

Hardware/Software Setup Required

Ubuntu Server 9.10 VMWare image (available at <http://www.thoughtpolice.co.uk/vmware/>)
(available at <http://www.vmware.com/appliances/directory/cat/0?k=Ubuntu%20Server%209.10>)
VMWare (available at http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0)

Chillispot (available at <http://www.chillispot.info/download.html>)

FreeRadius (available at <http://freeradius.org/>)

Problem Description

A hotspot is a place offering Internet connection via a wireless network. Hotspots can be either public (free) or private (paid). A private hotspot features a captive portal for users to authenticate in order to gain access to the hotspot.

In this lab, you will explore the steps for creating your own private hotspot using ChilliSpot and FreeRadius over Ubuntu.

Estimated completion time: 90 minutes.

Outcome

Report the steps to perform the tasks

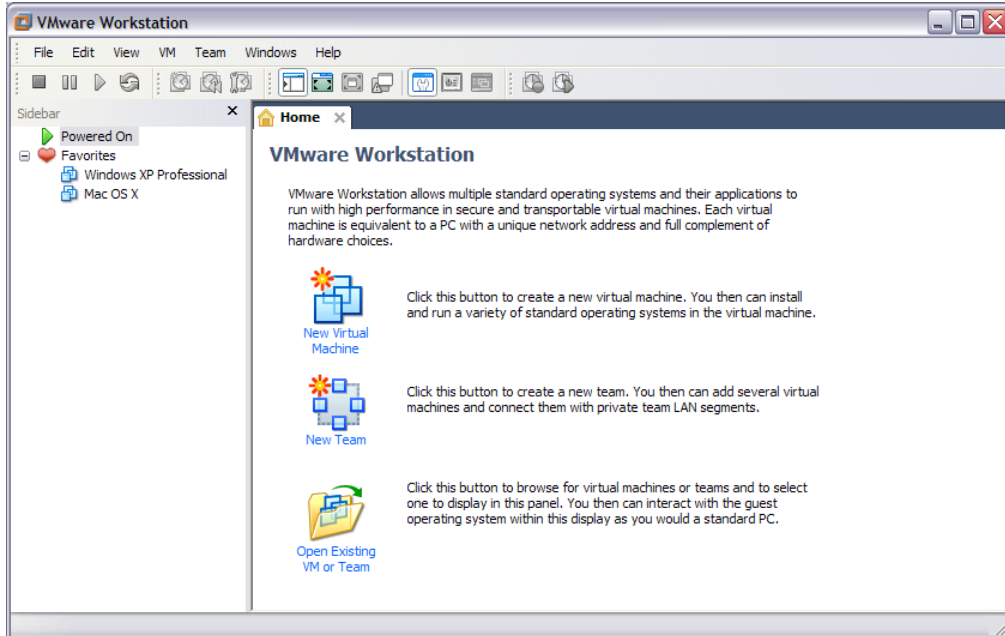
Validation/Evaluation

Be able to:

- Install and configure a LAMP server
- Install and configure ChilliSpot
- Install and configure FreeRadius with MySql
- Create an SSL site with Apache2
- Configure Internet Connection Sharing

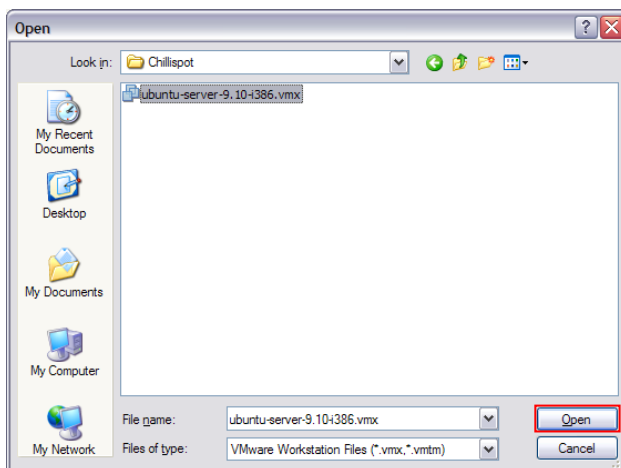
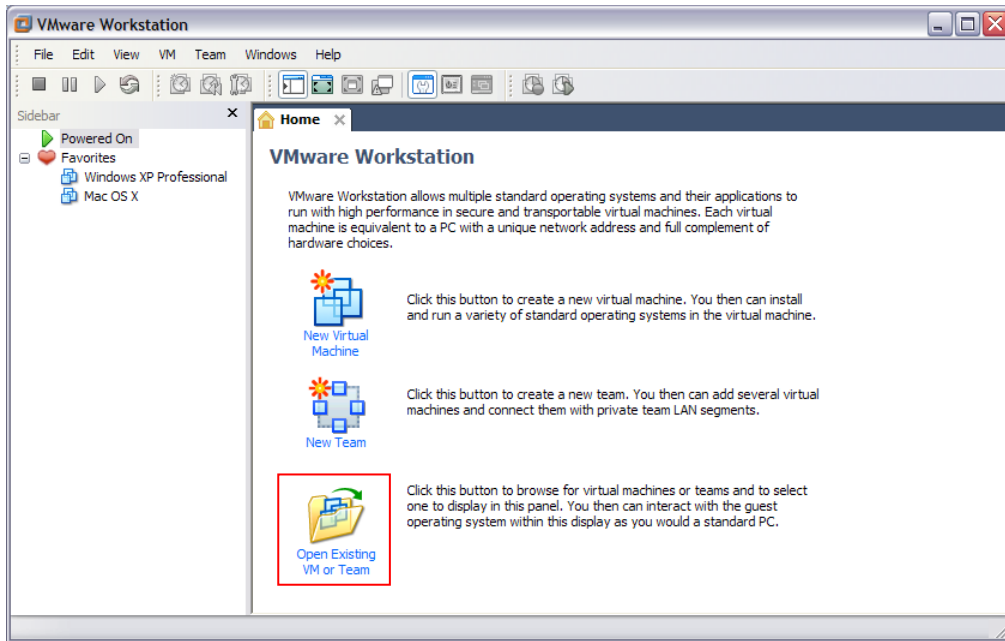
Lab Solution

1. In this lab, you will use the Ubuntu Server 9.10 VMWare appliance. Ubuntu Server 9.10 VMWare appliance is a free VMWare virtual machine running Ubuntu Server 9.10.
2. Download the ubuntu-server-9.10-i386.zip file from <http://www.thoughtpolice.co.uk/vmware/> and extract it on your computer.
3. Download and install VMWare on your computer. You can either use VMWare Workstation (trial version) or VMWare Player (free product). You can find both installer files at <http://www.vmware.com/>.
4. Start VMWare.

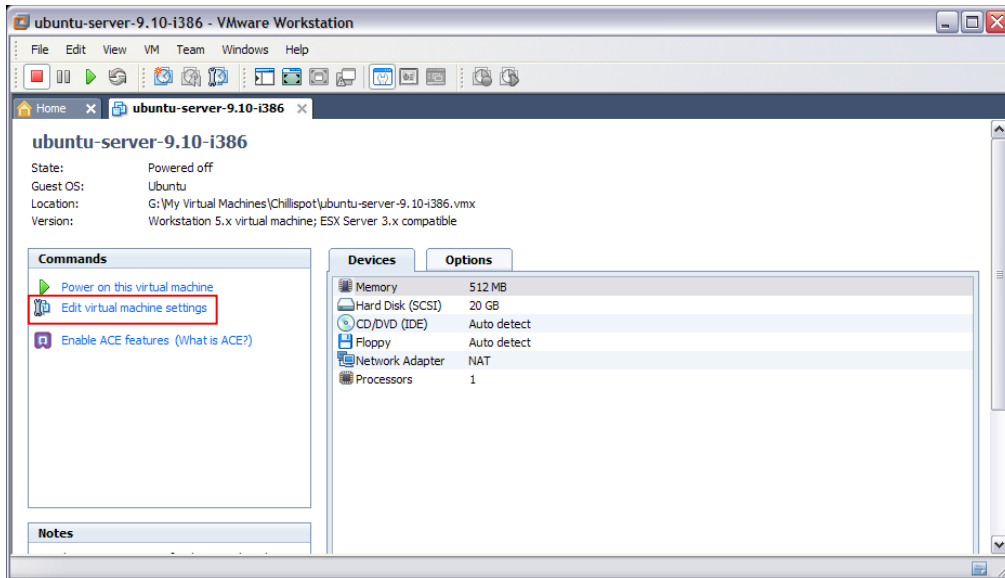


Note: We are using VMWare Workstation 6.5 for this exercise but the following steps are similar for VMWare Player.

