Configuración da seguridade no servidor Web

Prácticas

Índice

1. Control de acceso en Linux

Configurar o servidor web en Linux para permitir ou denegar o acceso a distintos directorios en función da IP desde a que se trate de acceder.

- Crea o directorio `/var/www/html/proba1` e crea nel un ficheiro HTML co contido que queiras.
- Crea o directorio `/var/www/html/proba2` e crea nel un ficheiro HTML co contido que queiras.
- Configura o control de acceso para o directorio `proba1` de forma que só poida acceder a el unha máquina en concreto.
- Configura o control de acceso para o directorio `proba2` de forma que só poida acceder a el unha máquina en concreto.
- Conecta o servidor Apache en Ubuntu en modo ponte (mesma configuración de rede que a da túa máquina real) de forma que sexa visible para todas as máquinas reais da aula e configura o control de acceso para que só poidan acceder dúas máquinas da túa elección a `proba1` e outras dúas a `proba2`.

Pasos a seguir

- Crea o directorio `/var/www/html/proba1` e crea nel un ficheiro HTML co contido que queiras.
- Crea o directorio `/var/www/html/proba2` e crea nel un ficheiro HTML co contido que queiras.

```
sudo cd /var/www/html
sudo mkdir proba1 proba2
sudo gedit proba1/index.html
sudo gedit proba2/index.html
```

O contido dos ficheiros pode ser similar a:

```
<html>
   <body>
      <h1>index.html en /var/www/html/proba1</h1>
   </body>
</html>
```

- Configura o control de acceso para o directorio `proba1` de forma que só poida acceder a el unha máquina en concreto, por exemplo, a máquina cliente e non a máquina Windows Server.

Como indicamos en actividades anteriores, a IP da máquina cliente é `192.168.0.3` e a da máquina Windows Server é `192.168.0.2`. Si non é así, emprega as IP´s axeitadas.

Editamos o ficheiro `/etc/apache2/sites-available/000-default.conf`:

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

E engadimos o seguinte bloque:

```
<Directory /var/www/html/proba1>
   Require  ip 192.168.0.3
</Directory>
```

- Configura o control de acceso para o directorio `proba2` de forma que só poida acceder a el unha máquina en concreto, por exemplo, a máquina Windows Server e non a máquina cliente.

  Editamos o ficheiro `/etc/apache2/sites-available/000-default.conf`:

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

  E engadimos o seguinte bloque:

```
<Directory /var/www/html/proba2>
   Require  ip 192.168.0.2
</Directory>
```

- Conecta o servidor Apache en Ubuntu en modo ponte (mesma configuración de rede que a da túa máquina real pero sumar 100 ao último byte do teu IP) de forma que sexa visible para todas as máquinas reais da aula e configura o control de acceso para que só poidan acceder dúas máquinas da túa elección a `proba1` e outras dúas a `proba2`.

A configuración é a mesma que a anterior pero indicando as dúas IP correspondentes, por exemplo:

```
<Directory /var/www/html/proba2>
   Require  ip 192.168.1.102 192.168.1.103
</Directory>
```

2. Control de acceso en Windows

Configurar servidor web en Windows para permitir ou denegar o acceso a distintos directorios en función da IP desde a que se trate de acceder.

- Crea o directorio `C:\Apache24\htdocs\proba1` e crea nel un ficheiro HTML co contido que queiras.
- Crea o directorio `C:\Apache24\htdocs\proba2` e crea nel un ficheiro HTML co contido que queiras.
- Configura o control de acceso para o directorio `proba1` de forma que só poida acceder a el a máquina cliente e non a máquina servidor Ubuntu.
- Configura o control de acceso para o directorio `proba2` de forma que só poida acceder a el a máquina servidor Ubuntu e non a máquina cliente.
- En ambos os casos, proba a listar o directorio raíz do servidor, que sucede?
- Conecta o servidor Apache en Windows en modo ponte (mesma configuración de rede que a da túa máquina real pero sumar 100 ao último byte do teu IP) de forma que sexa visible para todas as máquinas reais da aula e configura o control de acceso para que só poidan acceder dúas máquinas da túa elección a `proba1` e outras dúas a `proba2`.

Pasos a seguir

- Crea o directorio `C:\Apache24\htdocs\proba1` e crea nel un ficheiro HTML co contido que queiras.
- Crea o directorio `C:\Apache24\htdocs\proba2` e crea nel un ficheiro HTML co contido que queiras.

Creamos os directorios desde o navegador de arquivos e creamos os ficheiros co seguinte contido, por exemplo:

```
<html>
   <body>
      <h1>index.html en C:\Apache24\htdocs\proba1</h1>
   </body>
</html>
```

- Configura o control de acceso para o directorio `proba1` de forma que só poida acceder a el a máquina cliente e non a máquina servidor Ubuntu.

