SERVIDOR WEB HTTP APACHE

Un servidor *HTTP* es el programa que atiende las peticiones de los clientes Web y proporciona las páginas solicitadas. Utiliza de forma general el puerto *80 TCP* para atender las peticiones de los clientes, aunque también puede atender peticiones a través del puerto *443 TCP* utilizado para conexiones seguras

Apache es un servidor *HTTP* de código libre, que funciona en GNU/Linux, Windows y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continua siendo el servidor *HTTP* más utilizado. ***Apache*** es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por *Apache Software Foundation*

Instalación

Para instalar apache desde los repositorios *sudo apt-get install apache2* o *sudo apt-get install apache2-mpm-prefork*

Inicio, parada y reinicio del servicio apache

sudo service apache2 start | *stop* | *restart*

Archivos y directorios de configuración

Todos los archivos de configuración se encuentran en el directorio */etc/apache2*

apache2.conf	Archivo principal de configuración
ports.conf	Directivas de configuración que indican puertos y direcciones *IP* donde ***apache*** escucha peticiones
conf.d/	Contiene archivos de configuración asociados a módulos específicos Los archivos de este directorio son incluidos mediante la directiva *Include /etc/apache2/conf.d*
mods-available/	Directorio que contiene los archivos de los diferentes módulos que puede utilizar ***apache***
mods-enabled/	Directorio que contiene los módulos activos de apache
site-available/	Directorio que contiene los archivos de configuración de los diferentes hosts virtuales (sitios)
site-enabled/	Directorio que contiene los hosts virtuales activos
/var/www/	Directorio por defecto para alojar las páginas Web

El archivo de configuración */etc/apache2/apache2.conf* está dividido en tres secciones, configuración global, configuración general del servidor y configuración de los servidores virtuales. Como en todos los archivos de configuración de *GNU/Linux* el símbolo *#* indica un comentario y no será tenido en cuenta

SERVIDOR WEB APACHE

CONFIGURACIÓN GLOBAL

Estas directivas especifican el funcionamiento del servidor Web, indicando el directorio de los ficheros de configuración, el modo de funcionamiento del servidor, etc. Las entradas básicas son

ENTRADAS	DESCRIPCIÓN	VALOR POR DEFECTO
ServerRoot	Indica el directorio raíz donde se encuentran los ficheros de configuración, error y logs En esta entrada NO puede añadirse el carácter / al final del mismo	/etc/apache2
TimeOut	Tiempo en segundos que espera el servidor entre la realización de una conexión y el envío de la petición de la página Web mediante *GET*, el envío de la información mediante *POST* o la recepción de *ACK*s de los paquetes de datos enviados	300
KeepAlive	Indica el modo de funcionamiento del servidor en el intercambio de peticiones y envíos con los clientes Un valor *Off* indica que el funcionamiento será *HTTP/1.0*, esto es, cada petición de una página necesita una conexión TCP nueva y las entradas *MaxKeepAliveRequest* y *KeepAliveTimeout* serán ignoradas Un valor *On* indica que el funcionamiento será *HTTP/1.1*, por lo que múltiples peticiones serán enviadas a través de una misma conexión *TCP*	On
MaxKeepAliveRequest	Número máximo de peticiones por conexión *0* indica ilimitadas	100
KeepAliveTimeout	Segundos en espera entre peticiones antes de cerrar una conexión	15
<IfModule mpm_prefork_module> … </IfModule>	Indican el modo de ejecución del servidor y las características del mismo En el modo *prefork* un proceso padre lanza procesos hijo para que estos atiendan las peticiones de páginas Web recibidas, procurando que siempre existan algunos procesos hijo Es el modo de ejecución por defecto de *apache*	-
<IfModule mpm_worker_module> … </IfModule>	Indican el modo de ejecución del servidor y las características del mismo En el modo *worker*, el proceso padre lanza procesos hijos, los cuales a su vez ejecutan hilos, permaneciendo uno de ellos a la escucha de peticiones y el resto atendiendo a las mismas	-
Listen	Indica la dirección *IP* y puerto en el que el servidor Web escucha peticiones Si no se indica ninguna dirección *IP* o se indica con el símbolo * el servidor Web escuchará las peticiones de todas las interfaces de red existentes Pueden utilizarse múltiples directivas *Listen* para indicar diferentes puertos e interfaces de red Esta entrada se encuentra en el archivo */etc/apache2/ports.conf*	80 443
Include	Entrada que indica la ruta a diferentes ficheros de configuración que *apache* cargará al iniciarse	conf.d/ sites-enabled/ mod-enabled/ ports.conf
User	Usuario con el que se ejecutará *apache*	www-data
Group	Grupo con el que se ejecutará *apache*	www-data

SERVIDOR WEB APACHE

CONFIGURACIÓN GENERAL DEL SERVIDOR

Estas directivas definen el comportamiento del servidor por defecto y de todos los servidores virtuales excepto en los que se definan otras opciones

NOTA En *DEBIAN/UBUNTU*, las entradas *ServerTokens*, *ServerSignature* y las opciones de control de acceso sobre el directorio principal *<Directory />…</Directory>* se encuentran en el archivo */etc/apache2/conf.d/security*