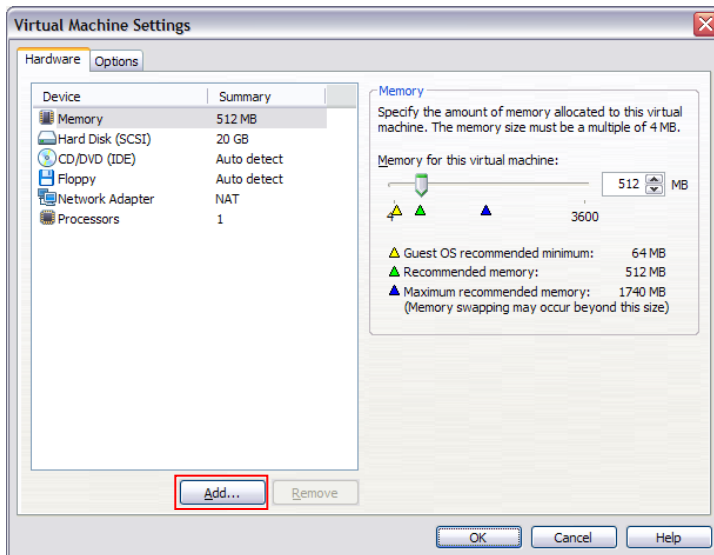
5. Click on “Open Existing VM or Team” to open an existing virtual machine and browse to the directory where you extracted the Ubuntu Server 9.10 image. Select the Ubuntu-server-9.10-i386.vmx file and click on Open.



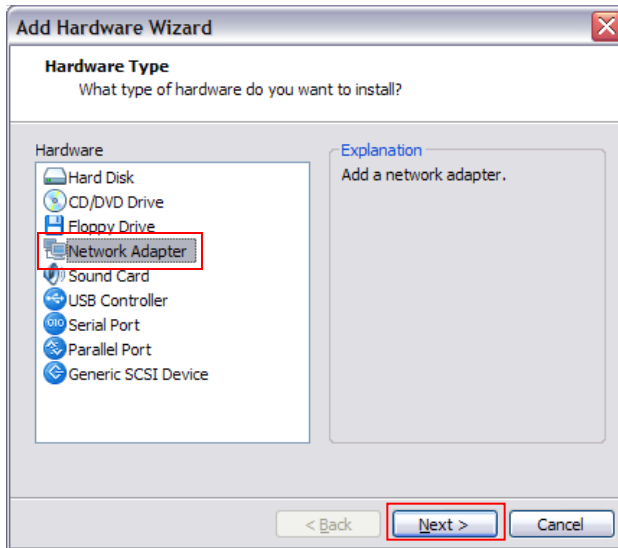
6. Your hotspot will be controlled on a single machine (represented here by the virtual machine). This machine requires at least two network interfaces, so click on Edit virtual machine settings to add a new network card.



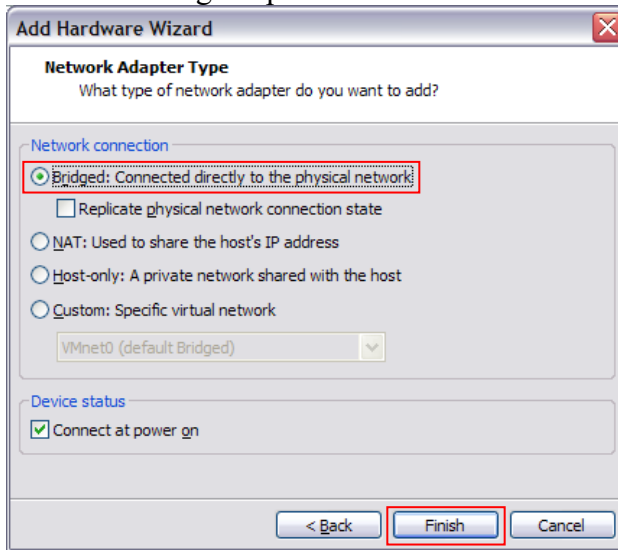
7. Click on Add...



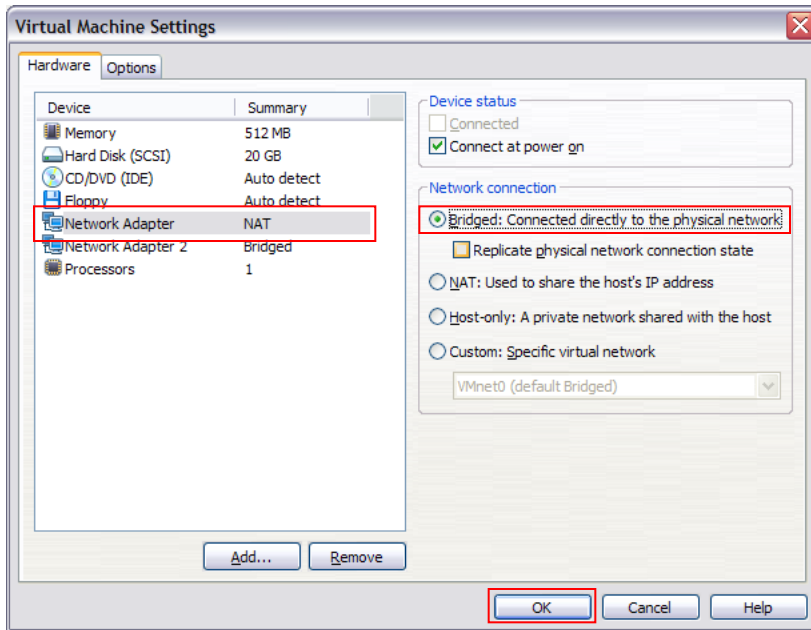
8. Select Network Adapter and click on Next >.



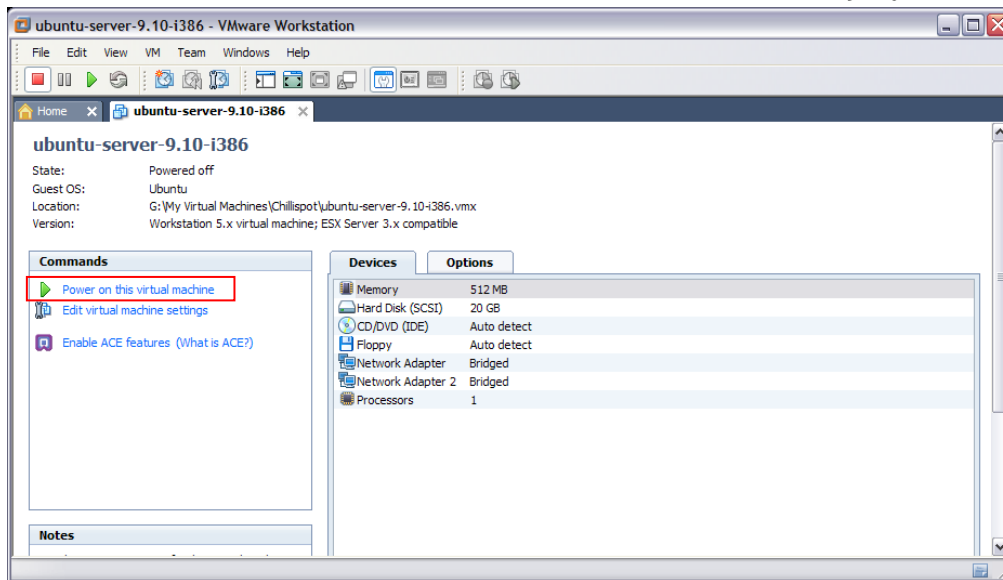
9. Select the Bridged option and click on Finish.



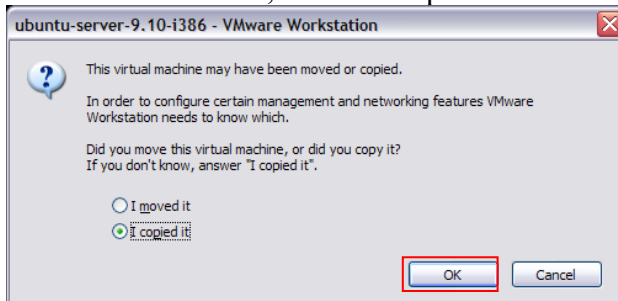
10. Additionally, change the network connection type to Bridged for the first Network Adapter and click on OK.