  Editamos o ficheiro `C:\Apache24\conf\httpd.conf` e engadimos o seguinte bloque:

```
<Directory C:\Apache24\htdocs\proba1>
   Require  ip 192.168.0.3
</Directory>
```

- Configura o control de acceso para o directorio `proba2` de forma que só poida acceder a el a máquina servidor Ubuntu e non a máquina cliente.

  Editamos o ficheiro `C:\Apache24\conf\httpd.conf` e engadimos o seguinte bloque:

```
<Directory C:\Apache24\htdocs\proba2>
   Require  ip 192.168.0.1
</Directory>
```

- En ambos os casos, proba a listar o directorio raíz do servidor, que sucede?

  Que o cartafol que está bloqueado para unha IP, non aparece listado.

- Conecta o servidor Apache en Windows en modo ponte (mesma configuración de rede que a da túa máquina real pero sumar 100 ao último byte do teu IP) de forma que sexa visible para todas as máquinas reais da aula e configura o control de acceso para que só poidan acceder dúas máquinas da túa elección a `proba1` e outras dúas a `proba2`.

A configuración é a mesma que a anterior pero indicando as dúas IP correspondentes, por exemplo:

```
<Directory C:\Apache24\htdocs\proba2>
   Require ip 192.168.1.102 192.168.1.103
</Directory>
```

3. Autenticación en Linux.

Configurar o servidor en Linux para restrinxir o acceso aos recursos a usuarios e grupos.

- Con autenticación HTTP Basic:
  - Crea o arquivo de contrasinais con dous usuarios: `profesor1` e `profesor2`.
  - Crea o grupo `profesores` que contén a `profesor1` e `profesor2`.
  - Crea o cartafol `/var/www/html/profesor` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:
    - Que poida acceder `profesor1`, pero non `profesor2`.
    - Que poidan acceder ambos profesores.
    - Que poidan acceder os membros do grupo `profesores`.
    - Que poida acceder calquera usuario válido.
- Con autenticación HTTP Digest:
  - Crea o arquivo de contrasinais con dous usuarios: `admin1` e `admin2`, pertencentes ao dominio `xestion`.
  - Crea o cartafol `/var/www/html/administradores` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:
    - Que poida acceder `admin1`, pero non `admin2`.
    - Que poidan acceder ambos administradores.
    - Que poida acceder calquera usuario válido.

Pasos a seguir

- Con autenticación HTTP Basic:
  - Crea o arquivo de contrasinais con dous usuarios: `profesor1` e `profesor2`.

Creamos o arquivos de contrasinais e o usuario `profesor1`, cando se nos solicite, introducimos o seu contrasinal. Empregamos –c no momento de crear o arquivo:

```
sudo htpasswd -c /etc/apache2/contr_basic profesor1
```

Creamos o usuario profesor2, cando se nos solicite, introducimos o seu contrasinal:

```
sudo htpasswd /etc/apache2/contr_basic profesor2
```

  - Crea o grupo `profesores` que contén a `profesor1` e `profesor2`.

Creamos un arquivo de grupos:

```
sudo gedit /etc/apache2/grupos
```

E editámolo co seguinte contido:

```
profesores: profesor1 profesor2
```

  - Crea o cartafol `/var/www/html/profesores` con algún arquivo HTML co contido que prefiras

```
sudo mkdir /var/www/html/profesor
sudo gedit /var/www/html/profesor/index.html
```

O contido do ficheiro pode ser similar a:

```
<html>
   <body>
      <h1>index.html en /var/www/html/profesores</h1>
   </body>
</html>
```

– e configura adecuadamente Apache de tres maneiras diferentes:

Editamos o ficheiro `/etc/apache2/sites-available/000-default.conf`:

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

– Que poida acceder `profesor1`, pero non `profesor2`.

Engadimos o seguinte bloque:

```
<Directory /var/www/html/profesores>
    AuthType Basic
    AuthName "Para invitados"
    AuthBasicProvider file
    AuthUserFile "/etc/apache2/contr_basic"
    Require user profesor1
 </Directory>
```

– Que poidan acceder ambos profesores.

Modificamos a directiva `Require` do seguinte xeito:

```
    Require user profesor1 profesor2
```

– Que poidan acceder os membros do grupo profesores.

Habilitamos o modulo `authz_groupfile` que non está habilitado por defecto:

```
sudo a2enmod authz_groupfile
```

Engadimos no bloque a liña:

```
AuthGroupFile "/etc/apache2/grupos"
```

E modificamos a directiva `Require` do seguinte xeito:

```
Require group profesores
```

– Que poida acceder calquera usuario válido.