ENTRADAS	DESCRIPCIÓN	VALOR POR DEFECTO
ServerAdmin	Dirección de correo del administrador del servidor	webmaster@localhost
ServerName	Indica el nombre y puerto con el que el servidor se identificará ante las peticiones que se realicen Si no se indica el nombre, este se obtiene mediante consulta inversa DNS Si no se especifica el puerto se utilizará el puerto por el que se recibió la petición	En distribuciones *DEBIAN/UBUNTU* no existe esta directiva y *apache* lanzará un error la primera vez que se ejecute, por tanto la indicaremos de forma *ServerName 127.0.0.1*
UseCanonicalName	Con valor *Off* Indica si se responde siempre con el nombre y puerto por el que se recibió la petición Con valor *On* indica que se responderá con el nombre y puerto especificado en la entrada *ServerName*	-
DocumentRoot	Indica el directorio a partir del cual se encuentran las páginas Web del servidor	/var/www
<Directory /> … </Directory>	Indican opciones de control de acceso sobre las páginas Web	-
<Directory /var/www> … </Directory>	Indican opciones de control de acceso sobre las páginas Web	-
DirectoryIndex	Indica los nombres y orden de las páginas por defecto que apache buscará si en la petición de un cliente no se especifica	-
AccessFileName	Indica el nombre del fichero que controla el acceso a determinados directorios	.htaccess
<Files ~ "^\.ht"> … </Files>	Indica las reglas para evitar que el fichero especificado con la entrada *AccessFileName* pueda ser accedido por un cliente Web	-
HostnameLookups	Indica si se almacenará en los ficheros logs el nombre del cliente *On* o su dirección IP *Off*	Off
ErrorLog	Indica donde se almacenarán los mensajes de error del servidor Web	/var/log/apache2/error.log
LogLevel	Indica el nivel de detalle de los mensajes de error Los niveles de mayor a menor detalle son *emerg*, *alert*, *crit*, *error*, *warn notice*, *info* y *debug* Establecer un nivel implica que también se almacenaran en el log de errores los mensajes de los niveles superiores	warn
LogFormat	Indica que información, como se guardará y como se llamarán en los ficheros logs de acceso del sistema	-
CustomLog	Indica donde se almacenarán los mensajes de acceso al sistema	/var/log/apache2/other_vhosts_access.log
ServerSignature	El valor *On* indica que apache añadirá el nombre y versión del servidor indicado en la entrada *ServerName* al final de cualquier documento de error generado El valor *Off* no añadirá el nombre y versión del servidor El valor *EMail* envía una línea de código *HTML mailto:ServerAdmin* En *DEBIAN/UBUNTU* esta directiva se encuentra en el fichero */etc/apache2/conf.d/security*	On
ServerTokens	Indica la información que da *apache* sobre sí mismo a los clientes en las peticiones Los valores de mayor a menor información son *Full*, *OS*, *Minimal*, *Minor*, *Major* y *Prod* En *DEBIAN/UBUNTU* esta directiva se encuentra en el fichero */etc/apache2/conf.d/security*	OS
Alias	Indica un camino distinto al camino por defecto para un recurso al que debe acceder el servidor	-
ScriptAlias	Igual que la directiva *Alias* pero sirve para especificar que el contenido serán scripts *CGI*	-
AddDefaultCharset	Indica el conjunto de caracteres por defecto a utilizar	UTF-8

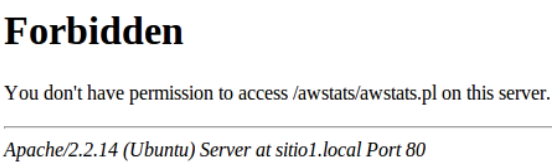
SERVIDOR WEB APACHE

Paso a paso. Configuración básica para evitar *Fingerprinting*

NOTA *Fingerprinting* es una técnica que permite identificar las características, versión, SO, etc… de un servidor desde el exterior

Ejemplos de *Fingerprinting*

Intentando acceder a un recurso que no existe



Intentando acceder a un recurso del servidor

1. Editar el archivo de configuración */etc/apache2/conf.d/security*
   #ARCHIVO SECURITY
   #/ETC/APACHE2/CONF.D/SECURITY

   ServerTokens Prod
   ServerSignature Off

2. Reiniciar ***apache*** y comprobar que ahora no se muestra ninguna información relativa al propio servidor

Paso a paso. Gestión de módulos en ***apache***. Configuración del módulo *status* para acceder desde la dirección *IP 192.168.1.XX* y habilitación del módulo *info*

Comandos

a2enmod módulo	Habilita el *módulo* indicado
a2dismod módulo	Deshabilita el *módulo* indicado

Directorios

/etc/apache2/mods-available	Archivos de configuración de los *módulos*
/etc/apache2/mods-enables	*módulos* habilitados por *apache*

El módulo *status*, que viene activado en la propia instalación de *apache*, nos permite averiguar de forma remota, sólo a usuarios autorizados, información del estado actual del servidor (servicio prestado, carga de trabajo actual, etc.)

El módulo *info*, que viene desactivado por defecto, permite ver información sobre la configuración del servidor y los módulos cargados en este

1. Modificar el archivo de configuración del módulo *status*
   #ARCHIVO STATUS.CONF
   #/ETC/APACHE2/MODS-AVAILABLE/STATUS.CONF

   ExtendedStatus On
   <Location /server-status>
   …
   Allow from 192.168.1.XX
   </Location>

2. Reiniciar *apache* y probar la nueva configuración desde el cliente Web de dirección *IP 192.168.1.XX* a través de la URL *http://IP_servidorWeb/server-status*

3. Habilitar el módulo *info*
   sudo a2enmod info

4. Reiniciar *apache* y probar el módulo desde el navegador a través de la URL *http:// IP_servidor/server-info*

CONFIGURACIÓN DE SERVIDORES VIRTUALES

Los servidores virtuales permiten que un solo ordenador pueda alojar múltiples dominios y páginas web, de forma que una sola dirección IP puede responder a diferentes nombres de dominio