11. Click on Power on this virtual machine to start the Ubuntu Server 9.10 virtual machine.



12. On the next window, select "I copied it" and click OK.



13. Wait while the virtual machine starts.

```

fsck from util-linux-ng 2.16
/dev/sda5 was not cleanly unmounted, check forced.
Filesystem checks are in progress (ESC to cancel):
[ 5.116260] ACPI: I/O resource piix4_smbus [0x1040-0x1047] conflicts with ACPI
I region SMB_ [0x1040-0x104b]
/dev/sda5: 164/124496 files (2.4% non-contiguous), 29909/248976 blocks
mountall: fsck /boot [404] terminated with status 1
fsck from util-linux-ng 2.16
/dev/mapper/ubuntu-root: clean, 45351/1237888 files, 251240/4948992 blocks

```

14. Type *notroot* as the username and *thoughtpolice* as the password.

```

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/

System information as of Thu Feb 25 11:55:02 EST 2010

System load: 0.25          Memory usage: 4%    Processes:      64
Usage of /:  3.6% of 18.58GB Swap usage:   0%    Users logged in: 0

Graph this data and manage this system at https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

notroot@ubuntu:~$ _

```

15. Now, type *sudo su* and press Enter to gain root privileges.

```
notroot@ubuntu:~$ sudo su
[sudo] password for notroot:
root@ubuntu:/home/notroot# _
```

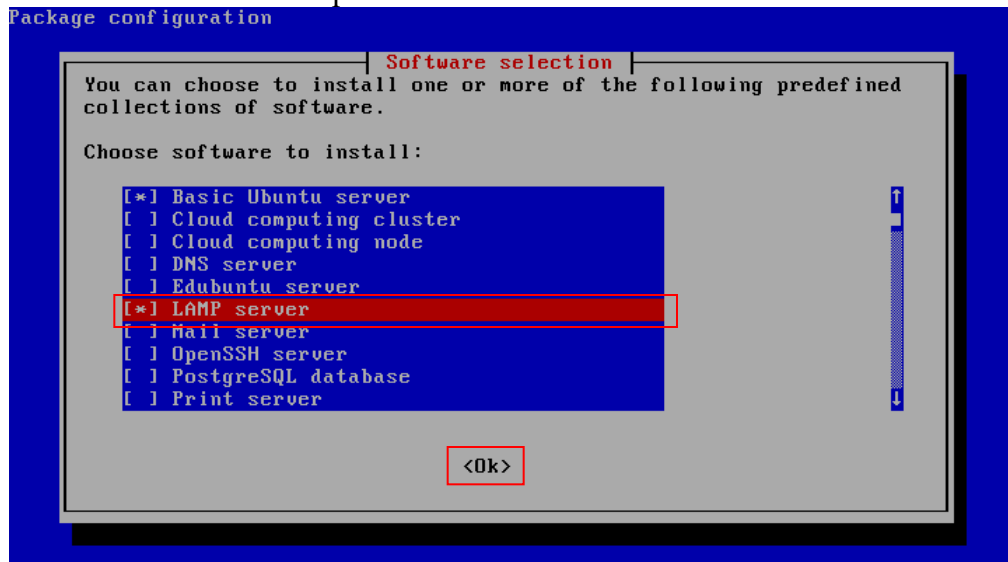
16. Type *apt-get update* and press Enter to update the available packages for Ubuntu Server 9.10.

```
root@ubuntu:/home/notroot# apt-get update
Get:1 http://security.ubuntu.com karmic-security Release.gpg [189B]
Ign http://security.ubuntu.com karmic-security/main Translation-en_US
Get:2 http://us.archive.ubuntu.com karmic Release.gpg [189B]
Ign http://us.archive.ubuntu.com karmic/main Translation-en_US
Ign http://security.ubuntu.com karmic-security/restricted Translation-en_US
Ign http://security.ubuntu.com karmic-security/universe Translation-en_US
Ign http://security.ubuntu.com karmic-security/multiverse Translation-en_US
Get:3 http://security.ubuntu.com karmic-security Release [44.1kB]
Ign http://us.archive.ubuntu.com karmic/restricted Translation-en_US
Ign http://us.archive.ubuntu.com karmic/universe Translation-en_US
Ign http://us.archive.ubuntu.com karmic/multiverse Translation-en_US
Get:4 http://us.archive.ubuntu.com karmic-updates Release.gpg [189B]
Ign http://us.archive.ubuntu.com karmic-updates/main Translation-en_US
Ign http://us.archive.ubuntu.com karmic-updates/restricted Translation-en_US
Ign http://us.archive.ubuntu.com karmic-updates/universe Translation-en_US
Ign http://us.archive.ubuntu.com karmic-updates/multiverse Translation-en_US
Get:5 http://us.archive.ubuntu.com karmic Release [65.9kB]
75% [3 Release gpgv 44056] [5 Release 38700/65.9kB 58%]_
```

17. Next, type *tasksel* and press Enter to make your installation a LAMP (Linux-Apache-Mysql-PHP) server.


```
root@ubuntu:/home/notroot# tasksel_
```

18. Select LAMP server and press OK.



19. Wait while the packages are being downloaded and installed on your computer.

Package configuration

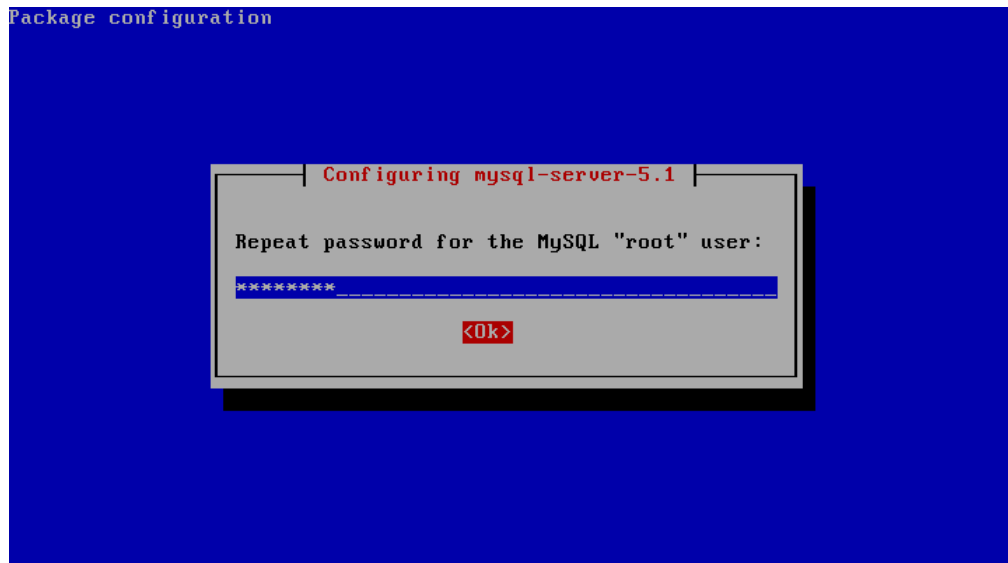
	Installing packages	
Retrieving file 4 of 26		
0%		

20. Set a new root password for MySQL and press Ok.

Package configuration

	Configuring mysql-server-5.1	
While not mandatory, it is highly recommended that you set a password for the MySQL administrative "root" user.		
If this field is left blank, the password will not be changed.		
New password for the MySQL "root" user:		

<Ok>		



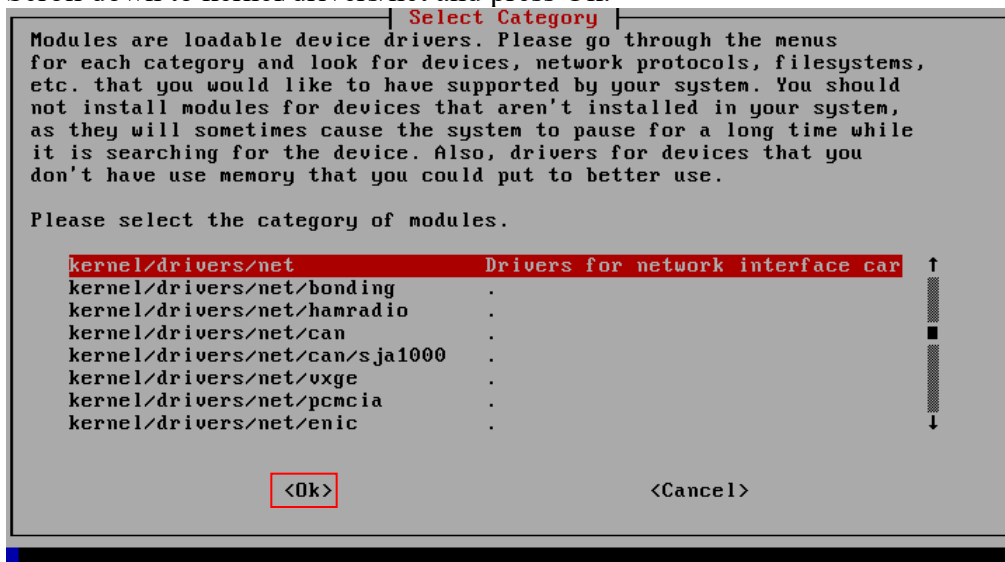
21. Next, you need to add TUN/TAP support to your kernel. Type *apt-get install modconf* and press Enter to install modconf.

```
root@ubuntu:/home/notroot# apt-get install modconf
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  modconf
0 upgraded, 1 newly installed, 0 to remove and 62 not upgraded.
Need to get 1,460kB of archives.
After this operation, 4,768kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com karmic/universe modconf 0.3.9ubuntu1 [1,460kB]
Fetched 1,460kB in 6s (226kB/s)
Selecting previously deselected package modconf.
(Reading database ... 41310 files and directories currently installed.)
Unpacking modconf (from .../modconf_0.3.9ubuntu1_all.deb) ...
Processing triggers for man-db ...
Setting up modconf (0.3.9ubuntu1) ...
root@ubuntu:/home/notroot# _
```

