OpenWrt Wiki

# Configure a guest WLAN

People looking for a way to configure a guest WLAN through the web interface should read the LuCI guest WLAN recipe.

Guest WLAN provides internet access to your network members. It also provides firewall security rules to isolate your guest network from the rest. This recipe contains information provided by our forums members and one blogger as shown below:

- Access Point with a second SSID for Guest Access to Internet only [https://forum.openwrt.org/viewtopic.php?id=28317]
- Atheros DIR-825 enabling the Guest wifi zone [https://forum.openwrt.org/viewtopic.php?id=26223]
- How to keep users of guest wlan out of lan ? [https://forum.openwrt.org/viewtopic.php?id=28926]
- Limiting bandwidth of the guest connection [http://jwalanta.blogspot.com/2012/03/multiple-ssid-on-openwrt-with-bandwidth.html]

## Manual configuration

The changes below assume an OpenWrt default configuration, the relevant files are:

- /etc/config/network
- /etc/config/wireless
- /etc/config/dhcp
- /etc/config/firewall
- /etc/config/wshaper

A guest WLAN (or plain guest network) across multiple network devices requires a separate VLAN.

### Step 1: Define a new network

Edit /etc/config/network and define a new `interface` section:

```
config interface 'guest'
       option proto 'static'
       option ipaddr '10.0.0.1'
       option netmask '255.255.255.0'
```

### Step 2: Copy the existing wireless network

In /etc/config/wireless, define a new wifi-iface section by copying the existing one and changing its network option to point to the newly created interface section.

```
config wifi-iface
       option device     '???'
       option mode       'ap'
       option network    'guest'
       option ssid       'guest'
       option encryption 'none'
```

For *`option 'device' '???'`* you should put the device listed in your 'wifi-device' section. For example, if your 'wifi-device' says *`config 'wifi-device' 'wifi0'`* then the wifi-iface section should be *`option 'device' 'wifi0'`*

Some hardware or drivers might not support multiple SSIDs. This is the case for e.g. the FOSS b43 driver for Broadcom hardware. If you want multiple SSIDs with Broadcom you'll need to use the proprietary wl driver.

To prevent connections between clients of the guest network, add the following line at the end of the configuration block

```
       option isolate 1
```

Some hardware or drivers might not support this option.

### Step 3: Define a new DHCP pool

In order to support DHCP on 'guest' wireless, a new `dhcp` pool must be defined in `/etc/config/dhcp`:

```
  [..]
  config dhcp 'guest'
    option interface 'guest'
    option start '50'
    option limit '200'
    option leasetime '1h'
```

```
[..]
```

### Step 4a: Adjust firewall settings

Edit `/etc/config/firewall` and add new zone section covering the 'guest' interface, allow internet, DNS and DHCP to guests:

```
[..]
config zone
  option name 'guest'
  option network 'guest'
  option input 'REJECT'
  option forward 'REJECT'
  option output 'ACCEPT'

# Allow Guest -> Internet
config forwarding
  option src 'guest'
  option dest 'wan'

# Allow DNS Guest -> Router
# Client DNS queries ordinate from dynamic UDP ports (>1023)
config rule
  option name 'Allow DNS Queries'
  option src 'guest'
  option dest_port '53'
  option proto 'tcp udp'
  option target 'ACCEPT'

# Allow DHCP Guest -> Router
# DHCP communication uses UDP ports 67-68
config rule
  option name 'Allow DHCP request'
  option src 'guest'
  option src_port '67-68'
  option dest_port '67-68'
  option proto 'udp'
  option target 'ACCEPT'
[..]
```

### Step 4b: Different modifications to firewall settings

I created this small set of firewall rules to completely isolate guests on the guest SSID. I had some devices that only worked with WEP or no authentication at all. WEP doesn't like to run on .11n devices in HT mode, so the only option was a wide-open SSID. I also didn't want my neighbors to quickly steal my internet… This firewall config will only allow specific, known source MACs to connect to the internet, with zero access to the rest of the network.

