Autenticación y manejo de sesiones con PHP

Autenticación y manejo de sesiones

Autentificación de usuarios y control de acceso

- Para identificar a los usuarios que visitan un sitio web, se pueden utilizar distintos métodos como el DNI digital o certificados digitales de usuario, pero el más extendido es solicitar al usuario cierta información que solo él conoce: la combinación de un **nombre de usuario y una contraseña**.
- En la mayoría de los casos es importante implantar en este tipo de aplicaciones web, las que acceden a bases de datos, algún **mecanismo de control de acceso** que obligue al usuario a identificarse. Una vez identificado, se puede limitar el uso que puede hacer de la información.
- En los ejemplos de esta unidad, la información de autentificación (nombre y contraseña de los usuarios) se envía en texto plano desde el navegador hasta el servidor web. Esta práctica es altamente insegura y nunca debe usarse sin un protocolo como **HTTPS** que permita cifrar las comunicaciones con el servidor web.

Autenticación y manejo de sesiones

Mecanismos de autenticación

- El protocolo HTTP ofrece un método sencillo para autentificar a los usuarios. El proceso es el siguiente:
    - El servidor web debe proveer algún método para definir los usuarios que se utilizarán y cómo se pueden autentificar.
    - Cuando un usuario no autentificado intenta acceder a un recurso restringido, el servidor web responde con un error de "Acceso no autorizado" (código 401).
    - El navegador recibe el error y abre una ventana para solicitar al usuario que se autentifique mediante su nombre y contraseña.
    - La información de autenticación del usuario se envía al servidor, que la verifica y decide si permite o no el acceso al recurso solicitado. Esta información se mantiene en el navegador para utilizarse en posteriores peticiones a ese servidor.
    - Si las credenciales son erróneas Apache muestra "Acceso no autorizado" (código 401).

Autenticación y manejo de sesiones

Mecanismos de autenticación

- En el servidor web Apache existe una utilidad en línea de comandos, htpasswd, que permite almacenar en un fichero una lista de usuarios y sus respectivas contraseñas.
- La información relativa a las contraseñas se almacena cifrada; aun así, es conveniente crear este fichero en un lugar no accesible por los usuarios del servidor web.
- Para crear el fichero de usuarios "proyecto.pass" y añadirle el usuario "gestor", podemos ejecutar:

```
sudo htpasswd -c proyecto.pass gestor
```

Autenticación y manejo de sesiones

Mecanismos de autenticación

- Para indicarle al servidor Apache qué recursos tienen acceso restringido, una opción es crear un fichero .htaccess en el directorio en que se encuentren, con las siguientes directivas:

```
AuthName "Contenido restringido"
AuthType Basic
AuthUserFile /etc/apache2/users/proyecto.pass
require valid-user
```

- Además tendrás que asegurarte de que en la configuración de Apache se utiliza la directiva AllowOverride para que se aplique correctamente la configuración que figura en los ficheros .htaccess.

Autenticación y manejo de sesiones

Mecanismos de autenticación

- Desde PHP puedes acceder a la información de autentificación HTTP que ha introducido el usuario utilizando el array "superglobal" $_SERVER. Los valores se muestran en la tabla adjunta.

Valores del array $_SERVER relacionados con la autentificación

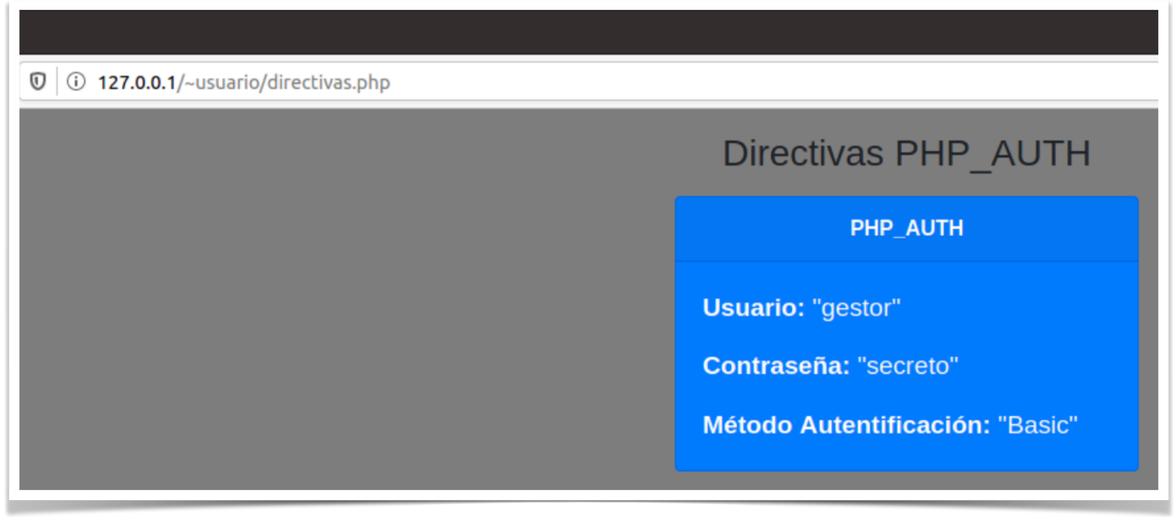
Valor	Contenido
$_SERVER['PHP_AUTH_USER']	Nombre de usuario que se ha introducido.
$_SERVER['PHP_AUTH_PW']	Contraseña introducida.
$_SERVER['AUTH_TYPE']	Método HTTP usado para autentificar. Puede ser **Basic** o **Digest**.

