



MASTER YOUR PI (EBOOK)

BEGINNER COURSE

MASTER PYTHON (EBOOK)

START HERE

RECOMMENDED PRODUCTS

How to use Raspberry Pi as a Wireless Router with Firewall?



now it's ok, I finally found how to do it, and I'll share this with you

The Raspberry Pi only have one Ethernet socket, so it's not possible to create a firewall with two RJ45 interfaces.

But there is a Wi-Fi interface that can be used for one side (LAN for example).

One way to build a firewall is to use the hostapd and iptables services.



If you are looking to quickly progress on Raspberry Pi, [you can check out my e-book here](#). It's a 30-day challenge, where you learn one new thing every day until you become a Raspberry Pi expert. The first third of the book teaches you the basics, but the following chapters include projects you can try on your own.



Linux Device Drivers Part 1.1 - Setup Ubuntu and Raspberry Pi

Begin with the end in mind

Download the Pi Glossary!

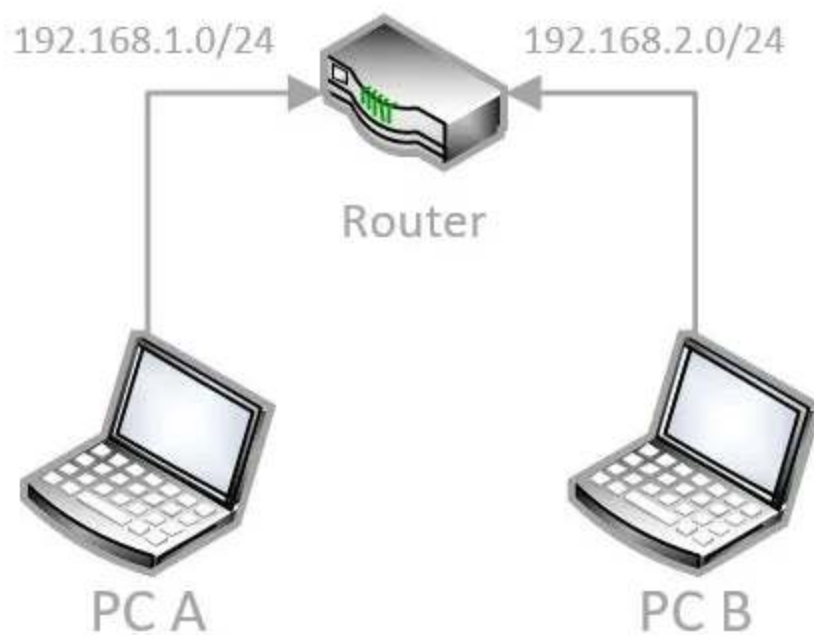
If you are lost in all these new words and abbreviations, request my free Raspberry Pi glossary here (PDF format)!

[DOWNLOAD IT NOW](#)

Router

A router is a network device that connects two networks together.





In this schema, we have two different networks, connected with a router: 1.0 and 2.0. If the router is well configured, it allows A and B to see each other, while on a different network.

The Raspberry Pi have only one Ethernet card, but we can use the Wi-Fi card to create a second network.

So, the router part in this tutorial will allow us to connect the Wi-Fi network to the Ethernet network.

For instance, in the previous example, we can configure that A can ping B, but not access the HTTP server on B.

I'll use a software called "iptables" for this, but you can use any other firewall software if you prefer.

My goal

If you use your Raspberry Pi at home, you probably don't need to connect two networks. But my goal is to create a new wireless access point with a firewall and other cool software to monitor the network and filter some kind of traffic.

Here is my current network:

And I want to turn it like this:

So, here are the steps you need to follow to do the same:

- Install your Raspberry Pi on the network
- Enable Wi-Fi access point with a different network subnet



I'll explain you each step in detail.

Let's move to the first step of this process 😊

Install your Raspberry Pi

The first thing to do is to install your Raspberry Pi on the network:

- Install Raspberry Pi OS by following [this tutorial](#)
You don't need the Desktop version, except if you want to use the Raspberry Pi for other things too
- Plug the Raspberry Pi on the network with an RJ45 cable
We'll use the Wi-Fi later, so you need to let him available
- A static IP address is not mandatory but it can help
You can check the end of [this article](#) to know how to configure it
- Once done, update your system by doing:

```
sudo apt update
sudo apt upgrade
sudo reboot
```

- Enable SSH in raspi-config > Interfacing options

```
sudo raspi-config
```



That's it, you have done the preparation step, we can start with the router installation.

