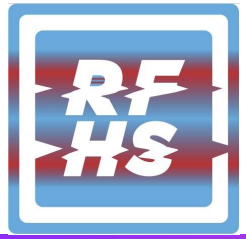


No Radios, No Problem

Hacking WiFi in a Virtual World

Nishant Sharma

Nishant Sharma

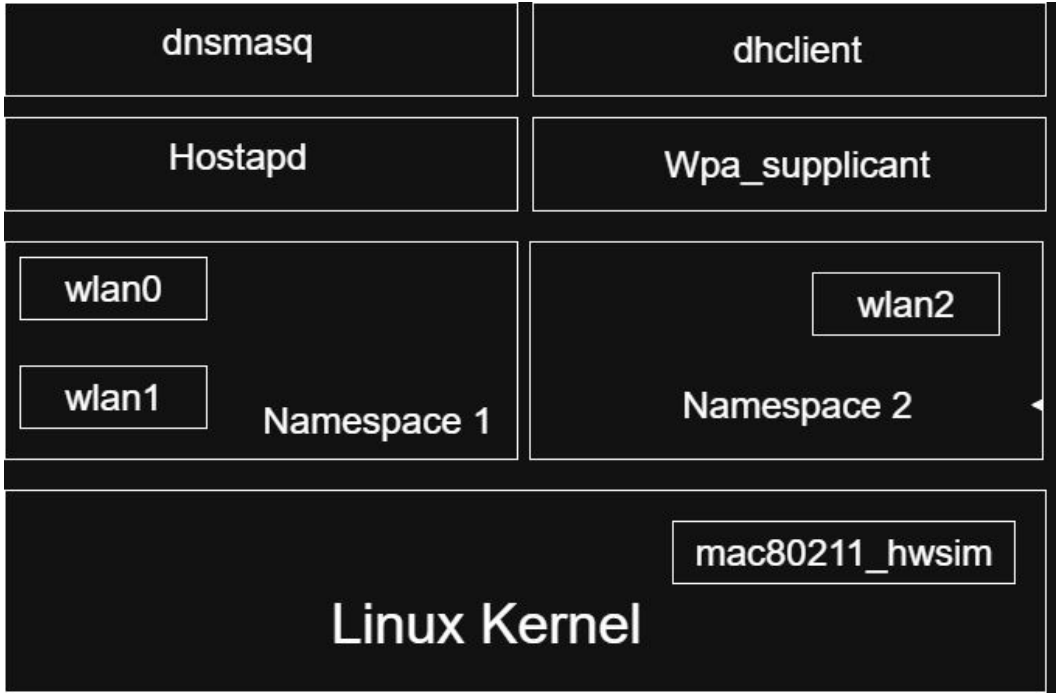
- Head of Cybersecurity Research @ SquareX (Browser Detection-Response)
- 10+ years in Cybersecurity education/training
  - VP, Labs and R&D for ine.com
  - Head R&D for Pentester Academy
- 25+ talks at Black Hat, DEF CON, HITB, OWASP
- 9+ Trainings and workshops at Blackhat, DEFCON over last 6 years
- 10+ open source tools including AWSGoat, Azuregoat, GCPGoat totalling 3500+ stars
- https://nishantsharmax.com

Why?

- Practice in VM, safe, controlled, predictable and scalable
- Easy for beginners
- Easy to replicate complicated, multi-device, multi-network, multi-config scenarios
- Practice Anywhere and Anytime

