Autenticación de Redes Inalámbricas usando chillispot.
Ing. David Omar Guevara Aulestia

En la actualidad conocemos, la importancia y facilidad del uso de las redes inalámbricas, estas nos permiten de una manera muy fácil tener conectividad, pero existen una serie de inconvenientes debido a no prestar atención a aspectos básicos de seguridad.

Las redes inalámbricas cuentan con diferentes estándares de seguridad pero así mismo existen diferentes formas de violentar estos mecanismos que permiten un acceso no autorizado a este tipo de redes.

Para asegurar la interconexión de una red inalámbrica se puede a más de utilizar los mecanismos a nivel de dispositivos, autenticar la interconexión a nivel de usuario, para lo cual es necesario establecer mayores mecanismos de seguridad en la interconexión de dispositivos de una red inalámbrica LAN (WLAN).

Para la autenticación de redes inalámbricas LAN a nivel de usuarios se requiere la instalación de un servidor que permita la interconexión con los dispositivos inalámbricos, para permitir el acceso de acuerdo a una lista de usuarios los mismos que estén acorde a políticas de acceso y seguridad para brindarles los servicios de conexión tanto a la red inalámbrica como a la red cableada, y también a otras redes como el Internet de la forma más cómoda y fácil para el usuario.

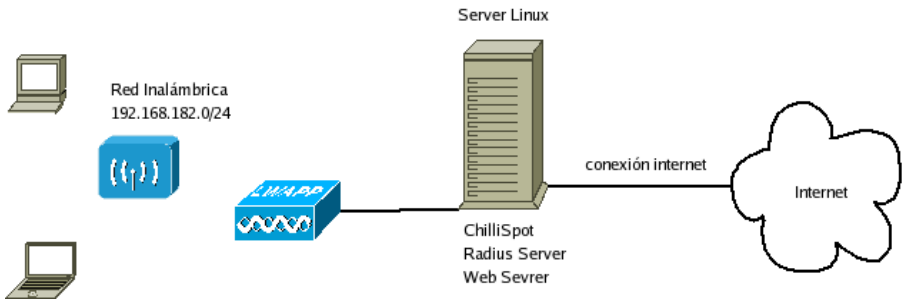

Figura1. Esquema de la interconexión.

Requerimientos:
- Cualquier Distribución Linux: (yo he utilizado Centos)
- Access Point: Cualquier Access Point
- Cliente red Inálambrica
- Acceso a Internet.

Servicios y aplicaciones a Instalar y configurar en el Servidor.
- Apache Web Server
- Mysql
- Radius Server
- chillispot

Una vez instalado el servidor Linux, y actualizado sus paquetes, debemos configurar los servicios y aplicaciones disponibles con la distribución, además descargar las aplicaciones adicionales como chillispot y freeradius, este último a pesar que si existe el correspondiente paquete para la distribución elegida, es necesario instalar la última versión disponible desde sus fuentes, para habilitar elementos adicionales que nos darán mayores prestaciones.

Configuración del Sistema:

Para utilizar el software chillispot es necesario conocer si se posee soporte en nuestro sistema del módulo de kernel TUN/TAP, que permitirá realizar una conexión segura a través de VPN entre el cliente inalámbrico y el server, para eso se realiza:

```
<<LISTING>>
Listado 1. Revisión de soporte TUN/TAP
[root@linux2 ~]# lsmod
Module                  Size  Used by
ipt_MASQUERADE          3649  1
iptable_nat             23037  2 ipt_MASQUERADE
ipt_REJECT              6593  1
ipt_state               1857  1
ip_conntrack            40565  ipt_MASQUERADE,iptable_nat,ipt_state
tun                     9153  0
parport_pc              24577  1
<</LISTING>
```

Vemos que ya existe el soporte de forma automática en el kernel del módulo mencionado. En caso de que no este presente es necesario actualizar el kernel, y para eso se debe contar con el código fuente del kernel para actualizarlo.

