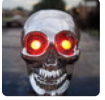




## Share WiFi With Ethernet Port on a Raspberry Pi



by WaltW

Do you have an old laser printer or scanner that still works great but isn't wifi compatible? Or maybe you want to connect an external hard drive as a backup device on your network and you've run out of ethernet ports on your home router. This instructable will help you create a bridge from the wifi connection to the ethernet port on a Raspberry Pi.

I needed a way to connect an older Xerox copier/printer that has a built in network adapter and network software but wasn't wifi compatible. This

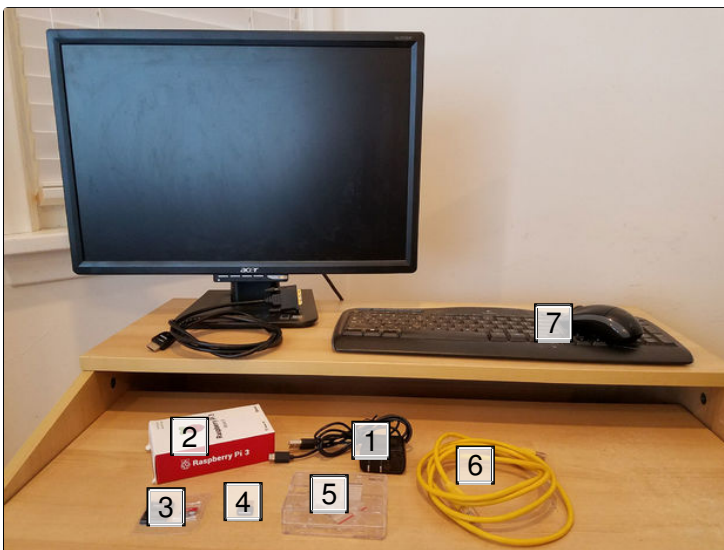
printer was in an older building and the printer was in a location that wasn't close to an ethernet punch down and couldn't be moved. With a few parts that I already had around my house I was able to put together a solution that solved my needs.

This easy DIY solution will give you the capability of adding a wifi connection to your older devices without breaking the bank by purchasing wireless print adapter.



## Step 1: Items You'll Need

1. Raspberry Pi (any model will do, but you'll see faster results with a model 3).
2. Power adapter for your Pi.
3. SD Card to install the operating system on (you could use an 8GB card up to whatever size you want. I usually go with a 32GB card in case I want to add more options to the RPi).
4. Wifi adapter
5. Ethernet cable
6. HDML cable (I have an old DVI monitor so I use this HDMI to DVI cable).
7. Keyboard and mouse
8. Card reader or computer with built in card reader.
9. A case for the Pi (optional)

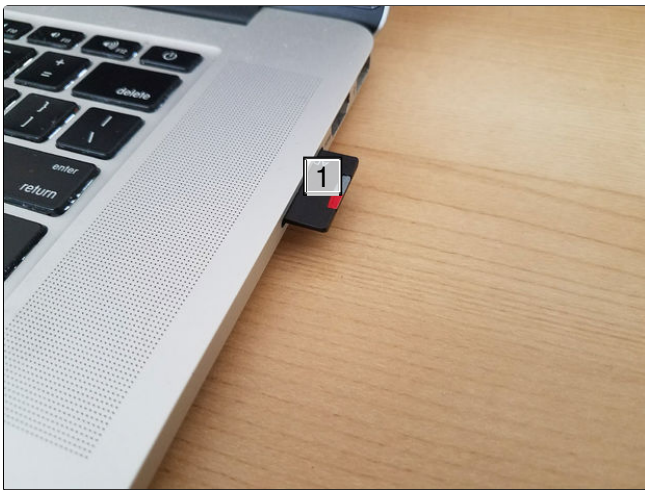


1. Power adapter
2. Rasberry Pi 3
3. SD Card
4. Wifi adapter
5. Case for Raspberry Pi
6. Ethernet Cable
7. Keyboard and mouse

