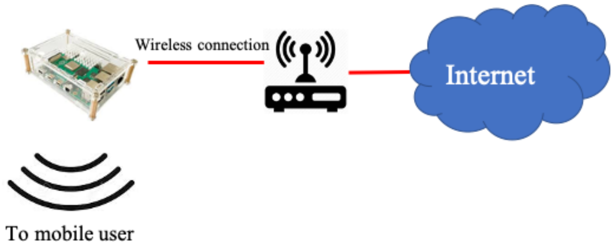
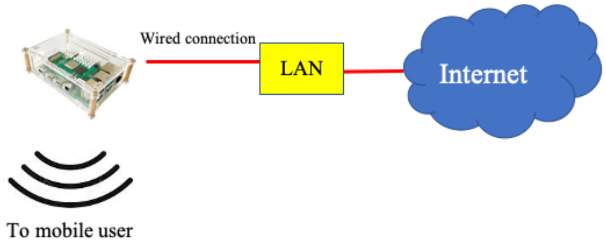
Supplementary Instruction of basic configurations

There are two possible network setups for this lab as shown in Figure 1.



(a) Lab Setup with Wi-Fi



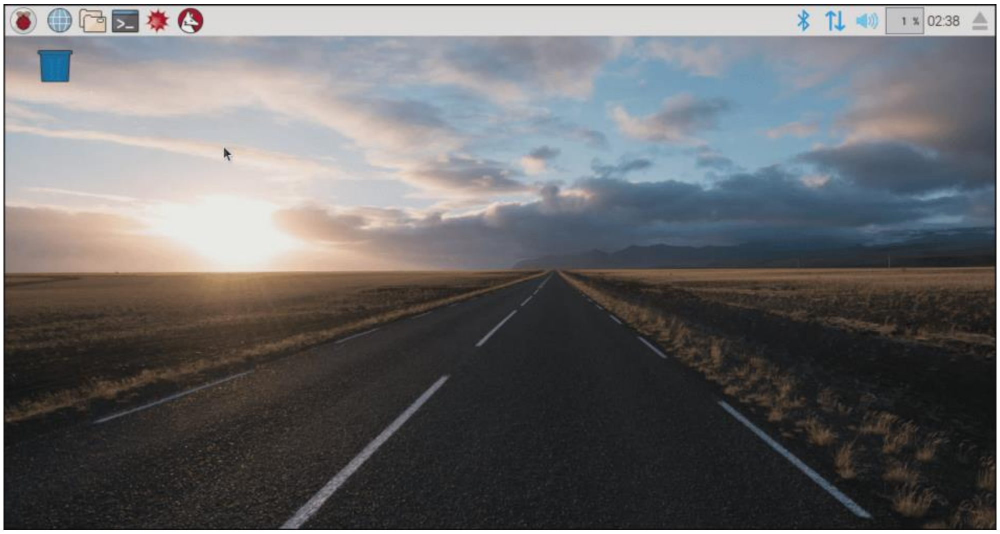
(b) Lab Setup without Wi-Fi

Figure 1. Two possible Wi-Fi Phishing Lab Setups

Network setup in Figure 1(a) is the actual network of Wi-Fi Phishing. However, we will use Figure 1(b) network setup instead for the convenience. Connect Ethernet cable to Ethernet port of Raspberry Pi to get an Internet access.

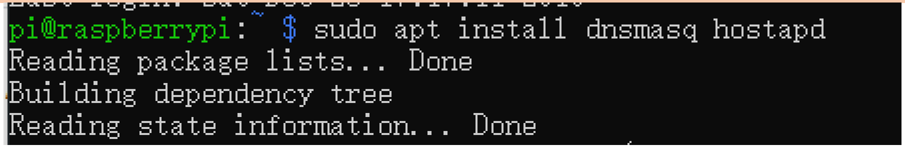
Set up Wireless Access point

You will have to run all commands in the terminal, which also called shell. To open terminal, click on the 4th icon from the left on the top-left corner menu. It should bring up the terminal and you can type and run the command.



