

**PROYECTO DE CICLO DE G.S. DE
ADMINISTRACIÓN DE SISTEMAS
INFORMÁTICOS EN RED**

FIREWALL PARA HTTPS



**MOUAD TAIEBI BOUMOHCHINE
CURSO 2018/19**

Índice

1. INTRODUCCION	3
2. OBJETIVOS	3
3. MATERIAL	3
4. MÉTODO	4
4.1 PREPARATIVOS INICIALES	5
4.2 BASE DE DATOS	7
4.3 IPTABLES	9
4.4 PHP	9
4.5 CÓDIGO	10
4.6 INSTALACIÓN DE SERVIDOR LAMPP EN RASPBERRY PI	18
4.7 CONFIGURACIÓN DE LAS PÁGINAS CON APACHE2	23
5. TIEMPO DE EJECUCIÓN	26
6. RESULTADOS	27
7. CONCLUSIONES	32
7.1 PROBLEMAS	32
7.2 PROPUESTA DE MEJORA	37
8. BIBLIOGRAFÍA Y WEBGRAFIA	37
9. ANEXOS	38

1.INTRODUCCION

Este proyecto consistirá en desarrollar un firewall que permita/deniegue el acceso a dominios previamente localizados en una base de datos.

Esto nos permitirá de forma fácil y sencilla, controlar el acceso a ciertas páginas web que no queremos que los usuarios visiten, ya sea por peligro o para evitar distracciones y porque los proxys no son capaces de filtrar páginas HTTPS.

Lo que utilizaremos para este proyecto será una Raspberry pi 3 modelo B, una base de datos mysql y dos máquinas virtuales que harán de clientes.

2.OBJETIVOS

El objetivo principal es conseguir prohibir el acceso a unas páginas y permitir otras y que a los clientes (máquinas virtuales) se les apliquen esas prohibiciones y se les impida el acceso a las páginas que intenten visitar

También se le pondrá una interfaz gráfica para a facilidad de uso de este proyecto y su futura aplicación en diferentes ámbitos de trabajo

3.MATERIAL

HARDWARE

- Raspberry pi 3b
- Fuente de alimentación de 3ª
- Tarjeta mini SD de 32GB de clase 10

SOFTWARE

- Ubuntu Mate como SO
<https://ubuntu-mate.org/download/>
- Iptables

- Win32DiskImage
<https://sourceforge.net/projects/win32diskimager>

4. MÉTODO

La información respecto a este proyecto en internet es casi nula.

Lo que haremos será quemar la imagen de la ISO de Ubuntu Mate y para ello es necesario el programa Win32DiskImage.

Actualizamos los repositorios
sudo apt-get update

Una vez arrancada la Raspberry, instalamos el servicio de SSH.

sudo apt install openssh-server

Entramos en el archivo de configuración de Linux para cambiar el puerto por defecto, eso añadirá una capa de seguridad. Línea **Port**.

sudo nano /etc/ssh/sshd_config

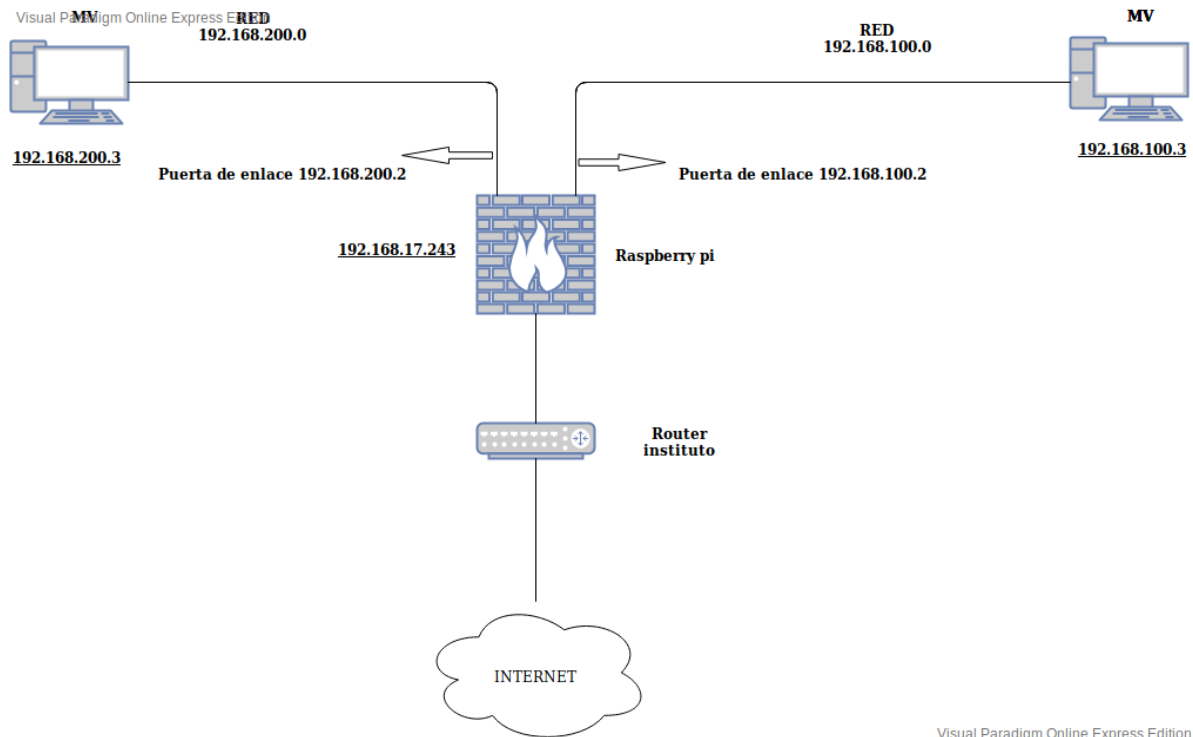
Guardamos e iniciamos el servicio.

sudo /etc/init.d/ssh start

Y para realizar el código php utilizaremos Aptana, pero no se puede instalar en la Raspberry, por lo que utilizaremos una máquina virtual para realizar pruebas del código antes de aplicarlo con un editor de textos como gedit o nano.

4.1 PREPARATIVOS INICIALES

Antes de empezar, lo que haremos será hacernos nuestro diagrama de cómo funcionará el proyecto:



La Raspberry hará de cortafuegos con 3 interfaces de red:

- Una será la principal, es decir la que conecta por internet mediante Ethernet.
- Y dos interfaces virtuales, una con la red 100 y otra con la red 200 en la que se conectarán dos máquinas virtuales.

Y la pregunta es, ¿cómo hago para que esas interfaces virtuales tengan conexión a internet? Habilitando el enrutamiento y utilizando DNAT (**Destination NAT**). Lo que hace es cambiar la dirección de destino del paquete, es decir, le dice hacia donde tiene que ir.

A continuación, están los comandos necesarios:

1. Creamos las interfaces virtuales

```
mouad@mouad-desktop:~$ sudo ifconfig eth0:0 192.168.100.2  
sudo ifconfig eth0:1 192.168.200.2
```

```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:35:3b:fb
          inet addr:192.168.17.243  Bcast:192.168.17.255  Mask:255.255.255.0
          inet6 addr: fe80::928a:5241:98b7:5649/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:639 errors:0 dropped:1 overruns:0 frame:0
          TX packets:451 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53174 (53.1 KB)  TX bytes:62051 (62.0 KB)

eth0:0    Link encap:Ethernet  HWaddr b8:27:eb:35:3b:fb
          inet addr:192.168.100.2  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth0:1    Link encap:Ethernet  HWaddr b8:27:eb:35:3b:fb
          inet addr:192.168.200.2  Bcast:192.168.200.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

2. Habilitamos el enrutamiento y DNAT para las dos redes

```
root@mouad-desktop:/home/mouad# echo '1' > /proc/sys/net/ipv4/ip_forward
root@mouad-desktop:/home/mouad# iptables -A FORWARD -j ACCEPT
iptables v1.6.0: unknown option "-j"
Try `iptables -h' or 'iptables --help' for more information.
root@mouad-desktop:/home/mouad# iptables -A FORWARD -j ACCEPT
root@mouad-desktop:/home/mouad# iptables -t nat -A POSTROUTING -s 192.168.100.0/
24 -o eth0 -j MASQUERADE
root@mouad-desktop:/home/mouad# iptables -t nat -A POSTROUTING -s 192.168.200.0/
24 -o eth0 -j MASQUERADE
```

Habilitar enrutamiento

Echo '1' > /proc/sys/net/ipv4/ip_forward

Habilitar DNAT

Iptables -t Nat -A POSTROUTING -s 192.168.100.0/24 -o eth0 -j MASQUERADE

Iptables -t nat -A POSTROUTING -s 192.168.200.0/24 -o eth0 -j MASQUERADE

4.2 BASE DE DATOS

Para crear la base de datos, lo que haremos será entrar en <http://localhost> y seleccionamos PHPMyAdmin.

Desde allí creamos una nueva base de datos que en caso la he llamado “dominios_bloquear”:



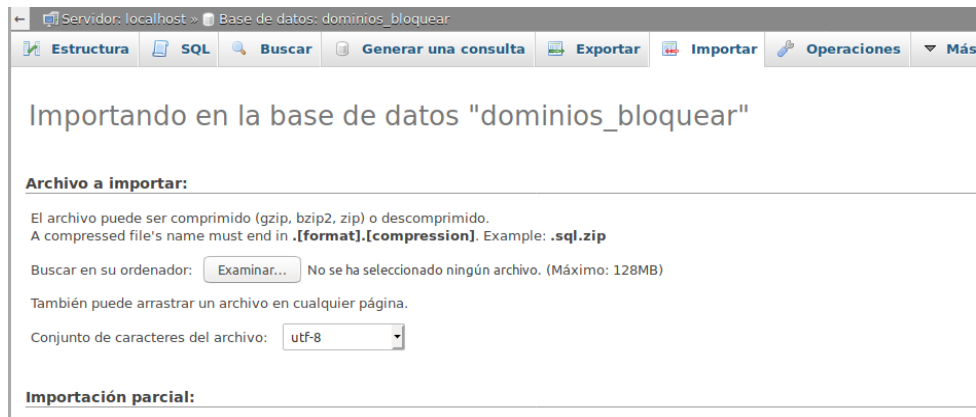
Una vez creada una base de datos, necesitaremos una tabla y filas, por lo que generamos este código SQL sencillo:

```
DROP TABLE IF EXISTS dominios;
create table if not exists dominios (
  dominio VARCHAR(50) PRIMARY KEY NOT NULL,
  activo char(2) DEFAULT 'NO'
);
insert into dominios (dominio) values ('as.com');
insert into dominios (dominio) values ('twitter.com');
insert into dominios (dominio) values ('twitter.es');
insert into dominios (dominio) values ('facebook.com');
insert into dominios (dominio) values ('marca.com');
insert into dominios (dominio) values ('twitch.tv');
insert into dominios (dominio) values ('charthitz.dr.ag');
insert into dominios (dominio) values ('mp3.com.au');
insert into dominios (dominio) values ('soundcloud.com');
insert into dominios (dominio) values ('digitaleradio.b');
insert into dominios (dominio) values ('belgiummp3.be');
insert into dominios (dominio) values ('angelgirls.biz');
insert into dominios (dominio) values ('topradio.be');
insert into dominios (dominio) values ('radio.dir.bg');
insert into dominios (dominio) values ('antenal.com.br');
insert into dominios (dominio) values ('365cast.com');
insert into dominios (dominio) values ('albania.com');
insert into dominios (dominio) values ('amsterdam-webcams.com');
insert into dominios (dominio) values ('mp3juices.gg');
```

He insertado las páginas más comunes en las que se suelen meter las personas en busca de entretenimiento, pero si se saben de más, fácilmente se puede entrar en la base de datos e insertar las filas que hagan falta.

La segunda columna indicará si la página está bloqueada o no.

Una vez tenemos el código, lo que tendremos que hacer será importarlo:



Servidor: localhost > Base de datos: dominios_bloquear

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Más

Importando en la base de datos "dominios_bloquear"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

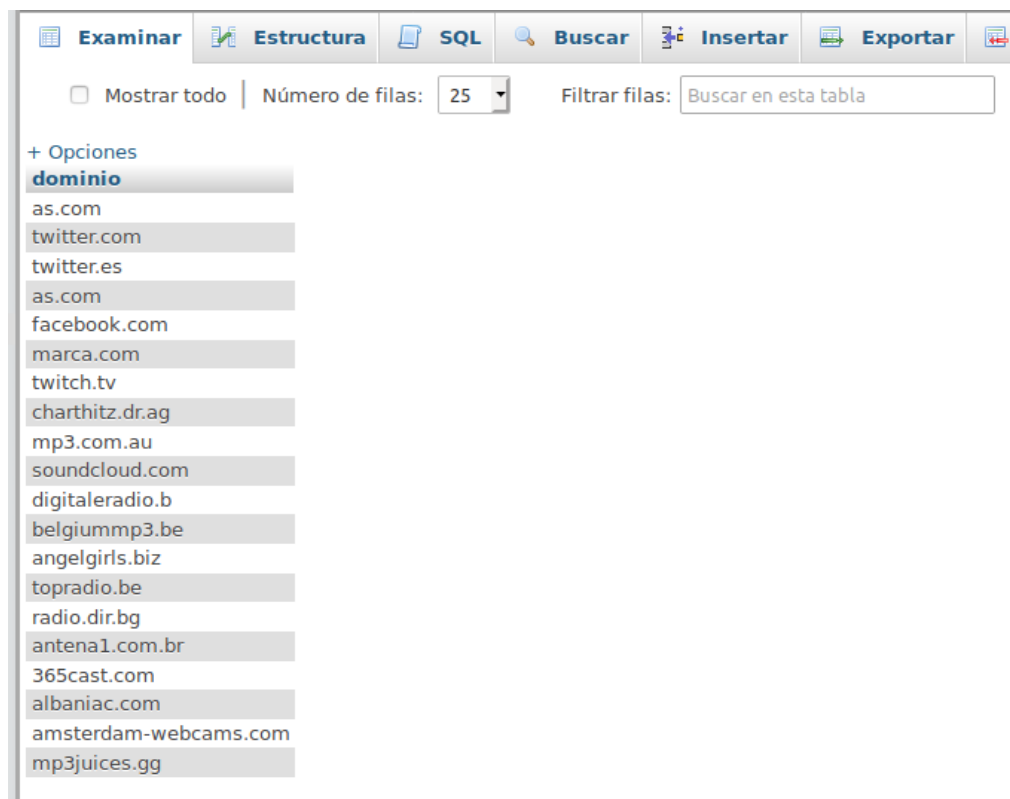
Buscar en su ordenador: No se ha seleccionado ningún archivo. (Máximo: 128MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

Importación parcial:

Y una vez importado, ya tenemos los dominios que necesitamos:



dominio
as.com
twitter.com
twitter.es
as.com
facebook.com
marca.com
twitch.tv
charthitz.dr.ag
mp3.com.au
soundcloud.com
digitaleradio.b
belgiummp3.be
angelgirls.biz
topradio.be
radio.dir.bg
antena1.com.br
365cast.com
albaniac.com
amsterdam-webcams.com
mp3juices.gg

4.3 IPTABLES

Iptables es una utilidad por comandos que se encarga de configurar el cortafuegos del kernel de Linux, que fue implementado por Netfilter.

Es muy fácil de usar y muy diverso a la hora de su uso.

Ahora debemos averiguar el comando que no permita el acceso a una página web segura.

Indagando por internet encontré el comando adecuado para esta función:

```
sudo iptables -A FORWARD -p tcp -s (la ip) --dport 443 -m string --string $d --algo bm -j REJECT
```

```
sudo iptables -A FORWARD -p tcp -s (la ip) --dport 843 -m string --string $d --algo bm -j REJECT
```

Analicemos el comando paso a paso:

- -A -> Append, coloca la regla debajo si hay otras más
- FORWARD -> Tráfico que atraviesa el firewall
- -p tcp -> El protocolo utilizado
- -s -> La ip o interfaz de entrada
- --dport -> Es el puerto de destino, en este caso el de https
- --string -> lo que se introduzca, lo buscará por la URL y si lo encuentra, aplica la regla
- REJECT -> A diferencia de DROP, lo que hará será bloquearlo y a la vez mandará error al cliente

Pero, ¿Por qué hay que bloquear también el puerto 843?

El puerto 843 se utiliza para determinar las comunicaciones de principio a fin. Es decir, si no se determina ese inicio o fin de conexión, no se podrá acceder a la página.

Ahora hay que aplicar este comando pero con PHP.

4.4 PHP

Hipertext PreProcessor es un lenguaje de código abierto utilizado normalmente para desarrollo web.











En vez de utilizar muchos comandos para mostrar HTML, este viene incrustado en PHP y viene encerrado entre las etiquetas especiales “<?php?>”.

Este código se ejecuta desde el lado del servidor, es decir, se mostrará, el resultado en el cliente sin mostrarle el código.

Es un lenguaje muy simple de utilizar y con muchas funciones para los profesionales, por lo que ¿por qué no utilizarlo?.

4.5 CÓDIGO

Para nuestro código, utilizaremos los siguientes archivos:

 añadir_dominio.php	14/05/2019 18:59	Archivo PHP	2 KB
 conexion.php	14/05/2019 18:59	Archivo PHP	1 KB
 formulario_borrado.php	14/05/2019 18:59	Archivo PHP	2 KB
 formulario_firewall.php	14/05/2019 18:59	Archivo PHP	2 KB
 index.php	14/05/2019 18:59	Archivo PHP	2 KB
 mclibre_php_soluciones_proyectos_com...	14/05/2019 18:59	Documento de ho...	3 KB
 pagina_principal.php	14/05/2019 18:59	Archivo PHP	4 KB
 tabla_dominios.php	14/05/2019 18:59	Archivo PHP	2 KB
 tabla_dominios_bloqueados.php	14/05/2019 18:59	Archivo PHP	2 KB
 tabla_dominios_bloqueados_no.php	14/05/2019 18:59	Archivo PHP	2 KB

- Conexión.php -> Este archivo se encargará de realizar una conexión con la BD.