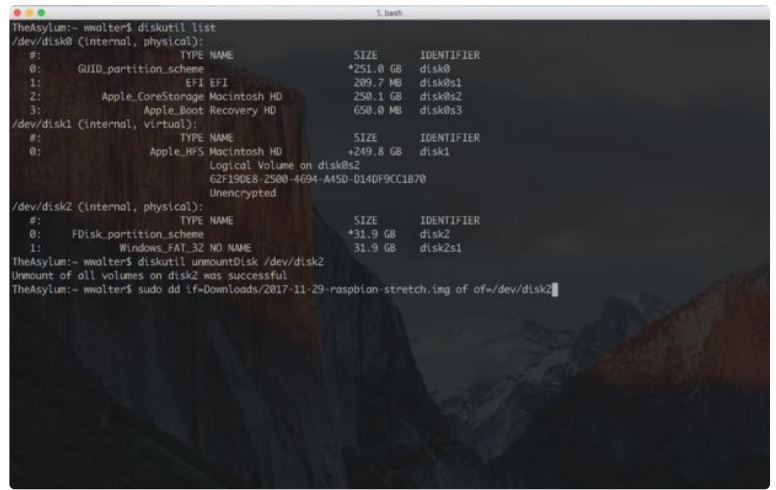
## Step 2: Install Raspbian Operating System

Download the latest version of Raspbian (Stretch as of this writing) operating system from [here](#). Insert the SD card in the adapter that ships with it and put the adapter with the SD card into your card reader. Copy the Raspbian operating system image onto the SD card by using these instructions:

- Windows instructions
- Mac OSX instructions
- Linux instructions



1. Insert the adapter with the SD card into the card reader



### Step 3: Assemble the Remaining Components

This will take some time to copy the image to your SD card. Assemble the rest of the RPi while you are waiting.

Insert the wifi adapter in one of the USB ports. Insert the keyboard and mouse dongle into one of the other USB ports. Connect a monitor to the Raspberry Pi with the HDMI cable.

When the Raspbian image has completed installing on the SD card, remove the SD card from the adapter and insert it into the SD card slot on the underside of the Raspberry Pi. Then insert the power adapter into the micro USB port and power up the Raspberry Pi.



1. Wifi adapter  
2. Keyboard and mouse dongle



1. SD card with Raspbian operating system



1. Connect monitor with the HDMI cable.
2. Connect power adapter to the mini USB port.

## Step 4: Set Up WiFi Connection

Once the Raspberry Pi has finished booting up set up your wifi connection on the Raspberry PI by opening a terminal window and edit the `wpa_supplicant.conf` file by using the following command:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Change the country to your two letter country code.

Add your Wifi access point SSID and password at the bottom of the file:

```
network={  
ssid="Your Wifi SSID"  
psk="yourWifiPassword"  
}
```

One thing to note: The default keyboard setting is to use a GB configuration. For those of us in the US it places some of the special characters in different places, mainly the @ and the " symbols are switched.

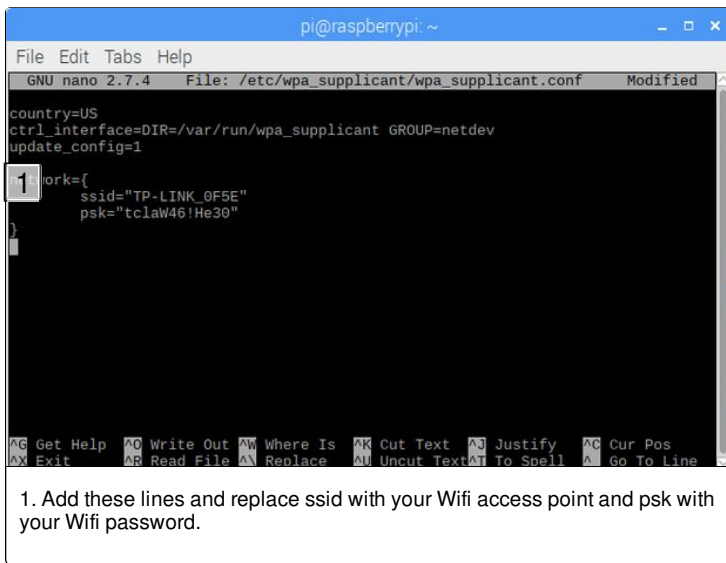
Save the file and exit nano.