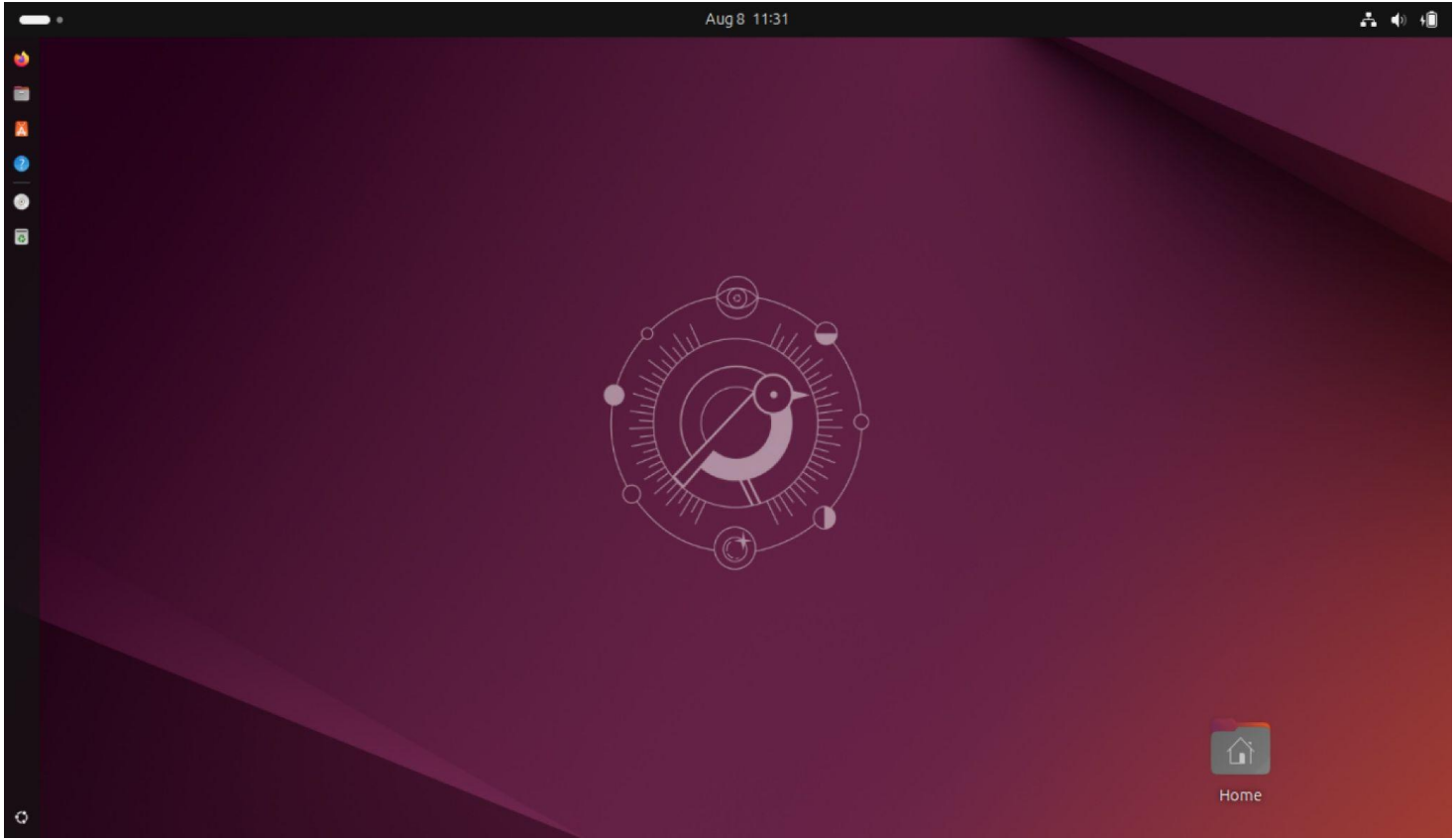
What all we need?

- Linux machine (or VM)
- 80211_hwsim
- Network Namespaces
- Hostapd
- Wpa_supplicant
- dnsmasq
- dhclient

