Sistemas Operativos

ICC329-1

Profesor: Carlos E. Méndez Dumestre
carlos.mendez@ufrontera.cl

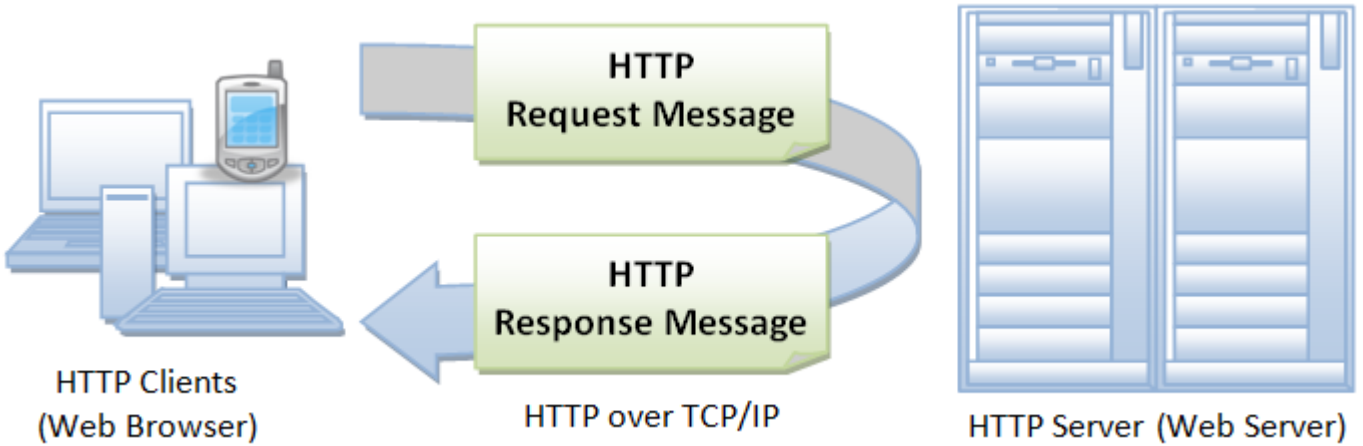
El Protocolo HTTP

HTTP (Hypertext Transfer Protocol, o Protocolo de Trasferencia de Hipertext), es el método utilizado para transferir o transportar información en la Red Mundial (***WWW***, World Wide Web). Su propósito original fue el proveer una forma depublicar y recupertar documentos ***HTML***.

El desarrollo del protocolo fue coordinado por World Wide Web Consortium y la IETF (Internet Engineering Task Force, o Fuerza de Trabajo en Ingeniería de Internet), culminando con la publicación de varios RFC (Request For Comments), de entre los que destaca el RFC 2616, mismo que define la versión 1.1 del protocolo, que es el utilizado hoy en día.

HTTP es un protocolo de solicitud y respuesta a través de TCP, entre agentes de usuario (Navegadores, motores de índice y otras herramientas) y servidores, regularmente utilizando el puerto 80. Entre la comunicación entre éstos puede intervenir como servidores Intermediarios (Proxies), puertas de enlace y túneles.

URL: http://tools.ietf.org/html/rfc2616



Apache

Apache Software Foundation (***http://www.apache.org/***) es una organización sin fines de lucro creada para dar soporte a los ***proyectos de software*** bajo la denominación Apache, incluyendo el popular servidor HTTP Apache. La ASF se formó a partir del llamado Grupo Apache y fue registrada en Delaware (Estados Unidos), en junio de 1999.

AFS soporta muchos proyectos en distintas áreas de software, entre los más populares están:
- Apache ***Derby*** (base de datos)
- Apache ***Tomcat*** (Servidor web de aplicaciones)
- Apache ***Batik*** (Librerías Gráficas SVG basadas en Java)
- Apache ***Http Server*** (Web Server)

https://projects.apache.org/

Apache Http Server:
Es un servidor HTTP, de código abierto y licenciamiento libre, que funciona en ***GNU/Linux***, sistemas operativos derivados de ***Unix***™, ***Windows***, Novell Netware y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continua siendo el servidor HTTP más utilizado, siendo además el servidor de facto contra el cual se realizan las pruebas comparativas y de desempeño para otros productos competidores. Apache es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por Apache Software Foundation.

