Foo, Bar and Foobar

A blog with a meaningless title

Hostapd : The Linux Way to create Virtual Wifi Access Point

NOTE: Although this guide should work in most cases, it is not flawless and still requires few minor modifications to make the process bug-free. Please do point out corrections and changes.

(After you are done with this post, please do checkout my Python Hostapd Client)

I was recently looking into ways to use my laptop's wifi adapter as a wireless access point to enable my phone (Nokia E63) and playstation portable to connect to the internet through it. Ad-hoc feature may be used to share internet throug wifi, but it doesn't work with many phones and my PSP. I found connectify and virtual router for Windows which served this purpose, unsatisfactorily. Other than the reasons like Virtual Router not detecting my 3g modem and Connectify (fr version) not allowing me to set desired ssid for my virtual access point, the biggest issue with these two was the limited modes available for the access point. Both the programs offered only WPA2-PSK encryption for infrastructure mode an WEP and open encryption for ad-hoc modes. Many devices connect only through infrastructure mode and support for WPA2-PSK is absent in few devices (including the PSP). Also, since I am a Linux user, I needed something else.

This is where **hostapd** kicks in.

HOSTAPD

"hostapd is a user space daemon for access point and authentication servers. "

In simple words, hostapd allows you to create software wifi access points allowing decent amount of configuration optic In rest of this post, I will show how to create a software access point in Linux using **hostapd** and share your internet to devices through it. I have used my Lenovo Z560 with ath9k wifi driver under Arch Linux and have also tested it under Ubuntu 11.10. But the method is also applicable for other Linux distros and supported hardware.

If the method works/doesn't work for a non-Atheros wifi card, please do comment.

REQUIREMENTS

- Supported Wireless Card (ie. supports master mode)
- An internet connection you want to share. (not strictly a neccessity)
- A linux distro

CHECKING WIFI CARD SUPPORT

First of all, you will need to find if your wireless card is supported by hostapd.

To check what kernel driver is in use for your wireless card, type the follwing in the terminal

```
lspci -k | grep -A 3 -i "network"
```

Look for the section in the output which corresponds to your Wireless controller. In my case, it is

06:00.0 Network controller: Atheros Communications Inc. AR9285 Wireless Network Adapter (PCI-Express) (rev 01)
Subsystem: Lenovo Device 30a1
Kernel driver in use: ***ath9k***

The Bold part is my kernel driver in use. It will vary depending on your wifi card and driver you are using.

Now get the interface details of your wireless driver by

```
modinfo ath9k | grep 'depend'
```

replace **ath9k** by your wifi kernel driver you determined in the last step. In my case, the output was

depends: ath9k_hw,mac80211,ath9k_common,ath,cfg80211

modinfo says my Kernel driver supports mac80211 interface which is supported by hostapd which implies that my wifi card is compatible with hostapd.

Supported wireless cards/drivers

- Linux mac80211 drivers
- Host AP driver for Prism2/2.5/3
- madwifi (Atheros ar521x)
- BSD net80211 layer (e.g., Atheros driver) (FreeBSD 6-CURRENT)

INSTALLING HOSTAPD

Install Hostapd from your distro's repo

```
#Arch Linux
sudo pacman -S hostapd
#Ubuntu
sudo  apt-get update && sudo apt-get install hostapd
#Should be available in official repo of your distro
```

Or Download Hostapd here and compile it.

CONFIGURING HOSTAPD

The */etc/hostapd/hostapd.conf* is the main configuration which you need to deal with in order to set up a **SoftAP**.

