Hot-Spot Wifi

Alessio Fanfani

13 maggio 2011

alessiofanfani@alice.it

http://alessiofanfani.altervista.org

1 Introduzione

Questo documento è stato realizzato per guidare passo passo un qualsiasi utente con medie conoscenze informatiche nella realizzazione di un hotspot wireless. Nel web la documentazione "buona" su questo argomento scarseggia e quindi ho pensato di fare questa guida cercando di elencare nel modo più semplice possibile tutti i passaggi.

Il mio obiettivo è creare un hotspot wireless che sia accessibile agli utenti attraverso delle credenziali di accesso. Una volta connessi, gli utenti potranno navigare liberamente in internet ad eccezione però di alcuni siti. L'hotspot è infatti dotato di un sistema di filtraggio di pagine web, ovvero se un utente prova a visualizzare i siti internet che l'amministratore di rete ritiene dannosi o non idonei. Infine la gestione degli utentu del hotspot dovrà essere il più semplice possibile ed avvenire attraverso interfaccia grafica.

In figura 1 è mostrato lo schema della rete in cui verrà inserito l'access point.

Questa guida è stata realizzata recuperando e rielaborando documenti e informazioni trovati su vari siti web (vedere bibliografia); un doveroso ringraziamento spetta agli autori di tali siti.

Il software utilizzato è tutto rilasciato con licenza Free Software/OpenSource e facilmente reperibile in Internet. In particolare, i programmi utilizzati sono:

Linux Ubuntu 10.04, Apache, ChilliSpot e Coova-Chilli, FreeRadius, MySQL, PhpMyAdmin, PhpMyPrepaid e/o EzRadius, Squid e SquidGuardian.

Nella guida verranno effettuate numerose modifiche a vari file, per questo motivo ho ritenuto utile caricare sul mio sito web un archivio contenente tutti i file necessari all'installazione del hotspot wireless. L'archivio è disponibile all'indirizzo: http://alessiofanfani.altervista.org/hotspot/file.zip

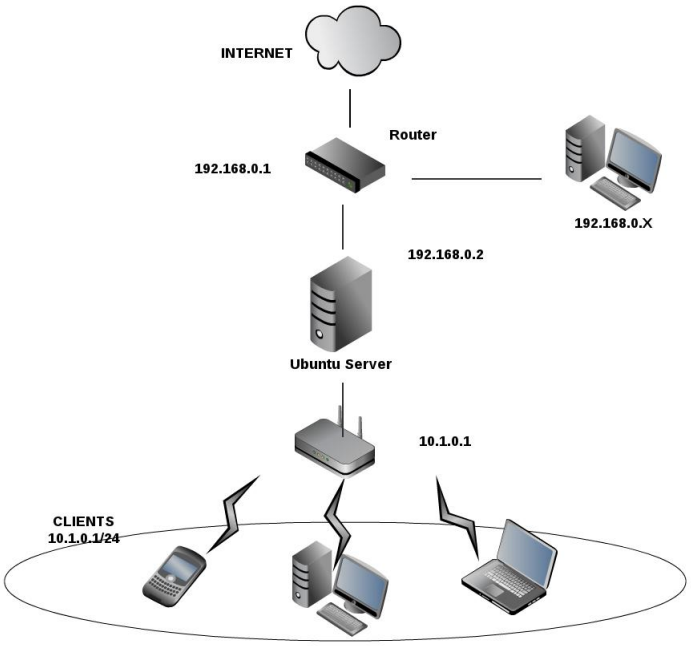

Figura 1: Schema di Rete

La password per decomprimere i file è *alessio.*

1.1 Requisiti

1. una macchina 386 in disuso
2. 2 schede di rete
3. 1 router (o modem)
4. 1 access point wireless

Procuriamoci un disco di installazione di Ubuntu Server 10.04 i386.

Installiamo Ubuntu, scegliendo la configurazione *LAMP* in modo da avere al termine già funzionanti il server web Apache , Mysql e PHP5.

E' consigliato installare anche *ssh* per poter amministrare il server tramite un client di rete.

La spiegazione della procedura di installazione di Ubuntu 10.04 la tralascio poichè la potete trovare su numerossisime guide online.

2 Setup del server

Aprite e modificate il file di configurazione delle interfacce di rete *"/etc/network/interfaces*", con il comando:

```
sudo nano /etc/network/interfaces
```

Il file deve essere simile al seguente:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
#iface eth1 inet dhcp
```

L'interfaccia *eth0* è collegata al router/gateway verso Internet, va quindi impostata o in DHCP (se il router o un altro PC svolge questo servizio) oppure impostando staticamente i parametri di rete.

L'*eth1* si trova collegata con la rete wireless/wired a cui imporre l'autenticazione. L'interfaccia va attivata ma non deve essere impostata ad un indirizzo IP che possa venire impegnato da uno dei dispositivi o host di questa rete. Quindi ci si limita ad attivare l'interfaccia con la direttiva auto.

Coova-Chilli si occuperà di gestire la connessione tramite un'interfaccia tun, che andrà a prendere l'indirizzo del server sulla sottorete fisicamente collegata a eth1.

Infine attivare l'IP Forward del router integrato nel Kernel assicurandosi che nel file *"/etc/sysctl.conf"* sia decommentata la seguente riga:

```
net.ipv4.ip_forward=1
```

ed effettuare la modifica con questo ulteriore comando:

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

Riavviamo adesso la configurazione di rete del server:

```
sudo /etc/init.d/networking restart
```

3 PhpMyadmin

Per gestire i database Sql può essere utile installare phpmyadmin e configurarlo con apache2. Per installarlo usare il comando:

```
sudo apt-get install phpmyadmin
```

quando viene proposta una scelta, selezionare (con il tasto space) la voce *apache2*.

Se non viene proposta nessuna scelta editare il file "/etc/apache2/apache2.conf" aggiungendo in fondo al file la riga:

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

e riavviare apache2 con il comando:

```
sudo /etc/init.d/apache2 restart
```

Testare il funzionamento di phpmyadmin, inserendo in un browser l'indirizzo:

```
http://domain/phpmyadmin
```

dove al posto di domain si deve inserire *localhost*, se il browser è sullo stesso pc in cui abbiamo installato phpmyadmin oppure l'indirizzo ip della macchina in cui è installato phpmyadmin.

Come utente inserite *root* e come password quella utilizzata nella configurazione di sql.

4 Problemi scheda di rete?

Nella fase di installazione si potrebbero verificare alcuni problemi quali:

4.1 Scheda non trovata da ifconfig

Per conoscere come Ubuntu nomina le schede di rete, digitare il comando:

```
ls /sys/class/net
```

Se volete modificare i nomi associati alle schede modificate il file:

```
/etc/udev/ruels.d/70-persistent-net.rules
```

Per una guida aggiornata ed esaustiva (in inglese) andare qui:
http://help.ubuntu-it.org/10.04/ubuntu/serverguide/it/network-configuration.html
Per una meno recente in italiano:
http://help.ubuntu-it.org/9.10/ubuntu/serverguide/it/network-configuration.html

4.2 Scheda non caricata all'avvio

Per caricare una scheda di rete bisogna dare il comando

```
sudo ifconfig ethX up
```

dova al posto di "X" inserire il numero della scheda di rete.

Per caricare la scheda automaticamente all'avvio modificare il file "*/etc/rc.local*" aggiungendo il comando

```
ifconfig ethX up
```

(senza *sudo*) prima di “exit 0”.

5 FreeRadius

Installare i pacchetti mysql-server freeradius e freeradius-mysql:

```
sudo apt-get install mysql-server freeradius freeradius-mysql
```

Usando Ubuntu 10.04 verrà installata la versione 2 di FreeRadius, la quale prevede alcune differenze nella configurazione rispetto alla versione 1.

Il processo di installazione del pacchetto mysql-server richiederà automaticamente la password dell’utente amministratore del database, l’utente "root" e la sua conferma.

Se si volesse modificare la password dare il comando seguente, sostituendo "mysqladminsecret" con la password desiderata:

```
mysqladmin -u root password 'mysqladminsecret'
```

A questo punto dobbiamo creare il database che sarà utilizzato per la gestione del sistema di autenticazione RADIUS, con il seguente comando (rispondere al prompt "Enter password: " con la password dell’utente DB root e poi digitare al prompt "mysql> " i comandi seguenti):

```
mysql -u root -p
Enter password: mysqladminsecret
mysql> CREATE DATABASE radius;
mysql> quit
```

Creare la struttura del database ed un utente con i privilegi di accesso e modifica del database radius:

```
mysql -u root -p radius < /etc/freeradius/sql/mysql/schema.sql
mysql -u root -p radius < /etc/freeradius/sql/mysql/nas.sql
mysql -u root -p
Enter password:mysqladminsecret
mysql> GRANT ALL PRIVILEGES ON radius.* TO 'radius'@'localhost' IDENTIFIED
    BY 'mysqlsecret';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Stare attenti a non confondere le password ’mysqladminsecret’ e ’mysqlsecret’.

Aggiustare il file /etc/freeradius/sql.conf con i dati di connessione al database appena settati:

```
server = "localhost"
login = "radius"
password = "mysqlsecret"
```

La configurazione del server Radius è un affare delicato e non è possibile far operare contemporaneamente una configurazione basata sul file degli utenti Radius memorizzati *"/etc/freeradius/users*" e una dove l'archivio degli utenti è salvato sul database. Si consiglia di procedere a piccoli passi e di tentare prima di configurare il server Radius con una configurazione basata su file (che è quella predefinita).

Configurare l'utente di test decommentando le seguenti righe del file *"/etc/freeradius/users*":

```
"John Doe"      Auth-Type := Local, User-Password == "hello"
                        Reply-Message = "Hello, %u"
```

Configurare il client 127.0.0.1 nel file *"/etc/freeradius/clients.conf"* con il valore della password per le connessioni al server Radius:

```
client localhost {
    ipaddr = 127.0.0.1
    secret = radiussecret
}
```

Per evitare confusione mettete *radiussecret* uguale a *myqlsecret*. A questo punto riavviamo il pc con il comando *sudo reboot.*

Dopo il riavvio testare la configurazione di freeradius con i comandi

```
sudo /etc/init.d/freeradius stop
sudo freeradius -X
```

se tutto è andato a buon fine dovremmo ricevere una risposta che finisce con "Ready to process requests".