Edit `/etc/config/firewall` and add new zone section covering the 'guest' interface, allow SSH, DNS and DHCP to guests, allow only specific source MAC addresses out to the WAN, drop broadcast traffic and deny the rest of orders:

```
[..]

# Enable logging
config zone
  option name 'guest'
  option network 'guest'
  option input 'REJECT'
  option forward 'REJECT'
  option output 'ACCEPT'

# Quick rule to allow SSH in
config rule
  option name 'Allow SSH in'
  option src 'guest'
  option dest_port '22'
  option proto 'tcp'
  option target 'ACCEPT'

# Allow DNS Guest -> Router
config rule
  option name 'Allow DNS Queries'
  option src 'guest'
  option dest_port '53'
  option proto 'tcpudp'
  option target 'ACCEPT'

# Allow DHCP Guest -> Router
config rule
  option name 'Allow DHCP request'
  option src 'guest'
  option src_port '67-68'
  option dest_port '67-68'
  option proto 'udp'
  option target 'ACCEPT'

# Allow only specific source MAC addresses out to the WAN
config rule
  option name 'Allow my Nintendo DS'
  option src 'guest'
  option dest 'wan'
  option proto 'all'
```

```
  option src_mac '00:ab:00:32:00:00'
  option target 'ACCEPT'

# Drop broadcast traffic, it just fills the logs :)
config rule
  option name 'Drop guest broadcast'
  option src 'guest'
  option dest_ip '172.16.62.255'
  option target 'DROP'

# Another explicit deny at the end.
config rule
  option name 'Deny guest -> WAN'
  option src 'guest'
  option dest 'wan'
  option proto 'all'
  option target 'REJECT'

[..]
```

### Step 4c: Block everything, but http and https

In this configuration I shared my guest SSID, but only provide http and https connections, and isolate GUEST network from LAN network.

Edit `/etc/config/firewall`, remove any *'guest'* configurations and add new zone section covering the 'guest' interface, allow SSH, DNS and DHCP to guests, and http and https to the outside world, then block the rest.

```
[..]
config zone
  option name 'guest'
  option network 'guest'
  option forward 'REJECT'
  option output 'ACCEPT'
  option input 'REJECT'

config forwarding
  option src 'guest'
  option dest 'wan'

config rule
  option src 'guest'
  option src_port '67-68'
  option dest_port '67-68'
  option proto 'udp'
  option target 'ACCEPT'
  option name 'Allow DHCP request'

config rule
  option src 'guest'
  option dest_port '53'
  option proto 'tcpudp'
  option target 'ACCEPT'
  option name 'Allow DNS Queries'

config rule
  option src 'guest'
  option dest 'lan'
  option name 'Deny Guest -> LAN'
  option proto 'all'
  option target 'DROP'

config rule
  option target 'ACCEPT'
  option src 'guest'
  option dest 'wan'
  option name 'Allow Guest -> WAN http'
  option proto 'tcp'
  option dest_port '80'

config rule
  option target 'ACCEPT'
  option src 'guest'
  option dest 'wan'
  option name 'Allow Guest -> WAN https'
  option proto 'tcp'
  option dest_port '443'

config rule
  option src 'guest'
  option dest 'wan'
  option name 'Deny Guest -> WAN'
  option proto 'all'
  option target 'DROP'

[..]
```

### Step 5: Limit bandwidth of the connection

> Wondershaper (wshaper) is not recommended. Better use `qos-scripts` or `sqm-scripts` (see the Bufferbloat project [http://www.bufferbloat.net/projects/cerowrt/wiki/Wondershaper_Must_Die]).

This is completely optional, but advised. Install package wshaper: `opkg install wshaper`

Edit `/etc/config/wshaper`:

```
config wshaper 'settings'
  option network 'guest'
  option downlink '64'
  option uplink '512'
```

> `Downlink` and `uplink` options are maximal limits, but in practice the speed will be slightly lower as wshaper also tries to prioritize traffic, so that the network stays responsive even when e.g. someone downloads a huge file - there are also wshaper options that can control this). The units are in `kbit`.

