

Contents

- [1 Public Free Hotspot and Private ESSID on VLAN's RC6 instructions](#)
 - ◆ [1.1 Comments](#)
- [2 Notes for Getting Encryption Working on the Virtual WiFi Interface](#)
 - ◆ [2.1 Comments](#)

Public Free Hotspot and Private ESSID on VLAN's RC6 instructions

I now have my WRT54GL V24 RC6 serving separate public and private networks. My private network 192.168.1.0/24 has WPA and my public network 192.168.2.0/24 has an open ESSID for any to use, it's also on its own VLAN on port 4 of the switch.

I put this together using things from this forum and the Wiki. Here's how....

For best results use a WRT54GL or suitable router with the BCM5352 chipset since this gives each virtual AP it's own MAC address. <http://en.wikipedia.orghttp://www.dd-wrt.com/wiki/WRT54G#WRT54GL>

First go into Wireless on the web interface and set up the main wireless which will be your private one w10. You can apply WPA to this in the usual way. Then set up your first virtual interface for public which will be w10.1. You can add another alias which can be included on the same public network. Leave these as bridged for the moment, in RC6 you can change them from the web interface anyway.

Now go into Setup/VLANs to put port 4 on VLAN2. This is not as straight forward as it appears since on my GL I had to select port 1 on the screen as this corresponds to port 4 in hardware.

- Early hardware: http://www.tonns.org/openwrt/WRTSL54GS_layout_rev1.png
- GL hardware: http://garycourt.com/wp-content/images/WRT54_sw2_internal_architecture.png

So in effect I had to move the tick on port 1 to VLAN2, leaving it set to none, to do port 4 on the back of the box.

In the DNS Masq box paste and save the following:

```
interface=br1
dhcp-range=br1,192.168.2.100,192.168.2.120,255.255.255.0,600
dhcp-option=net:br1,3,192.168.2.1
```

Having saved these settings open a shell to the router and login as root. Paste to shell:

```
nvrn set vlan0ports="1 2 3 5*"
nvrn set vlan2ports="0 5*"

nvrn set rc_startup='
#!/bin/ash
PATH="/sbin:/usr/sbin:/bin:/usr/bin:${PATH}"

brctl addbr br1
```

VLAN_Detached_Networks_each_with_Wireless_and_Internet

```
brctl delif br0 w10.1
brctl addif br1 w10.1
ifconfig br1 192.168.2.1 netmask 255.255.255.0
ifconfig vlan2 up
brctl addif br1 vlan2
ifconfig br1 up
'

nvram set rc_firewall='
iptables -I INPUT -i br1 -j ACCEPT
iptables -I FORWARD -i br1 -o vlan1 -m state --state NEW -j ACCEPT
iptables -I FORWARD -i br1 -o ppp0 -m state --state NEW -j ACCEPT
iptables -I FORWARD -i br0 -o br1 -j logdrop
'
nvram commit
```

You will now need to reboot your router.

The way this works is first to put port 4 on VLAN2 and create w10.1 then to create a bridge br1 and add VLAN2 and W10.1 to it. With the iptables rules this is then connected to the WAN.

Clients on the 192.168.1.0 LAN and 192.168.2.0 LAN cannot ping across although they can both see the router on either 192.168.1.1 or 192.168.2.1.

I have not implemented bandwidth shaping nor Hotspot splash page, but it's a start.

Comments

I implemented two wireless networks on the same router using the instructions at [Implementing Inexpensive Multiple SSID Networks](#), on which I based the recipe at [Navas Wireless Wiki:Wi-Fi How To:Two wireless networks on one router](#). --JNavas 21:01, 18 September 2009 (CEST)

Notes for Getting Encryption Working on the Virtual WiFi Interface

The above is an excellent guide, but I wanted to set up two private networks, one for guest use, one for my private use.

Everything appeared to be working until I enabled encryption on the virtual interface.

I discovered that the encryption agent, the `nas` daemon was started with the following options:

```
nas -P /tmp/nas.w10.1lan.pid -H 34954 -l br0 -i w10.1 -A -m 132 -k <passwd> -s <SSID> -w 6 -g 360
```

Where `<passwd>` is your WPA PSK, and `<SSID>` is the SSID you want for the virtual WiFi interface.

The issue is that `nas` is using the wrong bridge interface. It should be using `br1`.

The work around I'm using is to modify the `rc_startup` to look like:

```
rc_startup='
```

VLAN_Detached_Networks_each_with_Wireless_and_Internet

```
brctl addbr br1
brctl delif br0 wl0.1
brctl addif br1 wl0.1
ifconfig br1 192.168.1.1 netmask 255.255.255.0
ifconfig vlan2 up
brctl addif br1 vlan2
ifconfig br1 up
kill $( cat /tmp/nas.wl0.1lan.pid )
nas -P /tmp/nas.wl0.1lan.pid -H 34954 -l br1 -i wl0.1 -A -m 132 -k <passwd> -s <SSID> -w 6 -g 360
'
```

If the nas daemon for the virtual interface dies, it will be restarted with the incorrect bridge. I hope to ask the experts on the forum about this, but in case I don't get that far, the above does at least initially work.

Comments

This may be related to [Bug 0003729](#). --[JNavas](#) 20:52, 18 September 2009 (CEST)