Si no hay soporte en la versión de kernel es necesario comprobar todos los requerimientos para la actualización del kernel y luego activar en el mismo lo siguiente.

```
Linux Kernel Configuration: TUN/TAP support
Device Drivers  --->
 Network device support  --->
  <M>   Universal TUN/TAP device driver support
```

Recordemos que este proceso es solo cuando no tenemos por default el soporte para el drive TUN en nuestro kernel.

Una vez asignado el direccionamiento IP para las dos interfaces de red habilitamos la opción de reenvío de paquetes entre las dos interfaces de red, para que el sistema trabaje como un router.

El procedimiento es:

```
[root@linux2 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Es necesario realizar esta tarea cada vez que se reinicie el servidor, para hacerlo de forma automática habilitamos el ip_forward en el archivo `/etc/sysctl.conf`.

Ahora es necesario configurar las aplicaciones y servicios faltantes.

Apache Web Server

El Apache es instalado automáticamente cuando instalamos la distribución Linux.

Para configurar el server web se debe editar su archivo principal de configuración: `/etc/httpd/conf/httpd.conf`

Una configuración **básica** implica comentar la línea que contiene el atributo ServerName:

```
ServerName 192.168.10.22:80
```

Se asigna la dirección ip en caso de no tener configurado el servicio DNS.
Para iniciar el servicio se ejecuta el comando:

```
service httpd start
```

Además se habilita para que el servicio se inicie automáticamente cada vez que el servidor se reinicie:

```
chkconfig httpd on
```

Soporte SSL/TLS en Apache Web Server:

Para enviar información de forma segura entre los clientes y el servidor vía web habilitamos soporte SSL/TLS en el Apache.

El sistema operativo Linux, cuenta con una herramienta de código abierto para la implementación libre de los protocolos SSL (Secure Sockets Layer) y TLS (Transport Layer Security), como es el OpenSSL.

Además se cuenta con el módulo para el servidor web Apache mod_ssl, el cual provee soporte SSL versiones 2 y 3 y TLS versión 1.

Para la implementación de una conexión segura a nivel del servicio web en un equipo de producción es necesario contar con un dirección IP Pública para cada sitio de red que se quiera configurar el soporte SSL/TLS. Ya que cada certificado requiere una única dirección IP para su configuración.

El procedimiento para su implementación es el siguiente:

Verificamos si contamos con los paquetes necesarios

```
rpm -q openssl
openssl-0.9.7a-43.8
rpm -q mod_ssl
mod_ssl-2.0.52-22.ent.centos4
```

Si no estuvieren es necesario instalarlos:

```
yum install openssl mod_ssl
```

Se crea un directorio donde se almacenarán los certificados SSL para el servidor web. Por seguridad es recomendable que este directorio solo sea accesible por parte del superusuario root

```
mkdir -m 700 -p /etc/ssl/ddlinux.com
```

El subdirectorio ddlinux.com se lo crea para mantener cierta organización en caso que se necesite crear certificados para un subdominio o un sitio virtual para el servidor web Apache.

Dentro de la carpeta ddlinux.com se debe crear una clave pública RSA (Acrónimo de Ron Rivest, Adi Shamir y Len Adleman) de 1024 octetos, la cual utiliza Triple DES (Data Encrypton Standard), en formato PEM. Para mejorar la seguridad, se utiliza cinco archivos comprimidos con gzip, para mejorar la seguridad de la clave.

<<LISTING>>
Listado 2. Creación de una clave pública RSA

```
openssl genrsa -des3 -rand \
fichero1.gz:fichero2.gz:fichero3.gz:fichero4.gz:fichero5.gz \
-out server.key 1024
0 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
..++++++
.....++++++
```

```
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```
<</LISTING>>

Para evitar el ingreso de la clave cada vez que se reinicie el servidor web es conveniente generar una clave sin Triple Des. Así se automatiza el hecho de reinicar el server web sin que solicite una clave. Es necesario aclarar que es más seguro dar una clave cada vez que se reinicie el server web, pero para nuestra necesidad no es indispensable.

```
openssl rsa -in server.key -out server.pem
Enter pass phrase for server.key:
writing RSA key
```

Opcionalmente se crea un archivo de petición CSR (Certificate Signing Request) que se hace llegar a un RA (Registration Authority), como Verisign, quienes despues de obtener un pago, envían un certificado (server.crt) firmado por dicha autoridad.