Are you a bit lost in the Linux command line? [Check this article first](#), for the most important commands to remember, and a free downloadable cheat sheet so you can have the commands at your fingertips.

Install a wireless router

Raspberry Pi Bootcamp

Sale: 10% off today.

Take it to the next level.

I'm here to help you get started on Raspberry Pi.

Learn all the skills you need in the correct order.

WATCH NOW!

On Stretch (Debian 9), a script was available to do everything automatically, but it hadn't been updated and doesn't work anymore.

So, we'll do it manually, it's not so complex.

Configure the Wi-Fi country

If you are using a fresh new Raspberry Pi OS, you need to set a Wi-Fi country first.



```
sudo raspi-config
```

- Go to Localisation Options > Change WLAN country
- Select your country in the list
- Confirm and exit

Install the services

We'll mainly use two new services on our router:

- Hostapd: to create the wireless access point
- DNSmasq: to forward the DNS requests to another DNS server

Start by installing the required packages:

```
sudo apt install hostapd dnsmasq
```

That's it, we can now move to the configuration part.

Configure Hostapd


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

- Paste these lines (the file is probably empty):

```
interface=wlan0
driver=nl80211
ssid=RaspberryTips-Wifi
hw_mode=g
channel=6
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=<YOURPASSWORD>
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Don't forget to set a passphrase, and you can edit the line you want (for example to use another channel, security level or SSID)

- Save & exit (CTRL+O, CTRL+X)

Hostapd won't start automatically on boot, there are two changes to do to enable this:

- Edit the default configuration file:

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

- Add this line at the end:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

- Save & Exit
- Then, enable the service with:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
```



The next step is to configure DNSmasq:

- Open the configuration file:

```
sudo nano /etc/dnsmasq.conf
```

- Paste these lines at the end:

```
interface=wlan0  
bind-dynamic  
domain-needed  
bogus-priv  
dhcp-range=192.168.42.100,192.168.42.200,255.255.255.0,12h
```

Nothing to change here, except if you want to change the subnet.

- Save & exit (CTRL+O, CTRL+X)

Configure the DHCP server

The last configuration file to change is the DHCP server, set it on the same subnet:

- Open the configuration file:

```
sudo nano /etc/dhcpd.conf
```

- Scroll to the end of the file and paste these lines:



```
static routers=192.168.42.1
```

- Save & exit

We are almost ready for the first test.

Enable IP forwarding

If you have several network cards, the default behavior on Linux is to isolate them.

In our case, we want to enable the communication between the LAN and the Wi-Fi.

So, we need to change this:

- Open this file:

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

- Find this line (first page):

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

- And uncomment it:

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

- Save & exit



Note: I had to do this two times on my two tests because I was not getting an IP address on the first reboot. No idea why...

Once this is complete you should be able to see your Raspberry Pi access point in the networks list

In your Wi-Fi networks list, you should see something like this:

You can connect to it and check that everything is working as expected.

You should get an IP in the 192.168.42.0/24 subnet, the script created this network for you.

You'll not get any Internet connection for now, as we need to configure the firewall to allow the Internet traffic.

Understand the firewall concepts

I'll start with an introduction on the theory about firewall configuration.



The role of a firewall is to block or allow access from a specific IP to another.

And often we also use a port to set the exact permission.

Ex: We deny port 22 to everyone, except computer A that can access computer B with port 22.

Black or White

In a firewall configuration, you have the choice between two default rules:

- Blacklist: Allow all except ...
- Whitelist: Deny all except ...

Depending on what you want to do with your Raspberry Pi router, it's your choice to take the one you want.

The first option is probably ok if you are using it at home. You may want to block only certain things like the torrent protocol or specific IPs address.

But at work it's rather the second. The good practice is to block everything except what is allowed.

In, Out and Forward



- Input: Network packets coming into the firewall
- Output: Packets going out from the firewall
- Forward: Packets going through the firewall

On a hosted web server, you can block anything in input except HTTP and HTTPS.

But in output it's not a big deal what your server is doing on the Internet.

In our case, we'll mainly use FORWARD rules only as there is nothing on the Raspberry Pi.

There is no need to protect it more than that.