```

<?php
// function conexion_d(){
try {
    $user = "root";
    $pass = "";
    $dbname = "dominios_bloquear";
    $db = new PDO("mysql:host=localhost; dbname=$dbname", $user, $pass);

    /* $db = null;
    print "<p>... y se cierra la conexi&oacute;n<p>";
    *
    */
} catch (PDOException $e) {
    print "<p>Error: No se pudo conectar con la BBDD $dbname.</p>\n";
    print "<p>Error: " . $e->getMessage() . "</p>\n";
    exit();
}
//}
?>

```

Explicación

\$user -> Esta variable almacena el usuario de la base de datos

\$pass -> Aquí se almacena la contraseña de acceso a la base de datos

\$dbname -> Aquí se almacena el nombre de la base de datos a acceder

\$db -> Contiene la PDO para acceder, se almacenan todas las variables anteriores y el tipo de base de datos que es mysql.

PDO es una interfaz encargada de dar acceso a la base de datos a través de PHP.

Y se no consigue conectar pasará a la parte de catch, que mostrará en pantalla un error diciendo que no se pudo acceder a la base de datos.

- Formulario firewall.php -> Aquí esta el código del formulario para escoger el dominio a bloquear y aplicar la regla de iptables anteriormente vista.

(Parte 1)

```
<?php
require_once "pagina_principal.php";
require ("conexion.php");
//CABECERA
cabecera("Que vas a bloquear?", MENU_VOLVER);

/*SI SE DEJA EL FORMULARIO VACIO, SE VOLVERA A MOSTRAR*/
if(!$ _POST){
    echo '
    <form method="POST"> <!--El formulario utiliza el metodo POST-->
    <p>Dominios:';

        echo '<select name="domain">';

        foreach($db->query("select * from dominios WHERE activo = 'NO'") as $fila) {
            echo '<option value="' . $fila['dominio'] . '">' . $fila['dominio'] . '<option/>';
        }
        echo '</select>
        <table>
            <tr>
                <td>Escriba la ip:</td>
                <td><input type="text" name="ip" size="15" maxlength="15" /> </td>
            </tr>
        </table>

    ';
    echo '
    </p>
    <p>
        <input type="submit" value="Enviar" />
        <input type="reset" value="Borrar" />
    </p>
    </form>';
    // $db = conexion d();
```

Explicación

Require_once “pagina_principal.php” -> Este archivo php contiene la cabecera y el pie de página y lo solicitará una sola vez

Requiere (conexion.php) -> durante todo el proceso, se necesitará la conexión con la base de datos para poder acceder a los dominios previamente insertados.

if(!\$ _POST) -> si no se rellena el formulario y no se manda nada, lo que hará será volver a mostrarlo

Y dentro de la condición está la lista desplegable que se saca así:

echo '<select name="domain">'; -> Abrimos la etiqueta de select y le damos un nombre significativo.

foreach(\$db->query('select * from dominios') as \$fila) { -> Este bucle lo que hará será ejecutar la consulta y por cada fila que salga, lo irá almacenando en la variable \$fila

echo '<option value="".'\$fila['dominio'].''>'.\$fila['dominio'].'<option/>'; -> Y una vez se tienen los datos, se ira colocando un “option” por cada fila que contenga la variable \$fila.

Y otra parte del formulario donde se insertará la dirección ip de un host o de red a bloquear.

Y abajo están los botones de enviar el formulario y borrarlo.

(parte 2)

```
else(
    $d=$_POST["domain"];
    $ip=$_POST["ip"];
    //echo $d;
    if ($d != NULL) {
        if(filter_var($ip, FILTER_VALIDATE_IP))
        {
            exec("sudo iptables -A FORWARD -p tcp -s $ip --dport 443 -m string --string $d --algo bm -j REJECT");
            exec("sudo iptables -A FORWARD -p tcp -s $ip --dport 843 -m string --string $d --algo bm -j REJECT");
            echo exec("sudo iptables -L | grep $d ") ;
            $db->query("UPDATE dominios set activo = 'SI' WHERE dominio = '$d'");
        }
        else
        {
            echo 'IP no es valida';
        }
    }
    else {
        echo "Error al insertar o leer regla de iptables";
    }
}
pie ("2019-01-28");
?>
```

Explicación

Si no se cumple la primera condición y se selecciona y envía un valor, se hará lo siguiente:

\$d=\$_POST["domain"]; -> En la variable \$d se almacenará lo que se haya seleccionado de la lista desplegable.

\$ip=\$_POST["ip"]; -> En la variable \$ip se almacenará la ip que se intr

if (\$d != NULL) {

if(filter_var(\$ip, FILTER_VALIDATE_IP))

\$regla = exec("sudo iptables -A FORWARD -p tcp -s \$ip --dport 443 -m string --string \$d --algo bm -j REJECT");

echo exec("sudo iptables -L | grep \$d ") ; -> Si el valor almacenado en la variable no es nulo y que la ip se valide, se ejecutará la regla de iptables y en el “—string” se colocará lo seleccionado anteriormente y además mostrará por pantalla la regla ya aplicada, que la saca del comando “Iptables -L” filtrando por la variable \$d.

\$db->query("UPDATE dominios set activo = 'SI' WHERE dominio = '\$d'"); -> Y actualizará la base de datos poniendo la columna de activo a SI.

```
else {
echo "Error al insertar o leer regla de iptables";
} -> Y si da la casualidad de que el valor de $d es nulo, entonces mostrara ese mensaje
```

- Pagina_principal.php -> Se deposita el código de cabecera y pie de página.

```
<?php
define("MENU_PRINCIPAL", "menuPrincipal");
define("Eliminar_Regla", "menuborrado");
define("Añadir_regla", "menuañadir");
define("dominios_bloqueados", "menubloq");
define("dominios_no_bloqueados", "menuuno");
define("añadir_dom", "menu_añadir");
// Menú principal
define("MENU_VOLVER", "menuVolver"); // Menú Volver a inicio

function cabecera($texto, $menu)
{
    print "<!DOCTYPE html>
<html lang='es'>
<head>
<meta charset='utf-8' />
<title>Firewall para HTTPS - $texto</title>
<meta name='viewport' content='width=device-width, initial-scale=1.0' />
<link href='\"mclibre_php_soluciones_proyectos_comun.css\"' rel='\"stylesheet\"' type='\"text/css\"' />
</head>
<body>
<h1>Firewall HTTPS - $texto</h1>
<div id='\"menu\"'>
<ul>\n";
    if ($menu == MENU_PRINCIPAL) {
        print " <li><a href='\"tabla_dominios.php\"'>Ver dominios disponibles</a></li>\n";
        print " <li><a href='\"formulario_borrado.php\"'>Eliminar Regla</a></li>\n";
        print " <li><a href='\"formulario_firewall.php\"'>Añadir regla</a></li>\n";
        print " <li><a href='\"tabla_dominios_bloqueados.php\"'>dominios bloqueados</a></li>\n";
        print " <li><a href='\"tabla_dominios_bloqueados_no.php\"'>dominios sin bloquear</a></li>\n";
        print " <li><a href='\"añadir_dominio.php\"'>Añadir dominio</a></li>\n";
    }
}
```

Se crea la función que luego mostrara la cabecera.

Como se ve hay, con **define** se define el botón de menú principal, volver., añadir regla, eliminar regla, añadir, mostrar tabla de dominios bloqueados y no bloqueados...etc
En las condiciones se tendrá que poner a que página quieres que te lleve cuando pulsas sobre él.

```
function pie($fecha)
{
    print "</div>\n";
    $cadenaFecha = formatearFecha($fecha);
    echo <<< FINPIE
    <footer>
        <p class='\"ultmod\"'>
            Última modificación de esta página:
            <time datetime='\"$fecha\"'>$cadenaFecha</time> (Mouad Taiebi Boumohcine)</p>
        </footer>
    </body>
</html>
FINPIE;
}
```

Se crea la función del pie de página.

Aquí se llama a la a la función del pie de la página, lo que se muestra es la fecha y el nombre de la persona.

- Index.php -> Es el que mostrará la página de inicio.

```

4  *
5  * IES Virgen del Carmen de Jaén
6  * Implantación de Aplicaciones Web 2º ASIR
7  *
8  * Basado en el código de:
9  *
10 * @author Bartolomé Sintés Marco <bartolome.sintes+mclibre@gmail.com>
11 * @copyright 2012 Bartolomé Sintés Marco
12 * @license http://www.gnu.org/licenses/agpl.txt AGPL 3 or later
13 * @version 2012-11-27
14 * @link http://www.mclibre.org
15 *
16 * This program is free software: you can redistribute it and/or modify
17 * it under the terms of the GNU Affero General Public License as published by
18 * the Free Software Foundation, either version 3 of the License, or
19 * any later version.
20 *
21 * This program is distributed in the hope that it will be useful,
22 * but WITHOUT ANY WARRANTY; without even the implied warranty of
23 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 * GNU Affero General Public License for more details.
25 *
26 * You should have received a copy of the GNU Affero General Public License
27 * along with this program. If not, see <http://www.gnu.org/licenses/>.
28 */
29
30 require_once "pagina_principal.php";
31
32 cabecera("Inicio", MENU_PRINCIPAL);
33
34 pie("2017-01-09");
35 -?>

```

Mediante **require once** llama al archivo pagina_principal.php y se llama a las dos funciones con sus parámetros respectivos.

- Formulario borrado.php-> Hace lo contrario que el de formulario firewall.php.

Explicación

Lo que hará será buscar aquellas filas que contengan la columna de activo a SI y eliminar su regla iptables

```

<form method="POST"> <!--El formulario utiliza el metodo POST-->
<p>Dominios:';
    echo '<select name="domain_d">';

    foreach($db->query("select * from dominios where activo = 'SI'") as $fila) {
        echo '<option value="' . $fila['dominio'] . '">' . $fila['dominio'] . '<option/>';
    }
echo '</select>';
<table>
    <tr>
        <td>Escriba la ip de la regla a eliminar:</td>
        <td><input type="text" name="ip" size="15" maxlength="15" /> </td>
    </tr>
</table>';

```

```

$d=$_POST["domain_d"];
$ip = $_POST["ip"];
$c = exec("sudo iptables -L | grep $d | grep $ip") ;
if ($d != NULL) {
    if((filter_var($ip, FILTER_VALIDATE_IP)) && ($c != null))
    {
        $regla = exec("sudo iptables -D FORWARD -p tcp -s $ip --dport 443 -m string --string $d --algo bm -j REJECT");
        $db->query("UPDATE dominios set activo = 'NO' WHERE dominio = '$d'");
        echo "Regla iptables eliminada correctamente!";
    }
    else {
        echo "Error: No existe la regla con la ip buscada";
    }
}
else {
    echo "Error al borrar o leer regla de iptables";
}
}
)

```

La variable \$c se encargará de buscar en la lista de reglas, filtrando por el dominio y la ip.

Para la lista desplegable, si encuentra filas con la columna activo a SI, los pondrá en la lista desplegable.

Si el formulario no es nulo, la ip se valida y \$c no es nula, ejecutará la orden y además actualizará la base de datos poniendo la columna a NO.

- Tabla dominios bloqueados.php -> Esta tabla mostrará aquellas filas que tengan la columna de activo a SI.

Explicación