<<LISTING>>

Listado 3. Creación archivo de petición CSR

```
openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
 -----
 Country Name (2 letter code) [GB]:EC
 State or Province Name (full name) [Berkshire]:Tungurahua
 Locality Name (eg, city) [Newbury]:Ambato
 Organization Name (eg, company) [My Company Ltd]:DDLINUX
 Organizational Unit Name (eg, section) []:LINUX
 Common Name (eg, your name or your server's hostname)     []:David Guevara
 Email Address []:david@ddlinux.com

 Please enter the following 'extra' attributes
 to be sent with your certificate request
 A challenge password []:123456789
 An optional company name []:ddlinux
```
<</LISTING>>

Como no se desea adquirir un certificado firmado por un RA, se genera uno propio utilizando un archivo de petición CSR.

<<LISTING>>

Listado 4. Generar archivo de petición CSR.

```
 openssl x509 -req -days 730 -in server.csr \
> -signkey server.key -out server.crt
Signature ok
subject=/C=EC/ST=Tungurahua/L=Ambato/O=DDLINUX/OU=LINUX/CN=David
Guevara/emailAddress=david@ddlinux.com
Getting Private key
Enter pass phrase for server.key:
```
<</LISTING>>

De esta manera se ha generado un cerificado con una duración de 730 días.

Ahora es necesario asegurar que todos los archivos generados sean accesibles solo por el superusuario root.

```
chmod 400 /etc/ssl/ddlinux.com/server.*
```

Se crea la estructura de directorios para el sitio seguro que estará declarado como virtual en el web server.

```
mkdir -p /var/www/ddlinux/{cgi-bin,html,logs,etc,var}
```

En el directorio `/etc/httpd/conf.d` se crea el archivo `ddlinux.conf` con la siguiente información:
<<LISTING>>
Listado 5. Creación de Virtual Host en servidor Apache

```
NameVirtualHost 192.168.10.22:80
        <VirtualHost 192.168.10.22:80>
                ServerAdmin webmaster@ddlinux.com
                DocumentRoot /var/www/ddlinux.com/html
                ServerName www.ddlinux.com
                ServerAlias ddlinux.com
                Redirect 301 / https://www.ddlinux.com/
                CustomLog /var/www/ddlinux.com/logs/access_log combined
                Errorlog /var/www/ddlinux.com/logs/error_log
        </VirtualHost>
NameVirtualHost 192.168.10.22:443
        <VirtualHost 192.168.10.22:443>
                ServerAdmin webmaster@ddlinux.com
                DocumentRoot /var/www/ddlinux.com/html
                ServerName www.ddlinux.com
                ScriptAlias /cgi-bin/ /var/www/ddlinux.com/cgi-bin/
                SSLEngine on
                SSLCertificatefile /etc/ssl/ddlinux.com/server.crt
                SSLCertificateKeyfile /etc/ssl/ddlinux.com/server.pem
                SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
                CustomLog /var/www/fis.uta.edu.ec/logs/ssl_request_log \
                "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
                CustomLog /var/www/ddlinux.com/logs/ssl_access_log combined
                Errorlog /var/www/ddlinux.com/logs/ssl_error_log
        </VirtualHost>
```

<</LISTING>>
Finalmente se reinicia el server web y se comprueba su funcionamiento desde el navegador.

```
        service httpd restart
```

Radius Server:

Este servicio debido a la necesidad de usar opciones adicionales, no se lo instala a partir de archivos tipo rpm, o de repositorio para la distribución Centos. Es necesario descargar el archivo que contiene los fuentes y compilarlo e instalarlo manualmente.

```
        wget ftp://ftp.freeradius.org/pub/radius/freeradius-1.1.3.tar.bz2
```

luego:
<<LISTING>>
Listado 6. Instalación de Server freeradius

```
        bunzip2 freeradius-1.1.3.tar.bz2
        cd freeradius-1.1.3
        ./configure --with-experimental-modules
        make
        make install
        cp ./raddb/dictionary /etc/raddb/dictionary
        cd  /usr/local/etc/raddb
```

<</LISTING>>

Ya instalado el servicio se lo configura para que se conecte al servidor Mysql, así como también para que se conecte al servicio chillispot que se configurará más adelante.

```
        vi radiusd.conf
```

En este archivo se adicionan y cambian los parámetros necesarios para su correcta configuración

`$INCLUDE ${confdir}/sql.conf` (luego de esta línea se adiciona lo siguiente):
<<LISTING>>
Listado 7. Habilitar control de tiempo de conexión en Server freeradius

```
        sqlcounter noresetcounter {
        driver = "rlm_sqlcounter"
```

```
counter-name = Max-All-Session-Time
check-name = "Max-All-Session"
sqlmod-inst = sql
key = User-Name
reset = never
query = "SELECT SUM(AcctSessionTime) FROM radacct WHERE    UserName='%{%k}'"
}
```

<</LISTING>>

Estas líneas son importantes, ya que habilitan las características necesarias para realizar un control de acceso a los servicios de tal forma que se pueda estimar tiempo de conexión, ya sea por horas, por días o mensual, además de establecer parámetros para mejorar la seguridad y la gestión de los usuarios que se conectarán al sistema.

Los otros parámetros del archivo de configuración radiusd.conf a modificar son:

<<LISTING>>

Listado 8. Parámetros que deben constar en radius.conf y sql.conf

En la sección instantiate:

```
instantiate {
exec
expr
noresetcounter
}
```

En la sección authorize:

```
authorize {
preprocess
chap
mschap
suffix
sql
noresetcounter
}
```

En la sección Authentication:

```
authenticate {
Auth-Type PAP {
pap
}
Auth-Type CHAP {
chap
}
Auth-Type MS-CHAP {
mschap
}
}
```

En la sección PreAccounting:

```
preacct {
preprocess
suffix
}
```

En la sección Accounting:

```
accounting {
acct_unique
detail
unix
sql
```

```
}
```

En la sección Session:

```
session {
sql
}
```

Ahora se modifica el archivo sql.conf

```
sql {
       driver = "rlm_sql_mysql"
        server = "localhost"
        login = "freeradius"
        password = "12345"
        radius_db = "radius"
        acct_table1 = "radacct"
        acct_table2 = "radacct"
```

<</LISTING>>
Si se desea activar la opción de control para acceso simultáneo, se debe descomentar la línea:

```
simul_count_query = "SELECT COUNT(*) FROM ${acct_table1} WHERE    UserName='%{SQL-
User-Name}' AND AcctStopTime = 0 AND    CallingStationId<>'%{Calling-Station-Id}'"
```

En el archivo dictionary se adiciona la línea

```
$INCLUDE        /usr/local/share/freeradius/dictionary.wispr
```

Esto habilita el soporte a WISPR que permite configurar parámetros de ancho de banda y de caducidad a las conexiones.

Y finalmente se modifica el archivo clients.conf

```
client 127.0.0.1 {
        secret          = david123
         shortname       = localhost
```

Se realiza la respectiva configuración del servicio para que inicie cuando el Servidor se encienda.

```
echo 'radiusd' >> /etc/rc.d/rc.local
```

Como este servicio no se instalo a partir de archivo rpm, su inicio es diferente de los otros servicios.