Modificamos a directiva `Require` do seguinte xeito:

```
    Require valid-user
```

Para comprobar que a configuración se realizou correctamente, conectámonos desde o navegador da máquina cliente a `http://192.168.0.1/profesores` e debe solicitarse o usuario e contrasinal, no caso de que teñamos acceso, amosarase o arquivo creado, en caso contrario, aparecerá unha e outra vez a ventá de solicitude de usuario.

- Con autenticación HTTP Digest:

O primeiro que debemos facer é activar o módulo `auth_digest` que non está activado por defecto:

```
sudo a2enmod auth_digest
```

E reiniciamos o servidor:

```
sudo service apache2 restart
```

– Crea o arquivo de contrasinais con dous usuarios: `admin1` e `admin2`, pertencentes ao dominio `xestion`. Empregamos –c no momento de crear o arquivo.

Creamos o arquivo de contrasinais e o usuario `admin1`:

```
htdigest -c /etc/apache2/contr_digest xestion admin1
```

Cremos o usuario `admin2`:

```
htdigest /etc/apache2/contr_digest xestion admin2
```

– Crea o cartafol `/var/www/html/administradores` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:

Editamos o ficheiro `/etc/apache2/sites-available/000-default.conf`:

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

– Que poida acceder `admin1`, pero non `admin2`.

```
<Directory /var/www/html/profesores>
    AuthType Digest
    AuthName "xestion"
    AuthBasicProvider file
    AuthUserFile "/etc/apache2/contr_digest"
    Require user admin1
 </Directory>
```

– Que poidan acceder ambos administradores.

Modificamos a directiva `Require` do seguinte xeito:

```
Require user admin1 admin2
```

– Que poida acceder calquera usuario válido.

Modificamos a directiva `Require` do seguinte xeito:

```
Require valid-user
```

4. Autenticación en Windows

Configurar o servidor en Windows para restrinxir o acceso aos recursos a usuarios e grupos.

- Con autenticación HTTP Basic:
  - Crea o arquivo de contrasinais con dous usuarios: `profesor1` e `profesor2`.
  - Crea o grupo `profesores` que contén a `profesor1` e `profesor2`.
  - Crea o cartafol `C:\apache24\htdocs\profesor` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:
    - Que poida acceder `profesor1`, pero non `profesor2`.
    - Que poidan acceder ambos profesores.
    - Que poidan acceder os membros do grupo `profesores`.
    - Que poida acceder calquera usuario válido.
- Con autenticación HTTP Digest:
  - Crea o arquivo de contrasinais con dous usuarios: `admin1` e `admin2`, pertencentes ao dominio `xestion`.
  - Crea o cartafol `C:\apache24\htdocs\administradores` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:
    - Que poida acceder `admin1`, pero non `admin2`.
    - Que poidan acceder ambos administradores.
    - Que poida acceder calquera usuario válido.

Pasos a seguir

- Con autenticación HTTP Basic:
  - Crea o arquivo de contrasinais con dous usuarios: `profesor1` e `profesor2`.

Creamos o arquivos de contrasinais e o usuario profesor1, cando se nos solicite, introducimos o seu contrasinal:

```
cd C:\Apache24\bin\
htpasswd.exe -c C:\Apache24\contr_basic profesor1
```

Creamos o usuario `profesor1`, cando se nos solicite, introducimos o seu contrasinal:

```
htpasswd.exe C:\Apache24\contr_basic profesor2
```

  - Crea o grupo `profesores` que contén a `profesor1` e `profesor2`.

Creamos un arquivo de grupos, por exemplo, `C:\Apache24\grupos` e editámolo co seguinte contido:

```
profesores: profesor1 profesor2
```

  - Crea o cartafol `C:\Apache24\htdocs\profesores` con algún arquivo HTML co contido que prefiras, pode ser similar a:

```
<html>
   <body>
```

```
        <h1>index.html en C:\Apache24\htdocs\profesores</h1>
    </body>
</html>
```

– e configura adecuadamente Apache de tres maneiras diferentes:

Editamos o ficheiro `C:\Apache24\conf\httpd.conf`.

– Que poida acceder `profesor1`, pero non `profesor2`.

Engadimos o seguinte bloque:

```
<Directory "C:\Apache24\htdocs\profesores">
    AuthType Basic
    AuthName "Para invitados"
    AuthBasicProvider file
    AuthUserFile "C:\Apache24\contr_basic"
    Require user profesor1
</Directory>
```

– Que poidan acceder ambos profesores.