A questo punto possiamo uscire dalla modalità debug premendo Ctrl+C ed avviare e testare freeradius con i comandi:

```
sudo /etc/init.d/freeradius start
sudo radtest "John Doe" hello 127.0.0.1 0 radiussecret
```

Se il test è andato a buon fine possiamo cambiare la configurazione, attivando il nostro server MySQL.

Modifichiamo il file "/etc/freeradius/*radiusd.conf"* decommentantando la linea:

```
$INCLUDE sql.conf
```

Successivamente modifichiamo il file *"/etc/freeradius/sites-available/default"* aggiungendo o decommentando la parola *"sql*" nelle sezioni authorize{}, accounting{}, session{} e post-auth{}. Il posto migliore per metterla è nella riga successiva alla parola "files". Nella sezione authorize{} la parola "files" deve essere commentata aggiungendo # davanti ad essa.

E' buona norma testare questa nuova configurazione prima di andare avanti; stoppiamo dunque il servizio freeradius e riavviamolo nuovamente in modalità di debug.

```
sudo /etc/init.d/freeradius stop
sudo freeradius -X
```

Se riceviamo come risposta finale “Ready to process requests” possiamo continuare altrimenti dobbiamo ricontrollare i passaggi precedenti. Riavviamo Freeradius:

```
sudo /etc/init.d/freeradius restart
```

Come ultimo step aggiungiamo un utente test nel database; il comando generico è (su un’unica linea):

```
echo "INSERT INTO radcheck (UserName, Attribute, Value) VALUES ('radtest',
    'Password', 'pswtest');" | mysql -u radius -p radius
```

e scegliamo come utente e password di prova ’chillispot’; il comando diventa quindi:

```
echo "INSERT INTO radcheck (UserName, Attribute, Value) VALUES ('chillispot
    ', 'Password', 'chillispot');" | mysql -u radius -p radius
```

alla richiesta della password inserite ’mysqlsecret’.

E ritentiamo il test di prima con i nuovi valori:

```
sudo radtest chillispot pswtest 127.0.0.1 0 radiussecret
```

ovvero

```
sudo radtest chillispot chillispot 127.0.0.1 0 radiussecret
```

6 CoovaChilli

6.1 Installazione

Per installare CoovaChilli andare sul sito http://www.coova.org/CoovaChilli e copiare il link dell’ultima versione del programma compatibile con il nostro sistema operativo.

Questa guida si basa su CoovaChilli 1.2.5.

Scaricare il file .deb nella cartella /tmp con i comandi:

```
cd /tmp
sudo wget "link"
sudo dpkg -i *.deb
```

dove al posto di “link” si inserisce il link precedentemente trovato.

Dare i seguenti comandi per configurare Apache e CoovaChilli

```
sudo -s
cd /
cp /etc/chilli/defaults /etc/chilli/config
mkdir /var/www/hotspot
```

```
cp /etc/chilli/www/* /var/www/hotspot
sed -i 's/1.0.0.1/10.1.0.1/g' /etc/chilli/www/ChilliLibrary.js
sed -i 's/1.0.0.1/10.1.0.1/g' /var/www/hotspot/ChilliLibrary.js
```

Modificare il file "*/etc/default/chilli*" per abilitare CoovaChilli:

```
nano /etc/default/chilli
```

e modificarlo in

```
START_CHILLI=1
CONFFILE="/etc/chilli.conf"
HS_USER="chilli"
```

6.2 Configurazione

La configurazione di base è contenuta nel file "*/etc/chilli/defaults*". Questo file deve essere salvato nella stessa directory con il nome "*config*" poichè tutte le modifiche devono essere fatte su questo nuovo file. Ogni volta che il file "*config*" verrà modificato bisogna dare il comando "*sudo /etc/init.d/chilli restart*" che avvierà Chilli e genererà i nuovi file "*main.conf*", "*local.conf*" e "hs.conf" nella cartella "/etc/chilli/".

Per eseguire CoovaChilli in modalità debug bisogna dare il comando "*sudo chilli –debug –fg*". La modalità debug non modifica però nessun file tra quelli precedentemente elencati, perciò prima di eseguire la modalità debug dare i comandi "*sudo /etc/init.d/chilli restart*" e "*sudo /etc/init.d/chilli stop*".

Per default, la scheda di rete che è collegata verso l'esterno è *eth0* mentre l'interfaccia alla quale sono collegati i clienti della rete è *eth1*.

Iniziamo con i comandi:

```
sudo cp /etc/chilli/defaults /etc/chilli/config
sudo nano /etc/chilli/config
```

e modificare il file nel seguente modo:

```
HS_LANIF=eth1
HS_NETWORK=10.1.0.0
HS_NETMASK=255.255.255.0
HS_UAMLISTEN=10.1.0.1
HS_UAMPORT=3990
HS_NASID=nas01
HS_RADIUS=127.0.0.1
HS_RADIUS2=127.0.0.1
HS_UAMALLOW=10.1.0.0/24,192.168.0.0/24,208.67.222.222,208.67.220.220
HS_RADSECRET=radiussecret
HS_UAMSECRET=uamsecret
```

```
HS_TCP_PORTS="80 443"

HS_UAMSERVER=10.1.0.1
HS_UAMFORMAT=https://\$HS_UAMSERVER/uam/
HS_UAMHOMEPAGE=http://\$HS_UAMSERVER:\$HS_UAMPORT/www/coova.html
HS_UAMSERVICE=https://\$HS_UAMSERVER/cgi-bin/hotspotlogin.cgi

HS_MODE=hotspot
HS_TYPE=chillispot

HS_WWWDIR=/etc/chilli/www
HS_WWWBIN=/etc/chilli/wwwsh

HS_LOC_NAME="HotSpot"

HS_DEFBANDWIDTHMAXDOWN=512000   # Banda Massima Down in bit/s
HS_DEFBANDWIDTHMAXUP=64000   # Banda Massima Up in bit/s
```