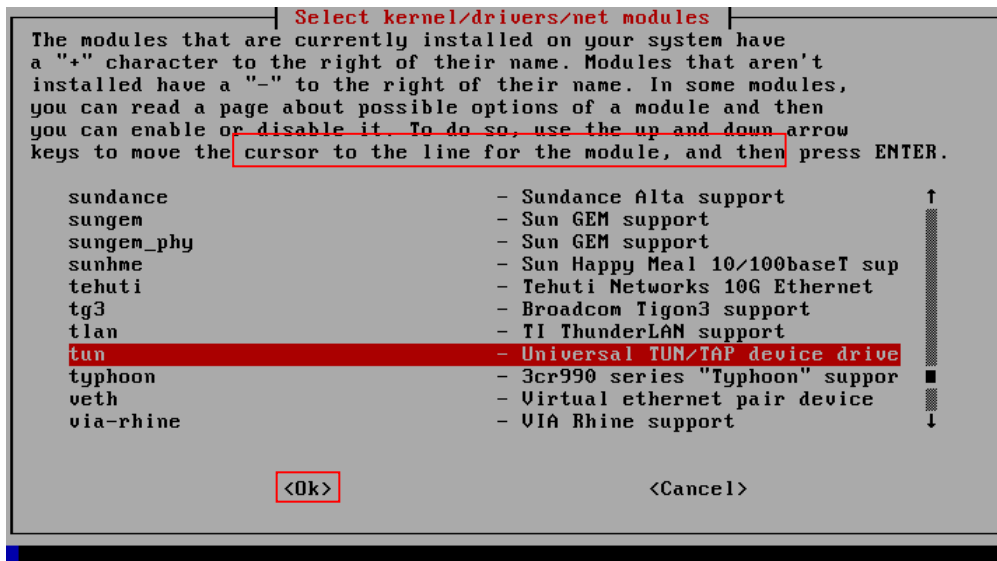
22. Type *modconf* and press Enter.

```
root@ubuntu:/home/notroot# modconf_
```

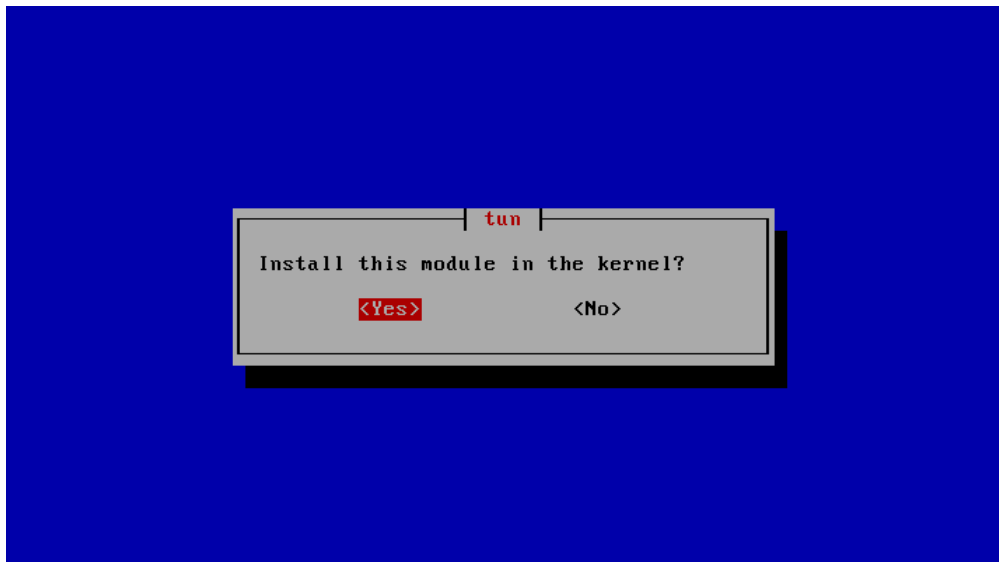
23. Scroll down to kernel/drivers/net and press Ok.



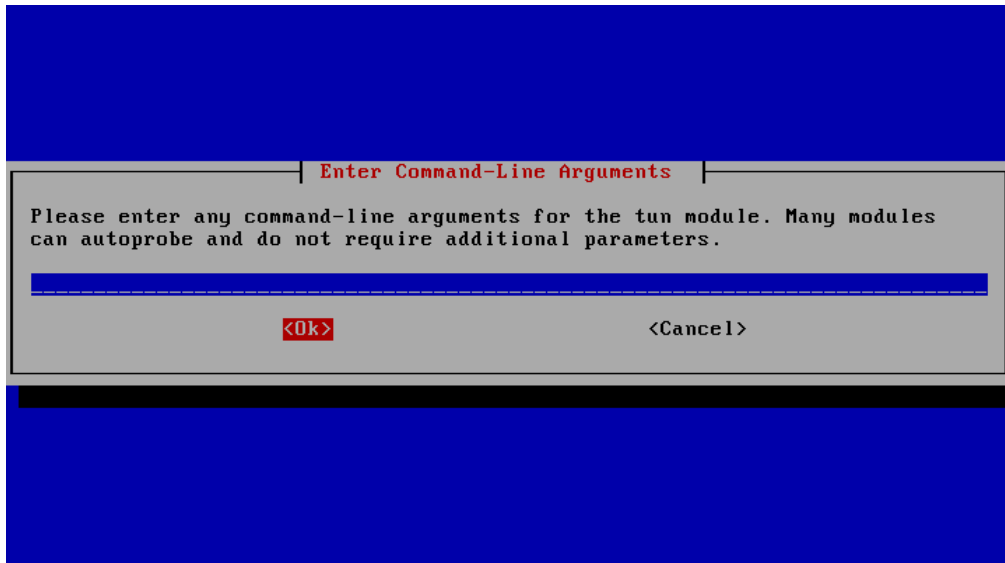
24. Select tun and press Ok.



Press Yes to install this module in the kernel.



Press Ok when prompted.



Installing module tun. If the device isn't there, or isn't configured correctly, this could cause your system to pause for up to a minute.

Installation succeeded.

Please press ENTER when you are ready to continue.

—

25. Exit modconf.



26. Next, type *modprobe tun* and press Enter.

```
root@ubuntu:/home/notroot# modprobe tun
root@ubuntu:/home/notroot# _
```

27. Now, you need to download and compile ChilliSpot. First, type *cd /usr/src/* and press Enter to change your location.

```
root@ubuntu:/home/notroot# cd /usr/src/  
root@ubuntu:/usr/src# _
```

28. Next, type `wget http://www.chillispot.info/download/chillispot-1.1.0.tar.gz` and press Enter to download ChilliSpot's source code.

```
root@ubuntu:/usr/src# wget http://www.chillispot.info/download/chillispot-1.1.0.  
tar.gz  
--2010-02-25 12:43:17-- http://www.chillispot.info/download/chillispot-1.1.0.ta  
r.gz  
Resolving www.chillispot.info... 195.228.254.184  
Connecting to www.chillispot.info:195.228.254.184:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 452749 (442K) [application/x-gzip]  
Saving to: 'chillispot-1.1.0.tar.gz'  
  
100%[=====>] 452,749      226K/s   in 2.0s  
2010-02-25 12:43:20 (226 KB/s) - 'chillispot-1.1.0.tar.gz' saved [452749/452749]  
root@ubuntu:/usr/src# _
```

29. Type `tar xzvf chillispot-1.1.0.tar.gz` and press Enter to extract the source code.


```
root@ubuntu:/usr/src# tar xzvf chillispot-1.1.0.tar.gz _
```

30. Type `cd chillispot-1.1.0` and press Enter to change your location.

```
root@ubuntu:/usr/src# cd chillispot-1.1.0
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

31. Next, type `apt-get build-essential` and press Enter to install gcc and other required packages.