Comandos

a2ensite sitio	Habilita el *VirtualHost* indicado
a2dissite sitio	Deshabilita el *VirtualHost* indicado

SERVIDOR WEB APACHE

Directorios

/etc/apache2/sites-available Archivos de configuración de los *VirtualHost*
/etc/apache2/sites-enables *VirtualHost* habilitados y servidos por *apache*

Opciones

DIRECTIVAS	DESCRIPCIÓN
NameVirtualHost dirección_IP:puerto	Indica la dirección IP y el puerto en el que se escucharán las peticiones para los servidores virtuales Se permite el uso del comodín * para indicar todas las direcciones IP del servidor web Tanto la dirección IP como el puerto deben haber sido habilitadas en la configuración global del servidor mediante la directiva *Listen*
<VirtualHost nombre:puerto></VirtualHost>	Indica mediante el nombre o dirección IP la dirección y puerto en que escucha el servidor virtual al que se refieren las directivas comprendidas entre ellas
ServerName	Nombre del servidor que debe solicitar el cliente para que sea atendido por este servidor web virtual
ServerAlias	Indica otros nombres que puede tener este mismo servidor virtual

NOTA Se supone que cada *VirtualHost* estará correctamente configurado en el archivo */etc/hosts* en caso de no disponer de un servidor *DNS* configurado de forma *IP_servidor* *nombre_VirtualHost*

Paso a paso. Crear un *VirtualHost* para el dominio sitio1.local

1. Crear directorio donde se almacenará el sitio web y la página de inicio
   sudo mkdir /var/www/sitio1
   sudo nano /var/www/sitio1/index.html

2. Crear el archivo de configuración de dicho sitio
   sudo nano /etc/apache2/sites-available/sitio1

   #VIRTUALHOST SITIO1.LOCAL
   #/ETC/APACHE2/SITES-AVAILABLE/SITIO1.LOCAL

   <VirtualHost 192.168.1.XX:80>
   ServerName sitio1.local
   DocumentRoot /var/www/sitio1
   DirectoryIndex index.html index.htm index.php

   CustomLog /var/log/apache2/sitio1.local/access_log combined
   ErrorLog /var/log/apache2/sitio1.local/error.log
   </VirtualHost>

3. Añadir la directiva *NameVirtualHost* en el archivo de configuración */etc/apache2/apache2.conf* o en un archivo de configuración dentro de */etc/apache2/conf.d/* del modo *NameVirtualHost 192.168.1.XX:80*

4. Habilitar el sitio web y reiniciar apache
   sudo a2ensite sitio1
   sudo service apache2 restart

CONTROL DE ACCESO A LOS RECURSOS

El control de acceso a los recursos puede realizarse de dos formas, no excluyentes, por **dirección IP** del cliente y por **usuario**

CONTROL DE ACCESO POR DIRECCIÓN IP DEL CLIENTE

El acceso a los recursos por **dirección IP** del cliente es controlado por **APACHE** a través de diferentes directivas que se aplican a un determinado directorio, una URL o un fichero del sistema

Entradas de control de acceso

Las entradas son *Directory*,*Location* y *Files*

Para aplicar reglas de control de acceso sobre un directorio y sus subdirectorios se utiliza la entrada *Directory* excepto que exista una regla más específica para alguno de los subdirectorios

Ejemplo. Aplicar reglas de control sobre el directorio */var/www/sitio1* y sus subdirectorios
<Directory "/var/www/sitio1">
…
</Directory>

La entrada *Location* aplica reglas de control de acceso a una URL determinada, por lo que su camino es relativo respecto al valor de la raíz de los documentos del servidor *DocumentRoot*

Ejemplo. Aplicar directivas de control de acceso al directorio y subdirectorios */var/www/sitio1/mi_directorio*
DocumentRoot /var/www/sitio1

<Location "/mi_directorio">
…
</Location>

La entrada *Files* se refiere al archivo especificado, independientemente del directorio donde se encuentre dicho archivo

Ejemplo. Aplicar reglas de control de acceso al archivo *mi_archivo.html*
<Files "mi_archivo.html">
…
</Files>

Todas las entradas anteriores tienen sus equivalentes para poder utilizar expresiones regulares, que son *DirectoryMatch*, *LocationMatch* y *FilesMatch*

Ejemplo. Aplicar reglas de control de acceso a los archivos que empiecen por *.ht*
<FilesMatch "^\.ht">
…
</FilesMatch>

Orden de aplicación de las entradas de control de acceso

1. Directory
2. DirectoryMatch
3. Files y FilesMatch (sin preferencia entre ellas)
4. Location y LocationMatch (sin preferencia entre ellas)

Directivas de control de acceso

Las directivas de control de acceso son *Order*, *Allow* y *Deny*

DIRECTIVA	DESCRIPCIÓN	VALORES
Order	Establece el orden en que se interpretan las directivas Allow y Deny	Allow,Deny Deny,Allow
Allow	Permite el acceso al host/s dirección/es IP o red/es	
Deny	Prohíbe el acceso al host/s dirección/es IP o red/es	

La directiva *Order* establece el orden en que se interpretan las directivas *Allow* y *Deny*, sus posibles valores son *Allow*, *Deny* o *Deny, Allow*

- Si la directiva *Order* tiene como valor *Allow,Deny* indica que la directiva *Allow* se evalúa antes que la directiva *Deny*, denegándose el acceso por defecto a todos los ordenadores que no cumplan alguna de las directivas
- Si la directiva *Order* tiene como valor *Deny,Allow* indica que la directiva *Deny* se evalúa antes que la directiva *Allow*, permitiéndose el acceso por defecto a todos los ordenadores que no cumplan alguna de las directivas