Mysql Server:

Como se va a interconectar el servidor de base de datos Mysql y el servidor Radius, se debe configurar el servidor Mysql para que se puedan comunicar y almacenar la información generada por el servidor Radius en la base de datos correspondiente.
<<LISTING>>
Listado 9. Craer la base de datos y establecer un usuario y contraseña para el acceso a la misma

```
mysql -u root -p
>CREATE DATABASE radius;
>GRANT ALL PRIVILEGES ON radius.* to    'freeradius'@'localhost'  IDENTIFIED     BY
'clave';
>FLUSH PRIVILEGES;
>exit
mysql -u root -p \  /usr/local/share/doc/freeradius/examples/mysql.sql
```

<</LISTING>>
Se procede a la creación de un usuario para las respectivas pruebas de conexión entre el servidor

Mysql y el servidor Radius, junto con chillispot.
<<LISTING>>
Listado 10. Creación de una cuenta para las pruebas del sistema

```
mysql -u root -p
>use radius;
>INSERT INTO radcheck (UserName, Attribute, Value) VALUES ('david',
'Password', 'laclave');
>exit
```

<</LISTING>>

Chillispot

Este servicio es el más importante para el propósito de autenticar a los usuarios de la red inalámbrica con el servidor linux, ya que establece una conexión vpn, con el access point y envía los usuarios y claves al servidor radius el mismo que almacenará todos los registros que éste genere en la base de datos. Cuando un usuario quiere acceder a un sitio en el internet el servidor reenvía de forma automática a una página segura en el servidor, que le pide un usuario y contraseña para permitirle acceso.

El proceso de instalación y configuración de este servicio es:

Se descarga el paquete estable más actualizado:

```
wget http://www.chillispot.org/download/chillispot-1.1.0.i386.rpm
```

Instalación

```
rpm -ivh chillispot-1.1.0.i386.rpm
```

Para configurar es necesario editar el archivo chilli.conf
<<LISTING>>
Listado 11. Configuración archivo chilli.conf

```
vi /etc/chilli.conf
net 192.168.182.0/24
(dirección de red que usaran los cliente wirelless)
dynip 192.168.182.0/24
(se establece que ese direccionamiento será dinámico)
dns1 192.168.10.17
(servidor dns a utilizar por los clientes)
domain ddlinux.com
(dominio que usaran los clientes)
ipup /etc/chilli.ipup
ipdown /etc/chilli.ipdown
radiuslisten 127.0.0.1
(dirección del servidor radius)
radiusserver1 127.0.0.1
radiusauthport 1812
radiusacctport 1813
(puertos udp del servidor radius)
radiussecret david123
(Contraseña compartida con el server radius)
dhcpif eth1
(Determinación del interfaz de red a utilizar)
uamserver https://www.ddlinux.com/cgibin/hotspotlogin.cgi
(Dirección que automáticamente se utilizará para la autenticación)
uamsecret ht2eb8ej6s4et3rg1ulp
(Contraseña compartida con el script de autenticación)
uamallowed 192.168.182.1
(Dirección permitida sin necesida de autenticación)
```

<</LISTING>>
Configurado el servicio chillispot, se copia los archivos adicionales para su funcionamiento:
<<LISTING>>
Listado 12. Pasos finales para chillispot

```
cp /usr/share/doc/chillispot-1.1.0/firewall.iptables /etc/rc.d
(se activa para que se ejecute automáticamente)
cd /etc/rc.d/
echo "firewall.iptables" >> /etc/rc.d/rc.local
cp /usr/share/doc/chillispot-1.1.0/hotspotlogin.cgi    \
/var/www/ddlinux.com/cgi-bin/
```

<<LISTING>>

Se puede editar el archivo hotspotlogin.cgi, para personalizar la ventana de autenticación.

Se establece el servicio para su inicio automático

```
chkconfig chilli on
```

Inicio del servicio

```
service chilli start
```

Configuración Access Point

La configuración del Access Point, utilizaremos claves WEP, para permitir a cualquier estación de trabajo inalámbrica, conectarse sin ningún inconveniente, sin importar sistema operativo, o tipo de tarjeta de red. Si deseamos mejorar la seguridad en el lado del Access Point podemos utilizar WPA, ya que el uso de claves WEP, no representan mucha seguridad debido a los mecanismos existentes para violentarlas.