```

$consulta="SELECT * FROM dominios where activo= 'SI'";
?>
</head>
<body>
    <table border="2" cellpadding="2" cellspacing="0">
        <tr><th colspan="4">bloqueados</th></tr>
    <?php
    foreach ($db->query($consulta) as $fila){
        $url=$fila['dominio'];
        $activo=$fila['activo'];
    ?>
    <tr>
        <td><?php echo ($url);?></td>
        <td><?php echo ($activo);?></td>
    </tr>

    <?php
    }
    echo "la tabla se ha creado existosamente";
?>

```

Es muy simple, la consulta sql tiene que tener como condición que la columna de activos este a SI y con un bucle foreach irá construyendo la tabla.

- Tabla dominios bloqueados no.php -> Esta tabla mostrará aquellos que aún no estén bloqueados.

Explicación

```

$consulta="SELECT * FROM dominios where activo= 'NO'";
?>
</head>
<body>
  <table border="2" cellpadding="2" cellspacing="0">
    <tr><th colspan="4">NO bloqueados</th></tr>
  <?php
  foreach ($db->query($consulta) as $fila){
    $url=$fila['dominio'];
    $activo=$fila['activo'];
  ?>
  <tr>
    <td><?php echo ($url);?></td>
    <td><?php echo ($activo);?></td>
  </tr>
  <?php

```

Es justo lo contrario que el anterior, la consulta sql tiene que tener como condición que la columna de activos este a NO y con un bucle foreach irá construyendo la tabla.

- Tabla dominios.php -> Esta tabla mostrará todos los dominios que existan en la base de datos.

Explicación

```

require("conexion.php");
$consulta="SELECT * FROM dominios";
?>
</head>
<body>
  <table border="2" cellpadding="2" cellspacing="0">
    <tr><th colspan="4">dominios disponibles</th></tr>
  <?php
  foreach ($db->query($consulta) as $fila){
    $url=$fila['dominio'];
  ?>
  <tr>
    <td><?php echo ($url);?></td>
  </tr>
  <?php
}
echo "la tabla se ha creado existosamente";
?>

```

Tan solo será una consulta simple que saque todas las filas y las vaya poniendo en una tabla.

- Añadir dominio.php -> Este archivo se encargará de insertar datos en la base de datos.

Explicación

(parte 1)

```
if (!$_POST) {
    echo '
<form method="POST"> <!--El formulario utiliza el metodo POST-->
<p>Dominio a insertar <input type="text" name="name"/></p>

';
    echo '
</p>
<p>
        <input type="submit" value="Enviar" />
        <input type="reset" value="Borrar" />
</p>
</form>';
}
```

En el formulario se introducirá el dominio a insertar en la base de datos.

(parte 2)

```
$añade=$_POST["name"];
$insertar = "INSERT into dominios values('$añade', 'NO')";
//$nuevo_dominio=$db->query("select dominio from dominios where dominio='$añade'");

if ($añade != NULL) {
    if (!preg_match('/.(com|net|org|biz|info|mobi|us|cc|bz|tv|ws|name|co|me|es)(\.[a-z]{1,3})?\z/i', $añade)){
        echo 'No has escrito bien el dominio.';
    }
    else {
        $db->query($insertar);
        echo 'Dominio insertado con éxito! :)' ;
    }
}
else {
    echo "No has insertado nada! Intentalo otra vez!";
}
```

En **\$añade** se guardará lo que se escriba en el formulario

\$insertar contendrá la orden SQL de inserción en la base de datos

if (\$añade != NULL) {

if(!preg_match('/.(com|net|org|biz|info|mobi|us|cc|bz|tv|ws|name|co|me|es)(\.[a-z]{1,3})?\z/i', \$añade)){

echo 'No has escrito bien el dominio.'; -> Si el formulario no está vacío, procederá a comprobar que lo que se inserta es un dominio con el formato “www.dominio.es” o “dominio.es”

Si todo está correcto, ejecutará la orden de inserción en la base de datos

4.6 INSTALACIÓN DE SERVIDOR LAMPP EN RASPBERRY PI

La instalación del servidor LAMPP en la Raspberry no se hace de la misma manera que en un PC normal. Como vimos anteriormente, para instalarlo solo era ejecutar un archivo .run que instalaba todos los servidores de una vez.

En la Raspberry es todo lo contrario. Hay que instalar el servidor web, mysql y PHP de uno en uno.

A continuación, explico los pasos para instalarlos:

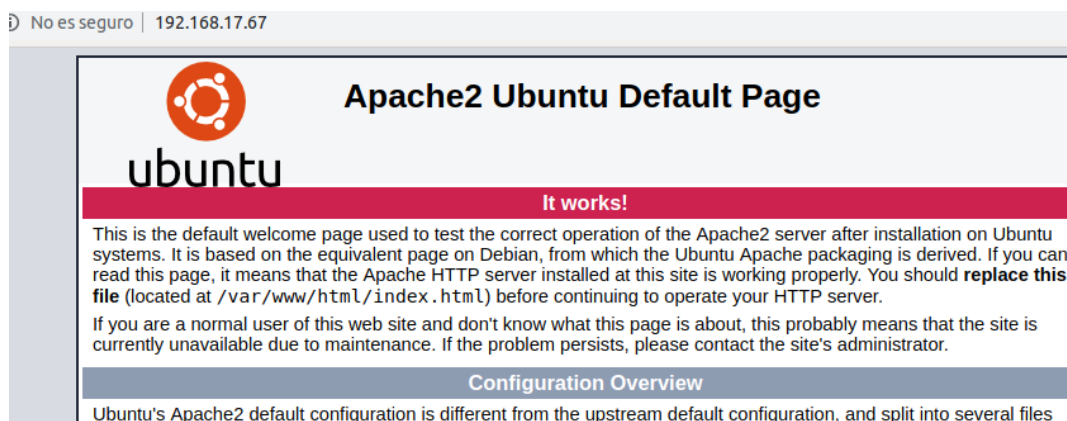
1. Procedemos a instalar Apache2

```
root@mouad-desktop:~# sudo apt-get install apache2
```

Una vez instalado, comprobamos la ip que tenemos actualmente. (Recomendable que tengamos esa IP estática de ahora en adelante).

```
root@mouad-desktop:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:35:3b:fb
          inet addr:192.168.17.67  Bcast:192.168.17.255
```

Esa IP la introducimos en el navegador, y si el servidor web se ha instalado correctamente, debería salir esta pantalla:



También se puede acceder utilizando la ip de loopback o introduciendo "localhost".

2. Instalamos el servidor mysql

```
root@mouad-desktop:~# sudo apt-get install mysql-server mysql-client
```

En el proceso de instalación nos pedirá una contraseña de root. En mi caso la he dejado vacía, pero no es seguro hacerlo.

Una vez finalizada la instalación, probamos que el servidor mysql funciona correctamente introduciendo este comando:

Mysql -u root -p

Con este comando se hace un login en la base de datos con el usuario root, si todo va bien, debería dejaros entrar en el servidor sin problema.

3. Instalamos la versión 7.0 de PHP

```
root@mouad-desktop:~# apt-get install php7.0
```

Y también este:

```
root@mouad-desktop:~# apt-get install php7.0-mysql
```

Este comando lo que instalará será la versión 7 de PHP para que la base de datos reconozca el código a la hora de que haga una conexión a través de PHP.

Y por último:

```
root@mouad-desktop:~# apt install php7.0 libapache2-mod-php7.0
```

Este último es importante y tuve problemas cuando no lo tenía instalado. Si no tenemos esto instalado, Apache no será capaz de interpretar código PHP, por lo que si ejecutáis código, aparecerá en el navegador.

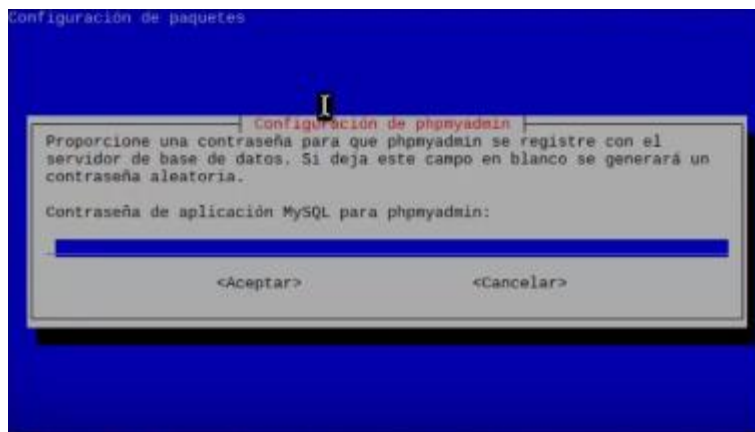
4. Instalación de PHPMyAdmin

```
root@mouad-desktop:~# apt-get install phpmyadmin
```

Durante la instalación nos pedirá el servidor web que se configurará, en nuestro caso Apache2.



También nos pedirá una contraseña para PHPMyAdmin, en mi caso no puse ninguna, y vuelvo a repetir que se debería poner:



Y antes de acceder a PHPMYADMIN, tenemos que incluir en la configuración de Apache que se ejecute PHPMyAdmin:

`Sudo nano /etc/apache2/apache2.conf`

Nos vamos a la última línea e incluimos esto:

```
Include /etc/phpmyadmin/apache.conf
```

Una vez que ha terminado la instalación, probamos que funciona:

`192.168.17.67/phpmyadmin`



Vemos que se ha instalado correctamente, pero nos da un error de acceso denegado, para solucionarlo haremos lo siguiente:

- Por defecto PHPMyAdmin no permite hacer login sin contraseña, por lo que editaremos el siguiente archivo:

`Sudo nano /etc/phpmyadmin/config.inc.php`

Y descomentaremos dos líneas que contiene esto:

```
//  
$cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
```

- Y lo siguiente que haremos será darle los permisos necesarios para que pueda acceder:
 - a. Iniciamos sesión en mysql

`Mysql -u root -p`

- b. Accedemos a la base de datos mysql

`Use mysql;`

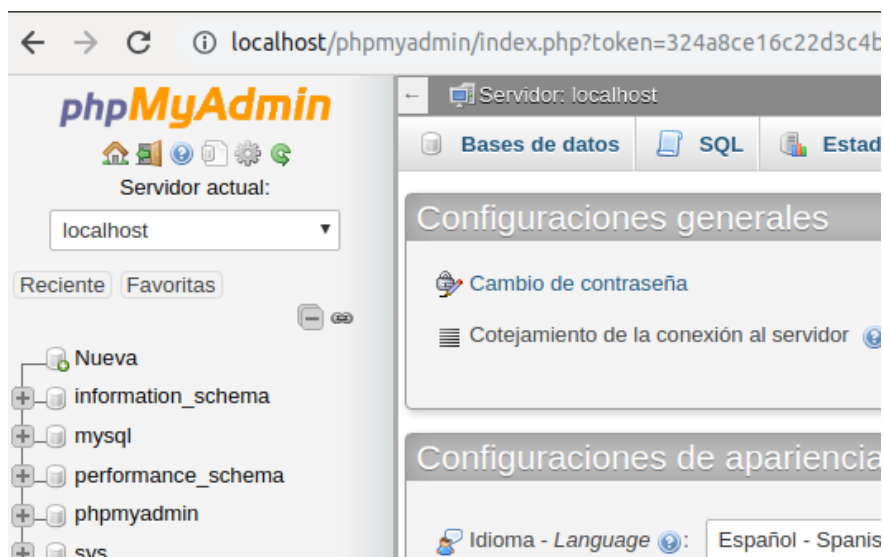
- c. Cambiamos el plugin del usuario root a `mysql_native_password` :

```
mysql> update user set plugin = "mysql_native_password" where user = 'root';
```

Y la tabla debería de quedar así:

```
Database changed
mysql> select user, plugin from user;
+-----+-----+
| user          | plugin          |
+-----+-----+
| root          | mysql_native_password |
| mysql.session | mysql_native_password |
| mysql.sys     | mysql_native_password |
| debian-sys-maint | mysql_native_password |
| phpmyadmin    | mysql_native_password |
+-----+-----+
5 rows in set (0,01 sec)
```

Y ahora sí nos permitirá acceder a la base de datos desde PHPMyAdmin:



4.7 CONFIGURACIÓN DE LAS PÁGINAS CON APACHE2

- Entramos en el archivo `ports.conf`, que se encuentra en la ruta `/etc/apache2/ports.conf`. Este archivo lo que hará será configurar los puertos que usará apache2.

```
GNU nano 2.5.3      File: /etc/apache2/ports.conf
# If you just change the port or add more ports here, you
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 8080
<IfModule ssl_module>
```

Por defecto, el puerto que utilizará será el 80, que la utilizaré para la página principal. Y he configurado otro puerto que usare para el de error.

- Ahora crearemos el directorio que contendrá los archivos php dentro de la ruta `/var/www`.

```
root@mouad-desktop:~# mkdir /var/www/firewall
```

- Una vez insertados los archivos, nos vamos a la ruta `/etc/apache/sites-available` y dentro:

Nano `firewall.conf`

Lo que hacemos es crear un archivo.conf para nuestra página.

```
GNU nano 2.5.3      File: firewall.conf
<VirtualHost 192.168.17.67:80>
    Alias /firewall "/var/www/firewall"
    DocumentRoot /var/www/firewall
    ServerAdmin mouadtaiebi@gmail.com
<Directory "/var/www/firewall">
    DirectoryIndex index.php
    AuthType Digest
    AuthName "Identificate!"
    AuthDigestProvider file
    AuthUserFile /etc/apache2/passwords.digest
    Require user mouad
</Directory>
</VirtualHost>
```

En Virtualhost se pone la ip del servidor, que como ya sabemos es la 67 y añadiendo dos puntos y el puerto que utilizará.

Se añade el **Alias**, que es como se llamará la ruta a la hora de ponerla en el navegador

DocumentRoot es el documento raíz de la página, apache sabrá que los recursos están en esa ruta

Añadimos la etiqueta **Directory** con la ruta de antes, para hacer referencia a esa carpeta

DirectoryIndex se le pondrá el archivo que hará de inicio cuando se abra la página. En este caso es index.php

El **AuthType** será **Digest**, que a diferencia del **basic**, este encriptará la información mientras se esté navegando.

Authname será el nombre de la autenticación

Authdigestprovider será el proveedor que dará la autenticación, que será un archivo.

Authuserfile será el archivo que contendrá las contraseñas y usuarios para la autenticación

Require user solo permitirá a los usuarios que le hayamos puesto.

- Una vez que todo esta bien configurado, guardamos el archivo, procederemos a activar la página:

```
root@mouad-desktop:/etc/apache2/sites-available# a2ensite firewall
Enabling site firewall.
To activate the new configuration, you need to run:
    service apache2 reload
root@mouad-desktop:/etc/apache2/sites-available# service apache reload
apache: unrecognized service
root@mouad-desktop:/etc/apache2/sites-available# service apache2 reload
```

El comando es:

A2ensite firewall

Al poner ese comando, pedirá reiniciar el servicio.

- Tenemos que habilitar el módulo de autenticación **digest**, ya que por defecto no está:

```
root@mouad-desktop:/home/mouad# a2enmod auth_digest.load
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
    service apache2 restart
root@mouad-desktop:/home/mouad#
```

- Ahora procederemos a crear la autenticación del usuario:

```
root@mouad-desktop:/home/mouad# htdigest -c /etc/apache2/passwords.digest Identificate! mouad
Adding password for mouad in realm Identificate!.
New password:
Re-type new password:
root@mouad-desktop:/home/mouad#
```

Procedo a explicar el comando:

El parámetro `-c` será crear el archivo **passwords.digest** si no lo encuentra en la ruta especificada.

“**Identificate!**” sería el nombre de **Authname** que pusimos anteriormente.

Y por último se pone el usuario.

Si el comando está correcto, pedirá que insertes una contraseña y que la confirmes.

Y si todo está bien configurado, debería de pedirte el usuario y contraseña y mostrarte la página web si es correcto:

The image shows a web browser window with the address bar displaying `192.168.17.67`. A login dialog box is overlaid on the page. The dialog is titled "Iniciar sesión" and shows the URL `http://192.168.17.67`. Below the URL, it says "Tu conexión con este sitio web no es privada". There are two input fields: "Nombre de usuario" with the text "mouad" and "Contraseña" with masked characters. At the bottom of the dialog are two buttons: "Cancelar" and "Iniciar sesión".

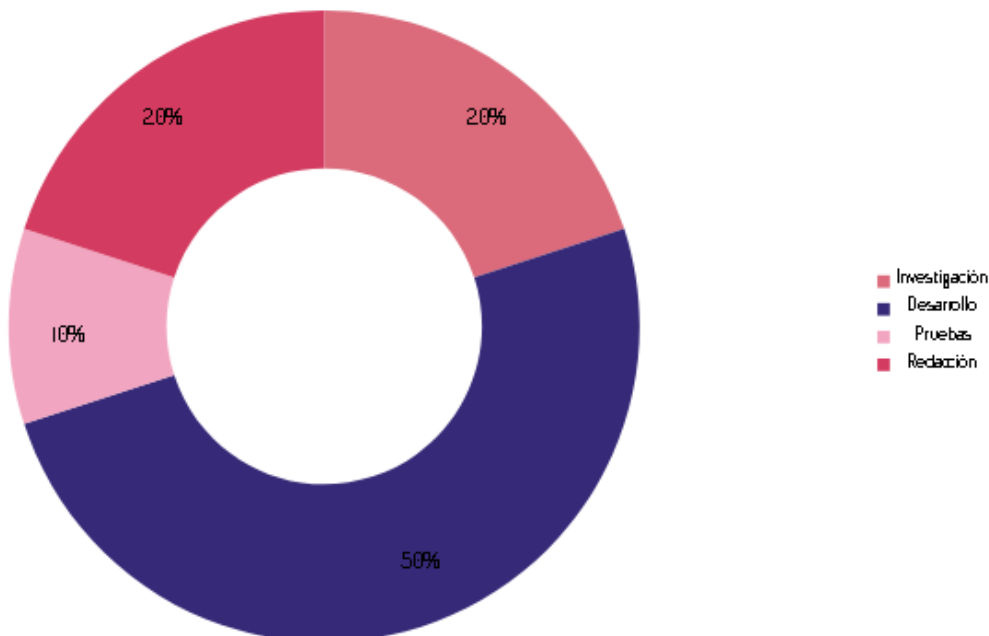
Below the dialog, the browser shows the "FIREWALL HTTPS - INICIO" page. The page has a navigation bar with four links: "Ver dominios disponibles", "Eliminar Regla", "Añadir regla", and "dominios bloqueados". At the bottom of the page, it says "Última modificación de esta página: 09 de enero de 2017 (Mouad Taiebi Boumohcine)".

5. TIEMPO DE EJECUCIÓN

El tiempo que me ha tomado dependía de las horas que iba a clase a realizarlo debido a que en mi casa no tenía una conexión a internet estable para poder realizar este proyecto.

Que normalmente iba 3 días a la semana, durante 2 horas, por lo que el total de realización del proyecto fue de 6 horas semanales desde que empecé las prácticas. Sumándole el tiempo de una hora diaria de redacción de este.

Lo que me ha tomado más tiempo fue la realización del código php de este proyecto, ya que tenía que hacerlo en otro equipo, errores que fueron saliendo a la hora de probarlo, instalación de los servidores necesarios en la Raspberry Pi para hacer funcionar el código y encontrar el comando correcto de IPTABLES que funcionara.



6. RESULTADOS

Tabla de dominios:



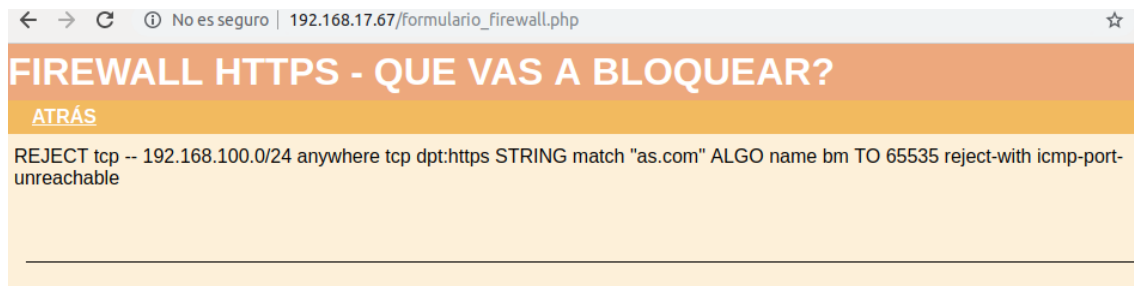
Formulario de bloqueo:

Dominios:

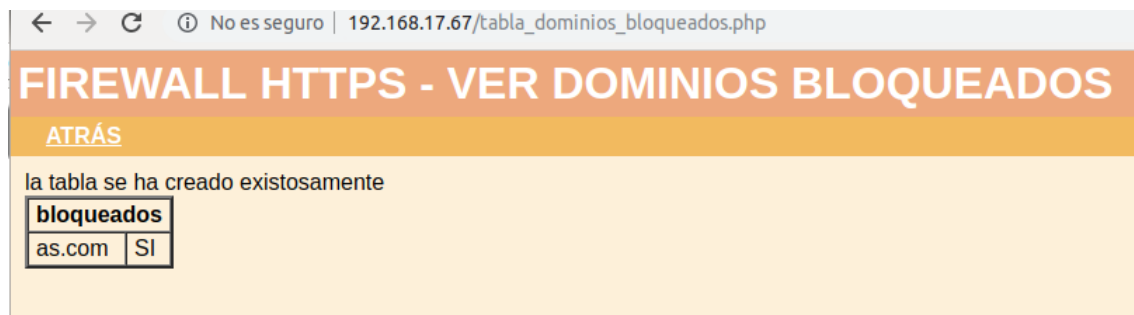
Escriba la ip:

De prueba, bloquearemos el acceso a toda la red 100 a la página as.com.

Si todo sale bien, debería dar esta pantalla:



Y en la tabla de bloqueados se deberá añadir dicho dominio:



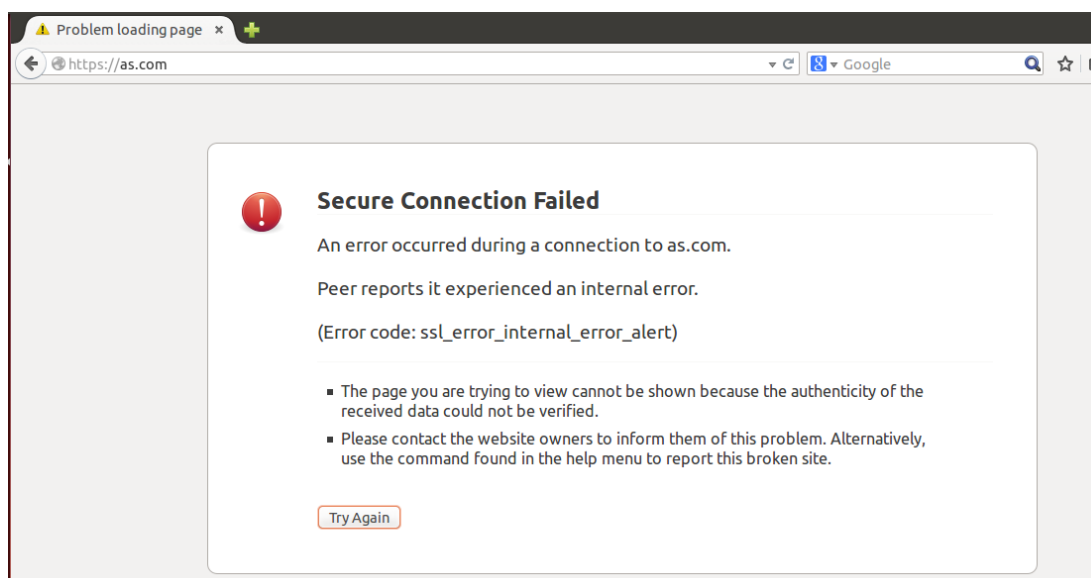
Y al hacer `iptables -L`:

```

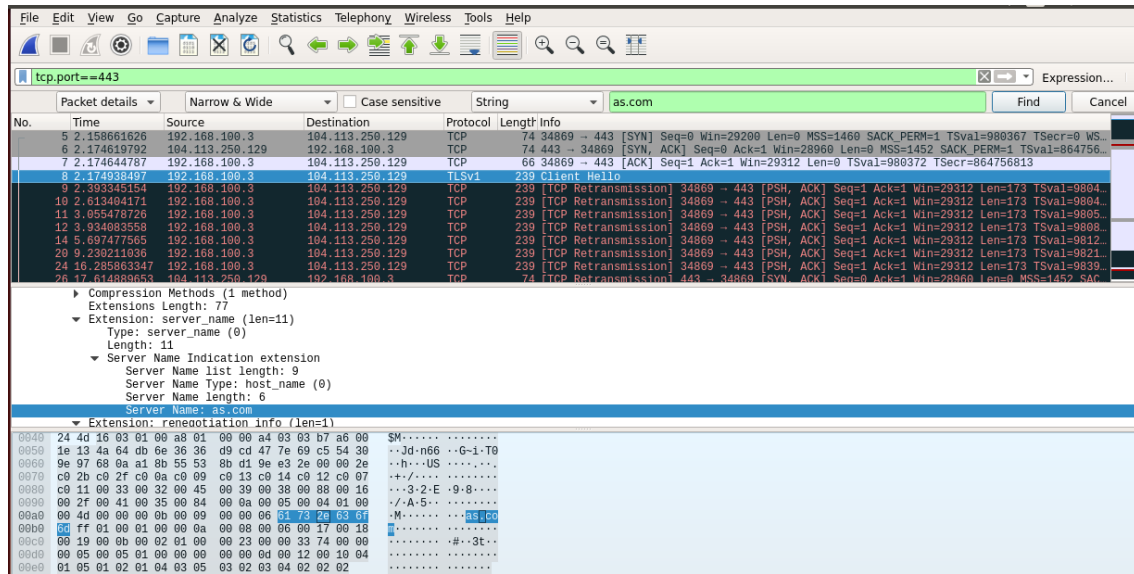
mouad@mouad-desktop:~$ sudo iptables -L
[sudo] password for mouad:
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- !192.168.17.71 192.168.17.67 tcp dpt:5673

Chain FORWARD (policy ACCEPT)
target prot opt source destination
REJECT tcp -- 192.168.100.0/24 anywhere tcp dpt:843 STRING match "as.com" ALGO name bm TO 65535 reject-with icmp-port-unreachable
REJECT tcp -- 192.168.100.0/24 anywhere tcp dpt:https STRING match "as.com" ALGO name bm TO 65535 reject-with icmp-port-unreachable
DROP tcp -- anywhere anywhere tcp dpt:843 STRING match "facebook" ALGO name bm TO 65535
DROP tcp -- anywhere anywhere tcp dpt:https STRING match "facebook" ALGO name bm TO 65535
ACCEPT all -- anywhere anywhere
ACCEPT all -- anywhere anywhere
  
```

Y al intentar acceder a dicha página, dará este error:



Y es porque cuando se empieza a encriptar la información de la página, una parte contiene el nombre de dominio y cuando se cumple la condición, iptables lo deniega, la encriptación falla y da error de conexión a servidor seguro.



Formulario de borrado:

FIREWALL HTTPS - QUÉ REGLA VAS A ELIMINAR?

[ATRÁS](#)

Dominios: as.com

Escriba la ip de la regla a eliminar:

Enviar
Borrar

Si se introduce una ip que no coincide con la regla insertada, dará error:
Por ejemplo:

FIREWALL HTTPS - QUÉ REGLA VAS A ELIMINAR?

[ATRÁS](#)

Dominios: as.com

Escriba la ip de la regla a eliminar: 192.168.200.3

Enviar
Borrar

Última modificación de esta página: 28 de enero de 2019 (Mouad Taiebi Boumohcine)

Anteriormente, bloqueamos as.com para toda la red 100. Si insertamos algo distinto.....

FIREWALL HTTPS - QUÉ REGLA VAS A ELIMINAR?

[ATRÁS](#)

Error: No existe la regla con la ip buscada o ip mal escrita

Última modificación de esta página: 28 de enero de 2019 (Mouad Taiebi Boumohcine)

Y no se eliminará:

```
Chain FORWARD (policy ACCEPT)
target prot opt source destination
REJECT tcp -- 192.168.100.0/24 anywhere tcp dpt:843 STRING match "as.com" ALGO name bm TO 65535 reject-with icmp-port-unreachable
REJECT tcp -- 192.168.100.0/24 anywhere tcp dpt:https STRING match "as.com" ALGO name bm TO 65535 reject-with icmp-port-unreachable
```

Si se inserta la ip correcta:

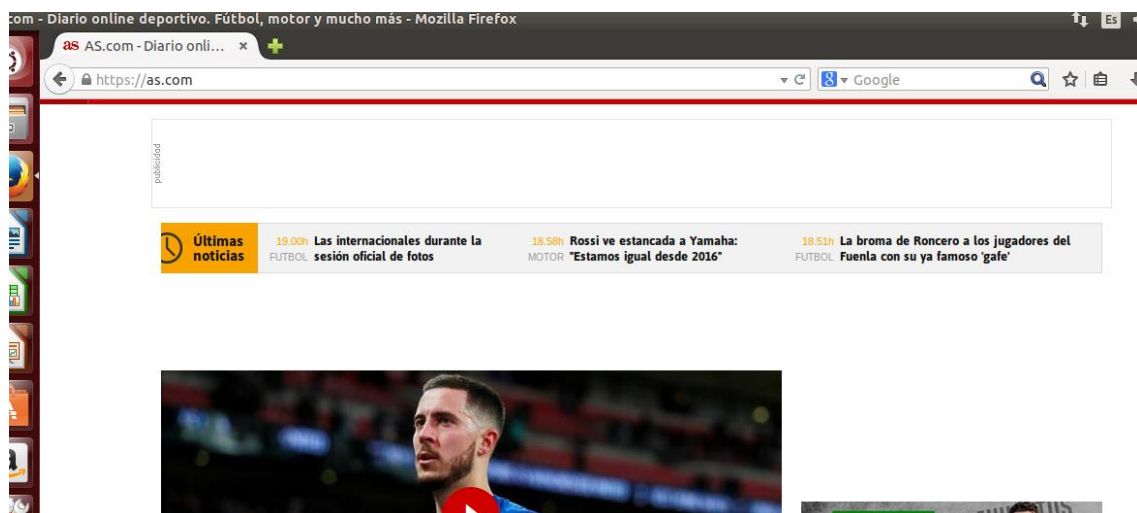
FIREWALL HTTPS - QUÉ REGLA VAS A ELIMINAR?

[ATRÁS](#)

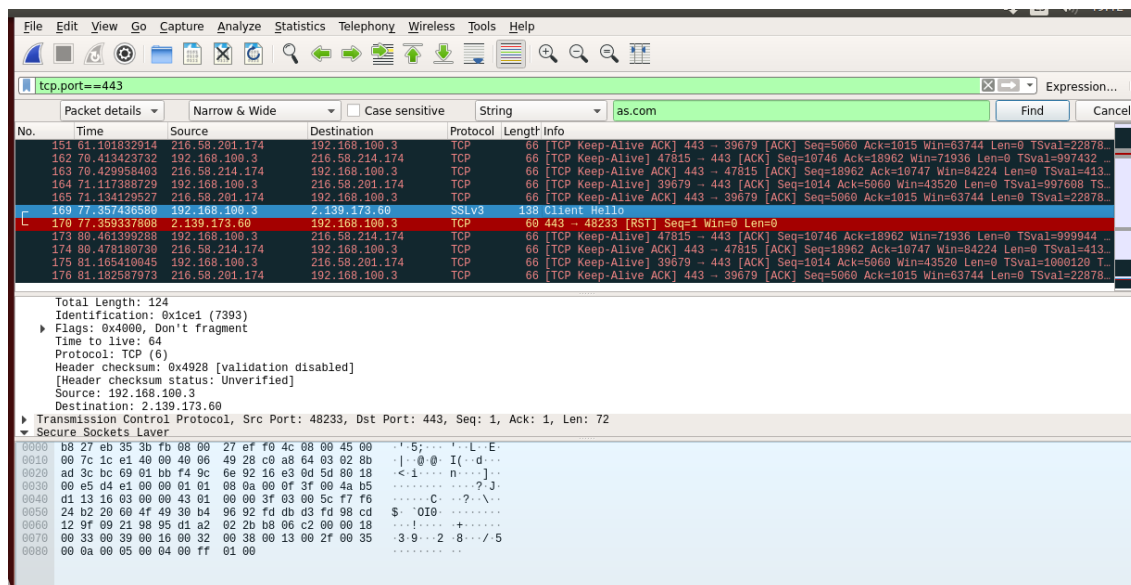
Regla iptables eliminada correctamente!

Última modificación de esta página: 28 de enero de 2019 (Mouad Taiebi Boumohcine)

Y permitirá el acceso a dicha página si intentamos acceder a ella:



Y si echamos un vistazo ahora en Wireshark:



Vemos que toda la información no es visible ya que esta encriptada. Eso quiere decir que la conexión a servidor seguro ha sido realizada con éxito.

Tabla de dominios no bloqueados:

FIREWALL HTTPS - VER DOMINIOS SIN BLOQUEAR	
ATRÁS	
la tabla se ha creado existosamente	
NO bloqueados	
365cast.com	NO
albaniac.com	NO
amsterdam-webcams.com	NO
angelgirls.biz	NO
animeflv.net	NO
antena1.com.br	NO
belgiummp3.be	NO
charthitz.dr.ag	NO
digitalradio.b	NO
facebook.com	NO
google.es	NO
marca.com	NO
mp3.com.au	NO
mp3juices.gg	NO
radio.dir.bg	NO
soundcloud.com	NO

Aquí se mostrará los dominios con los que no se les ha creado ninguna regla.

7.CONCLUSIONES

7.1 PROBLEMAS

El problema que hay es que no se le puede instalar un entorno de desarrollo como Aptana en la Raspberry Pi, hay que utilizar los editores de texto de Linux. Pero para mayor rendimiento a la hora de hacer la interfaz del proyecto, es mejor tenerlo.

La solución es tener una máquina virtual (Linux o Windows, aunque preferiblemente Linux) e instalarle Aptana. A continuación, pondré los pasos a seguir para instalarlo:

1. Lo primero será instalar XAMPP que es un software totalmente gratuito que gestiona los servidores Mysql, Apache y FTP de una manera gráfica.

<https://www.apachefriends.org/es/download.html>

Una vez descargado el archivo, lo que haremos será darle permisos de ejecución:

`chmod 755 xampp-linux-x64-7.2.11-0-installer.run`

Y ejecutarlo:

`sudo ./xampp-linux-x64-7.2.11-0-installer.run`

Una vez hecho eso y completar la instalación siguiendo los pasos, nos saldrá la interfaz:



Desde aquí ya podemos administrar los tres servidores.
Pero también se puede administrar por comandos:

Para iniciar/parar/reiniciar los servicios:

Sudo /opt/lampp/lampp start/stop/restart

Y para saber el estado:

Sudo /opt/lampp/lampp status

Y para ejecutar la interfaz gráfica desde comandos:

Sudo /opt/lampp/manager-linux-x64.run

Y para comprobar que funcionan los servidores:

<http://localhost>

2. Una vez instalado XAMPP, pasamos a instalar el entorno Aptana:

Nos descargamos el programa: <https://github.com/aptana/studio3>

Aptana requiere tener una versión de Java **1.5.x o posterior**

Para comprobar la versión:

Java -version

Y si no lo tenemos, procedemos a instalarlo:

Sudo apt install openjdk-1-jre

Una vez comprobado Java, descomprimos el zip de Aptana en la carpeta del usuario en el que estemos:

unzip Descargas/aptana.studio-linux.gtk.x86_64.zip -d Aptana_Studio_3

Al descomprimirlo, generará una carpeta, que en su interior contiene el ejecutable del programa. Le damos los permisos necesarios:

Chmod 755 Aptana_Studio_3/AptanaStudio3

Creamos un directorio que contendrá los archivos php que vayamos creando para nuestro proyecto y le otorgamos permisos:

Sudo mkdir /opt/lampp/htdocs/proyecto

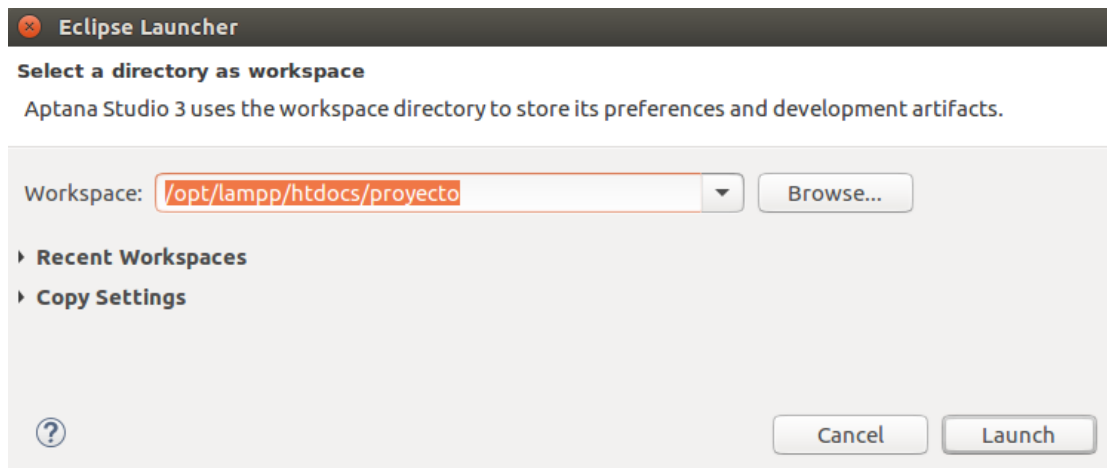
Sudo chmod 777 /opt/lampp/proyecto

Una vez hecho esto, procedemos a ejecutar el ejecutable:

Cd Aptana_Studio_3/

./AptanaStudio3

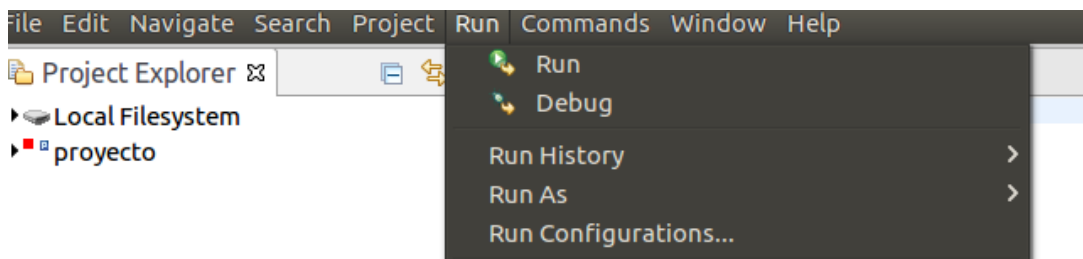
Al ejecutarlo, nos pedirá la ruta del espacio de trabajo (Workspace), que será la carpeta que hemos creado anteriormente



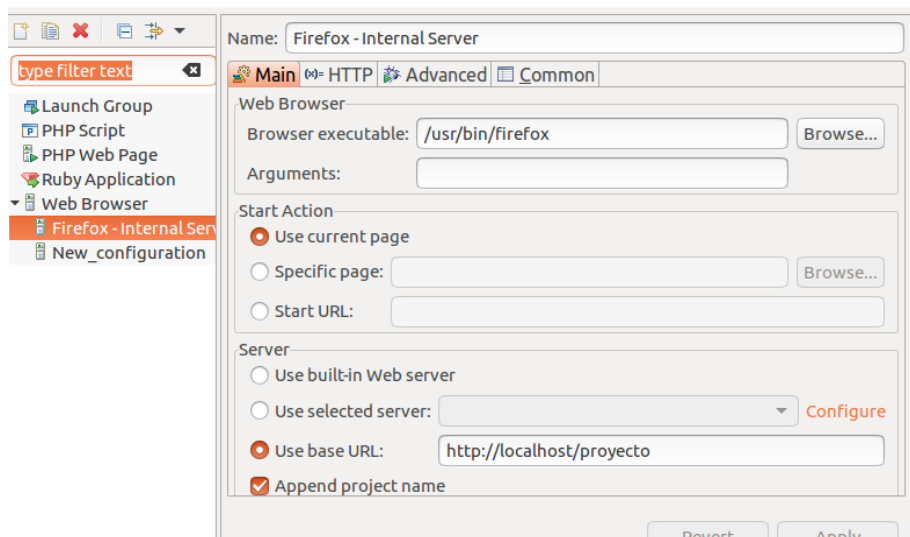
Ahora hemos de probar que todo funciona correctamente, por lo que crearemos un proyecto para phpinfo().

OJO!! Hay que tener el servidor web apache funcionando.

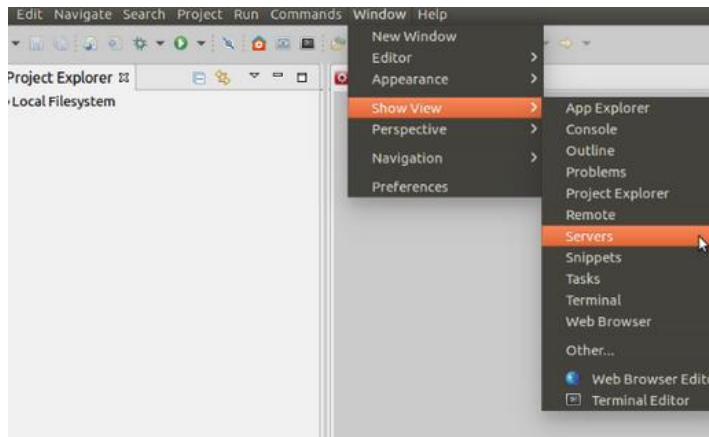
Entramos en Run Cobnfigurations:



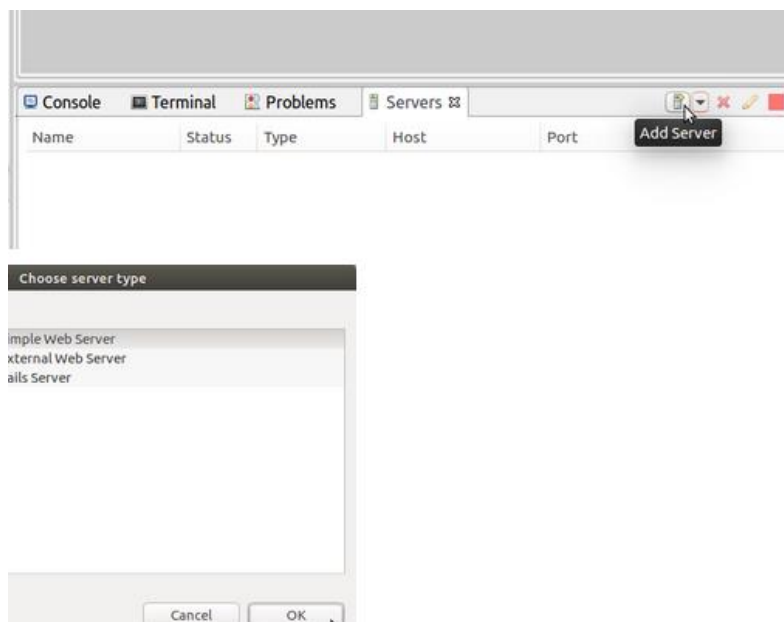
Y en Base URL ponemos http://localhost/*El nombre de vuestra carpeta* que en mi caso es proyecto.



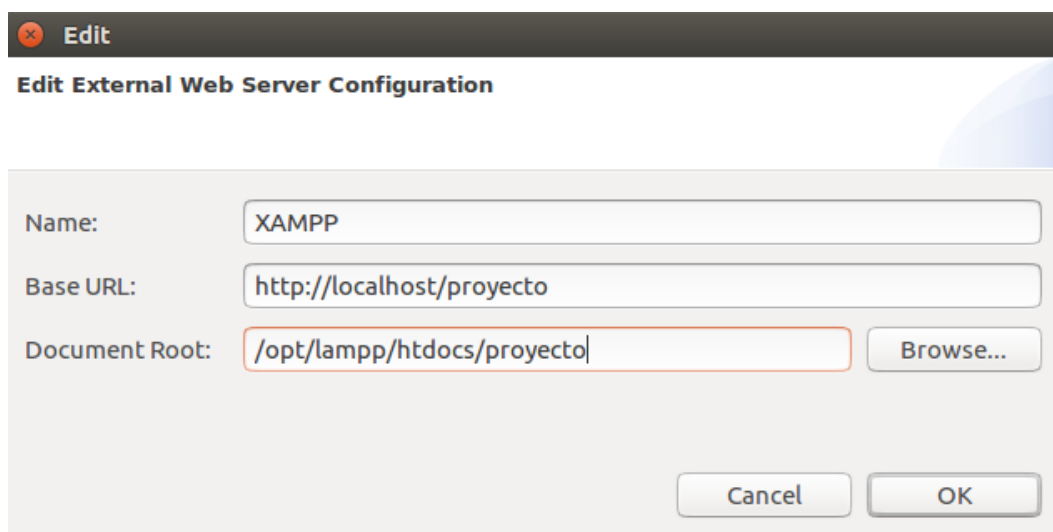
Ahora añadiremos el navegador que se encargará de visualizar nuestros códigos.
Nos vamos a la ruta Show view → Servers, y abajo aparecerá la pestaña de Servers.



Y ahora le damos a añadir servidor y elegimos Simple Web Server:



Y ponemos lo siguiente:



7.2 PROPUESTA DE MEJORA

Al ser la primera vez que hago algo como esto, la optimización de código es un poco malo e interfaz y rendimiento mejorables.

Por lo que como propuesta de proyecto para el curso 2019/2020, propongo mejorar la interfaz de este proyecto, optimizar el código aún más, conseguir que al intentar entrar a una página previamente bloqueada, se redirija a otro puerto en el que estará una página de error personalizada y añadir funcionalidades que el profesor/a crea convenientes.

8. BIBLIOGRAFÍA Y WEBGRAFIA

Configuración de red de la Raspberry:

https://bricolabs.cc/wiki/guias/raspi_-_configuracion_de_red

<https://linuxconfig.org/configuring-virtual-network-interfaces-in-linux>

<http://www.penguintutor.com/raspberrypi/networking-ip-alias-tutorial>

Instalación entorno de desarrollo Aptana:

<https://ubuntulife.wordpress.com/2008/09/06/instalar-aptana-studio-en-ubuntu-y-que-funcione/>

Aprendizaje mínimo para realizar el proyecto:

https://www.ibm.com/support/knowledgecenter/es/SS42VS_7.3.1/com.ibm.qradar.doc/c_qradar_adm_cnfig_advanced_iptables.html

Tutorial para instalar servidor LAMP en Raspberry

https://www.youtube.com/watch?v=YggXN_xJKbs

Como bloquear HTTPS con iptables:

<https://www.enmimaquinafunciona.com/pregunta/140566/como-bloquear-el-sitio-web-https-con-iptables>

<http://linux.dokry.com/iptables-para-bloquear-sitios-web-https.html>

Y también hago referencia a Jorge Sánchez, dueño del código del menú y el css. Al ser código libre, lo he utilizado para mi proyecto.

9. ANEXOS

Toda la información necesaria para el proyecto se podrá descargar aquí:

https://github.com/mouadmtb/firewall_https