This is the minimal configuration setting which will let you test if hostapd is working. Create a file ~/**hostapd-test.conf** with the following content:

```
#change wlan0 to your wireless device
interface=wlan0
driver=nl80211
ssid=test
channel=1
```

start hostapd by

```
sudo hostapd ~/hostapd-test.conf
```

Use a wifi device to check if the access point is being detected. You won't be able to connect to it at present.
Once hostapd is working fine, its time to configure hostapd with more options.
Here is a brief overview of some of its options:

```
#sets the wifi interface to use, is wlan0 in most cases
interface=wlan0
#driver to use, nl80211 works in most cases
driver=nl80211
#sets the ssid of the virtual wifi access point
ssid=dontMessWithVincentValentine
#sets the mode of wifi, depends upon the devices you will be using. It can be a,b,g,n. Setting to g ensur
hw_mode=g
#sets the channel for your wifi
channel=6
#macaddr_acl sets options for mac address filtering. 0 means "accept unless in deny list"
macaddr_acl=0
#setting ignore_broadcast_ssid to 1 will disable the broadcasting of ssid
ignore_broadcast_ssid=0
#Sets authentication algorithm
#1 - only open system authentication
#2 - both open system authentication and shared key authentication
auth_algs=1

#####Sets WPA and WPA2 authentication#####
#wpa option sets which wpa implementation to use
#1 - wpa only
#2 - wpa2 only
#3 - both
wpa=3
#sets wpa passphrase required by the clients to authenticate themselves on the network
wpa_passphrase=KeePGuessinG
#sets wpa key management
wpa_key_mgmt=WPA-PSK
#sets encryption used by WPA
wpa_pairwise=TKIP
#sets encryption used by WPA2
rsn_pairwise=CCMP

################################

#####Sets WEP authentication#####
```

```
#WEP is not recommended as it can be easily broken into
wep_default_key=0
wep_key0=qwert    #5,13, or 16 characters
#optionally you may also define wep_key2, wep_key3, and wep_key4

################################
#For No encryption, you don't need to set any options
```

So, here is my complete */etc/hostapd/hostapd.conf* with WPA authentication options.

```
interface=wlan0
driver=nl80211
ssid=dontMessWithVincentValentine
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=KeePGuessinG
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

SETTING UP THE DHCP SERVER

> ***Alternative Method****: I recommend using dnsmasq over dhcpd for this scenario mainly due to the ease in configuring it. I have continued this post from this point in a new separate post which uses dnsmasq instead of dhcpd. If you have any reason to choose dhcpd over dnsmasq or if dnsmasq isn't working for you, then carry on.*

Now that hostapd is running fine, you need to setup a DHCP server to run along with hostapd in order to assign ip address to the devices connecting to the access point. Setting up a dhcp server is quite straightforward.

Install dhcp server from your distro's repo.

```
#Arch Linux
sudo pacman -S dhcp
```

```
#Ubuntu
sudo  apt-get update && sudo apt-get install isc-dhcp-server
#Fedora
sudo yum -y install dhcp
```

edit /etc/dhcpd.conf (for arch linux) or /etc/dhcp/dhcpd.conf (for Ubuntu) to

```
ddns-update-style none;
ignore client-updates;
authoritative;
option local-wpad code 252 = text;

subnet
10.0.0.0 netmask 255.255.255.0 {
# --- default gateway
option routers
10.0.0.1;
# --- Netmask
option subnet-mask
255.255.255.0;
# --- Broadcast Address
option broadcast-address
10.0.0.255;
# --- Domain name servers, tells the clients which DNS servers to use.
option domain-name-servers
10.0.0.1, 8.8.8.8, 8.8.4.4;
option time-offset
0;
range 10.0.0.3 10.0.0.13;
default-lease-time 1209600;
max-lease-time 1814400;
}
```

options are easy to understand and you may change it according to your needs (if required).

FINAL STEPS

The final steps involves enabling NAT to share internet in one network interface with the clients connected through hostapd.

I have included all the steps to configure wlan interface, enable NAT, start DHCP server and hostapd in the BASH scrip below

Let the name of this file be ***initSoftAP.***

Copy the BASH file below to the file *initSoftAP.*(and make changes to file according to your system)