Autenticación y manejo de sesiones

Mecanismos de autenticación

- Si creas una página web que muestre los valores de estas variables, y preparas el servidor web para utilizar autentificación HTTP, cuando accedas a esa página con el usuario "gestor" y contraseña "secreto" podrás observar los datos de la imagen izquierda de abajo..

- Si no introduces un usuario/contraseña válidos, el navegador te mostrará el error 401.



Autenticación y manejo de sesiones

Mecanismos de autenticación

- En PHP puedes usar la función header para forzar a que el servidor envíe un error de "Acceso no autorizado" (código 401). De esta forma no es necesario utilizar ficheros .htaccess para indicarle a Apache qué recursos están restringidos. En su lugar, puedes añadir las siguientes líneas en tus páginas:

```
<?php
if (!isset($_SERVER['PHP_AUTH_USER'])) {
    header('WWW-Authenticate: Basic Realm="Contenido restringido"');
    header('HTTP/1.0 401 Unauthorized’);
    echo “<h2>Usuario no reconocido</h2>”;
    exit();
}
?>
```

Autenticación y manejo de sesiones

Cookies

- Una cookie es un fichero de texto que un sitio web guarda en el entorno del usuario del navegador.
- Su uso más típico es el almacenamiento de las preferencias del usuario (por ejemplo, el idioma en que se deben mostrar las páginas), para que no tenga que volver a indicarlas la próxima vez que visite el sitio.
- En PHP, para almacenar una cookie en el navegador del usuario, puedes utilizar la función setcookie. (solo es obligatorio el primer parámetro)

```
setcookie("nombre_usuario", $_SERVER['PHP_AUTH_USER'], time()+3600);
```

Autenticación y manejo de sesiones

Cookies

- Las cookies se transmiten entre el navegador y el servidor web utilizando los encabezados del protocolo HTTP. Por lo tanto se debe invocar a setcookie antes de mostrar el primer elemento HTML en la página.
- Cuando accedes a un sitio web, el navegador le envía de forma automática todo el contenido de las cookies que almacene relativas a ese sitio en concreto.
- Desde PHP puedes acceder a esta información por medio del array $_COOKIE.
- En última instancia la disponibilidad está controlada por el cliente (puede bloquearlas, reinstalar el equipo, utilizar otro navegador, etc.)

Autenticación y manejo de sesiones

Cookies

- Podemos modificar una cookie simplemente asignándole un nuevo valor.
- Para eliminar una cookie tenemos que invocar a setcookie pasándole una fecha de caducidad anterior a la actual.

```
setcookie("nombre_usuario", "", time()-10);
```

Autenticación y manejo de sesiones

Manejo de sesiones

- Las sesiones permiten almacenar información de usuarios individuales a través de un identificador de sesión (ID).
- Esto permite mantener accesible dicha información a medida que el usuario navega por diferentes páginas.
- Los ID de sesión son enviados generalmente al navegador mediante cookies. El ID se utiliza para recuperar los datos asociados a la sesión.
- La ausencia del ID de sesión permite informar a PHP de que tiene que crear una nueva sesión y por lo tanto su ID asociado.

Autenticación y manejo de sesiones

Manejo de sesiones

- Funcionamiento de las sesiones:
    - Cuando se intenta crear una sesión:
        - Si se detecta que ya existe un ID de sesión, se recupera (se cargan sus datos en el array superglobal $_SESSION).
        - Si no existe, se crea la sesión.
    - Cuando finaliza la ejecución del PHP, se guarda el contenido de $_SESSION en ficheros. Por defecto, en la ruta especificada en “session.save_path”.
- Las sesiones se pueden iniciar manualmente con la función session_start().
- Si el parámetro “session.auto-start” está definido a “1”, la sesión se creará automáticamente al ejecutarse el PHP.

Autenticación y manejo de sesiones

Inicio de una sesión

- Si has activado la directiva session.auto_start en la configuración de PHP, la sesión comenzará automáticamente en cuanto un usuario se conecte a tu sitio web.
- En caso de que ese usuario ya haya abierto una sesión con anterioridad, y esta no se haya eliminado, en lugar de abrir una nueva sesión se reanudará la anterior.
- Si por el contrario, decides no utilizar el inicio automático de sesiones, deberás ejecutar la función session_start para indicar a PHP que inicie una nueva sesión o reanude la anterior.
- Para poder iniciar una sesión utilizando session_start, tendrás que hacer las llamadas a esta función antes de que la página web muestre información en el navegador.

Autenticación y manejo de sesiones

Fin de una sesión

- Mientras la sesión permanece abierta, puedes utilizar la variable superglobal $_SESSION para añadir información a la sesión del usuario, o para acceder a la información almacenada en la sesión.
- session_destroy() elimina los datos asociados a la sesión.