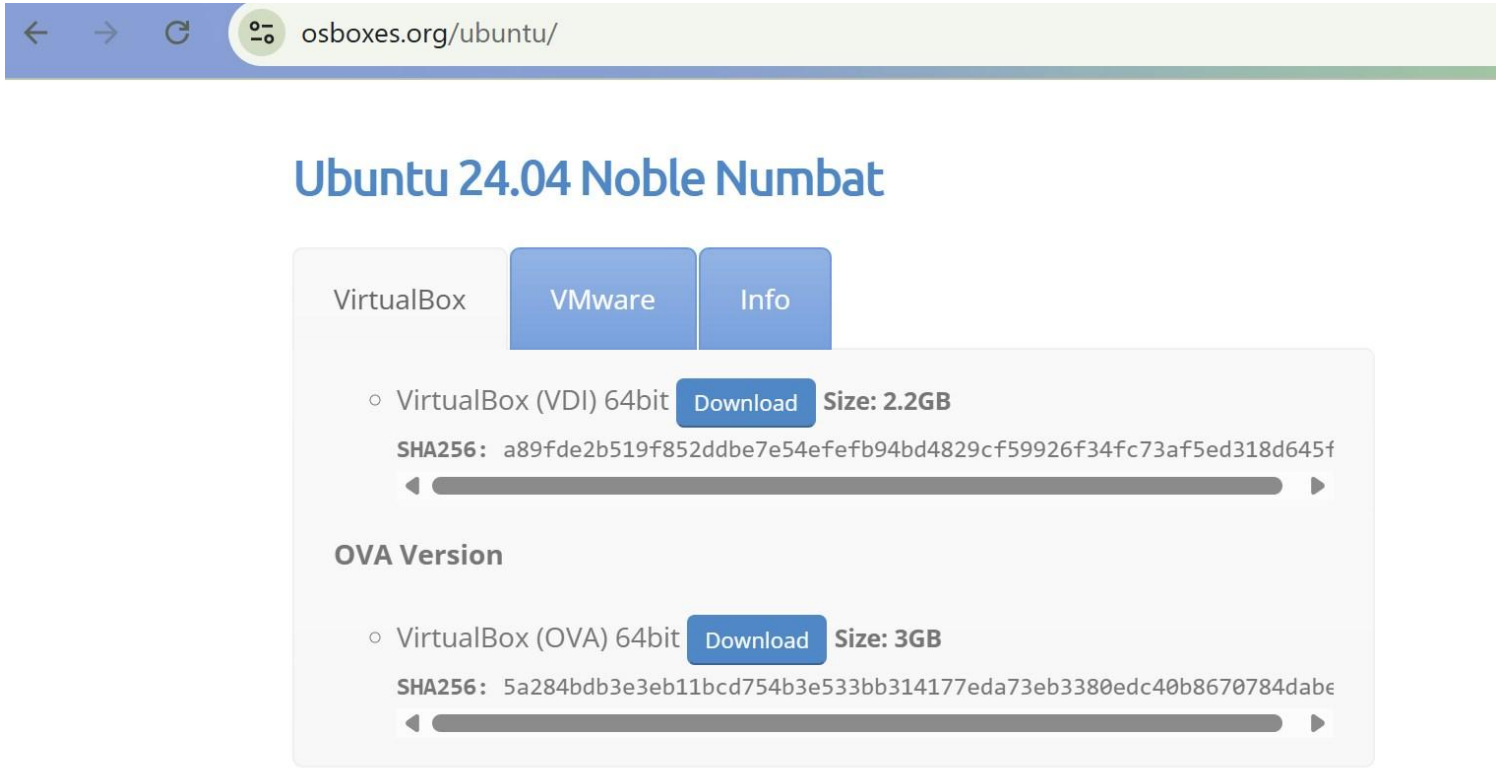


Ubuntu Desktop 24.04

SquareX

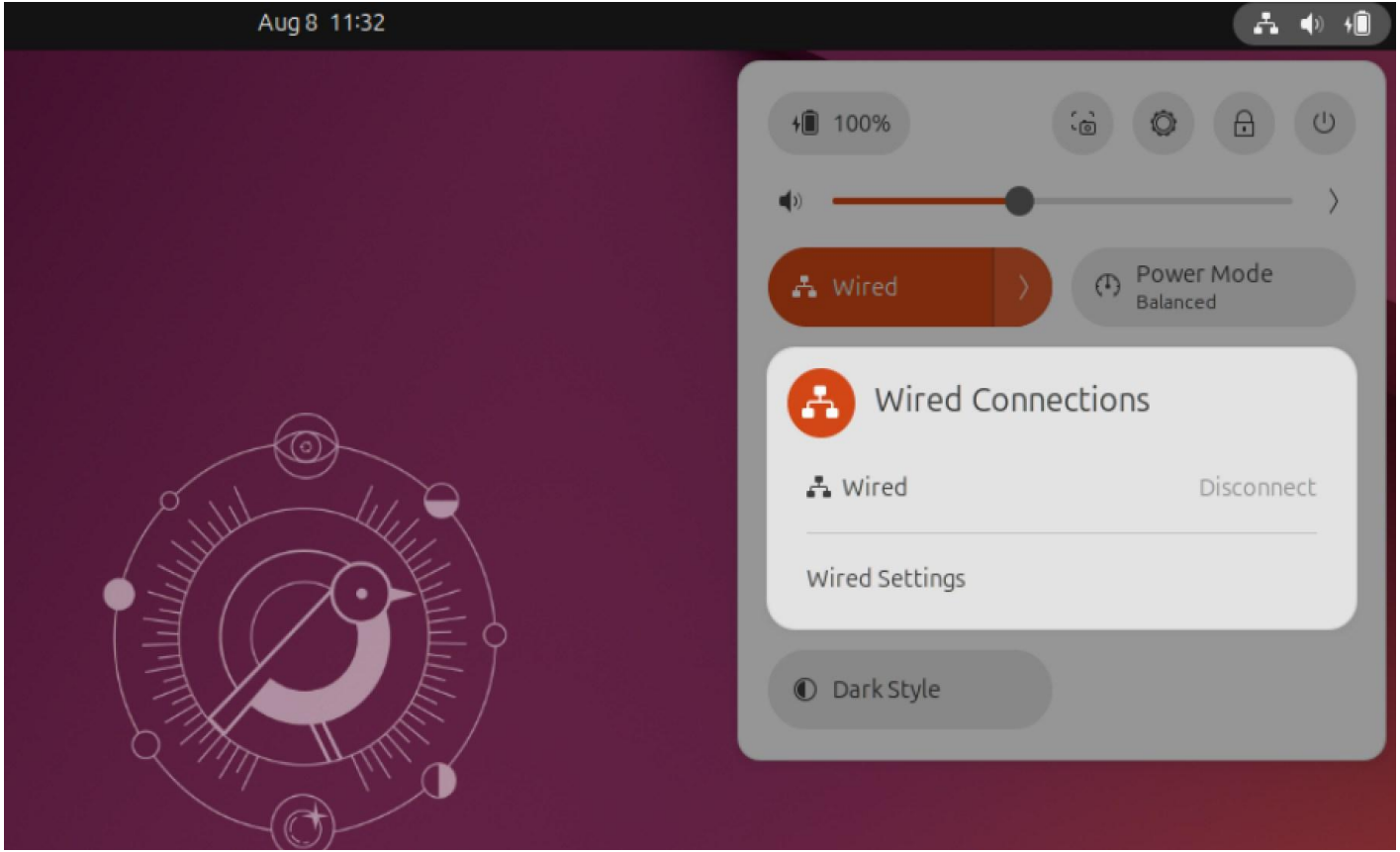


osboxes.org



Link: https://www.osboxes.org/ubuntu/

No WiFi Interface



mac80211_hwsim – Linux Wi-Fi Hardware Simulator

- Simulates IEEE 802.11 radios in software
- Part of Linux mac80211 stack
- Creates virtual PHY (phyX) & WLAN (wlanX) interfaces
- Supports managed, AP, monitor, mesh, IBSS modes
- Works with iw, hostapd, wpa_supplicant
- Ideal for Wi-Fi development, testing & research without hardware
- Ability to create multiple simulated Wi-Fi interfaces
- Works perfectly inside a VM (no PCI passthrough needed)

mac80211_hwsim

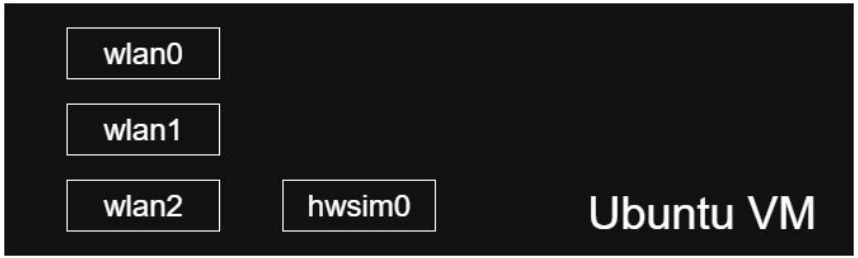
SquareX

- Make sure your kernel version is ≥ 3.6 (most modern Ubuntu versions are fine).
  Command: uname -r

- Install build-essential and linux-modules-extra
  Command: sudo apt install build-essential linux-modules-extra-$(uname -r)