### Multiple network devices

The basics are already covered by points 1 to 5 above. For a network setup that involves two or more network devices (e.g. a router, one or more switches, one or more access points) you need to provide a separate VLAN. It is recommended to configure your VLANs through the web UI, since this offers an easier overview, but if you know what you are doing, the configuration files themselves are easy to edit as well.

**Some basics about VLANs:**

- Most devices only use one VLAN by default (VLAN ID 1). The instructions that follow assume this is the case. Double check before proceeding.
- VLAN IDs match the virtual interfaces listed by ifconfig, i.e. a VLAN with ID 3 will show as ethX.3 (where X is your real interface, e.g. eth0).
- VLAN IDs should be identical across all network devices.
- A port can have three states: Off (not part of a specific VLAN), Untagged (when part of a VLAN), Tagged (when part of two or more VLANs).
- If a port is part of multiple VLANs, it needs to be set to Tagged in every single VLAN it is part of.
- Every VLAN should also include the CPU (tagged by default).

Furthermore, for a guest WLAN, only the port(s) connecting a network device to another one should be part of the VLAN. In practice, this means that in a router - access point setup, on each device only the port connecting to the other network device goes into the VLAN.

> Make sure you have identified the ports correctly. The VLAN definitions use **internal port numbers**. Quite a few devices (e.g. the Netgear WNDR3700 v1) number their internal ports differently from the numbers on the enclosure! Tagged ports are clearly visible as such in the UI, in the configuration file they're marked by a **t** behind the port number.

> Creating a new VLAN with port X already part of another VLAN means port X should be set to tagged **in all the existing VLAN(s) it is part of. Failure to do so may render your switch inoperable.** We cannot stress this enough! Only certain switches support ports being untagged in one and tagged in another VLAN at the same time.

**New VLAN**

Edit /etc/config/network and define a new VLAN with ID 2 (change if ID 2 is already taken). Make sure to use the right device name - check your existing VLAN stanza. Do this on every router, switch or AP.

```
config switch_vlan
  option device      'switch0'
  option vlan        '2'
  option ports       '3t 5t'
```

Here, internal ports 3 and 5 are part of the VLAN with ID 2, port 5 being the CPU. Both ports are tagged (indicated by the trailing **t**). Make sure port 3 (if you keep it in VLAN 1 as well) is tagged there as well. Below how it looks in LuCI:


**Note how port 3 is now tagged in VLAN 1 as well.** As explained above, you cannot have port X untagged in one VLAN and tagged in another. A port needs to have the same status across all the VLANs it is part of.

**New interface**

Add the guest interface to your router just like in step 1. The important difference is that, unlike in a single router setup, **we define an interface**.

```
config interface     'guest'
  option ifname      'eth0.2'
  option proto       'static'
  option ipaddr      '10.0.0.1'
  option netmask     '255.255.255.0'
```

On each *access point*, we replicate the same stanza, but with two substantial differences:

- We add an interface type, namely *bridge* (this will put the LAN and WLAN interfaces on the AP in one network);
- We set a different static IP (typically incremented by one).

Your config on your access point should look like this:

```
config interface     'guest'
  option ifname      'eth0.2'
  option type        'bridge'
  option proto       'static'
  option ipaddr      '10.0.0.2'
  option netmask     '255.255.255.0'
```

**Guest WLAN**

Like in step 2, replicate the wireless interface config in /etc/config/wireless on each access point. Disable the wireless on all other network devices.

**DHCP, firewall and bandwidth settings**

The modifications above are the only points where a multi-device setup differs from a single-device setup. The DHCP server only runs on the main router, so you do not need to edit any related settings on any switches or access points; do make sure though that the static IPs do not conflict with the DHCP range you set. As for the firewall, you should replicate all stanzas provided above on every network device, if they have a firewall running.

To cap the bandwidth, you can use tc, or wshaper - the latter being more user-friendly. Wshaper also has a LuCI front-end. Keep in mind wshaper treats upload speed as download, and the other way around.