Configure the firewall service

If you want, you can add a firewall in your router to filter the traffic.

Maybe at home it's not mandatory, but for a company or a public place you need this.

IPTables

There are several firewall packages available on Raspberry Pi OS: iptables or ufw for example.

There is also OpenWRT, a Raspberry Pi compatible distribution, to create a router firewall.

Master your Raspberry Pi in 30 days

Sale: 10% off today.

Download the eBook.

Uncover the secrets of the Raspberry Pi in a 30 days challenge.

Learn useful Linux skills and practice multiples projects.

GET IT NOW!

In this post, I'll use iptables, the most used.

It's already installed on your Raspberry Pi, so there's nothing else to do.



To do this, use the command:

```
sudo iptables -L
```

You should get something like that:

We find the input, output and forward parts I talked about previously.

For each we see that the default policy is ACCEPT, so everything allowed except what we add (blacklist mode).

You can use this command later to check if the new rules you add correspond to what you want.

Enable Internet forwarding

Before going into more details, we'll just add some basic rules to allow the Internet traffic:

- Type these 3 commands:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

- You don't need to understand them right now, I will explain everything in the next parts. But once done, you should have an Internet access working on your access point.



Add a FORWARD rule

We'll use the iptables command to add new rules in the firewall.

Every network is different, so every firewall rules table is different.

I'll start by an example, and then I'll give you the whole syntax to add specific rules in your environment.

- Start by resetting the iptables configuration

```
sudo iptables -F
```

The order is important in the rules list, and as the router already accept everything in the forward section, we can't block specific connections.

- Add a DROP rule

```
sudo iptables -A FORWARD -p tcp --dport 80 -j DROP
```

This command specifies that:

- we add a new rule (-A)
- in the forward section (FORWARD)
- for the tcp protocol (-p tcp)
- for the HTTP port (-dport 80)
- and the action is to DROP everything (timeout connection)

- Use the 3 commands I gave you in the previous part to allow the other Internet traffic
- After that, you should not be able to access a website like [this one](#) for example (but sites in HTTPS still work)



Or you can use the same command with the -D operator instead of -A.

```
sudo iptables -D FORWARD -p tcp --dport 80 -j DROP
```

This command allows you to delete a specific rule and not all like with the -F.

Iptables command syntax

As you should already understand, you can now use the same command template to create the firewall rules you need.

The command template is:

```
iptables -<operation> <direction> -p <protocol> --dport <port> -j <action>
```

- operation:
 - -F: flush, remove all rules, it requires no other parameters
 - -A: append, add a new rule
 - -D: delete, remove an existing rule
- direction: INPUT, OUTPUT or FORWARD (see the previous section)
- protocol: mainly TCP or UDP
- port: the port number you want to create your rule for
You can find a [list of common ports here](#)
- action: Define the choice to make for the corresponding traffic
 - ACCEPT: Allow access (while in whitelist mode)
 - REJECT: Deny access and tell the sender it's not allowed
 - DROP: Deny access but don't tell the sender

This is the short introduction to what you'll mainly use.

If you need further information, use "man iptables" or check [this page](#) for all parameters.



If you are in a stricter environment, switch to whitelist mode.

For example, if you are creating a free Wi-Fi in a hotel or other business, you probably want to allow only a few ports (like web and mails).

To do this, you need to create a list of all ports you want to allow.

If you do all the commands manually, you'll lose access after the first one 🌑😊

So, the easiest way is to create a script that run all commands at once.

Create the firewall script

- Create a new file with nano

```
sudo nano /usr/local/bin/firewall.sh
```

- Paste these lines inside

```
#!/bin/sh
#Clear all rules
iptables -F

#Whitelist mode
iptables -P INPUT ACCEPT
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

#Allow PING for everyone
iptables -A FORWARD -p icmp -j ACCEPT

#Allow HTTP/HTTPS for WiFi clients
iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -p tcp --dport 443 -j ACCEPT

#Allow POP/IMAP/SMTP for WiFi clients
iptables -A FORWARD -p tcp --dport 25 -j ACCEPT
iptables -A FORWARD -p tcp --dport 110 -j ACCEPT
iptables -A FORWARD -p tcp --dport 993 -j ACCEPT

#Allow PING for WiFi clients
```



```
iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j  
ACCEPT
```