Las directivas *Allow* y *Deny* indican el ordenador/es al /a los que se aplica la directiva especificado por su nombre, por su dirección IP, por el nombre del dominio al que pertenezca, una dirección IP parcial o mediante direcciones de red y su máscara

Ejemplo. Formas válidas de asignar un ordenador o red
Allow from miordenador.local
Allow from 192.168.1.xx
Allow from local
Allow from 192.168.1.0/255.255.255.0 o *Allow from 192.168.1.0/24*

NOTA Es posible especificar todos los ordenadores utilizando el valor *all*

Ejemplo. Denegar el acceso a todos los ordenadores
Deny from all

Ejemplo. Permitir el acceso a todos los ordenadores de la red *192.168.1* y denegando el acceso al resto al directorio */var/www/web1*

```
<Directory "/var/www/web1">
        Order Allow,Deny
        Allow from 192.168.1.0/24
</Directory>
```

Ejemplo. Permitir el acceso a todos los ordenadores de la red *192.168.1* denegando el acceso al ordenador de dirección IP *192.168.1.XX*

```
<Directory "/var/www/web1">
        Order Allow,Deny
        Allow from 192.168.1.0/24
        Deny from 192.168.1.XX
</Directory>
```

Ejemplo. Igual que el anterior pero cambiando la directiva *Order*

```
<Directory "/var/www/web1">
        Order Deny,Allow
        Allow from 192.168.1.0/24
        Deny from 192.168.1.XX
</Directory>
```

Todos los ordenadores tendrán acceso al directorio */var/www/web1* incluso el ordenador *192.168.1.XX*

Ejemplo. Aplicar directivas de control de acceso a diferentes entradas

```
DocumentRoot /var/www/pagweb

<Location "/">
        Order Deny,Allow
        Allow from all
</Location>
<Directory "/var/www/pagweb">
        Order Allow,Deny
        Allow from all
        Deny from 192.168.1.XX
</Directory>
```

La entrada *Directory* deniega el acceso al ordenador *192.168.1.XX* pero la entrada *Location* al evaluarse en último lugar dejará la restricción sin efecto

Además de las condiciones de control de acceso, dentro de las entradas se pueden especificar algunas funcionalidades especiales mediante la directiva *Options*

Su sintaxis es *Options [+/-] opción [+/-] opción …*

Las posibles opciones son

VALOR	DESCRIPCIÓN
None	Ninguna funcionalidad adicional estará activa
All	Todas las funcionalidades adicionales estarán activas menos **MultiViews**
ExecCGI	Permite utilizar scripts CGI
FollowSymLinks	Permite seguir (acceder) a recursos apuntados por enlaces simbólicos
SymLinksIfOwnerMatch	Versión segura de **FollowSymLinks**. Solo se permiten seguir (acceder) a recursos que tengan los mismos permisos que el propietario del enlace simbólico
Includes	Permite incluir SSI (Server Side Includes)
IncludesNoExec	Permite incluir SSI pero excluyendo aquellos que ejecutan comandos o CGI´s
Indexes	Muestra un listado con el contenido del directorio si no existen los archivos especificados en la Directiva **DirectoryIndex**
MultiViews	Permite negociación de contenido

Ejemplo. Permitir listar el contenido de los directivos y negociación de contenido enviado solo a los ordenadores de la red *192.168.1*

```
<Directory "/var/www/web1">
        Options Indexes MultiViews
        Order Allow,Deny
        Allow from 192.168.1.0/255.255.255.0
</Directory>
```


SERVIDOR WEB APACHE

CONTROL DE ACCESO POR USUARIOS

El control de acceso por usuario viene establecido por la directiva *AllowOverride* la cual tiene que especificarse dentro de una entrada *Directory*. La directiva *AllowOverride* indica las directivas permitidas en los ficheros de control de acceso por usuario

Valores de la directiva *AllowOverride*

OPCIONES	DESCRIPCIÓN
None	No permite ninguna directiva
All	Permite todas las directivas
AuthConfig	Permite directivas de autenticación de usuarios
FileInfo	Permite directivas de control de tipo de documentos
Indexes	Permite directivas de indexado de directorios
Limit	Permite directivas que controlan el acceso por dirección IP del cliente
Options	Permite directivas que controlan funcionalidades de los directorios

Ejemplo. Permitir directivas de indexado de directorios
AllowOverride Indexes

Para el control de acceso por usuario, las opciones que nos permiten esto son *All* y *AuthConfig*. Así si indicamos dentro de una entrada *Directory* la directiva *AllowOverride AuthConfig* indicamos que el servidor busque dentro de ese directorio y subdirectorios un archivo, especificado en la directiva *AccessFileName*, con reglas para controlar el acceso a los usuarios

Así, en la configuración general del servidor la directiva *AccessFileName* y la entrada *<FilesMatch "^\.ht">…</FilesMatch>*

```
AccessFileName .htaccess
<FilesMatch "^\.ht">
        Order Allow,Deny
        Deny from all
</FilesMatch>
```

Especifican el nombre del fichero que controla el acceso por usuario a los directorios y las reglas para evitar que los ficheros cuyo nombre empieza por *.ht* puedan ser accedidos por un cliente Web y ver su contenido