Modificamos a directiva `Require` do seguinte xeito:

```
    Require user profesor1 profesor2
```

– Que poidan acceder os membros do grupo profesores.

Comprobamos que estea habilitado o modulo `authz_groupfile` (en Windows está habilitado por defecto), no ficheiro `C:\Apache24\conf\httpd.conf` debe estar sen comentar a liña:

```
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
```

Engadimos no bloque a liña:

```
AuthGroupFile "C:\Apache24\grupos"
```

E modificamos a directiva `Require` do seguinte xeito:

```
Require group profesores
```

– Que poida acceder calquera usuario válido.

Modificamos a directiva `Require` do seguinte xeito:

```
    Require valid-user
```

Reiniciamos desde a área de notificacións o servidor Apache e, para comprobar que a configuración se realizou correctamente, conectámonos desde o navegador da máquina cliente a `http://192.168.0.2/profesores` e debe solicitarse o usuario e contrasinal, no caso de que teñamos acceso, amosarase o arquivo creado, en caso contrario, aparecerá unha e outra vez a ventá de solicitude de usuario.

- Con autenticación HTTP Digest:

O primeiro que debemos facer é activar o módulo `auth_digest` que non está activado por defecto, facendo que non estea comentada no ficheiro `C:\Apache24\conf\httpd.conf` a liña:

```
LoadModule auth_digest_module modules/mod_auth_digest.so
```

E reiniciamos o servidor desde a área de notificacións.

– Crea o arquivo de contrasinais con dous usuarios: `admin1` e `admin2`, pertencentes ao dominio `xestion`.

Creamos o arquivo de contrasinais e o usuario `admin1`:

```
cd C:\Apache24\bin\
htdigest.exe -c C:\Apache24\contr_digest xestion admin1
```

Cremos o usuario `admin2`:

```
htdigest.exe -c C:\Apache24\contr_digest xestion admin2
```

– Crea o cartafol `C:\Apache24\htdocs\administradores` con algún arquivo HTML co contido que prefiras e configura adecuadamente Apache de tres maneiras diferentes:

Editamos o ficheiro `C:\Apache24\conf\httpd.conf`.

– Que poida acceder `admin1`, pero non `admin2`.

```
<Directory C:\Apache24\htdocs\administradores >
    AuthType Digest
    AuthName "xestion"
    AuthBasicProvider file
    AuthUserFile "C:\Apache24\contr_digest"
    Require user admin1
</Directory>
```

– Que poidan acceder ambos administradores.

Modificamos a directiva `Require` do seguinte xeito:

```
Require user admin1 admin2
```

– Que poida acceder calquera usuario válido.

Modificamos a directiva `Require` do seguinte xeito:

```
Require valid-user
```

5. Ficheros .htaccess en Linux

Habilitar o uso de ficheiros `.htaccess` no directorio `/home/administrador/propio` e probaremos que as directivas incluídas neste arquivo teñen efecto.

- Crea o directorio `/home/administrador/propio` e asignámoslle un alias para que sexa accesible desde `http://192.168.0.1/propio`. Creamos o arquivo `propio.html` nel co contido que queiramos.
- Permite o emprego de ficheiros `.htaccess` nese directorio para todas as directivas admitidas.

- Crea o ficheiro `/home/administrador/propio/.htaccess` no que incluiremos a directiva apropiada para que se sirva por defecto o arquivo `propio.html` e soamente se poida acceder desde a IP da máquina cliente.
- Modifica a directiva `AllowOverride` para que só poidan sobrescribirse as directivas de control de acceso.
- Modifica a directiva `AllowOverride` só poidan sobrescribirse as directivas de indexación de directorios.

Pasos a seguir

- Crea o directorio `/home/administrador/propio` e asignámoslle un alias para que sexa accesible desde `http://192.168.0.1/propio`. Creamos o arquivo `propio.html` nel co contido que queiramos.

```
sudo mkdir /home/administrador/propio
```

Comprobando que teña permisos `755`.

```
sudo gedit /home/administrador/propio/propio.html
```

O contido do ficheiro pode ser similar a:

```
<html>
   <body>
      <h1>propio.html en /home/administrador/propio</h1>
   </body>
</html>
```

- Permite o emprego de ficheiros `.htaccess` nese directorio para todas as directivas admitidas.

  Editamos o arquivo `/etc/apache2/sites-available/000-default.conf`:

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

E engadimos as seguintes liñas:

```
Alias /propio /home/administrador/propio
<Directory /home/administrador/propio/>
       AllowOverride All
       Require all granted
</Directory>
```

Reiniciamos o servidor Apache:

```
sudo service apache2 restart
```

- Crea o ficheiro `/home/administrador/propio/.htaccess` no que incluiremos a directiva apropiada para que se sirva por defecto o arquivo `propio.html` e soamente se poida acceder desde a IP da máquina cliente.