- You can load the module with a given number of simulated radios:
  Command: sudo modprobe mac80211_hwsim radios=3

- Verify
  Command: ip link show



mac80211_hwsim

SquareX

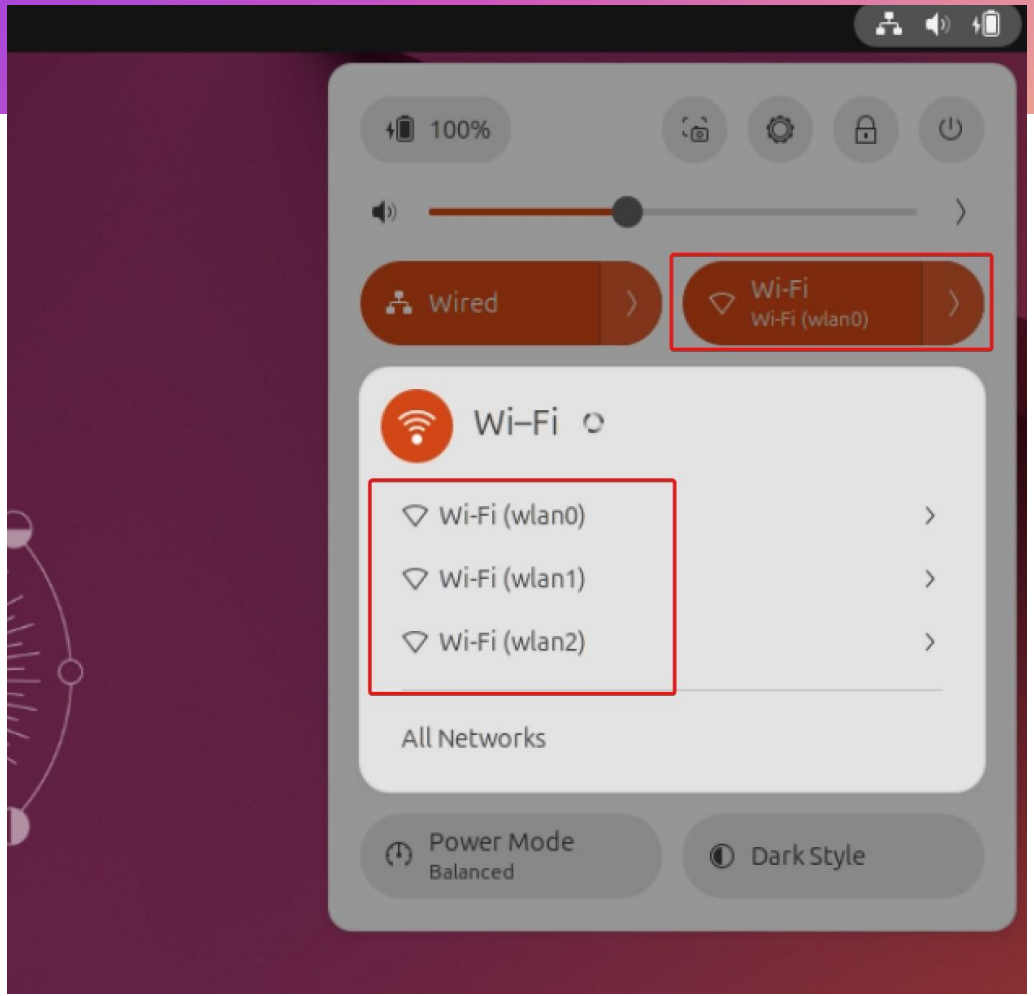
```
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo modprobe mac80211_hwsim radios=3
osboxes@osboxes:~$
osboxes@osboxes:~$
osboxes@osboxes:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    link/ether 08:00:27:a3:ac:aa brd ff:ff:ff:ff:ff:ff
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether 02:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
4: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
5: wlan2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether 02:00:00:00:02:00 brd ff:ff:ff:ff:ff:ff
6: hwsim0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode
    link/ieee802.11/radiotap 12:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
osboxes@osboxes:~$
```