Bring the Wifi connection up by typing:

```
ifup
```

or by rebooting the Raspberry PI with:

```
sudo reboot
```



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/wpa_supplicant/wpa_supplicant.conf Modified
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
1 network={
  ssid="TP-LINK_0F5E"
  psk="tc1aW46!He30"
}
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^S Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

1. Add these lines and replace ssid with your Wifi access point and psk with your Wifi password.

## Step 5: Configure Other Options With Raspi-config

When your Raspberry Pi has successfully connected to your Wifi you should see the Wifi symbol in the menu bar in the upper right corner of the screen for the Raspberry Pi.

You can now configure other options for your Pi. From a terminal window type:

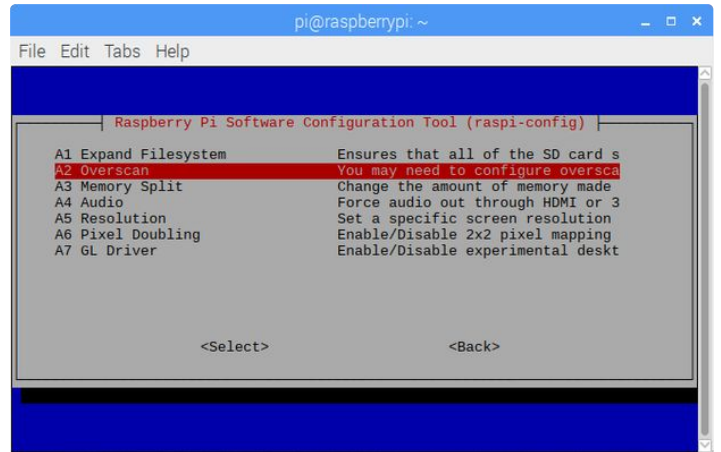
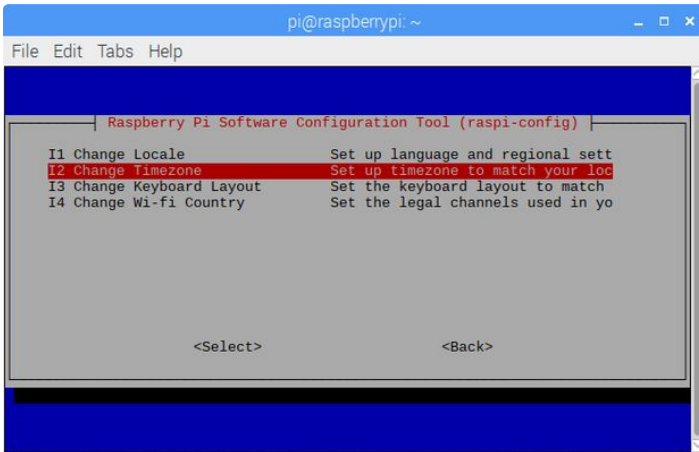
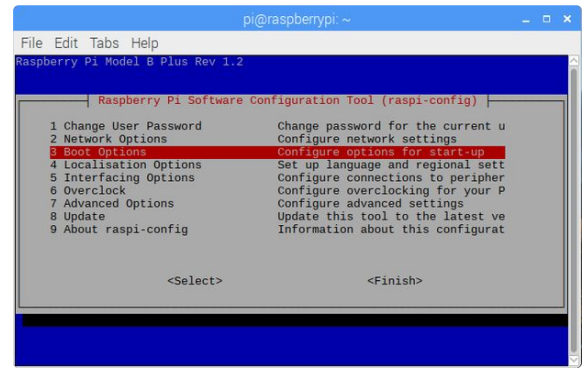
```
sudo raspi-config
```

This will bring up the raspi-config interface and allow you to configure other options your Raspberry Pi. You don't have to do this but there are a couple of things that you should do:

1. Change the default password. Don't leave devices on your network vulnerable by leaving the default passwords on your RPi for the pi and root users.
2. Set your location setting. This will give you the right keyboard settings, time settings and location for installing other software from the nearest repos. If you get the annoying PERL warnings about the locale cannot be set you can resolve it by using these instructions.
3. Expand the file system to use the entire SD card. This will give you access to the entire storage space on the HD card.