  Editamos o ficheiro `.htaccess`:

```
sudo gedit /home/administrador/propio/.htaccess
```

Co seguinte contido:

```
DirectoryIndex propio.html
Require ip 192.168.0.3
```

E comprobamos desde a máquina cliente que ao acceder a `http://192.168.0.1/propio` se amosa o arquivo `propio.html` e que se pode acceder desde a máquina cliente pero non desde, por exemplo, a máquina Windows Server.

- Modifica a directiva `AllowOverride` para que só poidan sobrescribirse as directivas de control de acceso.

  Editamos o arquivo `/etc/apache2/sites-available/000-default.conf`:

  ```
  sudo gedit /etc/apache2/sites-available/000-default.conf
  ```

  E modificamos a directiva `AllowOverride`:

  ```
  <Directory /home/administrador/propio/>
        AllowOverride AuthConfig
  </Directory>
  ```

  Reiniciamos o servidor Apache:

  ```
  sudo service apache2 restart
  ```

E comprobamos desde a máquina cliente, tratando de acceder a `http://192.168.0.1/propio/propio.html` (hai que indicar o arquivo xa que non está activa a opción `Indexes` e o nome do ficheiro non coincide co indicado no `DirectoryIndex` xeral indicado no arquivo `000-default.conf`) que, se non eliminamos a directiva `DirectoryIndex`, se nos amosa un erro interno do servidor. Se a eliminamos, se ten en conta a directiva `Require ip` e só se permite o acceso desde a máquina cliente.

- Modifica a directiva `AllowOverride` só poidan sobrescribirse as directivas de indexación de directorios.

  Editamos o arquivo `/etc/apache2/sites-available/000-default.conf`:

  ```
  sudo gedit /etc/apache2/sites-available/000-default.conf
  ```

  E modificamos a directiva `AllowOverride`:

  ```
  <Directory /home/administrador/propio/>
        AllowOverride Indexes
  </Directory>
  ```

  Reiniciamos o servidor Apache:

  ```
  sudo service apache2 restart
  ```

E comprobamos desde a máquina cliente que, se non eliminamos a directiva `Require ip`, se nos amosa un erro interno do servidor. Se a eliminamos, se ten en conta a directiva `DirectoryIndex` e se amosa `propio.html`.

6. Ficheros .htaccess en Windows

Nesta tarefa habilitaremos o uso de ficheiros `.htaccess` no directorio `C:\Users\administrador\propio` e probaremos que as directivas incluídas neste arquivo teñen efecto.

- Crea o directorio `C:\Users\Administrador\propio` e asignámoslle un alias para que sexa accesible desde `http://192.168.0.2/propio`. Creamos o arquivo `propio.html` nel co contido que queiramos.
- Permite o emprego de ficheiros `.htaccess` nese directorio para todas as directivas admitidas.
- Crea o ficheiro `C:\Users\Administrador\propio\.htaccess` no que incluiremos a directiva apropiada para que se sirva por defecto o arquivo `propio.html` e soamente se poida acceder desde a IP da máquina cliente.
- Modifica a directiva `AllowOverride` para que só poidan sobrescribirse as directivas de control de acceso.
- Modifica a directiva `AllowOverride` só poidan sobrescribirse as directivas de indexación de directorios.

Pasos a seguir

- Crea o directorio `C:\Users\Administrador\propio` e asignámoslle un alias para que sexa accesible desde `http://192.168.0.2/propio`. Creamos o arquivo `propio.html` nel co contido que queiramos.

  Creamos desde o navegador de arquivos o cartafol e o ficheiro `propio.html`. O seu contido pode ser similar a:

```
<html>
   <body>
      <h1>propio.html en C:\Users\Administrador\propio</h1>
   </body>
</html>
```

- Permite o emprego de ficheiros `.htaccess` nese directorio para todas as directivas admitidas.

  Editamos o arquivo `C:\Apache24\conf\httpd.conf`, engadindo as seguintes liñas:

```
Alias /propio "C:\Users\Administrador\propio"
<Directory "C:\Users\Administrador\propio ">
       AllowOverride All
       Require all granted
</Directory>
```

  Reiniciamos o servidor Apache desde a área de notificacións.

- Crea o ficheiro `C:\Users\Administrador\propio\.htaccess` no que incluiremos a directiva apropiada para que se sirva por defecto o arquivo `propio.html` e soamente se poida acceder desde a IP da máquina cliente.