Directivas del archivo de control de acceso

DIRECTIVA	DESCRIPCIÓN	VALORES
AuthType	Tipo de autenticación de usuarios	Basic Digest
AuthName	Cadena de texto que identifica la zona dominio a utilizar en la autenticación	-
AuthUserFile	Ruta y nombre del fichero que contiene los nombres y claves de los usuarios	-
AuthGroupFile	Ruta y nombre del fichero que contiene el nombre de los grupos de usuarios y los usuarios que conforman ese grupo Su sintaxis es *nombre_grupo:usuario1 usuario2 …*	-
Require	Indica los nombres de los usuarios, grupos o todos los usuarios a los que se le permite el acceso si proporcionan de forma correcta la contraseña La sintaxis es *Require user usuario1 usuario2 …* *Require group grupo1 grupo2 …* *Require valid-user*	user group valid-user
Satisfy	Como se deben satisfacer las condiciones de control de acceso, todas o alguna	all any

Tipos de autenticación

NOTA Según la documentación oficial de *apache* el tipo de autenticación *Digest* está en fase experimental

- *Basic* se envía la contraseña entre cliente y servidor sin cifrar, por lo que la seguridad dependerá del canal de comunicación
- *Digest* se envía la contraseña entre cliente y servidor cifrada (MD5), por lo que no es posible capturar la contraseña en texto plano, pero no es soportado por todos los clientes Web

Ejemplo. Fichero de control de acceso mediante usuarios permitiéndose el acceso a los usuarios *usu1* y *usu2* que se encuentren en el archivo *usuarios*

```
#ARCHIVO .HTACCESS
AuthType Basic
AuthName "control_acceso"
AuthUserFile /etc/apache2/passwd/usuarios
Require user usu1 usu2
```

Ejemplo. Fichero de control de acceso mediante grupos

```
AuthType Basic
AuthName "control acceso"
AuthUserFile /etc/apache2/passwd/usuarios
AuthGroupFile /etc/apache2/passwd/grupos
Require group grupo1 grupo2
```

La creación del archivo que contendrá los usuarios y sus contraseñas se realiza mediante el comando *htpasswd*, su sintaxis es

htpasswd -c [opciones] ruta_fichero usuario	Para crear el fichero, o si existe, para sustituirlo por uno nuevo
htpasswd [opciones] ruta_fichero usuario	Para añadir un nuevo usuario
htpasswd -D ruta_fichero usuario	Para eliminar un usuario del fichero

Las opciones son las posibles opciones de cifrado de contraseñas de los usuarios, que pueden ser

- *-p* cifrado de contraseñas en texto plano, sin cifrar, solo disponible para Windows
- *-d* cifrado CRYPT, si no se especifica nada se usará esta función para cifrar las contraseñas
- *-m* cifrado MD5
- *-s* cifrado SHA

Paso a paso. Crear un *VirtualHost* con una zona privada mediante autenticación básica

1. Crear un host virtual

```
#VIRTUALHOST SITIO2
#/ETC/APACHE2/SITES-AVAILABLE

<VirtualHost 192.168.1.XX:80>
  ServerName sitio2.local
  DocumentRoot /var/www/sitio2
  <Directory "/var/www/sitio2/privado">
    AllowOverride AuthConfig
  </Directory>
</VirtualHost>
```

2. Dentro del directorio */var/www/sitio2/privado* crear un archivo *.htaccess*

```
#ARCHIVO .HTACCESS
#/VAR/WWW/SITIO2/PRIVADO

AuthTipe Basic
AuthName "zona_privada"
AuthUserFile /etc/apache2/passwd/usuarios
Require valid-user
```

3. Crear el archivo */etc/apache2/passwd/usuarios*

```
sudo htpasswd -c /etc/apache2/passwd/usuarios usuario
```

MOD_AUTH_DIGEST

Para poder utilizar el modo de autenticación *Digest*, previamente hay que habilitar el módulo ***mod_auth_digest*** mediante el comando *sudo a2enmod auth_digest*

La creación del archivo donde se almacenarán los usuarios y sus contraseñas se realiza mediante el comando *htdigest*, su sintaxis es

htdigest -c ruta_archivo dominio usuario	Para crear el fichero, o si existe, para sustituirlo por uno nuevo
htdigest ruta_archivo dominio usuario	Para añadir un nuevo usuario

Donde *dominio* es el nombre del dominio de autenticación dado en la directiva *AuthName*

CONFIGURACIÓN DE APACHE CON SOPORTE SSL/TLS

HTTPS es la versión segura del protocolo *HTTP*. Se trata de una combinación del protocolo *HTTP* con el mecanismo de transporte *SSL* o *TLS* garantizando una protección razonable durante la comunicación cliente-servidor. Es ampliamente utilizado dentro de la red *WWW* en transacciones bancarias y pago de bienes y servicios

mod_ssl es el módulo que provee al servidor *apache* soporte para *SSL* y *TLS*, para habilitarlo teclea *sudo a2enmod ssl*

Opciones el módulo ***mod_ssl***

OPCIÓN	DESCRICPCIÓN	VALORES
SSLEngine	Habilita o deshabilita el uso de *SSL* en el *VirtualHost.*	on | off
SSLProtocol	Protocolos que pueden utilizarse	SSLv2 | SSLv3 | TLSv1 | all
SSLCertificateFile	Localización clave pública	
SSLCertificateKeyFile	Localización clave privada	

NOTA Para generar la clave pública y privada *SSL-TLS*
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout clavePrivada.key -out clavePublica.crt

Ejemplo. *VirtualHost* básico con soporte *SSL* o *TLS*

```
#VIRTUAL_HOST_SSL
#/ETC/APACHE2/SITES-AVAILABLE/SITIOSSL

<VirtualHost 192.168.1.XX:443>
        ServerName sitiossl.local
        DocumentRoot /var/www/sitiossl
        SSLEngine on
        SSLProtocol TLSv1
        SSLCertificateFile /etc/apache2/apache.crt
        SSLCertificateKeyFile /etc/apache2/apache.key
</VirtualHost>
```