```

root@ubuntu:/usr/src/chillispot-1.1.0# apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  binutils cpp-4.4 dpkg-dev fakeroot g++ g++-4.4 gcc gcc-4.4 gcc-4.4-base
  libc-bin libc-dev-bin libc6 libc6-dev libc6-i686 libgcc1 libgomp1 libstdc++6
  libstdc++6-4.4-dev linux-libc-dev
Suggested packages:
  binutils-doc gcc-4.4-locales debian-keyring debian-maintainers g++-multilib
  g++-4.4-multilib gcc-4.4-doc libstdc++6-4.4-dbg gcc-multilib manpages-dev
  autoconf automake1.9 libtool flex bison gdb gcc-doc gcc-4.4-multilib
  libmudflap0-4.4-dev libgcc1-dbg libgomp1-dbg libmudflap0-dbg libcloog-ppl0
  libppl-c2 libppl7 glibc-doc libstdc++6-4.4-doc
The following NEW packages will be installed:
  binutils build-essential dpkg-dev fakeroot g++ g++-4.4 gcc gcc-4.4
  libc-dev-bin libc6-dev libgomp1 libstdc++6-4.4-dev linux-libc-dev
The following packages will be upgraded:
  cpp-4.4 gcc-4.4-base libc-bin libc6 libc6-i686 libgcc1 libstdc++6
7 upgraded, 13 newly installed, 0 to remove and 55 not upgraded.
Need to get 26.9MB of archives.
After this operation, 61.6MB of additional disk space will be used.
Do you want to continue [Y/n]? _

```

Press Enter when prompted.

```

Setting up binutils (2.20-0ubuntu2) ...
Setting up libc-dev-bin (2.10.1-0ubuntu16) ...
Setting up linux-libc-dev (2.6.31-19.56) ...
Setting up libc6-dev (2.10.1-0ubuntu16) ...
Setting up libgomp1 (4.4.1-4ubuntu9) ...
Setting up gcc-4.4 (4.4.1-4ubuntu9) ...
Setting up gcc (4:4.4.1-1ubuntu2) ...
Setting up dpkg-dev (1.15.4ubuntu2) ...
Setting up fakeroot (1.12.4ubuntu1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroot (
fakeroot) in auto mode.
Setting up libstdc++6-4.4-dev (4.4.1-4ubuntu9) ...
Setting up g++-4.4 (4.4.1-4ubuntu9) ...
Setting up g++ (4:4.4.1-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mo
de.
Setting up build-essential (11.4) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@ubuntu:/usr/src/chillispot-1.1.0# _

```

32. Before compiling ChilliSpot, type `nano -w src/tun.c` and look for all instances of the string “(`__FreeBSD__`) defined (” and add “`||`” between “)” and “defined”. The resulting string should look like “(`__FreeBSD__`) `||` defined (”. Save the file with `Ctrl+O`.

```

GNU nano 2.0.9      File: src/tun.c

req.n.nlmseq = 0;
req.n.nlmflags |= NLM_F_ACK;

status = sendmsg(fd, &msg, 0); /* TODO Error check */

tun_sifflags(this, IFF_UP | IFF_RUNNING); /* TODO */
close(fd);
this->addr++;
return 0;

#elif defined (__FreeBSD__) defined (__OpenBSD__) || defined (__NetBSD__) || de$
int fd;
struct ifaliasreq areq;

/* TODO: Is this needed on FreeBSD? */
if (!this->addr) /* Use ioctl for first addr to make ping work */
    return tun_setaddr(this, addr, dstaddr, netmask); /* TODO dstaddr */

memset(&areq, 0, sizeof(areq));

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell

```

```

GNU nano 2.0.9      File: src/tun.c      Modified

tun_sifflags(this, IFF_UP | IFF_RUNNING); /* TODO */
close(fd);
this->addr++;
return 0;

#elif defined (__FreeBSD__) || defined (__OpenBSD__) || defined (__NetBSD__) ||$
int fd;
struct ifaliasreq areq;

/* TODO: Is this needed on FreeBSD? */
if (!this->addr) /* Use ioctl for first addr to make ping work */
    return tun_setaddr(this, addr, dstaddr, netmask); /* TODO dstaddr */

memset(&areq, 0, sizeof(areq));

/* Set up interface name */
strncpy(areq.ifra_name, this->devname, IFNAMSIZ);
areq.ifra_name[IFNAMSIZ-1] = 0; /* Make sure to terminate */

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell

```

33. Now, type `./configure` and press Enter to start compiling ChilliSpot.

```
root@ubuntu:/usr/src/chillispot-1.1.0# ./configure _
```

34. Next, type *make* and press Enter.

```
sr/local/sbin" -g -O2 -MT iphash.o -MD -MP -MF ".deps/iphash.Tpo" -c -o iphash
.o iphash.c; \
    then mv -f ".deps/iphash.Tpo" ".deps/iphash.Po"; else rm -f ".deps/iphas
h.Tpo"; exit 1; fi
if gcc -DHAVE_CONFIG_H -I. -I. -I.. -D_GNU_SOURCE -fno-builtin -DSBINDIR="/u
sr/local/sbin" -g -O2 -MT lookup.o -MD -MP -MF ".deps/lookup.Tpo" -c -o looku
p.o lookup.c; \
    then mv -f ".deps/lookup.Tpo" ".deps/lookup.Po"; else rm -f ".deps/looku
p.Tpo"; exit 1; fi
/bin/bash ../libtool --tag=CC --mode=link gcc -D_GNU_SOURCE -fno-builtin -DSBIND
IR="/usr/local/sbin" -g -O2 -o chilli chilli.o tun.o cmdline.o ippool.o ra
dius.o md5.o redir.o dhcp.o syserr.o iphash.o lookup.o
mkdir .libs
gcc -D_GNU_SOURCE -fno-builtin -DSBINDIR="/usr/local/sbin" -g -O2 -o chilli ch
illi.o tun.o cmdline.o ippool.o radius.o md5.o redir.o dhcp.o syserr.o iphash.o
lookup.o
make[2]: Leaving directory `/usr/src/chillispot-1.1.0/src'
Making all in doc
make[2]: Entering directory `/usr/src/chillispot-1.1.0/doc'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/usr/src/chillispot-1.1.0/doc'
make[2]: Entering directory `/usr/src/chillispot-1.1.0'
make[2]: Leaving directory `/usr/src/chillispot-1.1.0'
make[1]: Leaving directory `/usr/src/chillispot-1.1.0'
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

35. Finally, type *make install* and press Enter.

```
root@ubuntu:/usr/src/chillispot-1.1.0# make install_
```

```
Making install in src
make[1]: Entering directory '/usr/src/chillispot-1.1.0/src'
make[2]: Entering directory '/usr/src/chillispot-1.1.0/src'
test -z "/usr/local/sbin" || mkdir -p -- "/usr/local/sbin"
/bin/bash ../libtool --mode=install /usr/bin/install -c 'chilli' '/usr/local/s
bin/chilli'
/usr/bin/install -c chilli /usr/local/sbin/chilli
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/chillispot-1.1.0/src'
make[1]: Leaving directory '/usr/src/chillispot-1.1.0/src'
Making install in doc
make[1]: Entering directory '/usr/src/chillispot-1.1.0/doc'
make[2]: Entering directory '/usr/src/chillispot-1.1.0/doc'
make[2]: Nothing to be done for 'install-exec-am'.
test -z "/usr/local/man/man8" || mkdir -p -- "/usr/local/man/man8"
/usr/bin/install -c -m 644 './chilli.8' '/usr/local/man/man8/chilli.8'
make[2]: Leaving directory '/usr/src/chillispot-1.1.0/doc'
make[1]: Leaving directory '/usr/src/chillispot-1.1.0/doc'
make[1]: Entering directory '/usr/src/chillispot-1.1.0'
make[2]: Entering directory '/usr/src/chillispot-1.1.0'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/chillispot-1.1.0'
make[1]: Leaving directory '/usr/src/chillispot-1.1.0'
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

36. Type `nano -w /etc/chilli.conf` and press Enter to edit ChilliSpot's configuration file and add the following content.

```
radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
radiussecret radiusPassword
dns1 8.8.8.8
dhcpif eth1
uamallowed 192.168.182.1,8.8.8.8
uamserver https://192.168.182.1/cgi-bin/hotspotlogin.cgi
uamhomepage http://192.168.182.1
uamsecret uamPassword
```

```
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/chilli.conf_
```

```
GNU nano 2.0.9      File: /etc/chilli.conf      Modified

radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
radiussecret radiusPassword
dns1 8.8.8.8
dhcpif eth1
uamallowed 192.168.182.1, 8.8.8.8
uamserver https://192.168.182.1/cgi-bin/hotspotlogin.cgi
uamhomepage http://192.168.182.1
uamsecret uamPassword_

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

Note: You can also use the /usr/src/chillispot-1.1.0/doc/chilli.conf file as a template and adjust the lines specified above.