```
#!/bin/bash
#Initial wifi interface configuration
ifconfig $1 up 10.0.0.1 netmask 255.255.255.0
sleep 2
###########Start DHCP, comment out / add relevant section##########
#Thanks to Panji
#Doesn't try to run dhcpd when already running
if [ "$(ps -e | grep dhcpd)" == "" ]; then
dhcpd $1 &
fi
###########
#Enable NAT
iptables --flush
iptables --table nat --flush
iptables --delete-chain
iptables --table nat --delete-chain
iptables --table nat --append POSTROUTING --out-interface $2 -j MASQUERADE
iptables --append FORWARD --in-interface $1 -j ACCEPT

#Thanks to lorenzo
#Uncomment the line below if facing problems while sharing PPPoE, see lorenzo's comment for more details
#iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

sysctl -w net.ipv4.ip_forward=1
#start hostapd
hostapd /etc/hostapd/hostapd.conf 1>/dev/null
killall dhcpd
```

Script Changes (12/9/12) : Added check for already running dhcpd process (Thanks to Panji), Added an optional line t fix issues related to PPPoE connection sharing (See lorenzo's comment)

*It might be more convenient to use **hostapd -B /etc/hostapd/hostapd.conf** which runs hostapd in background. (Thanl to **Enda** for pointing out)*

Make this file executable, and run it. The syntax for executing it is

./initSoftAP wifi_card_interface interface_with_internet

```
chmod +x initSoftAP
./initSoftAP wlan0 eth0
```

The *"wifi_card_interface*" will be *wlan0* most of the cases. For "*interface_with_internet*", since I want to share internet fr my ethernet network interface, I used *eth0*. If I ever want to share internet from my 3g modem, I use *ppp0*. (These valu need not be same for everyone)
You may see available network interfaces by

```
ifconfig -a
```

Note:

- If dhcpd is failing to start and throwing errors like ***No subnet declaration for wlan0,*** take a look at these commer by Mahesh and Charlie. Either use dnsmasq, or try adding the following to the /etc/default/isc-dhcp-server file

INTERFACES="wlan0"
option netbios-name-servers 10.0.0.1

- Raspberry Pi users might want to take a look at Denis Kökeny's comment.

Problems, Errors, Feedback or any alternatives? Feel free to reply.

This entry was posted in Linux and FLOSS and tagged access point, dhcp, dhcpd, hostapd, hotspot, iptables, Linux, na wifi on April 27, 2012 [https://nims11.wordpress.com/2012/04/27/hostapd-the-linux-way-to-create-virtual-wifi-access-po .

---

475 thoughts on "Hostapd : The Linux Way to create Virtual Wifi Access Point"

Sha156

March 17, 2016 at 9:35 am

Hey,
We created a virtual access point as mentioned above. The problem is our WiFi is getting saved but not connecting. W
could be the problem? Could you please help us to solve this?

Pingback: Kako od Debian Linux-a napraviti ruter | sigurnost na mrezi

Pingback: Mininet-WiFi: Software defined network emulator supports WiFi networks | Open-Source Routing and Netwo
Simulation

dgn2200 firmware optus mobile
May 26, 2016 at 11:55 am

Fabulous, what a website it is! This weblog gives valuable
facts to us, keep it up.

Rafael
May 31, 2016 at 9:57 pm

how can i configure a hostapd with eap-sim (without external radius like freeradius)

Pingback: [ASK] server - DHCP option domain name server | Some Piece of Information

Nasri
July 17, 2016 at 11:59 am

Configuration Hostapd in Centos Linux with wireless WiFi adapter Dual Band 2.4 Ghz / 5 Ghz

It requires skilled professional who is able to configure a wireless adapter that works with Dual Band 2.4 GHz and 5 GH
simultaneously.

Adapters are available brand Intel and Atheros.

NOTE: We have a wireless adapter already running at 2.4 Ghz hostapd.
Implementation Time: Maximum 10 days.
Service mode: Remote Service via teamviewer to face technical support
Más información: S-K-Y-P-E: nazrifarran

Pingback: How To Create The Soft Link In Linux | Information

Pingback: A Debian wifi router – nullrend blogs