

1. Servizo web: Apache

Tócanos agora instalar un servidor web que proporcione páxinas ós nosos clientes. Nós traballaremos cun clásico: Apache pero non perdades de vista Nginx que se está convertindo no servidor web de referencia.

O teu traballo:



Unhas pequenas ideas aquí. Comezamos a montar o noso servidor web:

Precisamos tres máquinas: o noso vello

dserver00, o noso vello **ucliente** e unha debian nova que será o servidor web. Fai unha captura das tres máquinas en VirtualBox

1. Clona **dserver00** do primeiro día antes de instalar nada e chámalle á máquina **dserverXXWeb**

2. O nome de host será **dserverXXweb**. Lembra indicalo en `/etc/hosts/`

3. A IP estática desta máquina será **172.16.5.12**

3. Confirma que estás na nosa rede e que temos saída a internet

4. Tócanos instalar o *apache* (versión 2) en **dserverXXweb**:

```
apt-get update
```

```
apt-get install apache2
```

```
systemctl status apache2
```

5. Está activo? Estupendo. Mostra a túa pantalla.

6. Captura a saída de `tree -L 1 /etc/apache2`

7. Cotillea no ficheiro de configuración. Onde está o ficheiro que estás visualizando?

8. Modifícao poñendo o teu nome nun `<h1>`

Máis información aquí e tamén aquí (polo de agora non atenderemos o ufw)

2. Clientes para o servidor web

Se xa funciona o servidor Web facemos o de sempre: Comprobar se un cliente recibe resposta a unha petición de páxina web.

O teu traballo:

Imos comprobar se un **ucliente** é quen de conectarse e ver unha páxina web.

1. Arrancamos o noso servidor **dserver00** (funcionando DNS e DHCP) para que regale IP e resolva nomes (ou poñemos IP manualmente)

2. Arrancamos o **dserverXXweb**

3. Arrancamos unha máquina **ucliente** calquera (mellor *non* a de webmin) e imos ó noso navegador favorito. Tecleamos a URI seguinte:

`http://172.16.5.12`

Que obtemos?

4. Onde estará o ficheiro `.html` que se mostra no cliente? Pon o teu nome para demostrar nunha captura de pantalla que o tes.

5. Se en troques de poñer a IP queremos poñer o nome de host, que deberíamos modificar nos nosos servidores para usar o nome de host na barra de navegación?



3. Configuración de Apache2

Xa vimos que en `/etc/apache2` temos unha chea de ficheiros e directorios. Imos desmenuzar o que alí atopamos.

O teu traballo:

Consultando aquí

1. Cal é o ficheiro de configuración principal?

2. Cal é a diferenza entre o contido de *conf-enabled* e *conf-available*?

3. Cales son os ficheiros de configuración teus que estarán activos a próxima vez que arranques apache2? Fai unha captura de pantalla.

4. Como se crean as ligazóns simbólicas en apache2?

5. Como podo saber se a sintaxe dos meus ficheiros de configuración é correcta?

6. Onde podo averiguar a configuración do sitio web que trae de fábrica e que está en funcionamento?

7. Averigua e mostra o contido de `/var/log/apache2/` Fai unha captura de pantalla da saída deses ficheiros (chegan as dúas derradeiras liñas)

```
/etc/apache2/
├── apache2.conf
├── conf-available
├── conf-enabled
├── envvars
├── magic
├── mods-available
├── mods-enabled
├── ports.conf
├── sites-available
└── sites-enabled
```

4. Directivas de Apache

Xa temos instalado Apache2, arrancam os servizo e xa vimos a páxina web que trae de fábrica e a arbre de directorios. Para personalizar este servidor Web temos as **directivas**. Ocupan unha liña e non levan igual nen dous puntos. Tal cual o nome da directiva e o seu valor.

O teu traballo:

1. Bota un ollo ás directivas que ten Apache.

2. Mira teu ficheiro `/etc/apache2/apache2.conf` e anota algunha das directivas que tes sen comentar e indica o seu uso.

3. Consulta a directiva **DocumentRoot**, cal é o valor que trae na documentación? Cal é o que tes ti no teu ficheiro `/etc/apache2/sites-enabled/*.conf`

4. Consulta esta páxina e conta para que se usan as directivas: **Listen**, **ServerAdmin**, **ServerName** e **ServerAlias**. Pon unha ligazón en cada directiva á páxina correspondente da documentación.