Modificare il file "/etc/rc.local" per abilitare CoovaChilli:

```
nano /etc/rc.local
```

ed aggiungere il comando

```
/etc/init.d/chilli start
```

prima di *exit0.*

7 Firewall

Per il corretto funzionamento di CoovaChilli bisogna configurare il firewall di Ubuntu.

Come prima cosa disinstalliamo ufw:

```
sudo ufw disable
sudo apt-get remove ufw
```

A questo punto bisogna lavorare con iptables. All'avvio di CoovaChilli viene attivato il seguente script *"/etc/chilli/up.sh"*. Alla fine di questo script bisogna aggiungere le seguenti regole:

Facciamo prima una copia di backup e poi editiamo il file:

```
sudo cp  /etc/chilli/up.sh  /etc/chilli/up.sh.original
sudo nano /etc/chilli/up.sh
```

Ecco le righe da aggiungere:

```
[ -e "/var/run/chilli.iptables" ] && sh /var/run/chilli.iptables 2>/dev/null

iptables -I POSTROUTING -t nat -o $HS_WANIF -j MASQUERADE
```

```
iptables -P INPUT DROP iptables -I INPUT -m state --state RELATED,ESTABLISHED -j
    ACCEPT

iptables -I INPUT -p icmp --icmp-type 8 -s 192.168.0.0/24 -d 10.1.0.1   -m state --
    state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -I INPUT -p icmp --icmp-type 0 -s 0/0 -d 10.1.0.1      -m state --state
    ESTABLISHED,RELATED -j ACCEPT
iptables -I OUTPUT -p icmp --icmp-type 8 -s 10.1.0.1 -d 0/0     -m state --state
    NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -I OUTPUT -p icmp --icmp-type 0 -s 10.1.0.1 -d 0/0     -m state --state
    ESTABLISHED,RELATED -j ACCEPT
iptables -I FORWARD -p icmp -s 10.1.0.0/24 -d 0/0 -j ACCEPT

iptables -I INPUT -i eth0 -s 192.168.0.0/24 -p tcp -m tcp --dport 22 --dst
    192.168.0.0/24 -j ACCEPT

iptables -I INPUT -i lo -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 443 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 3990 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 4990 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 80 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 53 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 67:68 --syn -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 3306 --syn -j ACCEPT
iptables -A INPUT -i tun0 -j DROP
```

Nota: Tutti i comandi sono su una riga sola ed iniziano con iptables

Inoltre, possiamo andare a modificare lo script che viene lanciato allo spegnimento di CoovaChilli. Facciamo prima una copia di backup e poi editiamo il file:

```
sudo cp  /etc/chilli/down.sh  /etc/chilli/down.sh.original
sudo nano /etc/chilli/down.sh
```

Ecco le righe da aggiungere:

```
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -F POSTROUTING
```

Infine bisogna rendere eseguibili i due script

```
sudo chmod +rx /etc/chilli/up.sh
sudo chmod +rx /etc/chilli/down.sh
```

8 Apache Server

Andremo adesso a creare la pagina di login per il nostro hotspot, partendo da quella realizzata da *Liran Tal* [1] che ho caricato per praticità sul mio sito. Tale pagina può essere personalizzata a piacere editando i file contenuti nella cartella *template.*

```
sudo -s
cd /var/www/hotspot
wget http://alessiofanfani.altervista.org/hotspot/uam.tgz
tar xvf uam.tgz
rm uam.tgz
```

Modificare il file "*index.php*" con il comando:

```
nano /var/www/hotspot/uam/index.php
```

e sostituire la password uamsecret.

```
$uamsecret = "uamsecret";
$userpassword=1;
```

e rendiamo eseguibili gli script.

```
sudo chmod -R 555 /var/www/hotspot
```

Nota Facoltativa:

Nell'archivio http://alessiofanfani.altervista.org/hotspot/file.zip è presente la pagina di login da me realizzata che prevede anche altre pagine quali le informazioni per l'utente e le condizioni di servizio.

Questi file sono presenti nelle cartelle "*www*" e "*uam*" dell'archivio. Il contenuto della cartella "*www*" deve essere copiato in:

```
/var/www/
```

mentre il contenuto della cartella *"uam"* deve essere copiato in:

```
/var/www/hotspot/uam/
```

Dopo aver copiato questi file ricordarsi di dare i giusti permessi con il comando:

```
sudo chmod -R 555 /var/www/hotspot
```

[1] <liran@enginx.com>

9 SSL

Poichè l'hotspot richiede una connessione SSL bisogna installare e configurare i seguenti pacchetti:

```
sudo apt-get install libapache2-mod-auth-mysql ssl-cert
```

Creiamo un certificato ssl inserendo come *hostname* il risultato del comando "*hostname -f* ".

```
sudo mkdir /etc/apache2/ssl
sudo make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
```