To create an access point, we need DNSMasq and HostAPD. Install them using following command.

```
# sudo apt install dnsmasq hostapd
```



```
pi@raspberrypi:~ $ sudo apt install dnsmasq hostapd
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

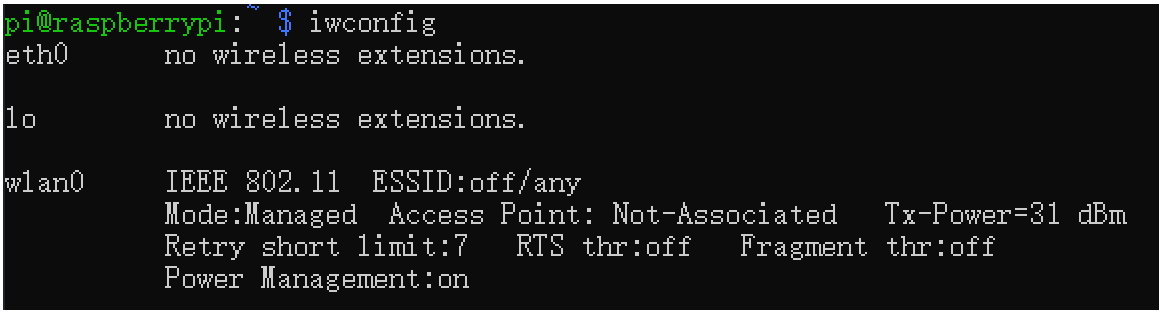
Stop both application for now since the configuration files are not ready yet. Use following commands to stop them

```
# sudo systemctl stop dnsmasq
# sudo systemctl stop hostapd
```

a) Configuring a static IP

Now we are going to assign static IP to the wireless interface or adaptor. First, we have to obtain the name of the wireless interface. Using command “iwconfig”, which shows the information of wireless interface as show below. It is “wlan0” in this output.

```
# iwconfig
```



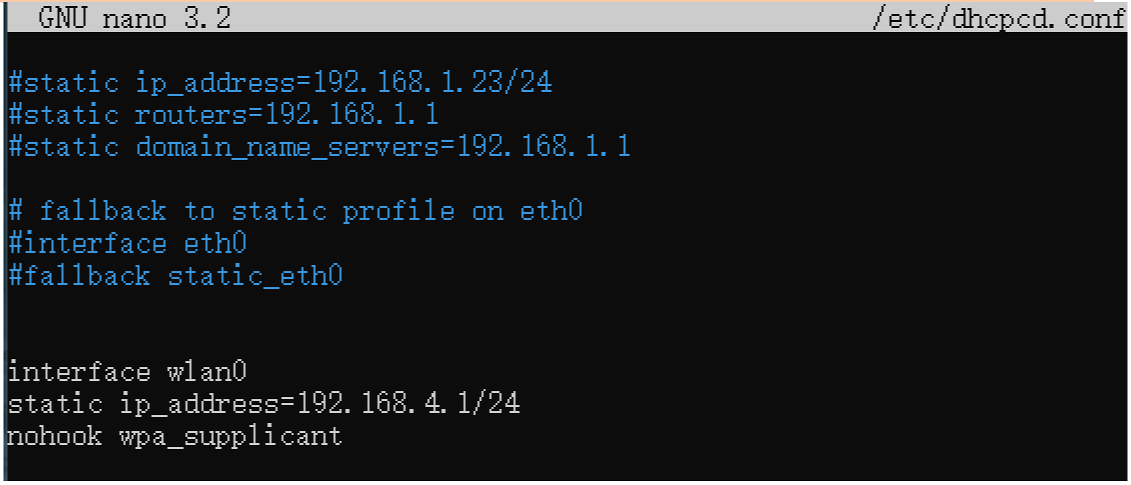
```
pi@raspberrypi:~ $ iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
```

We can assign static IP address at dhcpcd configuration file. Use following command to edit this file.

```
# sudo nano /etc/dhcpcd.conf
```

Go to the end of the file using arrow key “Down” and add following lines.