- You can adapt these lines to your needs

This is just an example

I don't drop on INPUT and OUTPUT in this sample script, but you can do it if you want a better control on your network usage

- Save and exit (CTRL+O, CTRL+X)
- Add execution right to this script

```
sudo chmod +x firewall.sh
```

- Run it

```
sudo /usr/local/bin/firewall.sh
```

- Check if everything works fine
If something goes wrong, reboot the Raspberry Pi to recover all access
Then find what you miss in your script

When it's ok, you can add it in the init tab to start it on boot

Make your configuration persistent

The GitHub script we installed before uses the file `/etc/iptables.ipv4.nat` to save the configuration

So once it's working, you can save your current configuration inside :

```
sudo iptables-save > /etc/iptables.ipv4.nat
```

This will load the configuration file on load and apply directly the changes

Monitor the network



Now that the router is working (with a firewall or not), we can add other packages to improve the Raspberry Pi capabilities

In this part, I suggest adding a web interface to monitor what happens on the Raspberry Pi (and on the network)

What's this tool?

The tool I chose is Webmin

It's a web interface, easy to install, that shows you all the current configuration, and several statistics and graphs about the system usage

You can even change the configuration from this interface

If you know other ones tell me in the comments

It's a tool that exists for a long time, but I don't know a more recent one to do this

Webmin Installation

- Check the latest Webmin version from [this page](#) (Source file)
- Download the file like this:

```
wget https://prdownloads.sourceforge.net/webadmin/webmin-1.941.tar.gz
```

- Extract the archive and move to the new folder

```
tar -zxvf webmin-1.941.tar.gz cd webmin-1.941
```

- Run the setup



Set a password and choose if you want to use SSL or not

Wait a few seconds for the installation to finish

- The tool is now available at <http://<IP>:10000>
Log in with the user name and password you just created

Webmin interface

I let you discover the web interface and browse through the menu

There are A LOT of tools inside, we don't need all of this

We'll mainly use those in the "Networking" part

For example, you can enable bandwidth monitoring or manage your firewall configuration from here

Master Python on Raspberry Pi

Sale: 10% off today.

Get the eBook.

Do more with your Raspberry Pi, learn the useful concepts and take the shortcuts.

You miss half of the fun of using a Raspberry Pi if you don't know anything about Python.



It's the next part

Add a proxy and a web filter

What's a proxy?

A proxy has three main roles:

- Creating a cache of all Internet pages, to increase web browsing speed
- Log all pages, to generate reports (top domain, top traffic, ...)
- And we can add a website blocker, to deny access to some kinds of content

To do this, we'll install Squid (the proxy) and SquidGuard (the filter) on the Raspberry Pi

Squid installation

- Squid is available in the repository. Install it with:

```
sudo apt install squid
```

- Backup the configuration file:

```
cd /etc/squid
```

```
sudo mv squid.conf squid.conf.old
```

- Switch to the root user a few seconds and remove all the comments quickly

```
sudo su  
cat squid.conf.old | egrep -v -e '^[[:blank:]]*#' | grep "\S" > squid.conf
```

```
exit
```

Then edit the configuration file:

```
sudo nano squid.conf
```



Squid works only on HTTP for the Wi-Fi network (42.X)

- Restart Squid to apply changes

```
sudo systemctl restart squid
```

Once Squid is configured, you have two choices to use it:

- Configure your web browser to use the Raspberry Pi as the HTTP Proxy
It depends on your browser, but you should find this option in Options > Network settings
- Configure iptables to redirect automatically all HTTP traffic to squid
It should be something like this:

```
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 80 -j DNAT --to  
192.168.42.1:3128  
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-  
port 3128
```

Then check that HTTP websites are working fine

You can see all the websites visited in the Squid log file :

```
sudo tail -f /var/log/squid/access.log
```

Webmin add-on

There is a Webmin add-on called “calamaris” you can install to monitor the proxy efficiency

Install it with apt:

```
sudo apt install calamaris
```

And then go back to webmin, in the unused modules to find new tools for squid monitoring

SquidGuard installation



```
sudo apt install squidguard
```

- Download a list of websites by category

```
wget http://squidguard.mesd.k12.or.us/blacklists.tgz
```