Attiviamo il modulo attraverso i comandi:

```
sudo a2enmod ssl
sudo /etc/init.d/apache2 force-reload
```

Creiamo un virtualhost, andando ad editare il file *"/etc/apache2/sites-available/hotspot*":

```
sudo nano /etc/apache2/sites-available/hotspot
```

e compiliamolo nel seguente modo

```
NameVirtualHost 10.1.0.1:443
<VirtualHost 10.1.0.1:443>
    ServerAdmin webmaster@domain.org
        DocumentRoot "/var/www/hotspot"
        ServerName "10.1.0.1"
        <Directory "/var/www/hotspot/">
                Options Indexes
                FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        Alias "/dialupadmin/" "/usr/share/freeradius-dialupadmin/htdocs/"

        <Directory "/usr/share/freeradius-dialupadmin/htdocs/">
                Options Indexes
                FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bi/ /var/www/hotspot/cgi-bin/
```

```
        <Directory "/var/www/hotspot/cgi-bin/">
                AllowOverride None
                Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog /var/log/apache2/hotspot-error.log

        LogLevel warn

        CustomLog /var/log/apache2/hotspot-access.log combined

        ServerSignature On
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/apache.pem
</VirtualHost>
```

Infine abilitiamolo con i comandi:

```
sudo a2ensite hotspot
/etc/init.d/apache2 reload
```

Configurazione Apache

Il servizio HTTPS sarà in ascolto sulla porta 443. Dovremmo quindi editare il file *"/etc/apache2/ports.conf*":

```
sudo cp /etc/apache2/ports.conf /etc/apache2/ports.conf.original
sudo nano /etc/apache2/ports.conf
```

e modificarlo così:

```
Listen *:443
Listen *:80
#<IfModule mod_ssl.c>
#    Listen 443
#</IfModule
```

Infine editiamo anche il file "*/etc/apache2/sites-available/default*" facendo prima una copia:

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/
    default.original
sudo nano /etc/apache2/sites-available/default
```

e modificando così le prime 2 righe:

```
NameVirtualHost *:80
<virtualhost *:80>
```

Configuriamo apache editando il file "*/etc/apache2/apache2.conf*":

```
sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.original
sudo nano /etc/apache2/apache2.conf
```

ed aggiungendo all'inizio del file:

```
ServerName 10.1.0.1
```

Ed infine editiamo *"/etc/hosts*"

```
sudo cp /etc/hosts /etc/hosts.original
sudo nano /etc/hosts
```

aggiungendo la riga:

```
10.1.0.1    host.name host #change to your host name
```

Infine riavviamo apache con:

```
sudo /etc/init.d/apache2 restart
```

10 Haserl

Per far funzionare lo script bisogna installare haserl; trovare il link dal quale scaricare l'ultima versione del programma sul sito http://haserl.sourceforge.net/.

Come prerequisito installare *"gcc"* con il comando*:*

```
sudo apt-get install gcc
```

Poi dare i seguenti comandi:

```
cd /tmp/
sudo wget http://sourceforge.net/projects/haserl/files/haserl-devel/haserl
    -0.9.28.tar.gz
tar xvf haserl*
cd haserl*
./configure
make
make install
```

Editare il file "*/etc/chilli/wwwsh*" con il comando *"sudo nano /etc/chilli/wwwsh"* e sostituiamo:

```
haserl=$(which haserl 2>/dev/null)
```

con

```
haserl=/usr/local/bin/haserl
```

11 PhpMyPrepaid o EzRadius

Dobbiamo adesso installare un programma che ci consenta una gestione facilitata degli utenti dell'hotspot. Online esistono vari programmi adatti a fare cio, nella guida spigherò l'installazioni di EzRadius e PhpMyPrepaid. Ovviamente è sufficiente l'installazione di uno solo dei 2 programmi; io consiglio il primo per la semplicità di utilizzo.

Prima di cominciare l'installazione facciamo un backup del database radius attraverso PhpMyAdmin. Un backup è disponibile anche nell'archivio dei file caricato sul mio sito.

11.1 EzRadius

Iniziamo l'installazione dando i seguenti comandi:

```
sudo -s
cd /var/www/hotspot/
wget http://sourceforge.net/projects/ezradius/files/ezradius-comm/ezradius-
    comm-0.2.1stable/ezradius-comm-0.2.1.tar.gz
tar xvf ezradius-comm-0.2.1.tar.gz
rm ezradius-comm-0.2.1.tar.gz
chmod -R 555 /var/www/hotspot/ezradius-comm
```

Accedere al seguente sito Web "http://192.168.0.2/hotspot/ezradius-comm/" ed autenticarsi con nome utente *admin* e password *admin* (sono predefiniti alla prima installazione).

Configurare l'interfaccia dal menù Tool->Config editor e inserire i segunti parametri:

- Username: admin
- Password: <una password per accedere a ezRadius>
- MySQL Host: localhost
- Database name: radius
- MySQL Username: radius
- MySQL Password: mysqlsecret

Poichè EzRadius è pensato per la versione 1 di Freeradius, dobbiamo modificare il database *radius* affichè si possano aggiungere correttamente gli utenti.

Tornare sul terminale e dare i comandi:

```
mysql -u radius -p radius
Enter password:mysqlsecret
mysql> CREATE TABLE IF NOT EXISTS usergroup LIKE radusergroup;
mysql> quit
```

A questo punto potete aggiungere in modo molto semplice gli utenti accedendo al sito: "http://192.168.0.2/hotspot/ezradius-comm/"

11.2 PhpMyPrepaid

Iniziamo l'installazione attraverso i seguenti comandi:

```
sudo -s
cd /var/www/hotspot/
wget http://sourceforge.net/projects/phpmyprepaid/files/phpmyprepaid/
    Phpmyprepaid-RC3/phpmyprepaidRC3.tgz
tar xvf phpmyprepaidRC3.tgz
rm phpmyprepaidRC3.tgz
chmod -R 555 /var/www/hotspot/phpmyprepaid
cd phpmyprepaid
```

Configurare il programma da un browser web inserendo come url:
http://192.168.0.2/hotspot/phpmyprepaid/www/.
Seguite questi punti in base al numero di schermata:

1. Cliccate start.
2. Accettate il contratto e cliccate next.
3. Se avete una schermata come in figura 2 cliccate next.

Figura 2: Schermata 3

4. Cliccate next.
5. Modificare come in figura 3 e cliccate next.

6. Inserire i dati che volete e cliccate next. La password dell'utente *admin* la chiameremo da ora in poi *"prepaidsecret"*.

7. Root password Msql è "*mysqladminsecret*" mentre nel campo PhpMyPrepaid Database Password inserite "mysqlsecret". IMPORTANTE: nel campo PhpMyPrepaid Database Name inserite "*radius*", come in figura 4.

8. Nelle prossime schermate leggete e Cliccate next.

9. Per accedere all'interfaccia di PhpMyPrepaid andare all'url:

http://192.168.0.2/hotspot/phpmyprepaid/www/.

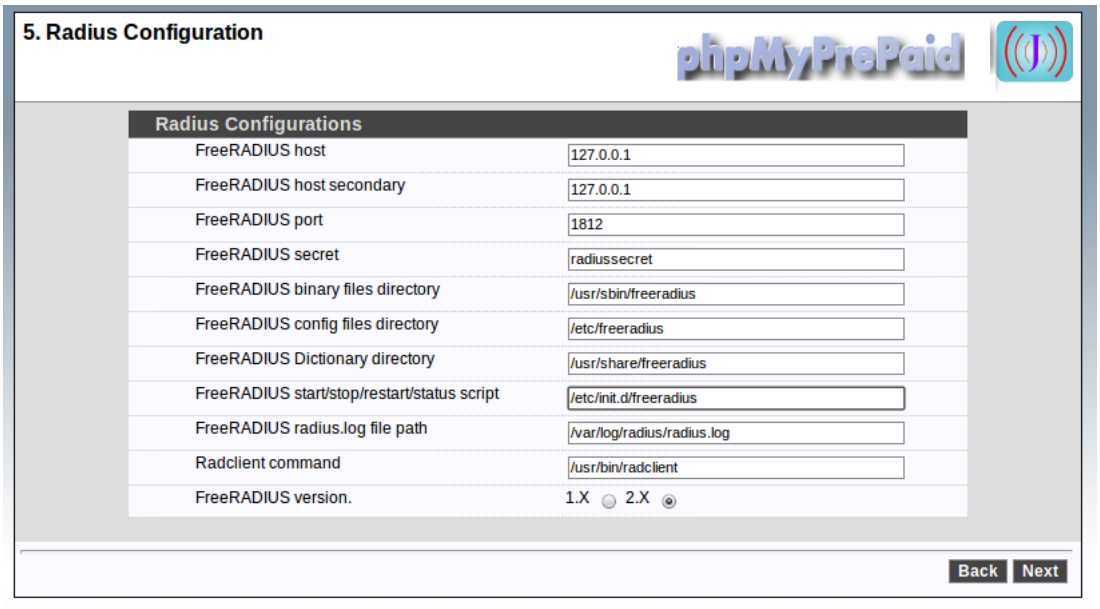

Figura 3: Schermata 5

Figura 4: Schermata 7

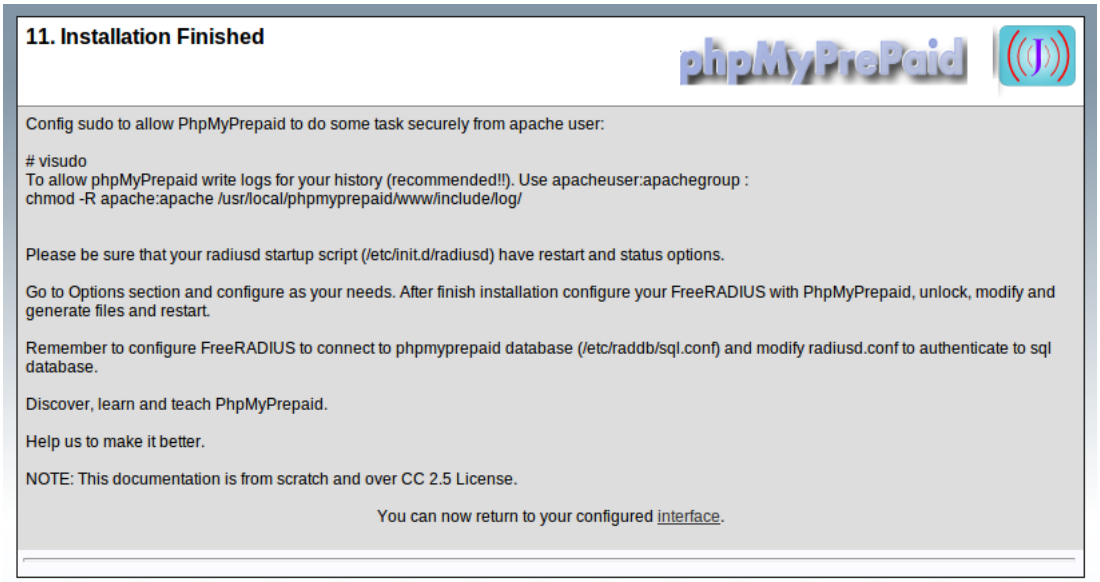

Figura 5: Schermata Finale

12 Squid & SquidGuard

Un'ultima funzione da implementare nel nostro sistema sarà il controllo dei siti visitabili dagli utenti del hotspot. Il programma scelto per fare ciò è squidGuard e permette di reindirizzare l'utente ad una pagina specifica ogni volta che esso prova ad accedere ad alcuni domini che sono presenti all'interno di una blacklist ovvero una lista contenente i siti da bloccare. SquidGuard basa il suo funzionamento sul proxy server Squid.

12.1 Installazione Squid

Come prima passo andiamo ad installare e configurare squid.

Installiamo Squid:

```
sudo apt-get install squid3
```

ed editiamo il file di configurazione *"/etc/squid3/squid.conf*", con il comando:

```
sudo nano /etc/squid3/squid.conf
```

aggiungendo le seguenti righe all'inizio del file:

```
http_port 10.1.0.1:3128 transparent
acl our_networks src 10.1.0.0/24
acl localnet src 127.0.0.1/255.255.255.255
http_access allow our_networks
http_access allow localnet
```

Configuriamo adesso CoovaChilli in modo che lavori correttamente insieme a Squid:

Abilitiamo la funzionalità di proxy dopo l'autenticazione integrata in chilli, aggingendo nel file di configurazione "*/etc/chilli/config*" le seguenti righe:

```
HS_POSTAUTH_PROXY=10.1.0.1
HS_POSTAUTH_PROXYPORT=3128
```

Infine modifichiamo il file *"/etc/chilli/up.sh*" aggiungendo dopo la riga:

```
iptables -I INPUT -p tcp -m tcp --dport 80 --syn -j ACCEPT
```

il comando:

```
iptables -I INPUT -p tcp -m tcp --dport 3128 --syn -j ACCEPT
```

Nota:

Un comando utile per visualizzare le opzioni attive nel file di configurazione di squid (che è molto lungo e complitato) è:

```
grep -v "^#" /etc/squid3/squid.conf | sed -e '/^$/d'
```

12.2 SquidGuard

Installiamo squidGuard con il comando:

```
sudo apt-get install squidguard
```

Scarichiamo i file contenenti le blacklist nella cartella "*/var/lib/squidguard/db/*" e scompattiamoli. Nella guida userò il l'archivio presente sul mio sito però vi consiglio di cercare online versioni delle blacklist più aggiornate. Un ottimo sito è http://urlblacklist.com/.

```
sudo -s
cd /var/lib/squidguard/db/
wget http://alessiofanfani.altervista.org/hotspot/blacklists.tar.gz
tar xvf blacklists.tar.gz
rm blacklists.tar.gz
```

Facciamo una copia di backup e poi editiamo il file di configurazione "/etc/squid/squidGuard.conf", con il comando:

```
sudo cp /etc/squid/squidGuard.conf /etc/squid/squidGuard.conf.original
sudo nano /etc/squid/squidGuard.conf
```

Il mio file di configurazioni è il seguente:

```
# CONFIG FILE FOR SQUIDGUARD #
dbhome /var/lib/squidguard/db/blacklists
logdir /var/log/squid3

src clients {
ip              10.1.0.0/24
}

dest abortion {
        domainlist      abortion/domains
        urllist         abortion/urls
        expressionlist  abortion/expressions
        redirect        http://10.1.0.1/block.html
}

dest ads {
        domainlist      ads/domains
        urllist         ads/urls
        expressionlist  ads/expressions
        redirect        http://10.1.0.1/block.html
}

dest adult {
        domainlist      adult/domains
        urllist         adult/urls
        expressionlist  adult/expressions
        redirect        http://10.1.0.1/block.html
}

dest aggressive {
        domainlist      aggressive/domains
        urllist         aggressive/urls
```

```
        redirect        http://10.1.0.1/block.html
}

dest drugs {
        domainlist      drugs/domains
        urllist         drugs/urls
        redirect        http://10.1.0.1/block.html
}

dest filesharing {
        domainlist      filesharing/domains
        urllist         filesharing/urls
        redirect        http://10.1.0.1/block.html
}

dest gambling {
        domainlist      gambling/domains
        urllist         gambling/urls
        redirect        http://10.1.0.1/block.html
}

dest games {
        domainlist      games/domains
        urllist         games/urls
        redirect        http://10.1.0.1/block.html
}

dest hacking {
        domainlist      hacking/domains
        urllist         hacking/urls
        redirect        http://10.1.0.1/block.html
}

dest manga {
        domainlist      manga/domains
        urllist         manga/urls
        redirect        http://10.1.0.1/block.html
}

dest mixed_adult {
        domainlist      mixed_adult/domains
        urllist         mixed_adult/urls
        redirect        http://10.1.0.1/block.html
}
```