```
interface wlan0
   static ip_address=192.168.4.1/24
    nohook wpa_supplicant
```



```
  GNU nano 3.2                                        /etc/dhcpcd.conf

#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0


interface wlan0
static ip_address=192.168.4.1/24
nohook wpa_supplicant
```

Use “ctrl + o” to save your setting and exit using “ctrl + x”.

b) Configuring DHCP server (dnsmasq)

The DHCP service is dynamic IP address assignment for clients and it is provided by dnsmasq. The default configuration file contains a lot of unnecessary information and it is much easier to start from scratch. Rename default configuration file and create new one.

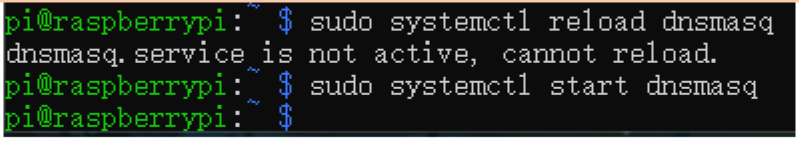
```
# sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
# sudo nano /etc/dnsmasq.conf
```

Type or copy following lines of statement into the dnsmasq configuration file and save it.

```
interface=wlan0      # Use the require wireless interface - usually is wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
```

Now, restart dnsmasq to make updated configuration effective. Use reload command first. It should run if dnsmasq is already running. If it complains that the service is not active, use start command. Example is shown below.

```
# sudo systemctl reload dnsmasq
# sudo systemctl start dnsmasq
```



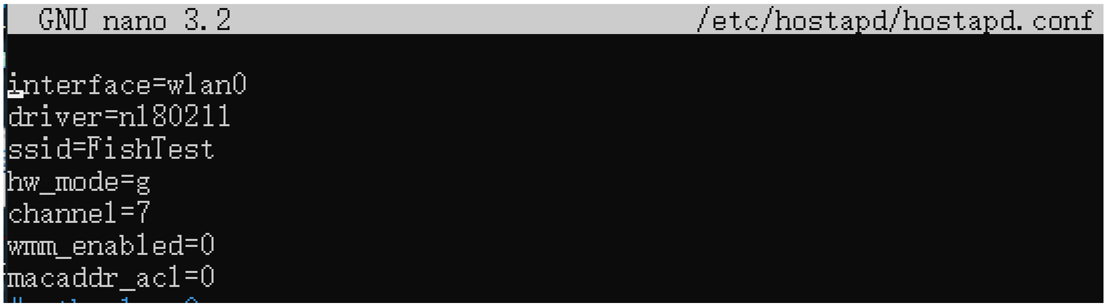
```
pi@raspberrypi:~ $ sudo systemctl reload dnsmasq
dnsmasq.service is not active, cannot reload.
pi@raspberrypi:~ $ sudo systemctl start dnsmasq
pi@raspberrypi:~ $
```

c) Configuring the access point host (hostapd)

Open configuration file using following command.

```
# sudo nano /etc/hostapd/hostapd.conf
```

If there's authentication, please remove it because we do not need it for the lab. Your file should look like below.

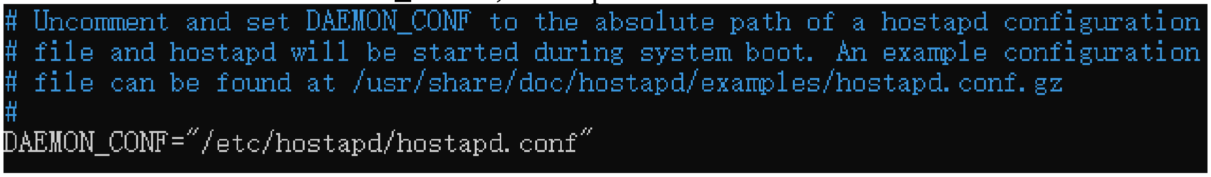


```
  GNU nano 3.2                          /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=FishTest
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
```

We need to let the system know where to find this configuration file. Open "hostapd" file to edit.