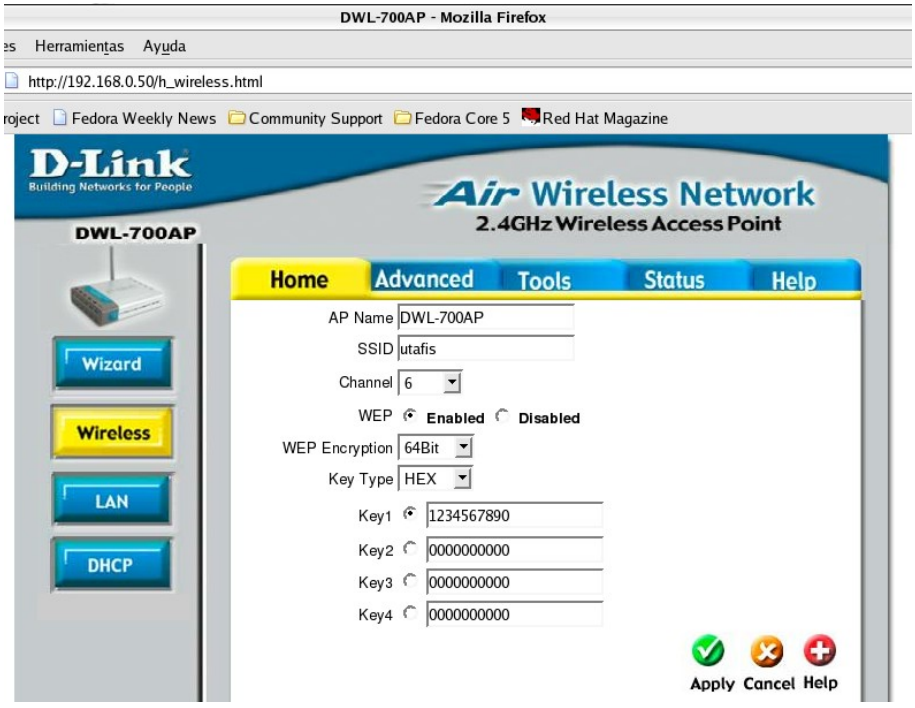

Figura 2. Configuración Access Point

Finalmente podemos crear las cuentas que tendrán acceso a la red, lo haremos desde mysql, usando phpmyadmin, o desde la consola de mysql, también podríamos escribir un programa usando php, para interactuar vía web, con la base de datos.

Creamos el usuario y la contraseña de acceso en la base datos radius en el serveridor Mysql.

```
INSERT INTO radcheck (id, UserName, Attribute, op, Value, CrDate, creator, Location,
activated, activeDate, status, rate) VALUES (NULL, 'david', 'User-Password', ':=',
'clavedavid', '2007-02-19', 'NULL', '0', '0', '0000-00-00 00:00:00', '0', '1');
```

Establecemos si deseamos un control de acceso por horas para el usuario, expresado en segundos.

```
INSERT INTO radcheck (id, UserName, Attribute, op, Value, CrDate, creator, Location, activated, activeDate, status, rate) VALUES (NULL, 'david', 'Max-All-Session', ':=', '10000', '2007-02-19', 'NULL', '0', '0', '0000-00-00 00:00:00', '0', '1');
```

Insertamos al usuario david en un grupo predefinido.

```
INSERT INTO usergroup VALUES(0,'david','grupodavid');
```

Podemos si queremos darle una fecha de caducidad a la cuenta david, para que no tenga acceso después de la fecha establecida.

```
INSERT INTO radreply VALUES(0,'david','WISPr-Session-Terminate-Time',':=','fecha-caduca');
```

Los parámetros que podemos dar a los grupos de usuarios:

Creamos el grupo

```
INSERT INTO radgroupcheck (id, GroupName, Attribute, op, Value) VALUES (NULL, 'grupodavid', 'Auth-Type', ':=', 'Local')";
```