iw dev

Command: iw dev

```
osboxes@osboxes:~$ iw dev
phy#2
        Interface wlan2
                ifindex 5
                wdev 0x200000001
                addr 02:00:00:00:02:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows
                        0       0       0
phy#1
        Interface wlan1
                ifindex 4
                wdev 0x100000001
                addr 02:00:00:00:01:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks
                        0       0       0       0       0
phy#0
        Unnamed/non-netdev interface
                wdev 0x2
                addr 42:00:00:00:00:00
                type P2P-device
                txpower 20.00 dBm
        Interface wlan0
                ifindex 3
                wdev 0x1
                addr 02:00:00:00:00:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks
                        0       0       0       0       0
```

WiFi Interfaces



SquareX

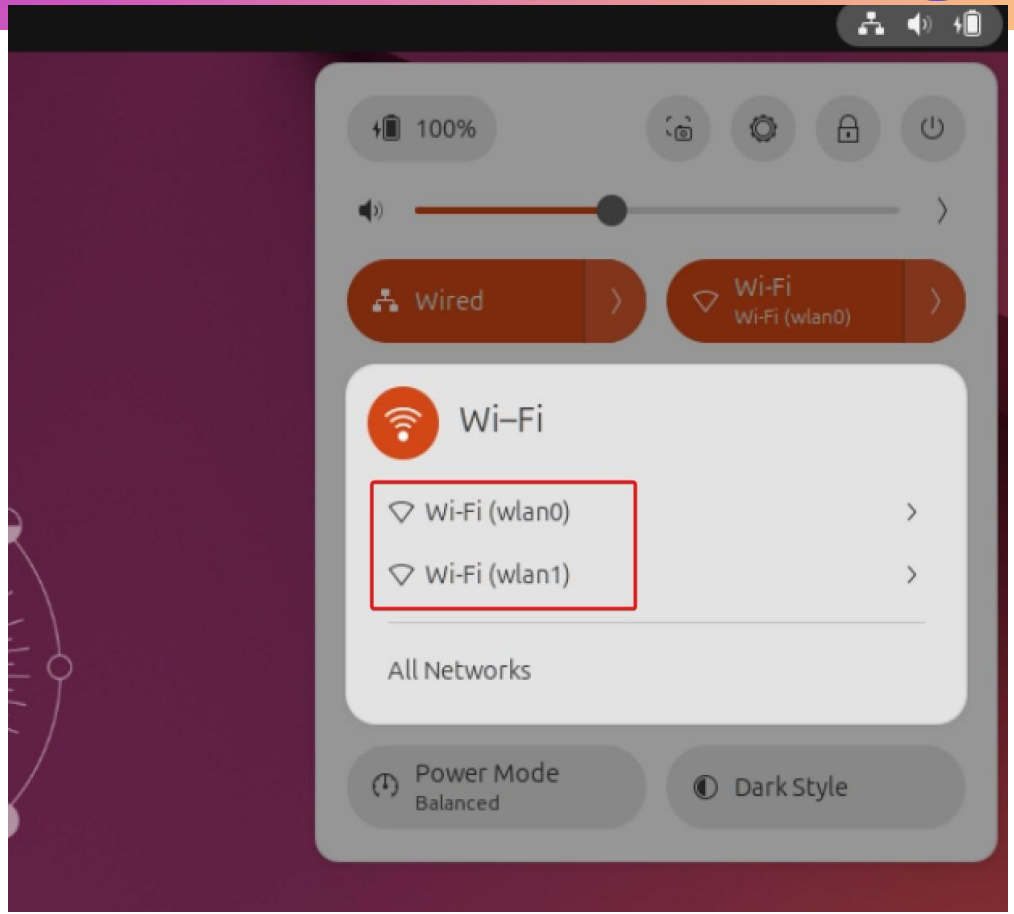
WiFi Interfaces

Install build-essential and linux-modules-extra

Command: sudo nmcli dev set wlan2 managed no

Load the module with a given number of simulated radios:

Command: sudo systemctl restart NetworkManager



Hostapd

open.conf

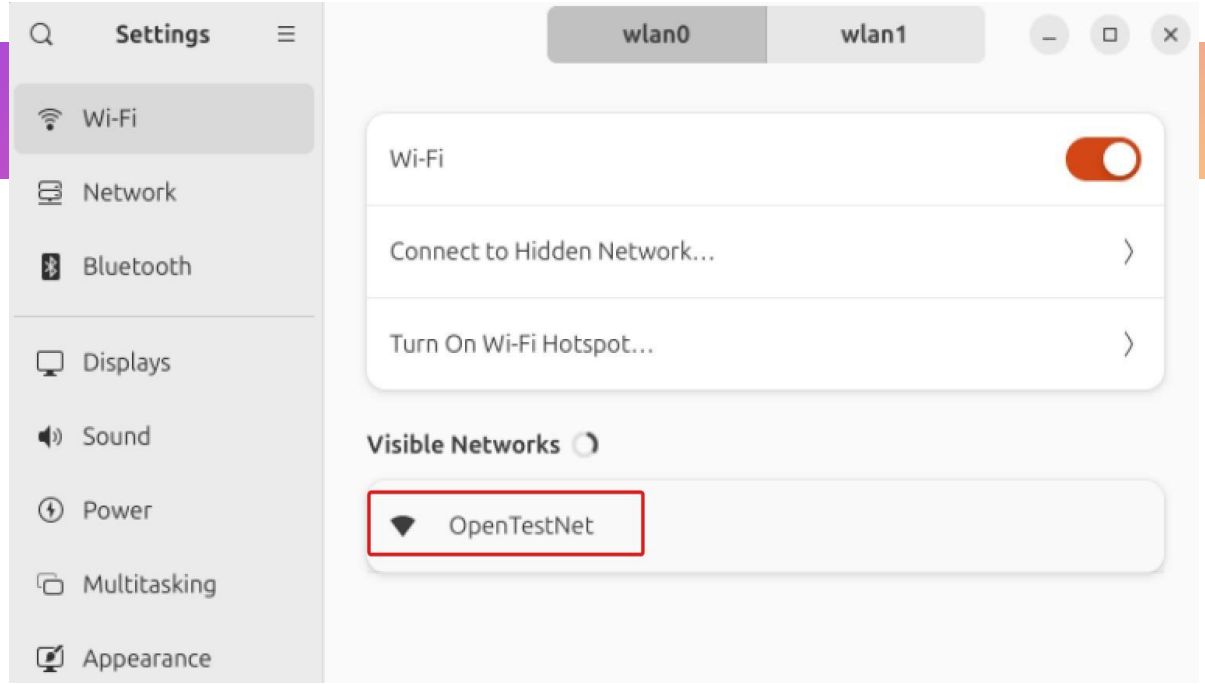
```
interface=wlan1
driver=nl80211
ssid=OpenTestNet
hw_mode=g
channel=6
bssid=02:12:34:56:78:9A
```

Start hostapd

Command: sudo hostapd open.conf

```
osboxes@osboxes:~$
osboxes@osboxes:~$ cat open.conf
interface=wlan2
driver=nl80211
ssid=OpenTestNet
hw_mode=g
channel=6
bssid=02:12:34:56:78:9A
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo hostapd open.conf
wlan2: interface state UNINITIALIZED->ENABLED
wlan2: AP-ENABLED
```

Hostapd




```
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo hostapd open.conf
wlan2: interface state UNINITIALIZED->ENABLED
wlan2: AP-ENABLED
wlan2: STA 02:00:00:00:00:00 IEEE 802.11: authenticated
wlan2: STA 02:00:00:00:00:00 IEEE 802.11: associated (aid 1)
wlan2: AP-STA-CONNECTED 02:00:00:00:00:00
wlan2: STA 02:00:00:00:00:00 RADIUS: starting accounting session CF73A97FAD605977
```

Network Namespace

Namespaces isolate system resources (processes, mounts, IPC, network, etc.) so each set appears to have its own instance.