```
# sudo nano /etc/default/hostapd
```

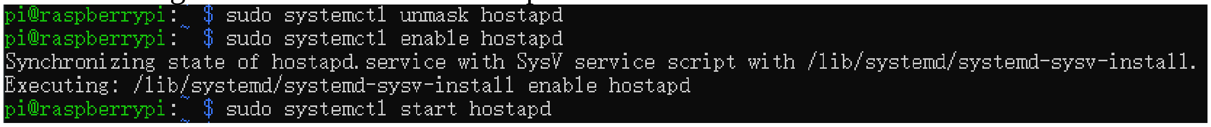
Find the line with #DAEMON_CONF, and replace it with this.



```
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

d) Start the access point

Use following commands to start access point.



```
pi@raspberrypi:~ $ sudo systemctl unmask hostapd
pi@raspberrypi:~ $ sudo systemctl enable hostapd
Synchronizing state of hostapd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable hostapd
pi@raspberrypi:~ $ sudo systemctl start hostapd
```

Now access point is active and you should be able to join the Wi-Fi. However, you cannot access the Internet yet.

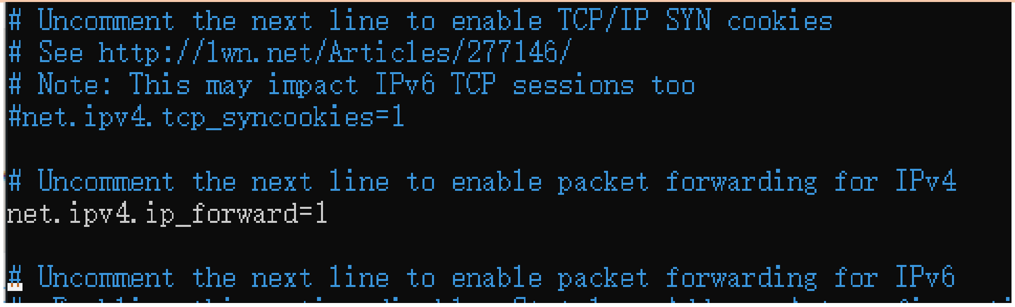
e) Add routing and masquerade

Open "sysctl.conf" file and edit.

```
# sudo nano /etc/sysctl.conf
```

Find and uncomment this line to allow IP forward:

```
net.ipv4.ip_forward=1
```



```
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
```

Add a masquerade for outbound traffic on eth0, where “eth0” is Ethernet port. Remember we use Ethernet port instead of wireless for Internet access for this lab.

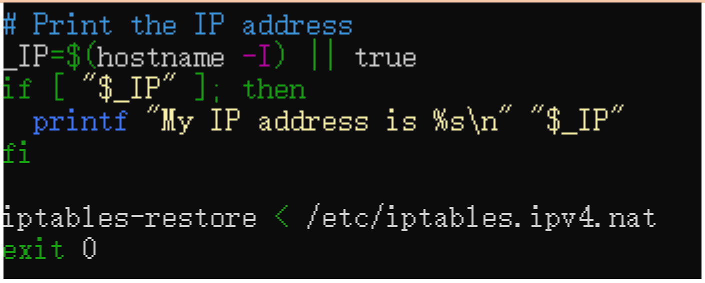
```
# sudo iptables -t nat -A  POSTROUTING -o eth0 -j MASQUERADE
```

The configuration is not permanent. It will be cleared when system is reboot. To make it permanent, you will need to save the iptables rules to your bash file.

```
# sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
# sudo nano /etc/rc.local
```

Once “rc.local” file is open, add following command just above "exit 0" to make it effective on system boot.

```
iptables-restore < /etc/iptables.ipv4.nat
```



```
# Print the IP address
_IP=$(hostname -I) || true
if [ “$_IP” ]; then
  printf “My IP address is %s\n” “$_IP”
fi

iptables-restore < /etc/iptables.ipv4.nat
exit 0
```

Reboot and restart the service using following command.

```
# sudo service dnsmasq restart
```

Now you should able to connect to the wire and browse internet.