37. Next, you will install and configure FreeRadius to handle the authentication process. FreeRadius will keep information about authorized users on a MySQL database. Type *apt-get install freeradius freeradius-mysql* and press Enter to install FreeRadius and configure it to work with MySQL.

```
root@ubuntu:/usr/src/chillispot-1.1.0# apt-get install freeradius freeradius-mys  
ql_
```

Press Enter when prompted and wait while the packages are being downloaded and installed on your system.

```
root@ubuntu:/usr/src/chillispot-1.1.0# apt-get install freeradius freeradius-mys  
ql  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  freeradius-common freeradius-utils libfreeradius2 libltdl7 libperl5.10  
Suggested packages:  
  freeradius-ldap freeradius-krb5 freeradius-postgresql  
The following NEW packages will be installed:  
  freeradius freeradius-common freeradius-mysql freeradius-utils  
  libfreeradius2 libltdl7 libperl5.10  
0 upgraded, 7 newly installed, 0 to remove and 55 not upgraded.  
Need to get 2,498kB of archives.  
After this operation, 6,455kB of additional disk space will be used.  
Do you want to continue [Y/n]? _
```

```

Unpacking freeradius-utils (from .../freeradius-utils_2.1.0+dfsg-0ubuntu7_i386.d
eb) ...
Processing triggers for man-db ...
Setting up libfreeradius2 (2.1.0+dfsg-0ubuntu7) ...
Setting up libltdl7 (2.2.6a-4) ...

Setting up libperl5.10 (5.10.0-24ubuntu4) ...

Setting up freeradius-common (2.1.0+dfsg-0ubuntu7) ...
Adding user freerad to group shadow

Setting up freeradius (2.1.0+dfsg-0ubuntu7) ...
stripping trailing /
stripping trailing /
stripping trailing /
* Starting FreeRADIUS daemon freeradius [ OK ]

Setting up freeradius-mysql (2.1.0+dfsg-0ubuntu7) ...
* Stopping FreeRADIUS daemon freeradius [ OK ]
* Starting FreeRADIUS daemon freeradius [ OK ]

Setting up freeradius-utils (2.1.0+dfsg-0ubuntu7) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@ubuntu:/usr/src/chillispot-1.1.0# _

```

38. Type `nano -w /etc/freeradius/clients.conf` and press Enter to edit this file.

```

root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/freeradius/clients.conf _

```

39. Look for “secret = testing123” under the “client localhost” section and change it to “secret = radiusPassword”. *Note: This secret value should match the radiussecret value of the chilli.conf file set up before.*


```
GNU nano 2.0.9      File: /etc/freeradius/clients.conf

#       lower case letters
#       numbers
#
# And is at LEAST 8 characters long, preferably 16 characters in
# length. The secret MUST be random, and should not be words,
# phrase, or anything else that is recognizable.
#
# The default secret below is only for testing, and should
# not be used in any real environment.
#
secret           = testing123
#
# Old-style clients do not send a Message-Authenticator
# in an Access-Request. RFC 5080 suggests that all clients
# SHOULD include it in an Access-Request. The configuration
# item below allows the server to require it. If a client
# is required to include a Message-Authenticator and it does
# not, then the packet will be silently discarded.
#

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

```
GNU nano 2.0.9      File: /etc/freeradius/clients.conf      Modified

#       lower case letters
#       numbers
#
# And is at LEAST 8 characters long, preferably 16 characters in
# length. The secret MUST be random, and should not be words,
# phrase, or anything else that is recognizable.
#
# The default secret below is only for testing, and should
# not be used in any real environment.
#
secret           = radiusPassword_
#
# Old-style clients do not send a Message-Authenticator
# in an Access-Request. RFC 5080 suggests that all clients
# SHOULD include it in an Access-Request. The configuration
# item below allows the server to require it. If a client
# is required to include a Message-Authenticator and it does
# not, then the packet will be silently discarded.
#

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

Save the file with Ctrl+O and exit nano with Ctrl+X.

40. Type `nano -w /etc/freeradius/sql.conf` and press Enter. Then, edit the login and password parameters using root as the login value and the MySQL root password you specified before as the password value. Save the file and exit nano.

```
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/freeradius/sql.conf_
```

Original values:

```
GNU nano 2.0.9      File: /etc/freeradius/sql.conf

# Which FreeRADIUS driver to use.
#
driver = "rlm_sql_${database}"

# Connection info:
server = "localhost"
login = "radius"
password = "radpass"

# Database table configuration for everything except Oracle
radius_db = "radius"
# If you are using Oracle then use this instead
# radius_db = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR$

# If you want both stop and start records logged to the
# same SQL table, leave this as is.  If you want them in
# different tables, put the start table in acct_table1
# and stop table in acct_table2
acct_table1 = "radacct"
acct_table2 = "radacct"

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

New values:

```
GNU nano 2.0.9      File: /etc/freeradius/sql.conf      Modified

# Which FreeRADIUS driver to use.
#
driver = "rlm_sql_${database}"

# Connection info:
server = "localhost"
login = "root"
password = "rootpass"

# Database table configuration for everything except Oracle
radius_db = "radius"
# If you are using Oracle then use this instead
# radius_db = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521)))"

# If you want both stop and start records logged to the
# same SQL table, leave this as is. If you want them in
# different tables, put the start table in acct_table1
# and stop table in acct_table2
acct_table1 = "radacct"
acct_table2 = "radacct"

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

Note: Take note of the value of the radius_db parameter. Later, you will create a database with the same name on MySql.

41. Type `nano -w /etc/freeradius/radiusd.conf` and press Enter. Then, uncomment the line `$INCLUDE sql.conf` by removing the leading `#`. Save the file and exit nano.

```
GNU nano 2.0.9      File: /etc/freeradius/radiusd.conf

# Extensible Authentication Protocol
#
# For all EAP related authentications.
# Now in another file, because it is very large.
#
$INCLUDE eap.conf

# Include another file that has the SQL-related configuration.
# This is another file only because it tends to be big.
#
#$INCLUDE sql.conf

#
# This module is an SQL enabled version of the counter module.
#
# Rather than maintaining separate (GDBM) databases of
# accounting info for each counter, this module uses the data
# stored in the raddacct table by the sql modules. This
# module NEVER does any database INSERTs or UPDATEs. It is
# totally dependent on the SQL module to process Accounting

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

```
GNU nano 2.0.9      File: /etc/freeradius/radiusd.conf      Modified

# Extensible Authentication Protocol
#
# For all EAP related authentications.
# Now in another file, because it is very large.
#
$INCLUDE eap.conf

# Include another file that has the SQL-related configuration.
# This is another file only because it tends to be big.
#
$INCLUDE sql.conf

#
# This module is an SQL enabled version of the counter module.
#
# Rather than maintaining separate (GDBM) databases of
# accounting info for each counter, this module uses the data
# stored in the raddacct table by the sql modules. This
# module NEVER does any database INSERTs or UPDATES. It is
# totally dependent on the SQL module to process Accounting

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

42. Type `nano -w /etc/freeradius/sites-available/default` and press Enter. Then, uncomment the sql option for the authorize, accounting, session, and post-auth sections. Save the file and exit nano.

```
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/freeradius/sites-available/default _
```

Enable the sql option for the authorize section:

```
GNU nano 2.0.9 File: /etc/freeradius/sites-available/default Modified

unix

#
# Read the 'users' file
files

#
# Look in an SQL database. The schema of the database
# is meant to mirror the "users" file.
#
# See "Authorization Queries" in sql.conf
sql

#
# If you are using /etc/smbpasswd, and are also doing
# mschap authentication, the un-comment this line, and
# configure the 'etc_smbpasswd' module, above.
#
etc_smbpasswd

#

^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Enable the sql option for the session sections:

```
GNU nano 2.0.9 File: /etc/freeradius/sites-available/default Modified

#
}

# Session database, used for checking Simultaneous-Use. Either the radutmp
# or rlm_sql module can handle this.
# The rlm_sql module is *much* faster
session {
    radutmp

    #
    # See "Simultaneous Use Checking Queries" in sql.conf
    sql
}

# Post-Authentication
# Once we KNOW that the user has been authenticated, there are
# additional steps we can take.
post-auth {

^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

43. Next, you need to configure MySql to work with FreeRadius. Type `mysql -u root -p` and press Enter to launch MySql. Use your MySql root password to log in.

```
root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

44. Type *CREATE DATABASE radius;* and press Enter to create the database.