Ejemplo. *VirtualHost* básico con soporte *SSL-TLS* redirigiendo peticiones

```
#VIRTUAL_HOST_SSL
#/ETC/APACHE2/SITES-AVAILABLE/SITIOSSL

<VirtualHost 192.168.1.XX:80>
        ServerName sitiossl.local
        DocumentRoot /var/www/sitiossl
        Redirect 301 / https://sitiossl.local
</VirtualHost>

<VirtualHost 192.168.1.XX:443>
        ServerName sitiossl.local
        DocumentRoot /var/www/sitiossl
        SSLEngine on
        SSLProtocol TLSv1
        SSLCertificateFile /etc/apache2/apache.crt
        SSLCertificateKeyFile /etc/apache2/apache.key
</VirtualHost>
```

MOD_EVASIVE

Los ataques ***DoS D***enial ***o***f ***S***ervice o ***DDoS D***istributed ***D***enial ***o***f ***S***ervice tienen la finalidad de provocar que un servicio o recurso sea inaccesible a los usuarios legítimos

mod_evasive aporta al servidor apache defensa ante este tipo de ataques

Instalación y activación de ***mod_evasive*** desde los repositorios

sudo apt-get install libapache2-mod-evasive | *sudo a2enmod mod-evasive*

Opciones de configuración del módulo ***mod_evasive***

OPCIÓN	DESCRIPCIÓN
DOSPageCount	Número de peticiones a una misma página dentro de un intervalo de bloqueo para que una dirección IP sea añadida a la lista de bloqueo
DOSSiteCount	Número de peticiones a un mismo sitio dentro de un intervalo de bloqueo para que una dirección IP sea añadida a la lista de bloqueo
DOSPageInterval	Umbral de bloqueo (en segundos) de la opción *DOSPageCount*
DOSSiteInterval	Umbral de bloqueo (en segundos) de la opción *DOSSiteCount*
DOSBlockingPeriod	Periodo de bloqueo para una dirección IP que haya superado alguno de los intervalos de bloqueo de las opciones *DOSPageInterval* o *DOSSiteInterval* Este parámetro es incremental, así, si una dirección IP bloqueada sigue enviando peticiones a una página o sitio Web, más tiempo seguirá bloqueado
DOSHashTableSize	Tamaño de la tabla Hash. A mayor tamaño más rápido será el rastreo de direcciones IP pero consumirá más memoria
DOSLogDir	Directorio donde se almacenarán los informes de *mod_evasive*
DOSWhitelist	Dirección o rango de direcciones IP que serán excluidas del rastreo de *mod_evasive* Pueden existir varias entradas *DOSWhitelist* en una misma configuración
DOSSystemCommand	Permite lanzar un comando *shell* cuando una dirección IP es bloqueada

Ejemplo. Lanzar una regla IPTables para bloquear IP´s atacantes

DOSSystemCommand iptables -I INPUT -p tcp --dport 80 -s %s -j DROP

Ejemplo. Archivo de configuración básico de ***mod_evasive***

```
#ARCHIVO CONFIGURACION MOD_EVASIVE
#/ETC/APACHE2/CONF.D/EVASIVE

<ifmodule mod_evasive20.c>
        DOSPageCount 5
        DOSSiteCount 10
        DOSPageInterval 2
        DOSSiteInterval 2
        DOSBlockingPeriod 10
</ifmodule>
```

MOD SECURITY

mod_security es un firewall *WEB* que se ejecuta como módulo del servidor web *apache*
Provee protección contra los ataques *WEB* más comunes

Instalación y activación de ***mod_security*** desde los repositorios

sudo apt-get install libapache2-mod-security2 | *sudo a2enmod mod-security*

Para habilitar ***mod_security*** se necesita indicar en el archivo *apache2.conf* el directorio donde se encuentran las reglas de filtrado

```
<ifModule mod_security2.c>
        Include conf.d/mod_security/*.conf
</ifModule>
```

Instalación reglas ***mod_secutity***

```
sudo mkdir /etc/apache2/conf.d/mod_security
cd /etc/apache2/conf.d/mod_security

sudo wget http://www.modsecurity.org/download/modsecurity-core-rules_2.5-1.6.1.tar.gz
sudo tar xzvf modsecurity-core-rules_2.5-1.6.1.tar.gz

sudo rm CHANGELOG LICENSE README modsecurity-core-rules_2.5-1.6.1.tar.gz
```

NOTA En *DEBIAN/UBUNTU* es necesario configurar correctamente la ruta de los ficheros logs del módulo ***mod_security***

```
#ARCHIVO MODSECURITY_CRS_10_CONFIG_CONF

SecAuditLog /var/log/apache2modsec_audit.log
SecDebugLog /var/log/apache2/modsec_debug.log
```

SERVIDOR WEB APACHE

MONITORIZACIÓN LOGS APACHE CON AWSTATS

Awstats es un analizador de ficheros logs a través de los cuales genera un archivo *HTML* para visualizar los datos mediante un cliente web

Instalación de *awastats* desde los repositorios

sudo apt-get install awstats

El archivo de configuración esta en */etc/awstats/awstats.conf*

NOTA Awstats no genera estadísticas automáticamente. Para esto tendrás que añadir un *crontab* de la forma

#GENERAR ESTADÍSTICASDEL SITIO1.LOCAL CADA DOS HORAS
*0 */2 * * * /usr/lib/cgi-bin/awstats.pl -update -config=sitio1.local*

Opciones de configuración

OPCIONES GENERALES

OPCIÓN	DESCRIPCIÓN
LogFile	Indica la ruta donde se encuentra el fichero *access.log* generado por apache
SiteDomain	Indica el nombre del dominio del cual generaremos las estadísticas