Feel free to look at the other options that are available to you through this interface. You can do other things like overclocking your CPU, set up ssh and ftp connections, and change your boot settings to boot to a command line or the desktop.





## Step 6: Configure the Network Bridge From Wifi to Ethernet

To do this we're going to use dnsmasq to set up the RPi to be a DHCP server and set up some custom DNS settings. This will allow the device connected to the RPi through ethernet to get an IP address from the RPi and also to for the RPi to pass DNS queries.

We'll also configure some iptables settings to make a NAT between the ethernet adapter and the Wifi connection.

First, install dnsmasq:

```
sudo apt-get install dnsmasq
```

Set your ethernet adapter to a static IP address

This will serve as a gateway for the device that you want to connect to the RPi ethernet port. Most Wifi routers use what's called a Private Network and set

attach to it. The NAT settings will route the ethernet requests through the Wifi connection.

There are several commands to run here:

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

Configure the dnsmasq settings

The first thing to do is to turn on IP forwarding. This is done by putting a single number 1 in the `/proc/sys/net/ipv4/ip_forward` file:

the IP range to something similar to:

192.168.1.1

For the ethernet adapter on your RPi you'll want to set that to an address that won't interfere with the routers ability to assign addresses, so we'll increment the subnet of the PRi to be:

192.168.2.1

Along with that you'll need to set up the the netmask to:

255.255.255.0

As well as the DHCP settings to broadcast what IP address are available:

```
network 192.168.2.0  
broadcast 192.168.2.255
```

Use iptables to configure a NAT setting to share the Wifi connection with the ethernet port  
NAT stands for Network Address Translation. This allows a single IP address to server as a router on a network. So in this case the ethernet adapter on the RPi will serve as the router for whatever device you

Now plug a CAT5 network cable into the device you want to include on the network and put the other end of the cable into the ethernet port on the RPi and you should be good to go! When we set up the ethernet interface we made it hot pluggable, so you should see the ethernet interface come up when you plug the device into the RPi.

```
sudo nano /proc/sys/net/ipv4/ip_forward
```

Put a 1 on the first line and then exit and save. *Hint: you also may have to edit /etc/sysctl.conf and uncomment this line:*

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

Next set up ip routing:

```
sudo ip route del 0/0 dev eth0 &> /dev/null  
a=`route | awk "/${wlan}/" '{print $5+1;exit}'`  
sudo route add -net default gw 192.168.2.1 netmask  
0.0.0.0 dev eth0 metric $a
```

The last thing to do is edit your /etc/dnsmasq.conf file and include these settings;

```
interface=eth0  
bind-interfaces  
server=8.8.8.8  
domain-needed  
bogus-priv  
dhcp-range=192.168.2.2,192.168.2.100,12h
```

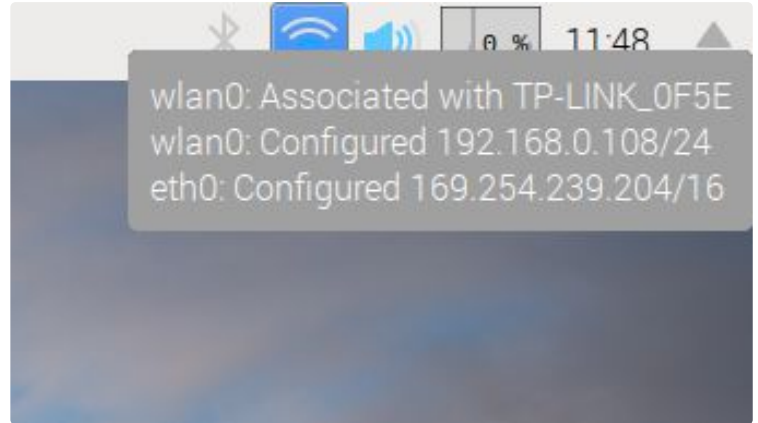
Then run this command to start your dnsmasq services:

```
sudo systemctl start dnsmasq
```

