2019

RaspberryAp

A BASIC ACCESS POINT CONFIGURATION, TRIED AND TESTED

BEN-AMI HALPERN

Follow the project/contribute on https://github.com/bnreplah/RouterPi-extender

The set up and creation of the Access point hosting our target machine is shown below. The initial scope of this report is creating an access point that is public facing and accessible. Upon connecting to the access point the computers connected should all be visible and reconnaissance for the CTF is underway. This project is the base of portable VPN clients, travel Routers, as well as a Wireless Bridge. Some configurations may require an external WiFi dongle to be attached and configured. In the case below a WiFi Dongle was not used, however it is needed for configuring a NAT in a Wireless Acess Point (WAP).

Goal:

The mission here is to create a raspberry pi Access Point. From there we will proceed to create a travel router, an access point that is able to piggyback public wifi and host a more secure network in public spaces.

Raspberry Pi has a Dual Band Wireless Network Interface Controller. Understanding what this means may prove crucial to our understanding of the Router set up. A question that comes to mind is if a Dual Band nic can both virtualize itself and split itself into two interfaces per Band. It seems possible however the only issue is that we find is that the NIC can't do both at the same time. It either connects to the WiFi or is broadcasting the signal. This is seen where the network manager will pick up the connected internet connection however will drop it after a few seconds and then attempt to try again.

https://www.lifewire.com/dual-band-wireless-networking-explained-818279

Modern Wifi Adapters have dual-band capability, meaning that they are able to broadcast both 2.5ghz and 5ghz. However most if not all wireless adapters are only able capable of receiving/broadcasting one signal at a time. So in order to set up our Wireless Access Point/ Router we need another wireless dongle.

First Step

We need to install Hostapd and dnsmasq to create the architecture needed for creating an AP as well as a Domain Name Server (DNS) masq, which will set up domain name forwarding. Both can be installed through the aptitude repository.

```
sudo apt-get install dnsmasq hostapd -y
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
```

We switch the interface into monitor mode to allow it to broadcast.

(I found this step not crucial)

```
ifconfig
ifconfig wlan0 down
iwconfig wlan0 mode monitor
ifconfig wlan0 up
```

Now we create the hostapd.conf file

```
nano hostapd.conf
```

In the file write the following

```
interface=wlan0
driver=nl80211
ssid=[AP NAME]
hw_mode=g
chanel=[AP Channe:6]
macaddr_acl=0
ignore_broadcast_ssid=0
#uncomment these lines if you'd like authentication
#auth_algs=1
#wpa=2
#wpa_key_mgmt=WPA-PSK
#rsn_pairwise=CCMP
############TKIP
#wpa_passphrase=pass
```

Now save the file

And edit the dnsmasq.conf

```
nano /etc/dnsmasq.conf
```

The in the dnsmasq.conf add the following

```
Interface=wlan0
dhcp-range=192.168.8.2,192.168.8.30,255.255.255.0,12h
dhcp-option=3,192.168.8.1
dhcp-option=6,192.168.8.1
server=8.8.8.8
log-queries
log-dhcp
listen-address=192.168.8.0
```

Now we have our AP configured

Now we create the hostapd script to activate the AP

```
#!/bin/bash
service hostapd start
service dnsmasq start
#echo run the dnsmasq start file in another terminal
hostapd /etc/hostapd/hostapd.conf


#EoF
```

```
#!/bin/bash
#you can customize this script so it the user can set their custom interface ip address
#To configure the interface type the following in the terminal

#ifconfig wlan0 up 192.168.8.1 netmask 255.255.255.0
#route add -net 192.168.8.0 netmask 255.255.255.0 gw 192.168.8.1

#uncomment the lines above if the ip isn't already configured
#not crucial


dnsmasq -C dnsmasq.conf -d


#Eof
```

Trouble Shooting

When I run the my "startHostapd" (shown above), the following error is returned:

```
Line 2: invalid/unknown driver 'nl80211'
```

This error is returned due to two main factors, generally that the NIC doesn't support that specific driver, or a spelling error or syntax error. In my case, it was both, when I attempted to set up the above configuration I was missing the appropriate driver, so I set up the AP on my native NIC. If you choose to set up the Hostapd configuration with a different driver you may need to figure out which driver is compatible with the NIC and download it separately. Upon remediating this error the response is given and all is right again in the world

```
Wlan0: AP-ENABLED
```

My Next issue I encountered was the passphrase and authentication of the AP. This wasn't such a prevalent issue at the moment, since for the current usage the AP is simple acting as a gateway to hide a local network. For the moment, I removed WPA and PSK encryption along with the passcode, however this issue will need to get solved upon moving forward in the project.

Another not prevalent but still import issue arises, which is the IP table forwarding. It seems that my issue lies in the syntax of the of the iptables command. Upon resolution of this issue, The AP will have bridge like capabilities and be able to relay the internet signal that it connects to it's clients.

Raspberry Pi Access Point

Sources, Articles, References

https://pimylifeup.com/raspberry-pi-wifi-extender/

https://www.shellvoide.com/wifi/setup-wireless-access-point-hostapd-dnsmasq-linux/

https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md

http://www.intellamech.com/RaspberryPi-projects/rpi3_simple_wifi_ap.html

https://seravo.fi/2014/create-wireless-access-point-hostapd