OPCIONES DE SEGURIDAD

OPCIÓN	DESCRIPCIÓN
AllowAccessFromWebToAuthenticatedUsersOnly	Indica si se habilita o no la seguridad a nivel de usuario Admite valores *0* deshabilitado o *1* habilitado
AllowAccessFromWebToFollowingAuthenticatedUsers	Indica el nombre o nombres de los usuarios que tendrán acceso a las estadísticas Solo admite valores si *AllowAccessFromWebToAuthenticatedUsersOnly* tiene valor *1*
AllowAccessFromWebToFollowingIPAddresses	Indica desde qué direcciones IP se permite el acceso a las estadísticas del sitio

Para que *awstats* trabaje con *apache* tendremos que añadir las siguientes líneas al archivo de configuración */etc/apache2/apache2.conf* o crear un archivo de configuración en el directorio */etc/apache2/conf.d*

#AWSTATS.CONF
#/ETC/APACHE2/CONF.D/AWSTATS.CONF

Alias /awstatsclasses /usr/share/awstats/lib/
Alias /awstats-icon /usr/share/awstats/icon/
Alias /awstatscss /usr/share/doc/awstats/examples/css/
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
ScriptAlias /awstats/ /usr/lib/cgi-bin/

<Directory "/usr/lib/cgi-lin">
Options +ExecCGI -MultiViews +SysmLinksIfOwnerMatch
</Directory>

Paso a paso. Crear un visor de estadísticas para un host virtual *sitio1.local*

1. Crear el archivo *awstats* para el virtual host *sitio1.local*
   sudo cp /etc/awstats/awstats.conf /etc/awstats/awstats.sitio1.local.conf

2. Modificar el archivo */etc/awstats/awstats.sitio1.local.conf*
   #AWSTATS.SITIO1.LOCAL.CONF
   #/ETC/AWSTATS/AWSTATS.SITIO1.LOCAL.CONF

   LogFile="/var/log/apache2/sitio1.local/access_log"
   SiteDomain="sitio1.local"

3. Generar las primeras estadísticas
   sudo /usr/lib/cgi-bin/awstats.pl -update -config=sitio1.local

4. Acceder desde el cliente web *sitio1.local/awstats/awstats.pl*

SERVIDOR WEB APACHE

WEB ANALYTIC PIWIK

PIWIK es una alternativa de código abierto a *google analytics* la cual permite, en tiempo real, obtener informes detallados de las visitas de tu web, los buscadores y palabras claves que usan, el idioma que hablan, las páginas más populares de la web, etc…

Paso a paso. Instalación y puesta en marcha de *PIWIK*

1. Descargar *PIWIK* desde su página Web
   wget http://piwik.org/latest.zip

2. Crear un *VirtualHost* en nuestro servidor *apache*
   #VIRTUALHOST PIWIK
   #/ETC/APACHE2/SITES-AVAILABLE/PIWIK

   <VirtualHost 192.168.1.XX:80>
   ServerName piwik.local
   DocumentRoot /var/ww/piwik

   ErrorLog /var/log/apache2/error.piwik.log
   CustomLog /var/log/apache2/access.piwik.log combined
   </VirtualHost>

3. Descomprimir y mover el contenido de la carpeta *piwik* a la carpeta del host virtual
   sudo unzip latest.zip
   sudo mv latest/piwik/ /var/www/piwik*

4. Modificar permisos a las carpetas *tmp/* y *config/*
   sudo chmod a+w /var/www/piwik/tmp
   sudo chmod a+w /var/www/piwik/config

5. Reiniciar *apache* y acceder a la instalación de *PIWIK* mediante la URL *piwik.local*

MOD SETENVIF

El módulo ***setenvif*** permite crear nuestras propias variables de entorno a través de expresiones regulares

DIRECTIVAS MOD SETENVIF

DIRECTIVA	DESCRIPCIÓN
BrowserMatch	Crear o modifica variables de entorno si una expresión regular dada coincide con el contenido de la cabecera *HTTP User Agent*
BrowserMatchNoCase	
SetEnvIf	Crear o modifica variables de entorno si una expresión regular dada coincide con el contenido de la cabecera *HTTP*
SetEnvIfNoCase	

CAMPOS CABECERA HTTP SOPORTADOS POR SETENVIF | SETENVIFNOCASE

CAMPO	DESCRIPCIÓN
User-Agent	Qué user-agent hace la petición
Referer	Desde donde se realiza la petición

ATRIBUTOS PETICIÓN HTTP SOPORTADOS POR SETENVIF | SETENVIFNOCASE

OPCIÓN	DESCRIPCIÓN
Remote_Host	Nombre del ordenador cliente que realiza una petición
Remote_Addr	Dirección *IP* del ordenador cliente
Server_Addr	Dirección *IP* del servidor cliente
Request_Method	Nombre del método utilizado en la petición, *GET*, *POST*, etc…
Request_Protocol	Versión del protocolo utilizado, *HTTP/1.1*, *HTTP/1.0*, etc…
Request_URI	Una parte de la *URL* solicitada

Ejemplo. Evitar peticiones a archivos *.jpg* y *.txt*

SetEnvIf Request_URI "\.txt$" archivos_prohibidos
SetEnvIf Request_URI "\.jpg$" archivos_prohibidos
Deny from env=archivos_prohibidos

Ejemplo. Evitar peticiones de la dirección *IP 192.168.1.XX*

SetEnvIf Remote_Addr "192\.168\.1\.XX" IP_prohibida
Deny from env=IP_prohibida

Ejemplo. Evitar peticiones del sitio Web *http://www.zen-cart.cn*

SetEnvIf Referer "http://www.zen-cart.con" spam
Deny from env=spam

NOTA Para encontrar conocidos *user-agents* maliciosos, *spammers*, etc… podéis acceder a la página *http://www.projecthoneypot.org*