5. Cales son os valores desas directivas no sitio web que trae de fábrica?



5. Apache2: aviso AH00558

Supoño que sabedes o que é un warning. Non é importante pero estorba. Cando facemos a instalación de apache2 e para comprobar a sintaxe executamos un dos comandos:

```
apachectl configtest
```

```
apachectl -t
```

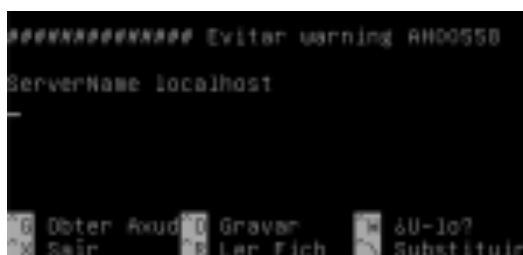
resulta que obtemos o que está catalogado como warning AH00558

O teu traballo:

Como contan neste blog engade ó final do ficheiro `/etc/apache2/apache2.conf` a directiva

ServerName localhost

É un warning, non é importante pero xa que tentamos facer un pouco ben as cousas non está de máis. Mostra que o fixeches no teu caderno electrónico.



Recargamos o ficheiro de configuración

```
systemctl reload apache2.service
```

Comprobamos se só dá *Syntax OK* cos comandos de comprobación da sintaxe.

6. Varios sitios web en apache

Xa vimos que un sitio web no fondo é unha carpeta e que apache2 trae un propio. O sitio que trae de fábrica ten un ficheiro de configuración no directorio *sites-available* e os seus *.html* na ruta que indica ese ficheiro. Facéndose unha idea de como é o que trae de modelo imos crear nós un par de sitios web no noso apache.

O teu traballo:

Comezamos creando un directorio para o novo sitio. Non imos complicarnos: no mesmo */var/www/html* creamos unha carpeta que se chame **www1.sitioXX.com**:

```
mkdir /var/www/html/www1.sitioXX.com
```

Quen é o propietario dese directorio? Non debería ser root !!!

No ficheiro *envvars* está o USER que deberíamos usar; ou ben averíguao co comando:

```
apachectl -S
```

Temos que dar permisos a ese usuario e grupo chamados **www-data**:

```
chown -R www-data:www-data /var/www/html/  
www1.sitioXX.com
```

e xa que estamos damos permisos ó pai para ter acceso a todo:

```
chmod -R 755 /var/www/html
```

Pode que xa estea listo: carpeta do sitio creada, permisos concedidos..... agardade, non hai sitio web con sentidiño se non ten un *index.html* !!!!!

Eso sabemos facelo ben:

```
nano /var/www/html/www1.sitioXX.com/index.html
```



vós veredes o que vos complicades co deseño web ?

Pero.... o plan era ter dous sitios web no mesmo apache.
Moito ánimo: facede outro tanto para un novo
www2.sitioXX.com

Máis información

7. Hosts virtuais en Apache (1/2)

Agora tocan os ficheiros de configuración. Xa temos un modelo que nos vai vir estupendamente: lembrades *000-default.conf* pois imos copialo e personalizalo para non ter que teclear todo de vez.

O teu traballo:

1. Imos inspirarnos no ficheiro de configuración que trae de fábrica, ou sexa, copiazoo directamente:

```
cd /etc/apache2/sites-available/  
cp 000-default.conf sitioXX.conf
```

2. Temos así un ficheiro de configuración *sitioXX.conf* con xusto o mesmo có de fábrica. Temos que preparar o noso:

Cal é o nome do noso sitio?

Cal é a ruta do noso sitio?



```
root@kali:~#  
ServerName www.sitio001.com  
ServerAdmin superadministrador@sitio001.com  
DocumentRoot /var/www/html/www1.sitio001.com  
ErrorLog ${APACHE_LOG_DIR}/error.log  
CustomLog ${APACHE_LOG_DIR}/access.log combined  
root@kali:~#
```

Revisa a sintaxe por si acaso e fai captura do teu.