- Extract files from the archive

```
tar -zxvf blacklists.tgz
```

While extracting, you'll see some blacklist categories appear on your screen

You can choose one and use it for the SquidGuard configuration later

You can open the files to get some websites examples

- Move files to the SquidGuard folder

```
sudo mv blacklists /var/lib/squidguard/db
```

- Archive the original SquidGuard configuration file

```
cd /etc/squidguard  
mv squidGuard.conf squidGuard.conf.old
```

- Create a new configuration file

```
sudo nano squidGuard.conf
```

- Paste these lines

```
dbhome /var/lib/squidguard/db logdir /var/log/squidguard dest violence {  
domainlist blacklists/violence/domains  
urllist blacklists/violence/urls  
log violenceaccess  
}  
acl {  
default  
{  
pass !violence  
redirect http://localhost/block.html  
} }
```


- Build the SquidGuard database

```
sudo squidGuard -C all -d
```

This can take a long time. If it's too long, remove files you don't need from the blacklists folder

Use a screen if you don't stay in front of your computer (screen -S build)

So if the SSH connection with your computer is lost, this will not stop the build process

- Restart Squid to apply changes

```
sudo service squid restart
```

It should be ok now, try to access a URL from the domain list and check that you are blocked by squidGuard

Related questions

Is it possible to add an Ad Blocker brick in this router? Yes, you can use Pi-Hole to do this, it's easy to install and you just need to set the Raspberry Pi as your DNS server (manually or in the DHCP configuration file, see Firewall > DNS issues for more information)

Is it possible to use a Raspberry to build a "full Ethernet" router? Yes, you can add an Ethernet hat to your Raspberry Pi like this expansion Hat ([more details on Amazon](#)). It's perfect for a firewall.

Thanks to [all my Patrons](#) for supporting me, helping me to keep doing this and producing even more tutorials for everyone! Thank you to everyone supporting me there.

You may also like:

- [25 awesome Raspberry Pi project ideas at home](#)
- [15 best operating systems for Raspberry Pi \(with pictures\)](#)
- [My book: Master your Raspberry Pi in 30 days](#)



Get My Cheat Sheet!

Grab your free PDF file with all the commands you need to know on Raspberry Pi!

[DOWNLOAD IT NOW](#)

That's it, you should now have better knowledge on how to build a complete firewall router with proxy on a Raspberry Pi

I hope it's working for you.

It took me a lot of time to write this post with many tests I didn't include here, but you have the most important things, with the best tools

If you have any issues, ask your question in the comments, we'll try to help you

Also, these tools are basically Linux stuff and you can find a lot of help on the Internet to go further

I give you all the documentation links here if needed:

- [RPI Wifi router](#)
- [Iptables](#)
- [Squid](#)
- [SquidGuard](#)
- [Pi Hole](#)

Raspberry Pi Resources



[Watch the Raspberry Pi Bootcamp course now.](#)

Master your Raspberry Pi in 30 days

Don't want the basic stuff only? If you are looking for the best tips to become an expert on Raspberry Pi, this book is for you. Learn useful Linux skills and practice multiple projects with step-by-step guides.

[Download the e-book.](#)

VIP Community

If you just want to hang out with me and show your support, you can also join the Patreon community. I share behind-the-scenes content there and give you early access to my content. You'll also get a shoutout when you join.

[More details here.](#)

Need help building something with Python?

Create, understand and improve any Python script for your Raspberry Pi.

Learn the essentials, step-by-step, without losing time understanding useless concepts.

[Get the e-book now.](#)

You can also find all my recommendations for tools and hardware [on this page](#).

This tutorial doesn't work anymore? [Report the issue here](#), so that I can update it!

0

Article Rating

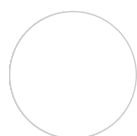


Patrick Fromaget

I'm the lead author and owner of RaspberryTips.com. My goal is to help you with your Raspberry Pi problems using detailed guides and tutorials. In real life, I'm a Linux system administrator with a web developer experience.

✉ Subscribe ▼

Login



Join the discussion



27 COMMENTS



Oldest ▼



Peter Stuart

🕒 3 years ago

Hi,
interesting way to use the RPi. Just wondering what the config would be using a USB to eth adapter instead of Wifi.
best regards





Monje

🕒 2 years ago

+1 to peter.

Also, is it possible to add a torrent client to the mix?