Create a network namespace
Command: sudo ip netns add wpsns

Setting localhost interface
Command: sudo ip -n wpsns link set lo up

Listing network namespaces
Command: ip netns list

```
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ip netns add wpsns
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ip -n wpsns link set lo up
osboxes@osboxes:~$
osboxes@osboxes:~$ ip netns list
wpsns
osboxes@osboxes:~$
osboxes@osboxes:~$
```

Moving to new Namespace

SquareX

```
osboxes@osboxes:~$ sudo iw phy phy2 set netns name wpsns
osboxes@osboxes:~$
osboxes@osboxes:~$
osboxes@osboxes:~$ iw dev
phy#1
        Interface wlan1
                ifindex 4
                wdev 0x100000001
                addr 02:00:00:00:01:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
phy#0
        Unnamed/non-netdev interface
                wdev 0x3
                addr 42:00:00:00:00:00
                type P2P-device
                txpower 20.00 dBm
        Interface wlan0
                ifindex 3
                wdev 0x1
                addr 02:00:00:00:00:00
                type managed
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
```

Interfaces across namespaces

SquareX

```
osboxes@osboxes:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:a3:ac:aa brd ff:ff:ff:ff:ff:ff
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
4: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
6: hwsim0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ieee802.11/radiotap 12:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
osboxes@osboxes:~$
osboxes@osboxes:~$
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ip netns exec wpsns ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
5: wlan2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 02:12:34:56:78:9a brd ff:ff:ff:ff:ff:ff permaddr 02:00:00:00:02:00
osboxes@osboxes:~$
```

Running ip link command in “wpsns” network namespace
Command: sudo ip netns add wpsns

Starting SSID

Assign IP address to wlan1
Command: sudo ip addr add 192.168.0.1/24 dev wlan1

Bring interface up
Command: sudo ip link set wlan1 up

Start hostapd
Command: sudo hostapd -B open.conf

```
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ip addr add 192.168.0.1/24 dev wlan1
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ip link set wlan1 up
osboxes@osboxes:~$
osboxes@osboxes:~$ nano open.conf
osboxes@osboxes:~$
osboxes@osboxes:~$ cat open.conf
interface=wlan1
driver=nl80211
ssid=OpenTestNet
hw_mode=g
channel=6
bssid=02:12:34:56:78:9A
osboxes@osboxes:~$
```

```
osboxes@osboxes:~$ sudo hostapd -B open.conf
wlan1: interface state UNINITIALIZED->ENABLED
wlan1: AP-ENABLED
osboxes@osboxes:~$
```

Starting SSID

SquareX

```
osboxes@osboxes:~$ iw dev
phy#1
        Interface wlan1
                ifindex 4
                wdev 0x100000001
                addr 02:12:34:56:78:9a
                ssid OpenTestNet
                type AP
                channel 6 (2437 MHz), width: 20 MHz (no HT), center1: 2437 MHz
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       29      0       0       0       0       12068           29
phy#0
        Unnamed/non-netdev interface
                wdev 0x3
                addr 42:00:00:00:00:00
                type P2P-device
                txpower 20.00 dBm
        Interface wlan0
                ifindex 3
                wdev 0x1
                addr 02:00:00:00:00:00
                type managed
                channel 6 (2437 MHz), width: 20 MHz (no HT), center1: 2437 MHz
                txpower 20.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
osboxes@osboxes:~$
```

Starting client

Run bash in “wpsns” network namespace
Command: sudo ip netns exec wpsns bash

Run wpa_supplicant
Command: sudo wpa_supplicant -i wlan2 -c /tmp/openwifi.conf -B

```
osboxes@osboxes:~$ sudo ip netns exec wpsns bash
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# cat /tmp/openwifi.conf
ctrl_interface=/run/wpa_supplicant
network={
    ssid="OpenTestNet"
    key_mgmt=NONE
}
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# sudo ip link set wlan2 up
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# sudo wpa_supplicant -i wlan2 -c /tmp/openwifi.conf -B
Successfully initialized wpa_supplicant
```

Too lazy to script?

SquareX

mininet-wifi.github.io/get-started/

Mininet-WiFi
Emulator for Software-Defined Wireless Networks
https://github.com/intrig-unicamp/mininet-wifi

Emulation Platform for Software-Defined Wireless Networks

Download/Get Started With Mininet-WiFi

Option 1: Mininet-WiFi VM Installation (easy, recommended)

Follow these steps for a VM install:

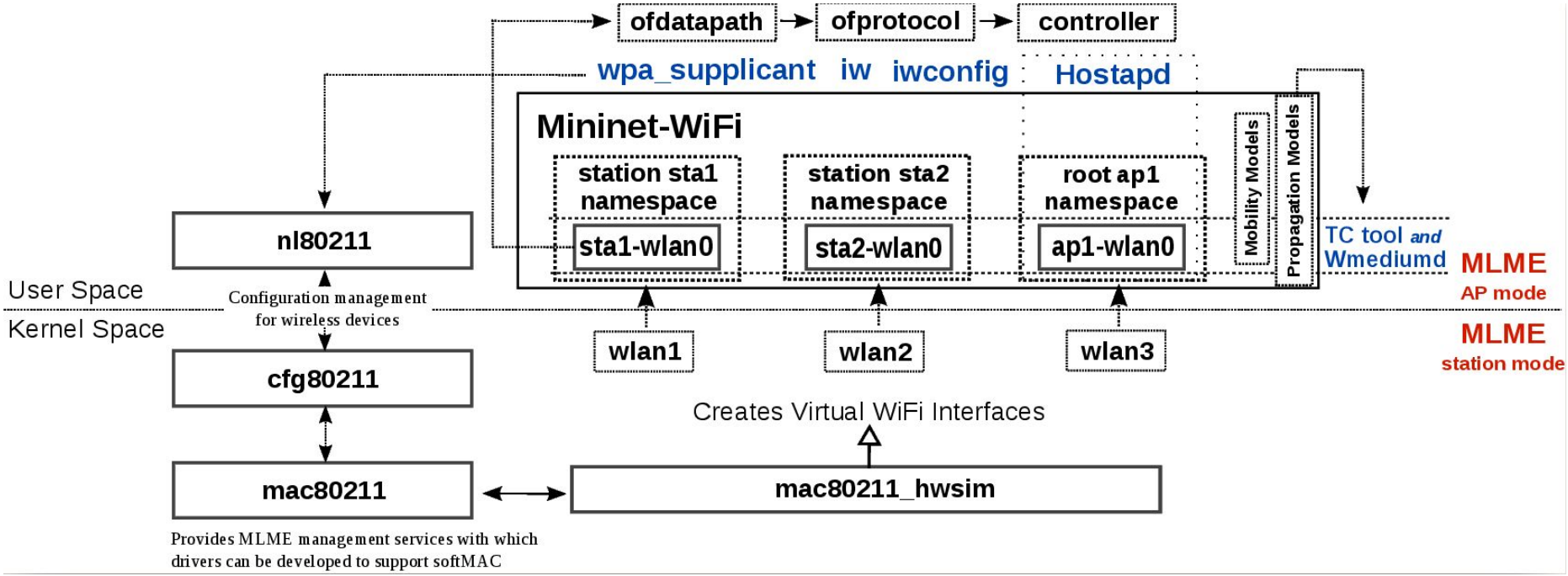
1. Download the Mininet-WiFi VM image
2. Download and install a virtualization system. We recommend VirtualBox (free, GPL) because it is free and works on OS X, Windows, and Linux (though it's slightly slower than VMware in our tests.) You can also use Qemu for any platform, VMware Workstation for Windows or Linux, VMware Fusion for Mac, or KVM (free, GPL) for Linux.
3. Sign up for the mininet-wifi-discuss mailing list. This is the source for Mininet-WiFi support and discussion with the friendly Mininet-WiFi community. ;-)

Option 2: Native Installation from Source

This option works well for local VM, remote EC2, and native installation. It assumes the starting point of a fresh Ubuntu, Debian (or, experimentally, Fedora) installation.

We strongly recommend more recent Ubuntu releases, because they support newer versions of Open vSwitch. (Fedora also supports recent OVS releases)

Mininet-WiFi



Wmediumd

The kernel module mac80211_hwsim uses the same virtual medium for all wireless nodes. This means all nodes are internally in range of each other and they can be discovered in a wireless scan on the virtual interfaces. Mininet-WiFi simulates their position and wireless ranges by assigning stations to other stations or access points and revoking these wireless associations. If wireless interfaces should be isolated from each other (e.g. in adhoc or mesh networks) a solution like wmediumd is required. It uses a kind of a dispatcher to permit or deny the transfer of packets from one interface to another.

Propagation Models

Mininet-WiFi supports the following propagation models:

- Friis Propagation Loss Model,
- Log-Distance Propagation Loss Model (default)
- Log-Normal Shadowing Propagation Loss Model
- International Telecommunication Union (ITU) Propagation Loss Model
- Two-Ray Ground Propagation Loss Model

Mininet-WiFi

SquareX

Containernet

Containernet is a fork of Mininet that allows to use Docker containers as Mininet hosts as well as Mininet-WiFi stations. This enables interesting functionalities to built networking/cloud testbeds. The integration is done by subclassing the original Host/Station classes.

Get started with Containernet and Mininet-WiFi

Follow these below for installing Containernet with Wi-Fi capabilities:

```
~$ sudo apt-get install ansible git aptitude
~$ git clone https://github.com/ramonfontes/containernet.
~$ cd containernet/ansible
~/containernet/ansible$ sudo ansible-playbook -i "localho
~$ cd ..
~/containernet$ sudo python setup.py install
```

Then, you can run containernet_wifi.py

```
~/containernet$ sudo python examples/containernet_wifi.py
```

What else?

Tool	Live Wireshark Capture	Capture 802.11 Frames	Notes
ns-3	❌ (only after sim ends)	✅ Yes, full PHY/MAC simulation	Uses built-in pcap output; no live sniff
CORE	✅ Yes	❌ Usually L2+ only unless you integrate `mac80211_hwsim`	Real kernel interfaces, namespaces
Mininet-WiFi	✅ Yes	✅ With `hwsim`	Can live capture with Wireshark
GNS3	✅ Yes	❌ Unless bridged to real Wi-Fi	Uses virtual/real interfaces

Thank You!