$ sudo openssl s_client -connect 127.0.0.1:443
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol    : TLSv1.3
    Cipher      : TLS_AES_256_GCM_SHA384
    Session-ID: 8E2B6B85EA253CDC5C43746D2F73915469FEDE8E396DD8BB57600E1427FE4B24
    Session-ID-ctx:
    Resumption PSK: 8D3511B91AAE853BC032DEF71F7D11B2512F05E100D094E38D791EFC314CE7E0DACFBE6877147547
    62AFE42B223DA950
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - d9 e9 07 22 48 20 0c d1-47 14 bd 20 82 24 a3 9a    ..."H ..G.. .$..
    0010 - cf a2 d4 a6 90 69 72 40-35 c6 bc ba a6 e0 ee ad    ....ir@5.....
    0020 - f2 85 cf d9 63 ac bf d1-2b a4 68 8d 3c f4 c3 d9    ....c...+.h.<...
    0030 - 72 ea 1b 3b f5 c8 98 ae-8d 2d 37 fc 3b c1 0c b1    r...;.....-7.;...
    0040 - 83 19 06 d9 b1 2e bc 6d-61 d4 d2 ef fe f4 71 39    .....ma.....q9
    0050 - 4e 40 10 fd 3e 6f a6 e9-59 98 a5 93 44 32 06 42    N@...>o..Y...D2.B
    0060 - d8 63 7a 20 41 c9 e2 a9-ab 1c 29 4c 70 80 c6 47    .cz A.....)Lp..G
    0070 - 0e e9 ef b4 fd 49 7b 3a-03 6d b5 c5 a4 84 a6 a7    .....I{:m.....
    0080 - 39 9b b0 f4 04 d5 20 82-91 a9 55 a2 af 92 f7 8e    9.....U.....
    0090 - 5d b1 80 bd 8c 70 49 1d-08 a7 d1 a2 7f dd 0f f4    ]....pI.....
    00a0 - a3 91 a9 35 a0 c5 99 10-38 76 6f 35 ed 16 bd 69    ...5....8vo5...i
    00b0 - 6e 0e 77 51 fb d4 a5 9e-ad 08 c8 3e 95 1c 27 a1    n.wQ.....>..'
    00c0 - d4 44 92 e3 0e ae 77 a2-cb cc 0b 6f f4 5e 42 cd    .D....w....o.^B.
    00d0 - 74 08 3b 1d 9b cd e0 02-59 39 e1 38 9a b9 b5 67    t.;.....Y9.8...g
    00e0 - 9a 44 88 43 c1 ab 7b f7-b4 ec 2d 05 5f 95 e4 15    .D.C...{-.-....

    Start Time: 1685981510
    Timeout    : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

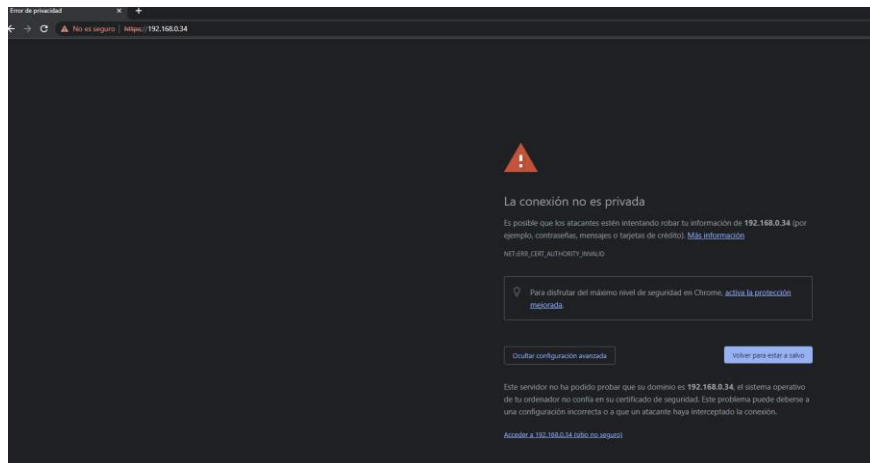
Explica en detalle qué significado tienen los campos siguientes dentro de SSL-Session:

- Protocolo: Indica el protocolo utilizado en este caso TLSv1.3
- Cipher: Indica el cifrado que utiliza el cliente en este caso TLS_AES_256_GCM_SHA384.
- Session-ID: Identifica la session actual.
- TLS session ticket: Hace referencia al estado de la sesión que se encripta en el cliente.

Ejercicio 10

Realiza una conexión SSL con tu navegador contra tu servidor mientras capturas el tráfico con el Wireshark (utiliza el modo puente del VirtualBox):

- a) Pega una captura de pantalla donde se vea la advertencia de tu navegador web al conectarse a un servidor con un certificado autofirmado.



- b) Explica qué algoritmos se han usado en la conexión
- c) Identifica y explica los tres pasos del SSL handshake en la captura