0



➡ Reply



Patrick Fromaget

Author

🔗 Reply to [Monje](#)

🕒 2 years ago

Hello,

I think it would only change the interface name, I can test it if you want

And yes, you can absolutely add a torrent client on this installation

Patrick



0



➡ Reply



Umur

🕒 2 years ago

Hello nice document , i have a question which version of raspi should i use? I have raspberry pi one B version and 3 only... Should i buy raspi 4 models? Thanks alot alot...



0



➡ Reply



Patrick Fromaget

Author

🔗 Reply to [Umur](#)

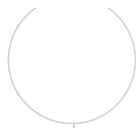
🕒 2 years ago

Hi Umur,

I've done this tutorial with a Raspberry Pi 3B+ so it's ok

If the question is general, the Raspberry Pi 4 only add extra performances, so it depends on what you want to do





James Fargo

🕒 2 years ago

Hi, I would only be interested in the raspberry router, leaving aside the rest that seems quite complicated to me, beginner. The raspberry router clearly coexists with a modem. Your site is really useful, good!



0



Reply



Patrick Fromaget

Author

🔗 Reply to [James Fargo](#)

🕒 2 years ago

Thanks!



0



Reply



James Fargo

🕒 2 years ago

Hi, I followed the instructions for the Wireless router installation.
But something went wrong, I probably did something wrong.
Fact is, the wireless raspberry doesn't work.
Could I get a little help?
Thank you in advance.



0



Reply



Patrick Fromaget

Author

🔗 Reply to [James Fargo](#)

🕒 2 years ago

Hello James,

Did you get an error?

Which part are you stuck in?

Patrick





Just JJ

🕒 2 years ago

Hi Patrick,

Thanks for the guide. I think with the latest rasp pi – hostapd.conf is masked. It was in my case. So, unmask, rerun the config and all good now

@James.

This might assist.

<https://raspberrypi.stackexchange.com/questions/95916/why-is-hostapd-masked-after-installation>



0



Reply



Patrick Fromaget

Author



Reply to [Just JJ](#)

🕒 2 years ago

Thanks for the information 😊



0



Reply



Kannan

🕒 2 years ago

Hi,

Nice Project

i have few questions

1. i am using ssh to login the pi with static IP so will i be able to still do the above wireless router part?

2. is there a way to implement IDS with alert notifications (email or sms)



0



Reply



Patrick Fromaget

Author



Reply to [Kannan](#)

🕒 2 years ago



have to use the IP on the Wi-Fi network)

2. I think that Snort would work on this installation, but I didn't try it

Patrick

+ 0 - ➔ Reply



Kannan

➔ Reply to [Patrick Fromaget](#)

🕒 2 years ago

Hi Patrick,

Thank you so much for the information

+ 0 - ➔ Reply



Hanif Rizal

🕒 2 years ago

Hi Patrick,

what a great tutorial! I would love to try this as soon as i can!

Can this be able to work with Pi-Hole?

+ 0 - ➔ Reply



Mohammad

🕒 2 years ago

Hi

Nice project

How can i force user to brows a local home page imidiatly after connecting to wifi?

Thanks,

+ 0 - ➔ Reply



Patrick Fromaget

Author

➔ Reply to [Mohammad](#)

🕒 2 years ago



Even if I never tested it on Raspberry Pi, it should be possible

You need to install a Radius server for the authentication

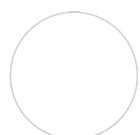
You will easily find tutorial for Linux

Nodogsplash seems a good solution: <https://github.com/nodogsplash/nodogsplash>

Let me know if you find a solution

And if others are interested, I could probably try it and share how to do this

+ 0 - ➔ Reply



Mohammad

🕒 2 years ago

Thanks Patrick,I'll share with you the results.

+ 0 - ➔ Reply



Patrick Fromaget

Author

🕒 2 years ago

Hi,

Just updated the first part as there is some changes on Buster with Hostapd, the script I gave previously is not updated anymore

So, I added the manual configuration

Let me know if you have any issue with it

Patrick

+ 0 - ➔ Reply



Jaime Donado

🕒 2 years ago

hi great tutorial i have a question, i config the acces point and install the webmin

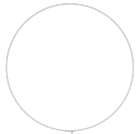


services in the same raspberry pi?

I searched for solutions on the internet but I can't find anything.

thanks you.

+ 0 - Reply



Francois le Roux

🕒 1 year ago

Excellent tutorial...

