NOTE: This is an old unmaintained and duplicate guide. You should use one of the following currently maintained guides instead.

For a GUI based method see: Separate LAN and WLAN.

If you're separating virtual interfaces then use the instructions from the Multiple WLAN Guide.

Contents

Introduction

The goal of this article is to separate the wireless LAN from the hardwired LAN, with an independent class c network for both interfaces. For security purposes, we will prevent communication between both networks, as well as between wireless clients. This way, the wireless network is able to be a little more lax on security without the threat of unsecured access to the wired network. This is a good implementation for wireless cafés and other businesses that provide wireless Internet access to their customers.

Setup

The configuration takes about 20 minutes to complete. For this tutorial, I used a WRT54GL v1.1 with DD-WRT NoKaid v24 sp1. I've also done this setup on a WRT54G v8.2 with DD-WRT Micro v24 sp1, so I imagine that any version of DD-WRT v24 sp1 would work with this. For this scenario, we will set the wired network to 192.168.1.1/24 and the wireless network to 192.168.2.1/24.

Configuration

Step 1

First, we will log into the router interface and go to Wireless > Basic Settings. Change Network Configuration from Bridged to Unbridged. This will bring up 3 new settings. For IP Address enter 192.168.2.1 and for Subnet Mask enter 255.255.255.0. Click Apply Settings. Now click on Wireless > Advanced Settings. Change AP Isolation from Disable to Enable and click Apply Settings. This will prevent wireless clients from communicating with each other. As far as wireless encryption goes, that's up to your preferences. I'm not going to cover it in this article.

Step 2

Now that we have the wireless network setup, we need to have the DHCP server issue IP addresses within its range. To do this, first go to Setup > VLANs. Scroll to the bottom and change Wireless from LAN to None. Click Apply Settings. Now go to Setup > Networking. Scroll to the bottom and click Add under Multiple DHCP Server. This will bring up several new options. Set the first option to eth1 (the wireless adapter) and leave the other default settings as they are. Click Apply Settings.

Step 3

Now we have the wireless and wired on their own class c networks, all we have to do now is prevent them from talking to each other. Goto Administration > Commands. Now put the following commands into the text box:

```
iptables  -t nat -I PREROUTING -i eth1 -d $(nvram get lan_ipaddr)/$(nvram get lan_netmask) -j DRO
iptables  -t nat -I PREROUTING -i br0 -d $(nvram get eth1_ipaddr)/$(nvram get eth1_netmask) -j DR
```

Click Save Firewall, reboot the router for good measure, and BAM! You now have your LAN on the 192.168.1.1/24 network and your wireless on 192.168.2.1/24, without access to each other.

Comments

--Jozevolf 13:15, 30 September 2008 (CEST)

I followed your article on v24 sp1. I had problems with wifi clients not being able to acquire IP from DHCP. I found out that there was a problem with the nas authentication service running without '-l' parameter. I then included the following two commands in post boot script:

```
killall nas
nas -P /tmp/nas.wl0lan.pid -H 34954 -l eth1 -i eth1 -A -m 132 -k yourpsk -s yourssid -w 6 -g 3600
# adjust other nas parameters according to your security needs
# use http:/http://wiki.dd-wrt.com/wiki.openwrt.org/OpenWrtDocs/nas as reference
```

For firewall part, I gues the following is more appropriate:

```
iptables -I FORWARD 1 -i eth1 -d $(nvram get lan_ipaddr)/$(nvram get lan_netmask) -j DROP
iptables -I FORWARD 2 -i br0 -d $(nvram get eth1_ipaddr)/$(nvram get eth1_netmask) -j DROP
```

--Nienberg 08 February 2010

- if you want to be able to control some of the wireless settings from the GUI, you can improve the startup script to something like the following. Remember you have to reboot the router though after you make a change in the GUI so the startup script will run. You could extend the concept for the m and w parameters, but you will have to convert them from the text stored in nvram to the integer expected by the command line.

```
PSK=$(nvram get wl0_wpa_psk)
SSID=$(nvram get wl0_ssid)
REKEY=$(nvram get wl0_wpa_gtk_rekey)

killall nas
nas -P /tmp/nas.wl0lan.pid -H 34954 -l eth1 -i eth1 -A -m 132 -k $PSK -s $SSID -w 6 -g $REKEY &
```

- if you have port redirection (Virtual Servers) on the LAN and if you want wireless users to be able to access the Virtual Servers then you must be more careful with the order of the firewall rules. The following script is an example:

```
LAN_IP=$(nvram get lan_ipaddr)/$(nvram get lan_netmask)

# we want users of our wireless to be able to access our virtual servers,
# so our rules must come after the rules that forward to our virtual servers.
# we insert them just before the last two rules.

# delete this rule created by the GUI. it is too high in the order
iptables -D FORWARD -i eth1 -j ACCEPT

# count the number of lines in the output of the list minus 3
INSERT_LINE=$(expr $(iptables -nL FORWARD | wc -l) - 3)

# prevent the wireless from sending to the LAN
iptables -I FORWARD $INSERT_LINE -i eth1 -d $LAN_IP -j logdrop

# accept anything else from the wireless (put back the deleted rule)
INSERT_LINE=$(expr $INSERT_LINE + 1)
iptables -I FORWARD $INSERT_LINE -i eth1 -j ACCEPT
```

--Cdufour 27 February 2010

Running V24 preSP2 build 13064 VINT/nokaid on a WRT54G v2.2, I found out that the culprit is not the missing '-l eth1' parameter for the 'nas' service, but rather a services order start issue. I found out that restarting the 'nas' service a few seconds after the end of the boot process solves the problem of clients not being able to obtain a DHCP lease (Note: I use DNSmasq DHCP, for I have the 'DHCP Server' disabled on the LAN segment; see Separate LAN and WLAN):

In 'Administration -> Commands -> Save Startup':

```
sh -c 'sleep 15; nas=$(ps w | grep "^ *$(pidof nas) " | sed "s/.*nas -P/nas -P/"); killall -9 nas
```

References

Separate LAN and WLAN