Práctica 2.2: Autenticación en Nginx

Tabla de Contenido

Requisitos antes de comenzar la práctica

1. La práctica 2.1 ha de estar funcionando correctamente.
2. No empezar la práctica antes de tener la 2.1 **funcionando y comprobada**.

1. Introducción

En el contexto de una transacción HTTP, la autenticación de **acceso básica** es un método diseñado para permitir a un navegador web, u otro programa cliente, proveer credenciales en la forma de usuario y contraseña cuando se le solicita una página al servidor.

La autenticación básica, como su nombre lo indica, es la forma más básica de autenticación disponible para las aplicaciones Web. Fue definida por primera vez en la especificación HTTP en sí y no es de ninguna manera elegante, pero cumple su función.

Este tipo de autenticación es el tipo más simple disponible pero adolece de importantes problemas de seguridad que no la hacen recomendable en muchas situaciones. No requiere el uso ni de cookies, ni de identificadores de sesión, ni de página de ingreso.

1.1. Paquetes necesarios

Para esta práctica podemos utilizar la herramienta `openssl` para crear las contraseñas.

En primer lugar debemos comprobar si el paquete está instalado:

```
dpkg -l | grep openssl
```

Y si no lo estuviera, instalarlo.

1.2. Creación de usuarios y contraseñas para el acceso web

Crearemos un archivo oculto llamado `.htpasswd` en el directorio de configuración `/etc/nginx` donde guardar nuestros usuarios y contraseñas :

```
sudo sh -c "echo -n 'vuestro_nombre:' >> /etc/nginx/.htpasswd"
```

Ahora crearemos un pasword cifrado para el usuario de forma interactiva:

```
sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"
```

o de forma no interactiva:

```
sudo sh -c "openssl passwd -apr1 'tupassword'>> /etc/nginx/.htpasswd"
```

Este proceso se podrá repetir para tantos usuarios como haga falta.

- Crea dos usuarios, uno con tu nombre y otro con tu primer apellido.
- Comprueba que el usuario y la contraseña aparecen cifrados en el fichero:

```
cat /etc/nginx/.htpasswd
```

1.3. Configurando el servidor Nginx para usar autenticación básica

Editaremos la configuración del server block sobre el cual queremos aplicar la restricción de acceso. Utilizaremos para esta autenticación el sitio web de *Perfect Learn*:

Recuerda que un *server block* es cada uno de los dominios (`server {⋯}` dentro del archivo de configuración) de alguno de los sitios web que hay en el servidor.