3. Agora que estamos en *sites-available* queremos habilitalo e que estea en *sites-enable*. Descubrimos o comando:

```
a2ensite sitioXX.conf
```

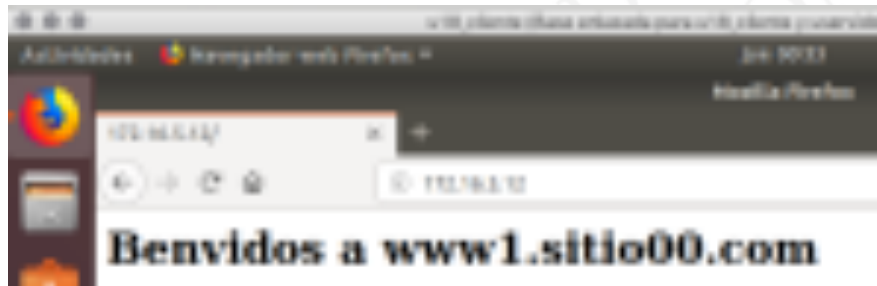
Está nos dous directorios?

4. Iremos deshabilitar a configuración de fábrica:

```
a2dissite 000-default.conf
```

Veña que xa vai!!! Decatádevos que nos manda facer un **reload**.

5. Correndo ó cliente por ver se vai. A min siiiii:??



6. Aínda non está ben de todo... alguén nota o que pasa?



Eso xa é cousa vosa: Pode que no servidor DNS ou pode que no `/etc/hosts`

Máis información aquí e tamén aquí se queredes cambiar as rutas ou aquí para usar alias , atopei esto polo tema dos subdominios.

8. Hosts virtuais en Apache (2/2)

E o noso segundo sitio web: www2.sitioXX.com funciona? Nesta páxina axudan a resolver problemillas

O teu traballo:

Vainos contando como fas todo o procedemento cun segundo sitio web. A min vaime quedar o ficheiro de configuración:

```

<VirtualHost <BB>
    ServerName www2.sitioXX.com
    ServerAlias superadministradordel sitioXX.com
    DocumentRoot /var/www/html/wwd2.sitioXX.com

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<VirtualHost <BB>
    ServerName www2.sitioXX.com
    ServerAlias superadministradordel sitioXX.com
    DocumentRoot /var/www/html/wwd2.sitioXX.com

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

Mostra

capturas dende o cliente Ubuntu cando no navegador indicas:

<http://172.16.5.12>

<http://dserverXXweb>

<http://www1.sitioXX.com>

<http://www2.sitioXX.com>

Todo esto que fixemos chámase Virtual Host baseado en nomes habería outra posibilidade que é baseado en IP. Aquí está moi ben contado

9. Logs en Apache

Todas as peticións que fai un cliente web e todos os erros que se producen mentres o servidor web está activo quedan rexistrados nos chamados ficheiros log.

O teu traballo:

1. Averigua o que se garda nos ficheiros *error.log* e tamén no *access.log* de apache2

2. Averigua onde están gardados os teus ficheiros log, tanto *error.log* coma *access.log* usando o comando:

```
apachectl -S
```

3. Vai onda eles e fai unha captura de pantalla.

4. Unha nova para aprender: o comando *tail* como sabes mostra o final dun ficheiro. Se engadimos *tail -f* daquela queda agardando algún evento e podemos ver en tempo real o que acontece no apache.

```
tail -f /var/log/apache/access.log
```

deixa o terminal e actualiza no cliente a páxina web. Que acontece no servidor?



10. Apache por IP ou portos

O xeito que temos de acceder a un sitio web aloxado no noso servidor é poñendo no navegador do cliente o nome do sitio tal e como está gardado no servidor. Existe outra forma de facelo: en troques de usar o nome empregar o porto no que está rexistrado. Alguén lembra algo de portos en redes de ordenadores?

O teu traballo:

- 1.No servidor web averigua en que porto estaba escoitando.
Faino consultando o ficheiro *ports.conf*
- 2.Engade outro porto, por exemplo 8080



- 3.Recarga **apache2**

4.Fabrica un novo sitio web que se chame *www3.sitioXX.com* que teña os permisos axeitados mailo seu *index.html*

5.Modifica no teu ficheiro de configuración *sitioXX.conf* engadindo ó final:

```
(/VirtualHost>

(VirtualHost 172.16.5.12:8080>

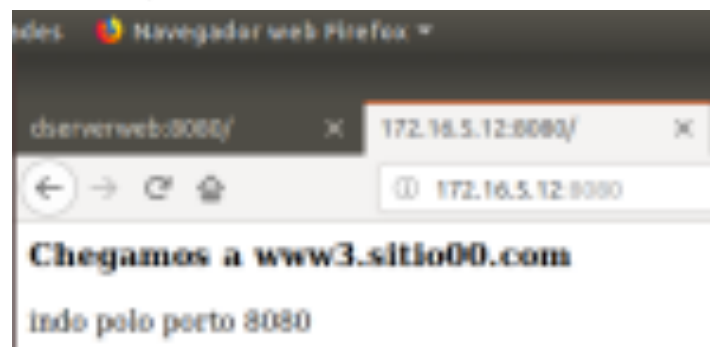
    ServerName www3.sitio00.com
    ServerAdmin saceradeadministradora@sitio00.com
    DocumentRoot /var/www/html/www3.sitio00.com

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

(/VirtualHost>
```

6. Habilita o novo sitio web con `a2ensite sitioXX.conf`, se tal mira os ficheiros de log ou confirma que todo está ben con `apachectl -s` tamén lembramos `apachectl configtest`

7.No meu cliente funciona moi ben. Fíxate que tes que empregar o número de porto tras dous puntíños.



ou o nome que teñas no teu servidor de DNS (ou o famoso `/etc/hosts` do cliente)

Máis información

11. Módulos en apache2

Os módulos veñen sendo algo así como as extensións de Apache, por facer unha comparación familiar unha especie de plug-in que nos permite engadir utilidades novas.

O teu traballo:

Busca nesta páxina información para averiguar:

1. Cal é a razón para instalar módulos en apache2?
2. Obtén a lista de módulos que tes instalados empregando `apachectl -M`
3. Averigua os módulos que tes habilitados mirando en `/etc/apache2/mods-enable`.
4. Cantos tes dispoñibles que teñan no seu nome `ssl*`?
5. Como son os comandos para habilitar e deshabilitar un módulo?
6. Cal é a utilidade do módulo **mod_ssl**?



Consulta dos módulos máis importantes

12. Certificados ssl

O noso servidor web non é seguro. Podemos recibir ataques e calquera pode ler o que recibimos ou enviamos. Para protexer as nosas comunicacións debemos tratar cun chisco de criptografía.



O teu traballo:

1. Que é un certificado ssl?
2. Para que se usan as chaves públicas e privadas?
3. Que é un certificado autoasinado?
4. En que casos pode empregarse?
5. Pon o nome dun par de empresas que vendan certificados dixitais
6. Imos fabricar un certificado ssl autoasinado para o noso apache.
 1. Instala openssl
 2. teclea a seguinte liña no teu **dserverXXweb**:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout sitioXX.key -out sitioXX.crt
```

7. Vai contestando. O importante é a liña na que pide o nome; abonda coa IP do teu **dserverXXweb**



8. Que significan os flags desta liña de comandos?
9. Mira o contido destes novos ficheiros. Cal é o certificado público? Cal é a chave privada?
10. Cal non debes XAMAIS compartir con amiguítos?

13. Apache seguro

Xa temos certificado ssl para asegurar as comunicacións entre cliente e servidor? Daquela imos indicarlle a apache2 que o use debidamente.

O teu traballo:

1. Move os ficheiros `.cert` e `.key` á carpeta `/etc/apache2`

2. Estes ficheiros teñen que ser só para o usuario **root** e ter os permisos 600:

```
chmod 600 /etc/apache2/
sitioXX.crt
```



3. Activamos o módulo para que apache2 dea traballado con ssl:

```
a2enmod ssl
```

4. Despois de activar un módulo cómpre reiniciar apache2

5. Non esqueza des confirmar que ten que ir polo porto seguro: 443

```
nano /etc/apache2/ports.conf
```



```
mkdir /var/www/html/seguero.sitioXX.com/
```

7. Iremos co ficheiro de configuración de apache2. Lembrades que de fábrica traía un ficheiro en `/etc/apache2/sites-available/default-ssl.conf`. Iremos copialo co nome **ssl-sitioXX.conf**

```

server@kali:~$ cd default-001.conf 001-001000.conf
server@kali:~$ ip -s
TOTAL 24
- - - - - 1.0000 port 13000 Nov 21 22:17 000-default.conf
- - - - - 1.0000 port 00000 Nov 2 12:04 default-001.conf
- - - - - 1.0000 port 777 Nov 26 13:14 001000.conf
- - - - - 1.0000 port 00000 Nov 27 13:00 001-001000.conf

```

[illegible]



9. Habilitamos coma sempre:

a2ensite ssl-sitioXX.com

10. Boas novas no cliente?? Pode que tarde un chisquiño de máis....