Instalación de Apache Web Server

La instalación de Apache Web Server es muy simple mediante el gestor de paquetes YUM:

yum install httpd httpd-devel

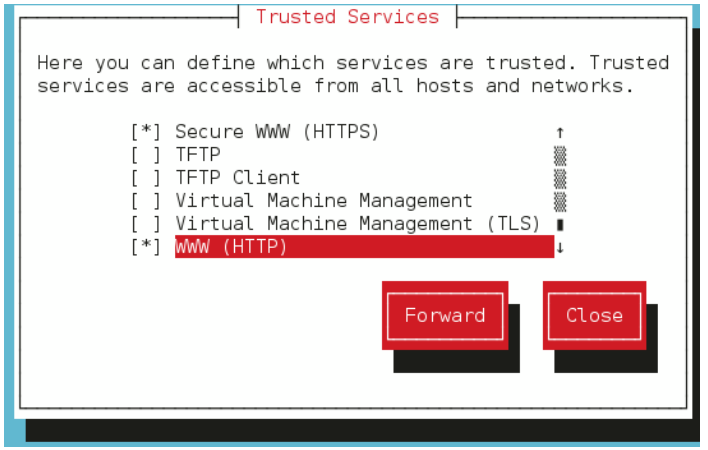
Una vez terminada la descarga e instalación de los paquetes desde los repositorios, es necesario revisar algunas configuraciones en el servidor antes de trabajar con el servidor web:

Firewall

Habilitar los permisos de conexión al puerto 80 – http (y opcionalmente al puerto 443 – https) en el firewall del servidor.
La manera más sencilla de hacer esto es usando la herramienta de configuración:

system-config-firewall-tui

Una vez cargada la consola, se selecciona la opción “Customize” y luego se selecciona (usando la barra espaciadora) los servicios WWW y Secure WWW.



Una opción más “genérica” es ingresar directamente estas reglas mediante el comando IPTABLES, que las incluye diréctamente en el firewall de linux:

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
service iptables save
```

Instalación de Apache Web Server

SELinux

Security-Enhanced Linux (SELinux) es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso al sistema. Es usado principalmente para restringir el acceso libre de servicios al sistema.

Para efectos de laboratorio, puede considerarse desabilitar SELinux (***setenforce 0***), pero en ambientes de producción, será necesario revisar las reglas necesarias para permitir el correcto funcionamiento del servidor web.

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/

Arranque del sistema

Debido a que el servidor web se instala como un servicio del sistema y que usualmente se desea tener el este servicio siempre operativo, es conveniente configurar el servicio httpd para que arranque automáticamente cada vez que arranque el sistema operativo.

```
chkconfig httpd on
```

Una vez revisadas las configuraciones anteriores, se puede iniciar el servicio web mediante el comando:

```
service httpd start
```

Obs:

En GNU/Linux, el comando ***service*** se utiliza para controlar la mayoría de los servicios del sistema.
Para el caso del servicio httpd su utlilización es:

```
service httpd {start|stop|restart|condrestart|try-restart|force-reload|reload|status|fullstatus|graceful|help|configtest}
```

Configuración de Apache Web Server

El archivo httpd.conf

El archivo de configuración del servidor Web Apache es ***/etc/httpd/conf/httpd.conf***. Este archivo está bien comentado y es bastante autoexplicativo.
Si necesita configurar Apache sólo tiene que modificar el archivo ***httpd.conf*** y después recargar o bien apagar y arrancar el proceso del comando httpd.
Si no se desea modificar diréctamente el archivo ***httpd.conf***, cualquier ajuste que se requiera realizar, ya sea para configurar ***virtual hosts***, u otra funcionalidad adicional, se puede realizar sin tocar el archivo principal de configuración (/etc/httpd/conf/httpd.conf), utilizando cualquier archivo con extensión ****.conf*** dentro del directorio ***/etc/httpd/conf.d/***.

Antes de modificar el archivo httpd.conf,es una buena práctica copiar el archivo original (dándole por ejemplo, el nombre httpd.conf-BKP) para disponer de una copia de seguridad en caso de errores.
Además, se dispone del comando ***`service httpd configtest`*** *para probar la configuración antes de cargarla o reiniciar el servicio.*

Secciones de httpd.conf

El archivo de configuración de apache está compuesto básicamente por 3 secciones:
Section 1: Global Environment:
The directives in this section affect the overall operation of Apache, such as the number of concurrent requests it can handle or where it can find its configuration files.

Section 2: 'Main' server configuration
The directives in this section set up the values used by the 'main' server, which responds to any requests that aren't handled by a <VirtualHost> definition.

Section 3: Virtual HostsVirtualHost
If you want to maintain multiple domains/hostnames on your machine you can setup VirtualHost containers for them.

Configuración de Apache Web Server

Principales directivas

ServerRoot
El comando ServerRoot es el directorio principal donde se encuentran todos los archivos del servidor. Tanto el servidor seguro como el no seguro utilizan un comando ServerRoot del /etc/httpd.

Timeout
El comando Timeout define, en segundos, el tiempo que el servidor esperará para recibir y enviar peticiones durante la comunicación. Específicamente, el comando Timeout define cuánto esperará el servidor para recibir peticiones GET, cuánto esperará para recibir paquetes TCP en una petición POST o PUT y cuánto esperará entre una ACK y otra respondiendo a paquetes TCP. El comando Timeout está ajustado a 300 segundos, que es el tiempo apropiado para la mayoría de las situaciones.

KeepAlive
El comando KeepAlive determina si el servidor permitirá varias conexiones a la vez (p.e., más de una petición por conexión). KeepAlive puede usarse para impedir que un cliente consuma muchos recursos del servidor. El comando KeepAlive aparece ya en on por defecto, lo que significa que se permiten varias conexiones a la vez. Puede ponerse en off para desactivarlas. Consulte la sección de nombre MaxKeepAliveRequests para conocer un método alternativo para limitar las peticiones.

MaxClients
El comando MaxClients establece un límite al total de los procesos del servidor (es decir, clientes conectados simultáneamente) que se ejecutan a la vez. Debe mantener el comando MaxClients a un valor alto (el valor por defecto es 150), porque no se permitirán nuevas conexiones una vez que se alcance el número máximo de clientes simultáneamente conectados. El valor del comando MaxClients no puede superar el 256 sin que se haya recompilado Apache. La principal razón de tener el parámetro MaxClients es evitar que un servidor errático vuelva inestable al sistema operativo.

Configuración de Apache Web Server

Principales directivas

Listen
El comando Listen establece los puertos en los que secure Web server acepta las peticiones entrantes. secure Web server está configurado para escuchar en el puerto 80 para comunicaciones no seguras y (en máquinas virtuales que define el servidor seguro) en el puerto 443 para comunicaciones seguras.

Para puertos por debajo de 1024, el comando httpd deberá ser ejecutado como root. Para el puerto 1024 y superiores, el comando httpd puede ser ejecutado como si se fuera un usuario cualquiera.

El comando Listen también se puede usar para especificar direcciones IP específicas en las cuales aceptará conexiones el servidor.

BindAddress
BindAddress es un modo de especificar en qué direcciones IP el servidor escuchará. Debería usarse la directiva Listen en su lugar si se necesita esta funcionalidad. El servidor no usa el comando BindAddress el cual ya aparece comentado en httpd.conf.

LoadModule
El comando LoadModule se usa para cargar módulos Dynamic Shared Object(DSO). Para más información sobre el soprte de los DSOs de Apache y cómo usar la directiva LoadModule, lea la sección de nombre Añadir módulos a su servidor. Nótese que el orden de los módulos es importante, así que mejor no tocarlo.

Configuración de Apache Web Server

Principales directivas

User
La directiva User establece el userid usado por el servidor para responder a peticiones. El valor de User determina el acceso al servidor. Cualquier archivo al que no pueda acceder este usuario será también inaccesible al visitante de la web. El comando predeterminado para User es apache.

Nota: A menos que sepa exactamente lo que está haciendo, no utilice el comando User como si fuese root. Usar root para User creará grandes problemas de seguridad en secure Web server.

Group
El comando Group es similar a User. Group establece el grupo en el que el servidor responde a las peticiones. El valor predeterminado del comando Group también es apache.

ServerAdmin
ServerAdmin debería ser la dirección de correo del administrador del secure Web server. Esta dirección de correo aparecerá en los mensajes de error generados por el servidor para páginas web,de tal manera que los usuarios pueden comunicar errores enviando correo al administrador. El comando ServerAdmin ya se encuentra en la dirección root@localhost.

ServerName
El comando ServerName puede usarse para establecer el nombre de la máquina del servidor diferente al nombre real de máquina como por ejemplo, usar www.your_domain.com aunque el nombre real del servidor sea foo.your_domain.com. Nótese que ServerName debe ser un nombre "Domain Name Service" (DNS) válido que se tenga derecho a usar (no basta con inventar uno).

Si se especifica ServerName, hay que asegurarse de incluir la pareja nombre-dirección IP en el archivo /etc/hosts.

Configuración de Apache Web Server

Principales directivas

DocumentRoot
DocumentRoot es el directorio que contiene la mayoría de los archivos HTML que se entregarán en respuesta a peticiones. El directorio predeterminado DocumentRoot para servidores seguros y no seguros es /var/www/html. Por ejemplo, el servidor puede recibir una petición para el siguiente documento:

http://dominio.cl/pagina1.html
El servidor buscará el archivo en el siguiente directorio por defecto:
/var/www/html/pagina1.html

Directory
Las etiquetas <Directory /path/to/directory> y </Directory> se usan para agrupar directivas de configuración que sólo se aplican a ese directorio y sus subdirectorios. Cualquier directiva aplicable a un directorio puede usarse en las etiquetas <Directory>. Las etiquetas <File> pueden aplicarse de la misma forma a un archivo específico.

DirectoryIndex
DirectoryIndex es la página por defecto que entrega el servidor cuando hay una petición de índice de un directorio especificado con una barra (/) al final del nombre del directorio.

Por ejemplo, cuando un usuario pide la página http://your_domain/this_directory/, recibe la página DirectoryIndex si existe, o un listado generado por el servidor. El valor por defecto para DirectoryIndex es index.html, index.htm index.shtml e index.cgi. El servidor intentará encontrar cualquiera de estos cuatro, y entregará el primero que encuentre. Si no encuentra ninguno y si Options Indexes se encuentra en el directorio, el servidor generará un listado, en formato HTML, de los subdirectorios y archivos del directorio.

Apache Alias

Normalmente cuando se indica una ruta como la siguiente

http://servidor.web/carpeta1/archivo.html

Estamos buscando el archivo llamado "***archivo.html***" en la carpeta llamada "***carpeta1***" que se debe encontrar en la carpeta principal del servidor web donde se alojan las páginas web (***DocumentRoot***).

En la configuración del servidor web Apache es posible crear un ***alias*** para redireccionar la ruta de una dirección web a una carpeta que no se encuentre forzosamente dentro de la especificada como DocumentRoot.

Para ello se debe usar la directiva alias. con el siguiente formato: **`Alias ruta-URL ruta-carpeta.`**
Una vez establecido el alias, se debe asignar los permisos adecuados usando la directiva ***<Directory>.***

Por ejemplo, supongamos que la carpeta principal del servidor web se encuentra en ***/var/www/html***, y queremos dar acceso web a la carpeta "***/var/galeria/mis_fotos***" usando la direccion: ***http://servidor.web/fotos.***

Para ello, se debe indicar la directiva ***alias*** junto con el nombre del sufijo indicado en la dirección web y la ruta a la carpeta real:

```
Alias /fotos "/var/galeria/mis_fotos/"

<Directory "/var/galeria/mis_fotos">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Apache seguridad básica con HTPASSWD y HTACCESS

Htpasswd y **htaccess** proporcionan una manera de implementar un control de acceso básico para los directorios de un servidor web Apache.
Este sistema ofrece control de acceso a nivel de usuario y grupo mediante tres archivos de configuración:

.htpasswd:

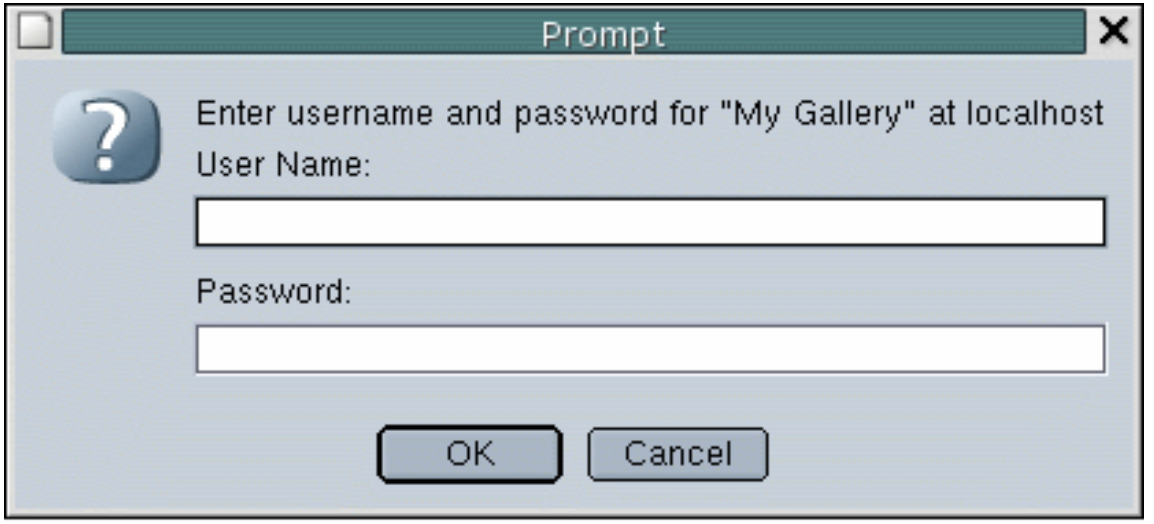
La base de datos de contraseñas, almacena los pares de nombre usuario/password.
Cuando los usuarios piden acceso a un directorio web protegido, el servidor solicita nombre de usuario y contraseña, los cuales compara contra el contenido del archivo .htpasswd para validarlos.