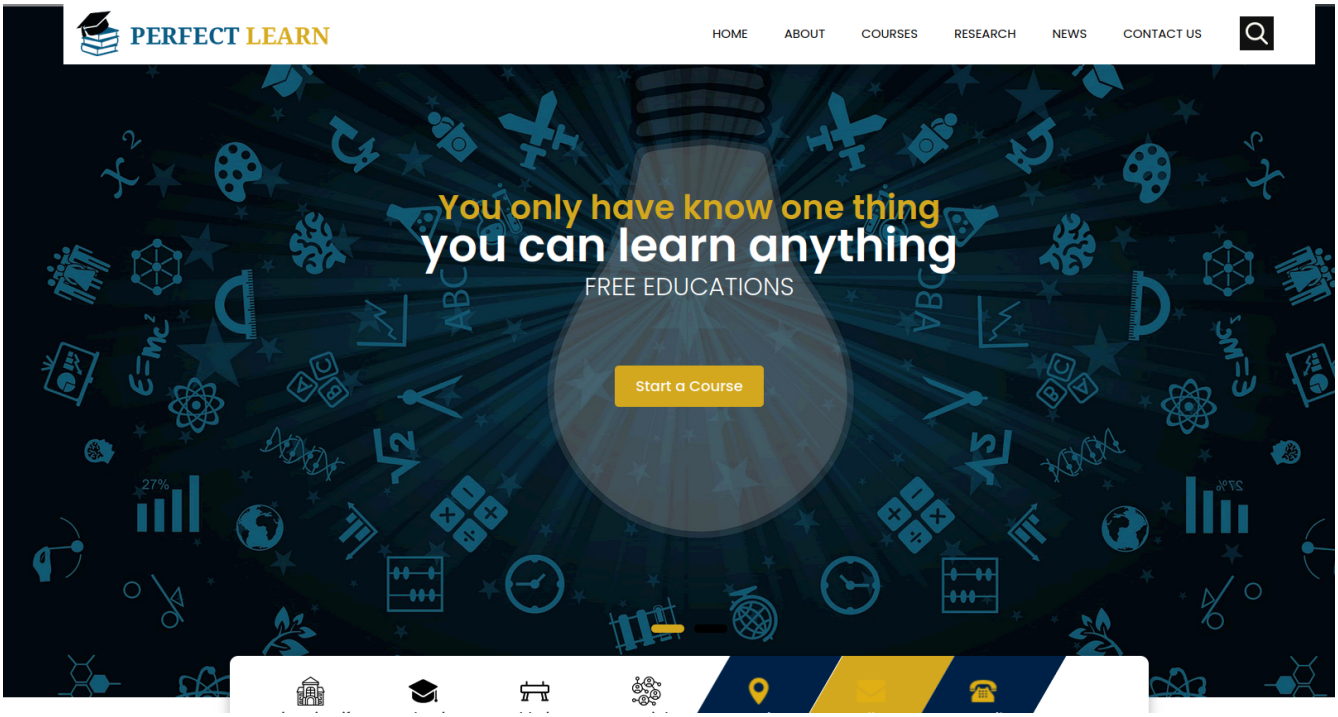

Figura 1. Perfect Learn

```
sudo nano /etc/nginx/sites-available/nombre_web
```

Debemos decidir qué recursos estarán protegidos. Nginx permite añadir restricciones a nivel de servidor o en un **location** (directorio o archivo) específico. Para nuestro ejemplo, vamos a proteger el **document root** (la raíz, la página principal) de nuestro sitio.

Utilizaremos la directiva `auth_basic` dentro del *location* y le pondremos el nombre a nuestro dominio que será mostrado al usuario al solicitar las credenciales. Por último, configuramos Nginx para que utilice el fichero que previamente hemos creado con la directiva `auth_basic_user_file` :

```
server {
    listen 80;
    listen [::]:80;

    root /var/www/deaw/html/simple-static-website;
    index index.html index.htm index.nginx-debian.html;

    server_name nombre_web;

    location / {
        auth_basic "Área restringida";
        auth_basic_user_file /etc/nginx/.htpasswd;
        try_files $uri $uri/ =404;
    }
}
```

Una vez terminada la configuración, reiniciamos el servicio para que aplique nuestra política de acceso.

```
sudo systemctl restart nginx
```

1.4. Probando la nueva configuración

1. Comprueba desde tu máquina física/anfitrión que puedes acceder a `http://nombre-sitio-web` y que se te solicita autenticación.
2. Comprueba que si decides cancelar la autenticación, se te negará el acceso al sitio con un error. ¿Qué error es?

Una vez os autenticáis con éxito, el navegador guardará esta autenticación exitosa y no volverá a pediros *usuario/contraseña*. Llegados a ese punto, si queréis volver a probar a autenticaros, tendréis que abriros una **nueva ventana privada** del navegador.

2. Tareas

2.1. T.1.

Intenta entrar primero con un usuario erróneo y luego con otro correcto. Puedes ver todos los sucesos y registros en los logs `access.log` y `error.log`.

Adjunta una captura de pantalla de los logs donde se vea que intentas entrar primero con un usuario inválido y con otro válido. Indica dónde podemos ver los errores de usuario inválido o no encontrado, así como donde podemos ver el número de error que os aparecía antes.

Cuando hemos configurado el siguiente bloque:

```
location / {
    auth_basic "Àrea restringida";
    auth_basic_user_file /etc/nginx/.htpasswd;
    try_files $uri $uri/ =404;
}
```

La autenticación se aplica al directorio/archivo que le indicamos en la declaración del `location` y que en este caso el raíz `/`.

Así pues, esta restricción se aplica al directorio raíz o base donde residen los archivos del sitio web y que es `/var/www/deaw/html/simple-static-website`

Y a todos los archivos que hay dentro, ya que no hemos especificado ninguno en concreto.

Ahora bien, vamos a probar a aplicar autenticación sólo a una parte de la web. Vamos a intentar que sólo se necesite autenticacíón para entrar a la parte de portfolio:

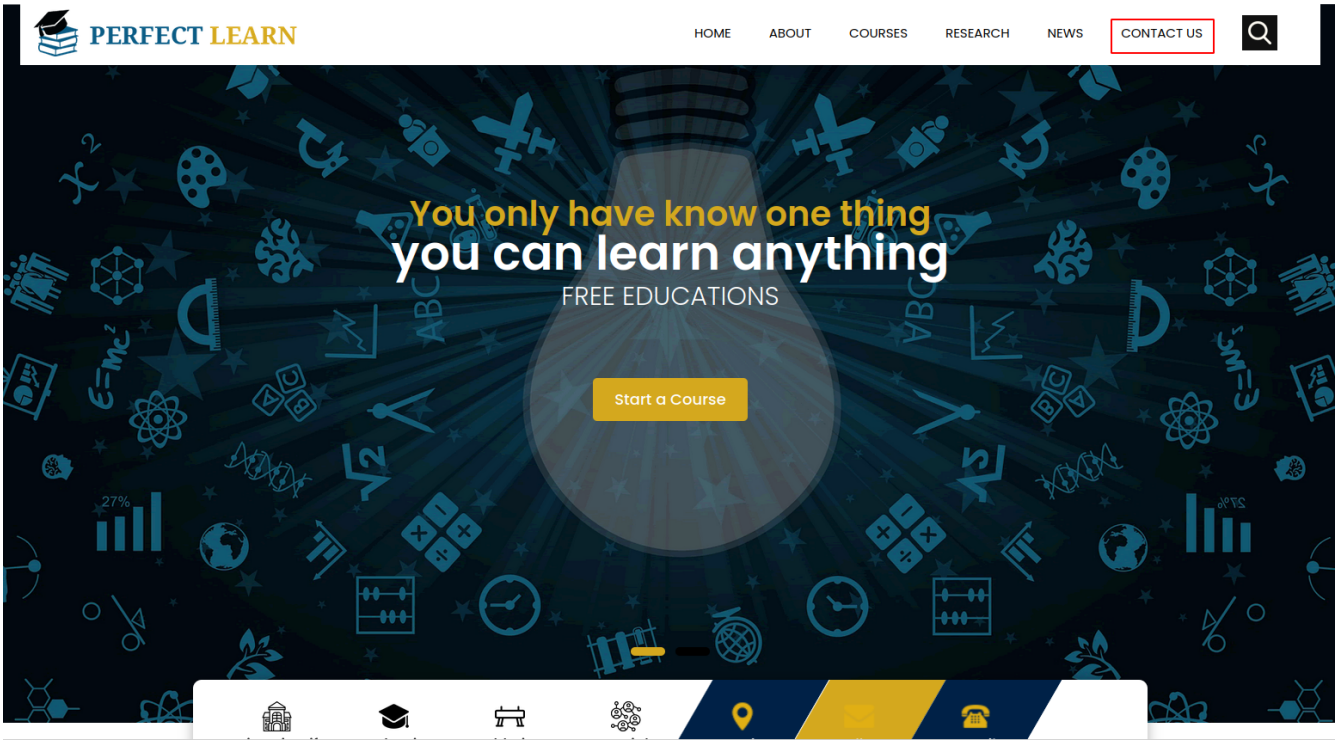

Figura 2. Acceso a Contact Us

Esta sección se corresponde con el archivo `contact.html` dentro del directorio raíz.

2.2. T.2.

Borra las dos líneas que hacen referencia a la autenticación básica en el *location* del directorio raíz. Tras ello, añade un nuevo *location* debajo con la autenticación básica para el archivo/sección `contact.html` únicamente.

2.3. Combinación de la autenticación básica con la restricción de acceso por IP

La autenticación básica HTTP puede ser combinada de forma efectiva con la restricción de acceso por dirección IP. Se pueden implementar dos escenario:

- Un usuario debe estar ambas cosas, autenticado y tener una IP válida
- Un usuario debe o bien estar autenticado, o bien tener una IP válida

Dentro del *block server* o archivo de configuración del dominio web, que recordad está en el directorio `sites-available`.

```
location /api {
  # ...
  deny 192.168.1.2;
  allow 192.168.1.1/24;
  allow 127.0.0.1;
  deny all;
```

```
}
```

El acceso se garantizará a la IP `192.168.1.1/24`, excluyendo a la dirección `192.168.1.2`.

Hay que tener en cuenta que las directivas `allow` y `deny` se irán aplicando en el orden en el que aparecen el archivo.

Aquí aplican sobre la *location* `/api` (esto es sólo un ejemplo de un hipotético directorio o archivo), pero podrían aplicar sobre cualquiera, incluida todo el sitio web, la location raíz `/`.

La última directiva `deny all` quiere decir que por defecto denegaremos el acceso a todo el mundo. Por eso hay que poner los `allow` y `deny` más específicos justo antes de esta, porque al evaluarse en orden de aparición, si los pusiéramos debajo se denegaría el acceso a todo el mundo, puesto que `deny all` sería lo primero que se evaluaría.

Combinar la restricción IP y la autenticación HTTP con la directiva `satisfy`.

Si establecemos el valor de la directiva a `all`, el acceso se permite si el cliente satisface ambas condiciones (IP y usario válido). Si lo establecemos a `any`, el acceso se permite si se satisface al menos una de las dos condiciones.

```
location /api {
  # ...
  satisfy all;

  deny 192.168.1.2;
  allow 192.168.1.1/24;
  allow 127.0.0.1;
  deny all;

  auth_basic "Administrator's Area";
  auth_basic_user_file /etc/nginx/.htpasswd;
}
```

3. Tareas

3.1. Tarea 1

Configura Nginx para que no deje acceder con la IP de la máquina anfitriona al directorio raíz de una de tus dos webs. Modifica su server block o archivo de configuración. Comprueba como se deniega el acceso:

- Muestra la página de error en el navegador
- Muestra el mensaje de error de `error.log`

3.2. Tarea 2

Configura Nginx para que desde tu máquina anfitriona se tenga que tener tanto una IP válida como un usuario válido, ambas cosas a la vez, y comprueba que sí puede acceder sin problemas

3.3. Evaluación

Criterio	Puntuación
Configuración correcta de la autorización básica de Nginx, comprobación e identificación del error	**2 puntos**
Capturas correctas del log	**2 puntos**
Configuración correcta de la autorización básica en `contact`	**2 puntos**
Correcta configuración y comprobación de las tareas de autenticación básica y restricción por IP	**2 puntos**
Se ha prestado especial atención al formato del documento, utilizando la plantilla actualizada y haciendo un correcto uso del lenguaje técnico	**2 puntos**