```
root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE radius;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

45. Type *quit* and press Enter to exit MySql.

```
root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE radius;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

46. Type `mysql -u root -p radius < /etc/freeradius/sql/mysql/schema.sql` to import the MySQL schema from `/etc/freeradius/sql/mysql/schema.sql`. Use your MySQL root password.

```
root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p radius < /etc/freeradius
/sql/mysql/schema.sql
Enter password:
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

47. Next, type `mysql -u root -p radius` and press Enter to launch MySQL again and work with the radius database. Use your MySQL root password to log in.

```

root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p radius
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _

```

48. Type `INSERT INTO radcheck(id, UserName, Attribute, op, Value) VALUES (NULL, 'test', 'User-Password', ':=', 'test');` and press Enter to add a testing user to the radcheck table.

Then, type `exit` to exit MySQL.

```

root@ubuntu:/home/notroot# mysql -u root -p radius
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> INSERT INTO radcheck(id, UserName, Attribute, op, Value)
-> VALUES (NULL, 'test', 'User-Password', ':=', 'test');
Query OK, 1 row affected (0.00 sec)

mysql> _

```

Note: Later, you can repeat this step to add authentication information for real users. It will also be a good idea to delete the information for the test user.

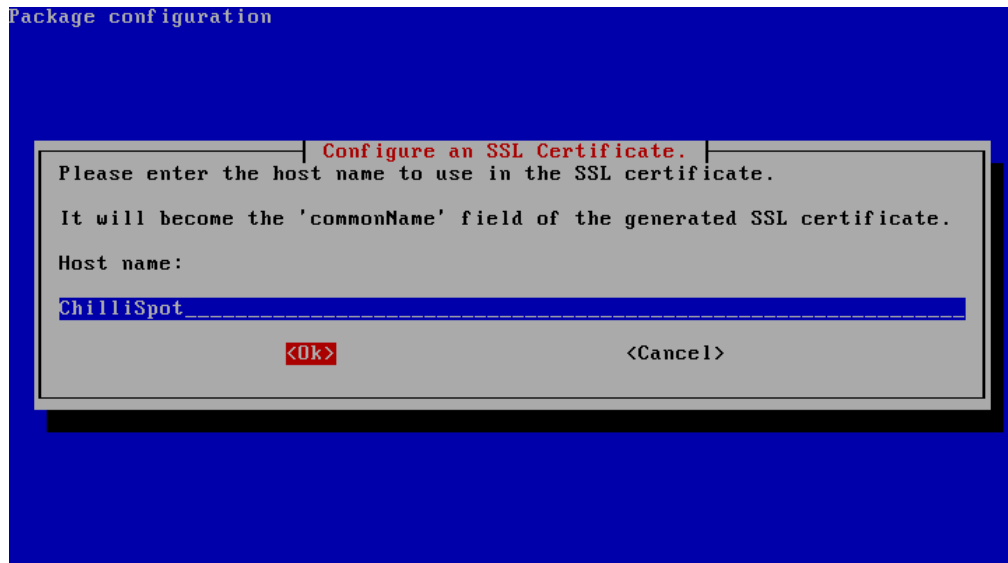
49. Type `mysql -u root -p radius < /etc/freeradius/sql/mysql/nas.sql` and press Enter.


```
root@ubuntu:/usr/src/chillispot-1.1.0# mysql -u root -p radius < /etc/freeradius
/sql/mysql/nas.sql
Enter password:
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

50. Next, you need to configure Apache and SSL. First, type *make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/private/ChilliSpot.crt* and press Enter to create the self-signed certificate for your site.

```
root@ubuntu:/usr/src/chillispot-1.1.0# make-ssl-cert /usr/share/ssl-cert/ssleay.
cnf /etc/ssl/private/ChilliSpot.crt_
```

Enter ChilliSpot as the host name and press Ok.



51. You need to split the certificate into two files. Type `cp /etc/ssl/private/ChilliSpot.crt /etc/ssl/private/ChilliSpot.pem` and press Enter. Then, type `cp /etc/ssl/private/ChilliSpot.crt /etc/ssl/private/ChilliSpot.key` and press Enter.

```
root@ubuntu:/usr/src/chillispot-1.1.0# cp /etc/ssl/private/ChilliSpot.crt /etc/s
sl/private/ChilliSpot.pem
root@ubuntu:/usr/src/chillispot-1.1.0# cp /etc/ssl/private/ChilliSpot.crt /etc/s
sl/private/ChilliSpot.key
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

52. Open the .key file with nano and leave only the part beginning with -----BEGIN RSA PRIVATE KEY----- and ending with -----END RSA PRIVATE KEY-----.


```
root@ubuntu:/usr/src/chillispot-1.1.0# chmod 600 /etc/ssl/private/ChilliSpot.key
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

55. Next, type *a2enmod ssl* and press Enter to enable Apache's SSL module.

```
root@ubuntu:/usr/src/chillispot-1.1.0# a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
create self-signed certificates.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

56. Create the directory for your Web site by typing *mkdir /var/www/ChilliSpot*.

```
root@ubuntu:/usr/src/chillispot-1.1.0# mkdir /var/www/ChilliSpot
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

57. Next, type `cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-available/ChilliSpot` and press Enter. Then, type `nano -w /etc/apache2/sites-available/ChilliSpot` and press Enter to edit the configuration file for the SSL site.

```
root@ubuntu:/usr/src/chillispot-1.1.0# cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-available/ChilliSpot
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/apache2/sites-available/ChilliSpot _
```

58. Adjust the configuration file so it looks like this:

```
<IfModule mod_ssl.c>
<VirtualHost 192.168.182.1:443>
    ServerAdmin webmaster@hostmauritius.com
    ServerName 192.168.182.1:443
    DocumentRoot /var/www/ChilliSpot
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/ChilliSpot/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
```

```

        Order allow,deny
        Allow from all
</Directory>
ErrorLog /var/log/apache2/error.log
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn
CustomLog /var/log/apache2/ssl_access.log combined
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
#    SSL Engine Switch:
#    Enable/Disable SSL for this virtual host.
SSLEngine on
#    A self-signed (snakeoil) certificate can be created by installing
#    the ssl-cert package. See
#    /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
#    If both key and certificate are stored in the same file, only the
#    SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/private/ChilliSpot.pem
SSLCertificateKeyFile /etc/ssl/private/ChilliSpot.key
#    Server Certificate Chain:
#    Point SSLCertificateChainFile at a file containing the
#    concatenation of PEM encoded CA certificates which form the
#    certificate chain for the server certificate. Alternatively
#    the referenced file can be the same as SSLCertificateFile
#    when the CA certificates are directly appended to the server
#    certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt
#    Certificate Authority (CA):
#    Set the CA certificate verification path where to find CA
#    certificates for client authentication or alternatively one
#    huge file containing all of them (file must be PEM encoded)
#    Note: Inside SSLCACertificatePath you need hash symlinks
#          to point to the certificate files. Use the provided
#          Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt
#    Certificate Revocation Lists (CRL):
#    Set the CA revocation path where to find CA CRLs for client
#    authentication or alternatively one huge file containing all
#    of them (file must be PEM encoded)
#    Note: Inside SSLCARevocationPath you need hash symlinks
#          to point to the certificate files. Use the provided
#          Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl
#    Client Authentication (Type):
#    Client certificate verification type and depth. Types are
#    none, optional, require and optional_no_ca. Depth is a
#    number which specifies how deeply to verify the certificate
#    issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10
#    Access Control:
#    With SSLRequire you can do per-directory access control based
#    on arbitrary complex boolean expressions containing server
#    variable checks and other lookup directives. The syntax is a
#    mixture between C and Perl. See the mod_ssl documentation
#    for more details.
#<Location />
#SSLRequire (
#    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#    and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#    and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#    and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#    and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20
#    ) \

```