.htgroup:

El archivo de grupos htpasswd. Almacena información de los grupos creados y de los miembros de cada grupo. Éste archivo es opcional, sólo si se desea implementar control de acceso por grupos.

.htaccess:

Almacena las normas de acceso (allow, deny – permitir, denegar), la ubicación de los archivos de configuración, el método de autentificación, etc.
(Éste archivo es obligatorio)



Apache seguridad básica con HTPASSWD y HTACCESS

Cómo implementar la seguridad:

Supongamos que tenemos un directorio “datos1” el que se desea implementar control de acceso por usuario y password. (Éste directorio debe tener previamente creado su alias en la configuración del servidor web – httpd.conf)

1. **Crear el archivo .htpasswd:**

   Para crear un nuevo archivo de contraseñas se debe ejecutar el comando htpasswd con el switch –c, el nombre del archivo de contraseñas y el nombre del usuario:

   ```
   /usr/bin/htpasswd –c .htpasswd usuario
   ```

   (El commando solicitará ingresar la password para el usuario y luego re-ingresarla.)

Observaciones:

Si ya existe el archivo ***.htpasswd*** *y se desea agregar nuevos usuarios, se debe omitir el switch “–c”:*

```
/usr/bin/htpasswd .htpasswd usuario2
```

El archivo ***.htpasswd*** *debe estar idealmente almacenado en un directorio al que sólo tengan acceso usuarios restringidos (por ej. El usuario root y el usuario de apache)*

La creación del archivo de grupos ***.htgroup****, es muy simple:*

Con un editor de texto (vi ó nano) se deben agregar las líneas indicando el nombre del grupo y sus integrantes (separados por un espacio – sin comas), por ej:

```
grupo1: usuario usuario1
```

Apache seguridad básica con HTPASSWD y HTACCESS

2. <u>Crear el archivo .htaccess:</u>

Con un editor de texto (vi ó nano) se debe crear el archivo con las siguientes líneas:

```
AuthUserFile /etc/security/.htpasswd
AuthGroupFile /etc/security/.htgroup
AuthName usuario
AuthType Basic

<Limit GET POST>
     Require user usuario
     #Require group grupo1
</Limit>
```

Este archivo debe estar almacenado dentro del directorio que se desea “segurizar”.

AuthUserFile *indica la ubicación del archivo de usuarios y passwords, para el ejemplo se ha puesto en el directorio /etc/security.*
AuthGroupFile *indica la ubicación del archive de usuarios grupos, para el ejemplo se ha puesto en el directorio /etc/security, si no se desea utilizar un archivo de grupos, puede dejarse apuntando a /dev/null.*
AuthName *indica el texto que aparecerá cuando muestre el cuadro de diálogo de autentificación. (si es más de una palabra debe ir entre comillas, por ej: “Autentificar Usuario”)*
AuthType *indica el tipo de autentificación que se utilizará, en este caso, Básica.*

<Limit></Limit> *controla qué usuarios o grupos tienen acceso permitido, qué tipo de acceso pueden obtener (GET, POST, PUT) y el orden en que se evalúan estas normas.*
Se puede usar alternativamente:

Require user*: para implementar acceso a cuentas de usuario individuales.*
Require group*: para implementar acceso a múltiples usuarios mediante la creación de grupos de usuarios.*

OBS: Se debe optar por sólo una de estas dos alternativas, ámbas alternativas al mismo tiempo no funcionarán. (Una de ellas tomará control y anulará a la otra)

Apache seguridad básica con HTPASSWD y HTACCESS

3. **Habilitar la seguridad para el alias** del directorio en el archivo de configuración del servidor web Apache (httpd.conf).
   Se debe editar el archivo httph.conf (/etc/httpd/conf/httpd.conf) y crear ó modificar el Alias de la siguiente forma:

Alias seguridad htaccess	Alias Con seguridad htaccess habilitada

`Alias /datos1 "/var/datos1/"`
 	
`<Directory "/var/datos1">`
`Options Indexes MultiViews`
	`Alias /datos1 "/var/datos1/"`
	
	`<Directory "/var/datos1">`
	`Options Indexes MultiViews`

`AllowOverride None`	**`AllowOverride AuthConfig`**

`Order allow,deny`
`Allow from all`
`</Directory>`
	`Order allow,deny`
	`Allow from all`
	`</Directory>`