```
dest onlinegames {
        domainlist      onlinegames/domains
        urllist         onlinegames/urls
        redirect        http://10.1.0.1/block.html
}

dest phishing {
        domainlist      phishing/domains
        urllist         phishing/urls
        redirect        http://10.1.0.1/block.html
}

dest porn {
        domainlist      porn/domains
        urllist         porn/urls
        expressionlist  porn/expressions
        redirect        http://10.1.0.1/block.html
}

dest proxy {
        domainlist      proxy/domains
        urllist         proxy/urls
        redirect        http://10.1.0.1/block.html
}

dest sect {
        domainlist      sect/domains
        urllist         sect/urls
        redirect        http://10.1.0.1/block.html
}

dest sexuality {
        domainlist      sexuality/domains
        urllist         sexuality/urls
        redirect        http://10.1.0.1/block.html
}

dest spyware {
        domainlist      spyware/domains
        redirect        http://10.1.0.1/block.html
}

dest violence {
```

```
        domainlist      violence/domains
        urllist         violence/urls
        redirect        http://10.1.0.1/block.html
}

dest virusinfected {
        domainlist      virusinfected/domains
        urllist         virusinfected/urls
        redirect        http://10.1.0.1/block.html
}

dest warez {
        domainlist      warez/domains
        urllist         warez/urls
        redirect        http://10.1.0.1/block.html
}

dest webmail {
        domainlist      webmail/domains
        urllist         webmail/urls
}

dest whitelist {
        domainlist      whitelist/domains
        urllist         whitelist/urls
}

acl {
        clients {
                pass    whitelist webmail  !abortion !ads !adult !
                    aggressive !drugs !filesharing !gambling !games !
                    hacking !manga !mixed_adult !onlinegames !phishing !
                    porn !proxy !sect !sexuality !spyware !violence !
                    virusinfected !warez !violence
        }

        default {
                pass     none
                redirect http://10.1.0.1/block.html
        }
}
```

Per poter utilizzare squidGuard bisogna modificare il file di configurazione di squid in modo da indicare al programma dove risiede squidGuard. Editiamo il file di configurazione *"/etc/squid3/squid.conf*",

con il comando:

```
sudo nano /etc/squid3/squid.conf
```

aggiungendo la seguente riga:

```
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

Creiamo la pagina alla quale saranno reindirizzati gli utenti che tentano di accedere ai siti bloccati.

```
sudo nano /var/www/block.html
```

Ed inseriamo il seguente codice html:

```
<html>
<head><title>Avviso</title></head>
<body><br><br>
<div style="text-align: center;"><big style="color: red;">!!! ATTENZIONE
    !!!</big>
<br><br>
Il sito web è stato bloccato per motivi di sicurezza.<br>
</div>
<br>
</body>
</html>
```

Diamo i giusti permessi ai file:

```
sudo chown proxy:proxy -R /var/lib/squidguard/db
sudo chown proxy:proxy /etc/squid/squidGuard.conf
sudo chown -R proxy:proxy /var/lib/squidguard/db
sudo chown -R proxy:proxy /var/log/squid/
sudo chown -R proxy:proxy /var/log/squid3/
sudo chmod 777 /etc/squid/squidGuard.conf
sudo chmod -R 777 /var/lib/squidguard/db
sudo chmod -R 777 /var/log/squid/
sudo chmod -R 777 /var/log/squid3/
sudo chmod 555 /var/www/block.html
```

Attiviamo SquidGuard con i comandi:

```
sudo squidGuard -C all
```

Infine riconfiguriamo Squid con il comando:

```
squid3 -k reconfigure
```

Nota:

Ecco alcuni comandi utili per controllare il funzionamento si Squid:

Per lanciare Squid in modalità verbosa:

```
sudo squid3 -NCd1
```

Nel caso di modifiche alla configurazione, per riavviare squid e squidGuard con le nuove impostazioni è sufficiente digitare il seguente comando:

```
squid3 -k reconfigure
```

Per vedere se il processi di squid sono avviati:

```
ps -e | grep squid
```

Riferimenti bibliografici

[1] http://faberlibertatis.org/wiki/Hotspot_Ubuntu_Hardy_Server_HOWTO

[2] http://old.nabble.com/Come-aggiungere-scheda-di-rete—(ubuntu-server)-td30064446.html

[3] https://help.ubuntu.com/community/WifiDocs/ChillispotHotspot

[4] http://manajung.blogspot.com/2010/09/ubuntu-wi-fi-hotspot-using-coovachilli.html

[5] http://www.blog.highub.com/linux/install-and-configure-phpmyadmin-on-ubuntu-lamp/

[6] http://web.mit.edu/rhel-doc/3/rhel-rg-it-3/s1-iptables-options.html

[7] http://dema.tv/2009/05/04/wifi-in-hotel-howto-tecnico/

[8] http://www.ubuntugeek.com/setting-up-ubuntu-10-04-lucid-server-with-squid-3-as-a-transparent-proxy.html

[9] http://wiki.ubuntu-it.org/Server/SquidGuard

[10] http://gionn.net/squidguard-su-squid3-filtraggio-web