## Configuration by shell script

Here is the script that makes minimal changes required to setup guest wifi network on openwrt.

```
#!/bin/sh

# This is supposed to be run on openwrt

# Written by Stanislav German-Evtushenko, 2014
# Based on [[:doc:recipes:guest-wlan]]

# Configure guest network
uci delete network.guest
uci set network.guest=interface
uci set network.guest.proto=static
uci set network.guest.ipaddr=192.168.101.21
uci set network.guest.netmask=255.255.255.0

# Configure guest Wi-Fi
uci delete wireless.guest
uci set wireless.guest=wifi-iface
uci set wireless.guest.device=radio0
uci set wireless.guest.mode=ap
uci set wireless.guest.network=guest
uci set wireless.guest.ssid=openwireless.org
uci set wireless.guest.encryption=none

# Configure DHCP for guest network
```

```
uci delete dhcp.guest
uci set dhcp.guest=dhcp
uci set dhcp.guest.interface=guest
uci set dhcp.guest.start=50
uci set dhcp.guest.limit=200
uci set dhcp.guest.leasetime=1h

# Configure firewall for guest network
## Configure guest zone
uci delete firewall.guest_zone
uci set firewall.guest_zone=zone
uci set firewall.guest_zone.name=guest
uci set firewall.guest_zone.network=guest
uci set firewall.guest_zone.input=REJECT
uci set firewall.guest_zone.forward=REJECT
uci set firewall.guest_zone.output=ACCEPT
## Allow Guest -> Internet
uci delete firewall.guest_forwarding
uci set firewall.guest_forwarding=forwarding
uci set firewall.guest_forwarding.src=guest
uci set firewall.guest_forwarding.dest=wan
## Allow DNS Guest -> Router
uci delete firewall.guest_rule_dns
uci set firewall.guest_rule_dns=rule
uci set firewall.guest_rule_dns.name='Allow DNS Queries'
uci set firewall.guest_rule_dns.src=guest
uci set firewall.guest_rule_dns.dest_port=53
uci set firewall.guest_rule_dns.proto=udp
uci set firewall.guest_rule_dns.target=ACCEPT
## Allow DHCP Guest -> Router
uci delete firewall.guest_rule_dhcp
uci set firewall.guest_rule_dhcp=rule
uci set firewall.guest_rule_dhcp.name='Allow DHCP request'
uci set firewall.guest_rule_dhcp.src=guest
uci set firewall.guest_rule_dhcp.src_port=68
uci set firewall.guest_rule_dhcp.dest_port=67
uci set firewall.guest_rule_dhcp.proto=udp
uci set firewall.guest_rule_dhcp.target=ACCEPT

uci commit

# Configure wshaper (optional)
opkg update
opkg install wshaper
uci set wshaper.settings=wshaper
uci set wshaper.settings.network=guest
uci set wshaper.settings.downlink=500
uci set wshaper.settings.uplink=2000
## Work around for https://github.com/openwrt/packages/issues/565 (wshaper: settings are not applied on boot)
echo -e '#!/bin/sh\n\n[ "$ACTION" = ifup ] && /etc/init.d/wshaper enabled && /etc/init.d/wshaper start || exit 0' > /etc/hotplug.d/iface/10-wshaper

uci commit
```

## Apply changes

1. Enable the new wireless network

   ```
   /etc/init.d/network restart
   ```

2. Restart the firewall

   ```
   /etc/init.d/firewall restart
   ```

3. Restart the DHCP service

   ```
   /etc/init.d/dnsmasq restart
   ```

4. Start traffic shaping

   ```
   /etc/init.d/wshaper start
   ```

5. Make traffic shaping permanent

   ```
   /etc/init.d/wshaper enable
   ```

## HotSpot (Captive Portal)

If you want to setup a simple Hotspot for your guest WLAN, take a look at Nodogsplash or WiFiDog.

For a captive portal to a commercial ChilliSpot compatible Hotspot service provider, look at CoovaChilli.

## Troubleshooting

If you don't get an IP from DHCP check if you listen to the interface.

```
[..] config 'dnsmasq' [..] list 'interface' 'lan' list 'interface' 'guest' [..]
```

doc/recipes/guest-wlan.txt · Last modified: 2016/08/23 22:31 by rseiler