  Editamos o ficheiro `.htaccess` co seguinte contido:

```
DirectoryIndex propio.html
Require ip 192.168.0.3
```

Para sermos capaces de crear en Windows un ficheiro sen extensión que comece por un punto, podemos facelo desde o caderno de notas, facendo Arquivo→Gardar como... e escollendo a opción de "Todos os arquivos *.*".

E comprobamos desde a máquina cliente que ao acceder a `http://192.168.0.2/propio` se amosa o arquivo `propio.html` e que se pode acceder desde a máquina cliente pero non desde, por exemplo, a máquina servidor Xubuntu.

- Modifica a directiva `AllowOverride` para que só poidan sobrescribirse as directivas de control de acceso.

Editamos o arquivo `C:\Apache24\conf\httpd.conf`, modificando a directiva `AllowOverride`:

```
<Directory "C:\Users\Administrador\propio">
	AllowOverride AuthConfig
</Directory>
```

Reiniciamos o servidor Apache desde a área de notificacións, e comprobamos desde a máquina cliente, tratando de acceder a `http://192.168.0.2/propio/propio.html` (hai que indicar o arquivo xa que non está activa a opción `Indexes` e o nome do ficheiro non coincide co indicado no `DirectoryIndex` xeral indicado no arquivo `httpd.conf`) que, se non eliminamos a directiva `DirectoryIndex`, se nos amosa un erro interno do servidor. Se a eliminamos, se ten en conta a directiva `Require ip` e só se permite o acceso desde a máquina cliente.

- Modifica a directiva `AllowOverride` só poidan sobrescribirse as directivas de indexación de directorios.
- Editamos o arquivo `C:\Apache24\conf\httpd.conf`, modificando a directiva `AllowOverride`:

```
<Directory "C:\Users\Administrador\propio">
	AllowOverride Indexes
</Directory>
```

Reiniciamos o servidor Apache desde a área de notificacións e comprobamos desde a máquina cliente que, se non eliminamos a directiva `Require ip`, se nos amosa un erro interno do servidor. Se a eliminamos, se ten en conta a directiva `DirectoryIndex` e se amosa `propio.html`.

7. Servidor virtual HTTPS en Linux.

Nesta tarefa crearemos unha clave privada e un certificado autofirmado que nos servirán para poñer en marcha un sitio virtual HTTPS no servidor Apache en Linux.

- Crea unha clave privada para o sitio.
- Crea un certificado autofirmado para o sitio.
- Configura un sitio virtual HTTPS chamado `daw-ssl.com`, os seus contidos estarán situados en `/var/www/daw-ssl`, e actívao.

Pasos a seguir

Para poder empregar SSL en Apache, o primeiro que debemos facer é activar o módulo `mod_ssl`:

```
sudo a2enmod ssl
```

E reiniciamos o servidor:

```
sudo service apache2 restart
```

- Crea unha clave privada para o sitio.

Situámonos dentro dun cartafol fo directorio persoal do usuario que estamos a empregar, por exemplo `/home/administrador/certificados` e executamos o comando:

```
openssl genrsa -out clave-privada.key 2048

Generating RSA private key, 2048 bit long modulus
..............+++
.....................................................................+++
e is 65537 (0x10001)
```

- Crea un certificado autofirmado para o sitio.

Primeiro xeramos a solicitude de certificado, completando os datos que nos solicite:

```
openssl req -new -key clave-privada.key -out solicitude-certificado.csr

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Pontevedra
Locality Name (eg, city) []:Vigo
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Xunta de Galicia
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:DAW
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Creamos o certificado autofirmado:

```
openssl x509 -req -days 365 -in solicitude-certificado.csr -signkey clave-privada.key -out certificado-autofirmado.crt
```

```
Signature ok
subject=/C=ES/ST=Pontevedra/L=Vigo/O=Xunta de Galicia/CN=DAW
Getting Private key
```

- Configura un sitio virtual HTTPS chamado `daw-ssl.com`, os seus contidos estarán situados en `/var/www/daw-ssl`, e actívao.

  Copiamos os certificados nos cartafoles apropiados:

```
sudo mv clave-privada.key /etc/ssl/private/
sudo mv certificado-autofirmado.crt /etc/ssl/certs/
```

  E modificamos adecuadamente os propietarios e permisos:

```
sudo chown root:ssl-cert /etc/ssl/private/clave-privada.key
sudo chmod 644 /etc/ssl/private/clave-privada.key
sudo chown root:ssl-cert /etc/ssl/certs/certificado-autofirmado.crt
```

  Creamos o directorio `/var/www/daw-ssl`

```
sudo mkdir /var/www/daw-ssl
```

  E dentro creamos, por exemplo, un arquivo `index.html`

```
sudo gedit /var/www/daw-ssl/index.html
```

  O contido do ficheiro pode ser similar a:

```
<html>
   <body>
      <h1>index.html en /var/www/daw-ssl</h1>
   </body>
</html>
```

  Creamos o ficheiro de configuración do sitio virtual:

```
sudo gedit /etc/apache2/sites-available/daw-ssl.conf
```

  Co seguinte contido:

```
<IfModule mod_ssl.c>
 <VirtualHost *:443>
        ServerName daw-ssl.com
        DocumentRoot /var/www/daw-ssl

        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/certificado-autofirmado.crt
        SSLCertificateKeyFile /etc/ssl/private/clave-privada.key
 </VirtualHost>
</IfModule>
```

  Habilitamos o sitio virtual:

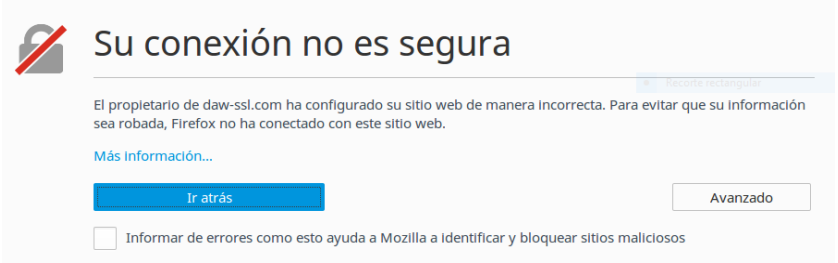
```
sudo a2ensite daw-ssl
```

Na máquina cliente, engadimos unha nova liña ao arquivo `/etc/hosts`

```
192.168.0.1 daw-ssl.com
```

E desde o navegador accedemos a `https://daw-ssl.com`

Como o certificado non foi emitido por ningunha autoridade de certificación de confianza do navegador, aparece o aviso:



Engadimos unha excepción de seguridade e xa podemos ver o contido do sitio.

8. Servidor virtual HTTPS en Windows.

Nesta tarefa crearemos unha clave privada e un certificado autofirmado que nos servirán para poñer en marcha un sitio virtual HTTPS no servidor Apache en Windows.

- Crea unha clave privada para o sitio.
- Crea un certificado autofirmado para o sitio.
- Configura un sitio virtual HTTPS chamado `daw-ssl.com`, os seus contidos estarán situados en `C:\Apache24\htdocs\daw-ssl`, e actívao.

Pasos a seguir

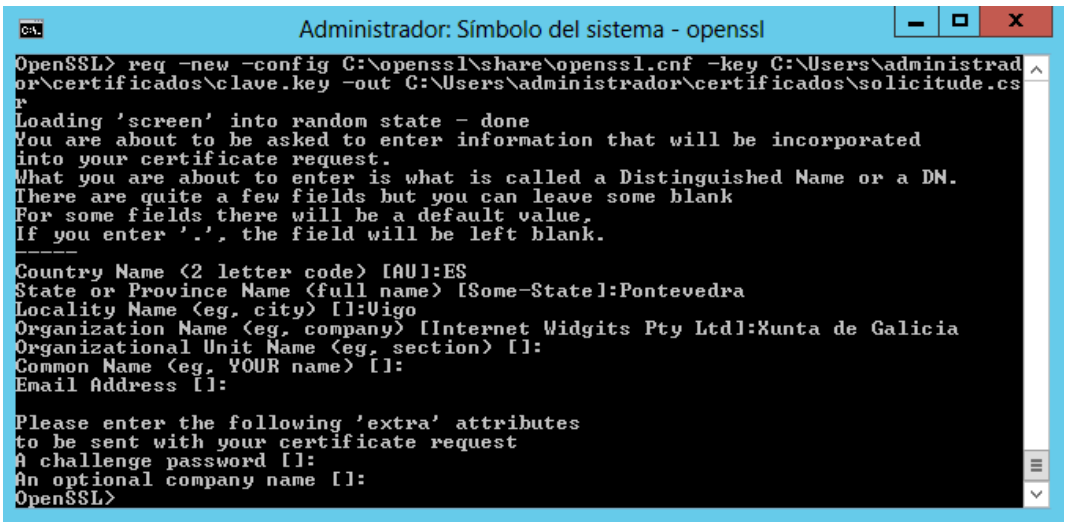
Para xerar as claves e certificados facemos uso do executable `C:\Apache24\bin\openssl.exe`.

- Crea unha clave privada para o sitio.

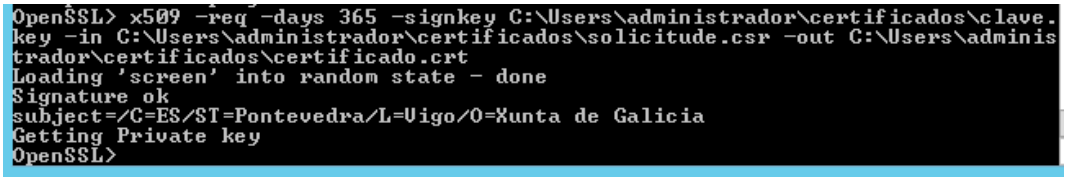
```
C:\openssl\bin>openssl
OpenSSL> genrsa -out C:\Users\administrador\certificados\clave.key
Loading 'screen' into random state - done
Generating RSA private key, 512 bit long modulus
.....................................++++++++++++
.................++++++++++++
e is 65537 (0x10001)
OpenSSL>
```

- Crea un certificado autofirmado para o sitio.

Xeraremos a solicitude de certificado (en Windows debemos indicar a localización do ficheiro de configuración de OpenSSL):



E agora autofirmaremos o certificado:



E agora copiamos desde o navegador de arquivos a clave e o certificado a, por exemplo, o novo directorio `C:\Apache24\ssl`.

- Configura un sitio virtual HTTPS chamado `daw-ssl.com`, os seus contidos estarán situados en `C:\Apache24\htdocs\daw-ssl`, e actívao.

Para poder empregar SSL en Apache, o primeiro que debemos facer é activar o módulo `mod_ssl` eliminando no arquivo `C:\Apache24\conf\httpd.conf` os comentarios da liñas:

```
#LoadModule ssl_module modules/mod_ssl.so
#Include conf/extra/httpd-ssl.conf
```

E reiniciamos o servidor desde a área de notificacións.

Creamos o directorio `C:\Apache24\htdocs\daw-ssl` e dentro creamos, por exemplo, un arquivo `index.html` co seguinte contido:

```
<html>
   <body>
      <h1>index.html en C:\Apache24\htdocs\daw-ssl</h1>
   </body>
</html>
```

Editamos a configuración do sitio virtual en `C:\Apache24\conf\extra\httpd-ssl.conf`

Eliminamos todo o seu contido menos a liña:

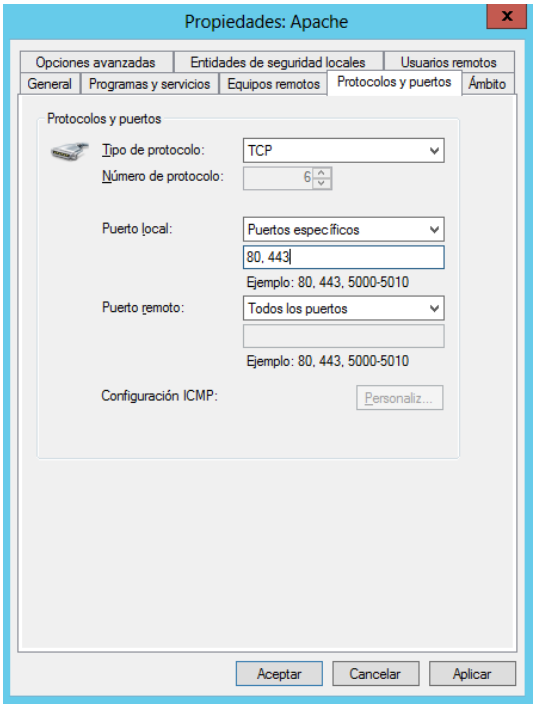
```
Listen 443
```

E engadimos:

```
<VirtualHost *:443>
   ServerName daw-ssl.com
   DocumentRoot "C:\Apache24\htdocs\daw-ssl"

   SSLEngine on
   SSLCertificateFile "C:\Apache24\ssl\certificado.crt"
   SSLCertificateKeyFile "C:\Apache24\ssl\clave.key"
</VirtualHost>
```

Para conseguir acceder desde a máquina cliente, temos que configurar a devasa de Windows para que acepte conexións entrantes no porto 443. Para elo, modificamos a regra que creamos para que permitise as conexións entrantes no porto 80 cando instalamos Apache, para que tamén as acepte nese porto. Para elo, facemos dobre clic na devandita regra e, na lapela "Protocolos e portos", engadimos tamén o porto 443 do seguinte xeito:



Na máquina cliente, engadimos unha nova liña ao arquivo `/etc/hosts`

```
192.168.0.2 daw-ssl.com
```

E desde o navegador accedemos a `https://daw-ssl.com`

Como o certificado non foi emitido por ningunha autoridade de certificación de confianza do navegador, aparece o aviso:

Engadimos unha excepción de seguridade e xa podemos ver o contido do sitio.