```

#         or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>
#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#       Translate the client X.509 into a Basic Authorisation. This means that
#       the standard Auth/DBMAuth methods can be used for access control. The
#       user name is the 'one line' version of the client's X.509 certificate.
#       Note that no password is obtained from the user. Every entry in the user
#       file needs this password: `xxj3lZMTZzkVA'.
#   o ExportCertData:
#       This exports two additional environment variables: SSL_CLIENT_CERT and
#       SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#       server (always existing) and the client (only existing when client
#       authentication is used). This can be used to import the certificates
#       into CGI scripts.
#   o StdEnvVars:
#       This exports the standard SSL/TLS related `SSL_*' environment variables.
#       Per default this exportation is switched off for performance reasons,
#       because the extraction step is an expensive operation and is usually
#       useless for serving static content. So one usually enables the
#       exportation for CGI and SSI requests only.
#   o StrictRequire:
#       This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#       under a "Satisfy any" situation, i.e. when it applies access is denied
#       and no other module can change it.
#   o OptRenegotiate:
#       This enables optimized SSL connection renegotiation handling when SSL
#       directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>
#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#       This forces an unclean shutdown when the connection is closed, i.e. no
#       SSL close notify alert is send or allowed to received. This violates
#       the SSL/TLS standard but is needed for some brain-dead browsers. Use
#       this when you receive I/O errors because of the standard approach where
#       mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#       This forces an accurate shutdown when the connection is closed, i.e. a
#       SSL close notify alert is send and mod_ssl waits for the close notify
#       alert of the client. This is 100% SSL/TLS standard compliant, but in
#       practice often causes hanging connections with brain-dead browsers. Use
#       this only for browsers where you know that their SSL implementation
#       works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
BrowserMatch ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
</VirtualHost>

```

```
GNU nano 2.0.9 File: /etc/apache2/sites-available/ChilliSpot Modified
<IfModule mod_ssl.c>
<VirtualHost 192.168.182.1:443>
    ServerAdmin webmaster@localhost
    ServerName 192.168.182.1:443_
    DocumentRoot /var/www/ChilliSpot
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/ChilliSpot/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    </Directory>

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

59. Now, type *a2ensite ChilliSpot* and press Enter to enable your new SSL site.

```
root@ubuntu:/usr/src/chillispot-1.1.0# a2ensite ChilliSpot
Enabling site ChilliSpot.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

60. Next, type */etc/init.d/apache2 reload* and press Enter to apply your new configuration to Apache.


```
root@ubuntu:/usr/src/chillispot-1.1.0# /etc/init.d/apache2 reload
* Reloading web server config apache2 [ OK ]
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

61. Type `/etc/init.d/apache2 restart` and press Enter to restart Apache with the new configuration.

```
root@ubuntu:/usr/src/chillispot-1.1.0# /etc/init.d/apache2 restart
* Restarting web server apache2 [ OK ]
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

62. Now, type `nano -w /var/www/index.html` and press Enter. Then, replace the file's content with the following line: `<a href="http://192.168.182.1:3990/prelogin">Click here to login</a>`. Save the file and exit nano.

```
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /var/www/index.html
```

```
GNU nano 2.0.9      File: /var/www/index.html
<a href="http://192.168.182.1:3990/prelogin">Click here to login</a>_

[ Wrote 1 line ]

^G Get Help      ^O WriteOut     ^R Read File    ^V Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is    ^U Next Page    ^U UnCut Text  ^T To Spell
```

63. Type `cp doc/hotspotlogin.cgi /usr/lib/cgi-bin/` and press Enter to copy the CGI with the login procedure.

```
root@ubuntu:/usr/src/chillispot-1.1.0# cp doc/hotspotlogin.cgi /usr/lib/cgi-bin/
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

64. Open the CGI file you just copied with nano and look for the parameter uamsecret. Set its value with the same value as the uamsecret parameter on the chilli.conf file. Save the file and exit nano.

```
GNU nano 2.0.9      File: /usr/lib/cgi-bin/hotspotlogin.cgi      Modified

# Redirects from ChilliSpot daemon:
#
# Redirection when not yet or already authenticated
#   notyet:  ChilliSpot daemon redirects to login page.
#   already: ChilliSpot daemon redirects to success status page.
#
# Response to login:
#   already: Attempt to login when already logged in.
#   failed:  Login failed
#   success: Login succeeded
#
# logoff:   Response to a logout

# Shared secret used to encrypt challenge with. Prevents dictionary attacks.
# You should change this to your own shared secret.
# $uamsecret = "ht2eb8ej6s4et3rg1ulp";

# Uncomment the following line if you want to use ordinary user-password
# for radius authentication. Must be used together with $uamsecret.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

```

GNU nano 2.0.9      File: /usr/lib/cgi-bin/hotspotlogin.cgi      Modified
# Redirects from ChilliSpot daemon:
#
# Redirection when not yet or already authenticated
#   notyet:  ChilliSpot daemon redirects to login page.
#   already: ChilliSpot daemon redirects to success status page.
#
# Response to login:
#   already: Attempt to login when already logged in.
#   failed:  Login failed
#   success: Login succeeded
#
# logoff:    Response to a logout

# Shared secret used to encrypt challenge with. Prevents dictionary attacks.
# You should change this to your own shared secret.
$uamsecret = "uamPassword";

# Uncomment the following line if you want to use ordinary user-password
# for radius authentication. Must be used together with $uamsecret.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell

```

65. Finally, you need to configure Internet Connection Sharing between your two network cards.

Type `nano -w /etc/network/interfaces` and press Enter. Configure your eth0 interface with static information so it is on the same network as your working access point.

```
root@ubuntu:/usr/src/chillispot-1.1.0# nano -w /etc/network/interfaces _
```

In our example, the AP is on 192.168.1.1.

```
GNU nano 2.0.9      File: /etc/network/interfaces      Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.21
netmask 255.255.255.0
gateway 192.168.1.1_

^G Get Help  ^O WriteOut  ^R Read File  ^V Prev Page  ^X Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

66. Now, create a backup copy of your resolv.conf file. Type `cp /etc/resolv.conf /etc/resolv.conf.bak` and press Enter.

```
root@ubuntu:/usr/src/chillispot-1.1.0# cp /etc/resolv.conf /etc/resolv.conf.bak
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

67. Edit your /etc/resolv.conf file so it looks like the following:

```
GNU nano 2.0.9      File: /etc/resolv.conf      Modified
nameserver 192.168.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4_

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

68. Type `nano -w nat.sh` and press Enter. Then, add the following lines:
- ```
iptables -A FORWARD -i eth0 -o eth1 -s 192.168.182.0/24 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A POSTROUTING -t nat -j MASQUERADE --source 192.168.182.0/24
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
GNU nano 2.0.9 File: /usr/src/chillispot-1.1.0/nat.sh
iptables -A FORWARD -i eth0 -o eth1 -s 192.168.182.0/24 -m conntrack --ctstate $
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A POSTROUTING -t nat -j MASQUERADE --source 192.168.182.0/24
echo 1 > /proc/sys/net/ipv4/ip_forward

[Read 4 lines]

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

69. Giving running permissions to your new file. Type `chmod +x nat.sh` and press Enter.

```
root@ubuntu:/usr/src/chillispot-1.1.0# chmod +x nat.sh
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

70. Type `./nat.sh` and press Enter to run your script.

```
root@ubuntu:/usr/src/chillispot-1.1.0# ./nat.sh
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

*Note: You will need to run this script every time you reboot your system.*

71. Edit `/etc/sysctl.conf` and add these lines:

```
net.ipv4.conf.default.forwarding=1
net.ipv4.conf.all.forwarding=1
```

```
GNU nano 2.0.9 File: /etc/sysctl.conf Modified

Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#
The contents of /proc/<pid>/maps and smaps files are only visible to
readers that are allowed to ptrace() the process
kernel.maps_protect = 1
net.ipv4.conf.default.forwarding=1
net.ipv4.conf.all.forwarding=1_

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^I To Spell
```

72. Run ChilliSpot on a different terminal by pressing Alt+F2. Then type `chilli --debug --fg` and press Enter. This will start ChilliSpot on debug mode so you can see any error messages.

```
root@ubuntu:/home/notroot# chilli --debug --fg
ChilliSpot version 1.1.0 started.
chillispot[153131]: ChilliSpot 1.1.0. Copyright 2002-2005 Mondru AB. Licensed under GPL. See http://www.chillispot.org for credits.
Waiting for client request...
-
```

73. Press Alt+F1 to change back to your original terminal window.



```
root@ubuntu:/usr/src/chillispot-1.1.0# _
```

74. Type `/etc/init.d/freeradius restart` and press Enter to restart FreeRadius.

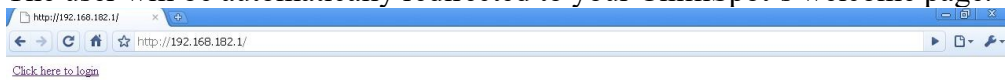
```
root@ubuntu:/home/notroot# /etc/init.d/freeradius restart
* Stopping FreeRADIUS daemon freeradius [OK]
* Starting FreeRADIUS daemon freeradius [OK]
root@ubuntu:/home/notroot# _
```

75. Now, all clients that will want to connect to your AP will have to authenticate first before joining your wireless network.  
A new user wants to visit a site.



---

The user will be automatically redirected to your ChilliSpot's welcome page.



---

The user will have to login using his/her credentials. Here, we are logging in with “test” as the username and password.



The user can now browse the Internet normally.



---

76. Close all windows.