MOD REWRITE

El módulo ***rewrite*** permite modificar *URL*s mediante reglas y condiciones en función de variables de entorno, variables de servidor, cabeceras *HTTP* o marcas de tiempo a través de expresiones regulares

Rewrite está instalado pero no habilitado, por tanto, para utilizarlo tecleamos *sudo a2enmod rewrite*

DIRECTIVAS REWRITE

DIRECTIVA	DESCRIPCIÓN
RewriteEngine	Habilita *On* o no *Off* el uso del módulo ***rewrite***
RewriteCond *condición cadena*	Indica la *condición* a cumplir por la *cadena* de texto La *condición* puede usar variables de entorno de la forma *%{VARIABLE}*
RewriteRule *patrón sustitución banderas*	Indica el *patrón* que hace el reenvío a la *URL* indicada por *sustitución*

VARIABLES DE ENTORNO

VARIABLE	DESCRIPCIÓN
HTTP_USER_AGENT	Contenido de la cabecera *HTTP user-agent*
HTTP_REFERER	Dirección de la página empleada por el *user-agent*
HTTP_HOST	Contenido de la cabecera *HTTP host*
DOCUMENT_ROOT	Directorio raíz de los documentos del servidor
REMOTE_HOST	Nombre del cliente
REMOTE_PORT	Puerto empleado para la comunicación con el servidor
REMOTE_USER	Usuario cliente autenticado
REMOTE_ADDR	Dirección *IP* del cliente

Las *banderas* de la directiva *RewriteRule* indican las opciones que queremos que se realicen. Las más comunes son *F*, prohibido, *L*, no continuar aplicando directivas *RewriteRule* y *R*, devolver al servidor un código *3XX*

Tanto la directiva *RewriteCond* como *RewriteRule* hacen uso de expresiones regulares *Perl*

RESUMEN EXPRESIONES PERL

EXPRESIÓN	DESCRIPCIÓN
.	Carácter simple
[AB…]	Cualquier carácter indicado
[^AB…]	Cualquier carácter no indicado
A | B	Caracteres alternativos
A?	0 o 1 carácter
A*	0 o más caracteres
A+	1 o más caracteres
(AB…)	Agrupación
^	Comienzo de línea
$	Final de línea
\A	Escape de caracteres
!	Negación

NOTA Estas directivas pueden incluirse dentro de un archivo *.haccess* o entre una entrada *<Directory>…</Directory>*

Ejemplo. Reenvío *403* de una petición desde la *IP 192.168.1.XX*

```
RewriteEngine On
RewriteCond %{REMOTE_ADDR} ^192\.168\.1\.XX$
RewriteRule ^(.*)$ [F]
```

Ejemplo. Reenvío de una conexión no segura a una conexión segura

```
RewriteEngine On
RewriteCond %{SERVER_PORT}  !^443$
RewriteRule ^(.*)$ http://%{SERVER_NAME}$1 [R,L]
```

SERVIDOR WEB APACHE

ANÁLISIS DE VULNERAVILIDADES WEB NIKTO

Nikto es un scanner Web de código abierto escrito en *Perl* utilizado para analizar vulnerabilidades en servidores *HTTP*. La URL del proyecto es *cirt.net/nikto2*

Para utilizarlo, se puede instalar desde los repositorios de *Ubuntu* mediante el comando *sudo apt-get install nikto*

La sintaxis básica del comando ***nikto*** es

nikto -h host [opciones]

OPCIONES BÁSICAS

OPCIÓN		DESCRIPCIÓN
-update		Actualiza los plugins y bases de datos de ***nikto***
-host	-h	Indica el servidor a escanear mediante su dirección *IP*, nombre o lista de host a escanear Si no se especifica utilizará el puerto 80 para la comunicación
-port	-p	Indica el puerto a escanear
-output	-o	Guardará un archivo de registro
-Format	-F	Indica el formato del archivo de registro indicado en la opción *-output* Los posibles valores son *htm csv txt* o *xml*
-evasion	-e	Habilita la detección de intrusiones
-Tuning	-T	Indica el tipo de escaneo a realizar Si no se especifica se realizarán todos los tipos de escaneo

TIPOS DE ESCANEO -TUNING | -T

OPCIÓN	DESCRIPCIÓN
0	Upload de archivos
1	Ficheros interesantes | visualizar los logs
2	Malas configuraciones | ficheros por defecto
3	Revelación de información
4	Inyección XSS Script HTML
5	Recuperación de archivos remotos | directorio Web raíz
6	Denegación de servicio
7	Recuperación de archivos remotos | todo el sistema
8	Ejecución de comandos | consola remota
9	Inyección SQL
a	Salto de autenticación
b	Identificación de software
c	Inclusión remota de código
x	negación

ANEXO. INSTALACIÓN DE UN SERVIDOR LAMP DESDE LOS REPOSITORIOS

Un servidor *LAMP LINUX*, *APACHE*, *MYSQL* y *PHP* es un servidor que da soporte a páginas Web dinámicas que utilicen *PHP* como lenguaje de servidor y *MySQL* como gestor de bases de datos

1. Instalar el servidor web *apache*
   sudo apt-get install apache2-mpm-prefork

2. Instalar el gestor de bases de datos *MySQL* y asegurar dicha instalación
   sudo apt-get install mysql-server
   sudo mysql_secure_installation

3. Instalar *PHP*
   sudo apt-get install php5 php-pear php5-mysql