To make this work for Ethernet replace wlan0 with eth1 (confirm second adaptor name assigned) to adaptor in all the config settings.

i found when rebooting Internet forwarding had to be enabled again.

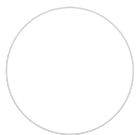
```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state  
RELATED,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

With my limited knowledge wonder how. Probably just creating a script.

+ 0 - Reply

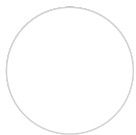


Dave

🕒 1 year ago

Can I run just the firewall?

+ 0 - Reply



Gerard

🕒 1 year ago

Hello! Awesome.

One question,

When I try the command "sudo iptables -A FORWARD -i eth0 -o wlan0 -m state -



Why this?

Also, when I reboot my raspberry, Internet forwarding had to be enabled again.

+ 0 - Reply



Patrick Fromaget

Author

🔗 Reply to Gerard

🕒 1 year ago

Replace - by --

+ 0 - Reply



alex

🕒 1 year ago

hola,

i've followed step by step the tutorial, yet i'm not able to see any access point related to the raspberry.

i've cheked if hostapd status and you can see there is a error.

- hostapd.service – Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP Authenticator

Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset: enabled)

Active: activating (auto-restart) (Result: exit-code) since Thu 2021-03-18 20:03:16 CET; 1s ago

Process: 1533 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B \$DAEMON_OPTS \${DAEMON_CONF} (code=exited, status=1/FAILURE)

root@raspberrypi:/home/pi# sudo service hostapd status -l

- hostapd.service – Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP Authenticator

Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset: enabled)

Active: activating (auto-restart) (Result: exit-code) since Thu 2021-03-18 20:03:25 CET; 830ms ago



- hostapd.service – Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP Authenticator

Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset: enabled)

Active: activating (auto-restart) (Result: exit-code) since Thu 2021-03-18 20:03:32 CET; 765ms ago

Process: 1553 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B \$DAEMON_OPTS \${DAEMON_CONF} (code=exited, status=1/FAILURE)

Mar 18 20:03:32 raspberrypi systemd[1]: Failed to start Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP Authenticator.

i've looked in internet, apparently a solution is to go back to a early version of hostapd or, add a line at :

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

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

which it's already implemented in your tutorial.

A little help would be welcome.

Gracias.

+ 0 - Reply



Patrick Fromaget Author

Reply to alex

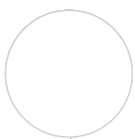
1 year ago

Can you check the log files to get more details about the exact error?

It would be difficult to help without more information.

Thanks

+ 0 - Reply



ay

1 year ago



-bash: /etc/iptables.ipv4.nat: Permission denied
For does who have the same problem this solution:
sudo bash -c "iptables-save > /etc/iptables/rules.v4"
has worked for me.



0



Reply

Recent Posts

5 Easy Ways To Reset A Forgotten Password On Raspberry Pi

Forgetting your password is never a good experience, especially when it's the main password, the one to unlock your operating system. But no worries, there are different ways to recover it (or rather...

CONTINUE READING



3 Easy Ways To Install Discord On Raspberry Pi

Discord is an app that has been trending in the last few years, especially since the 2020 pandemic, with everyone at home looking for communication tools. I started using it maybe 5 years ago, and it...

CONTINUE READING



report this ad

Welcome



Hi, I'm Patrick. I am a Linux system administrator, and I am passionate about the Raspberry Pi and all projects on this topic.

I created this site to share with you what I learned about it.

ABOUT PAGE



[Home](#)[Master your Raspberry Pi \(eBook\)](#)[Beginner Course](#)[Master Python on Raspberry Pi](#)[Resources](#)[Write for Us](#)[Contact](#)[Affiliate Program](#)[Privacy Policy](#)[About Me](#)[About RaspberryTips](#)[Terms and Conditions](#)[Version Française](#)

COPYRIGHT © 2022 RASPBERRYTIPS. ALL RIGHTS RESERVED.

[Become a Patron!](#)

This site is owned and operated by Patrick Fromaget. RaspberryTips.com is a participant in the Amazon Services LLC Associates Program, an affiliate advertising program designed to provide a means for sites to earn advertising fees by advertising and linking to Amazon.com. This site also participates in other affiliate programs and is compensated for referring traffic and business to these companies.

Raspberry Pi is a trademark of the Raspberry Pi Foundation