No permitimos que un usuario ingrese al sistema con su usuario más de una vez en forma simultanea.

```
INSERT INTO radgroupcheck (id, GroupName, Attribute, op, Value) VALUES(NULL,'grupodavid','Simultaneous-Use',':=','1')";
```

Le decimos a los usuarios que después de 300 segundos sin hacer nada, se desconecte de forma automática.

```
INSERT INTO radgroupreply (id, GroupName, Attribute, op, Value, prio) VALUES(NULL,'grupodavid','Idle-Timeout',':=','300',0)";
```

Ahora podemos establecer el ancho de banda de bajada y de subida

```
INSERT INTO radgroupreply (id, GroupName, Attribute, op, Value, prio) VALUES(NULL,'grupodavid','WISPr-Bandwidth-Max-Down',':=','128000',0)";

INSERT INTO radgroupreply (id, GroupName, Attribute, op, Value, prio) VALUES(NULL,'grupodavid','WISPr-Bandwidth-Max-Up',':=','32000',0)";
```

Ahora solo resta desde un cliente abrir un navegador y probar nuestra configuración.

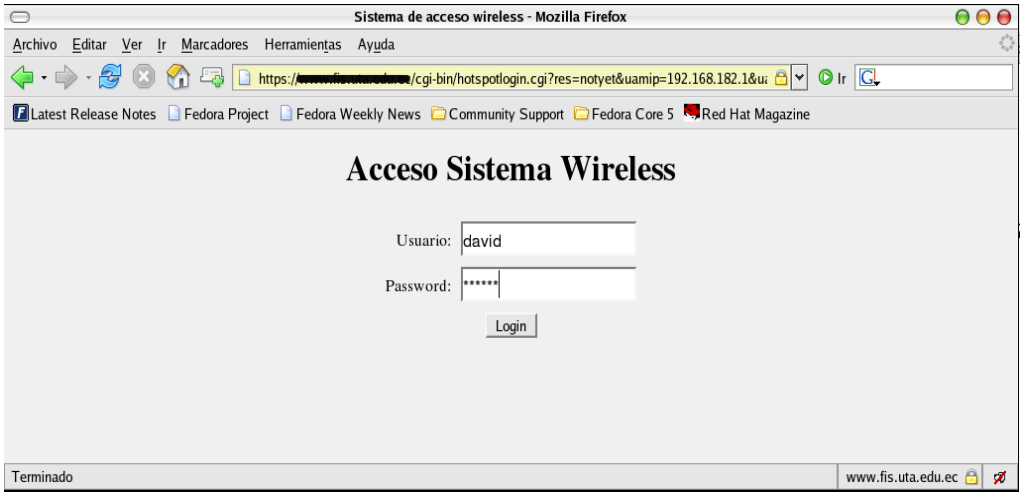

Figura 3. Página de inicio para autenticación de usuarios

Figura 4. Ventana de Acceso Autorizado

Conclusiones:
Al tener un mecanismo de autenticación a nivel de usuario, aseguramos de mejor manera nuestra red inalámbrica, de forma que si pueden acceder o violentar el control de acceso a nivel del Access Point, podrán tener acceso a la red inalámbrica, incluso obtener el direccionamiento IP, pero mientras no ingrese un usuario y una contraseña válidas, el firewall no le dará acceso a la red interna como al internet.

Además al guardar los registros en la base de datos, se puede revisar la dirección IP asignada, como también la MAC address de la estación para tener un control más detallado de lo que esta sucediendo en la red.

Existen varias aplicaciones tipo hotspot o portal cautivo, pero creo de forma personal que chillispot nos brinda mayores facilidades para personalizar el acceso de las redes inalámbricas a nuestros servicios de nuestra red local.

Links:
Sitio Chillispot
www.chillispot.org
HOWTO Chillispot with FreeRadius and MySQL - Gentoo Linux Wiki
http://gentoo-wiki.com/HOWTO_Chillispot_with_FreeRadius_and_MySQL
HOWTO Chillispot with FreeRadius and MySQL - Gentoo Linux Wiki
ttp://www.howtoforge.com/wireless_hotspot_howto