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/sysctl.conf Modified
#####3
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=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
```

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
allow-hotplug eth0
iface eth0 inet static
address 192.168.2.1
netmask 255.255.255.0
network 192.168.2.0
broadcast 192.168.2.255
```

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/dnsmasq.conf
interface=eth0
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=192.168.2.2,192.168.2.100,12h
```



## Step 7: Automate All Those Commands With a Script

This was a lot of work to get the network bridge up and running. You'll probably want this to run automatically every time your RPi boots up, so to do that we're going to need a script to run all of these commands for us. Luckily Arpit Agarwal has already created a script and is available for download here.

Don't worry about typing all those commands above and run this command from your home directory to download the script file:

<https://raw.githubusercontent.com/arpitjindal97/raspbian-recipes/master/wifi-to-eth-route.sh>

To get this file to run every time you boot your RPi

you'll need to add a directive to your session autostart file:

```
nano /home/pi/.config/lxsession/LXDE-pi/autostart
```

and add this to the bottom of the file:

```
sudo bash /home/pi/wifi-to-eth-route.sh
```

Then just reboot the RPi and the script does all the work for you. You can also run this set up any time you want by running this command from a terminal:

```
sudo bash /home/pi/wifi-to-eth-route.sh
```



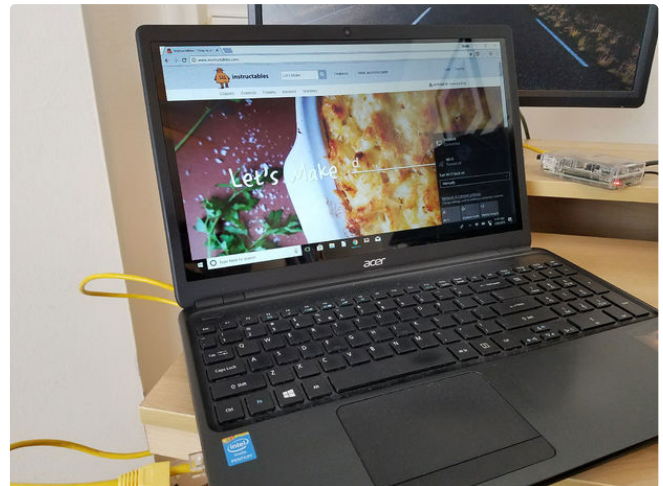
```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /home/pi/.config/lxsession/LXDE-pi/autostart
lxpanel --profile LXDE-pi
pcmanfm --desktop --profile LXDE-pi
xscreensaver -no-splash
@point-rpi
sudo bash /home/pi/wifi-to-eth-route.sh
Read 5 lines
AG Get Help  AO Write Out  AW Where Is  AK Cut Text  AJ Justify  AC Cur Pos
AX Exit      AR Read File  AL Replace  AU Uncut Text AT To Spell  AA Go To Line
```

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ wget https://raw.githubusercontent.com/arpitjindal97/raspbian-recipes/master/wifi-to-eth-route.sh
--2018-01-20 19:24:52-- https://raw.githubusercontent.com/arpitjindal97/raspbian-recipes/master/wifi-to-eth-route.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.128.133, 151.101.192.133, 151.101.0.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.128.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1246 (1.2K) [text/plain]
Saving to: 'wifi-to-eth-route.sh'

wifi-to-eth-route.sh 100%[=====] 1.22K --.-KB/s in 0s
2018-01-20 19:24:54 (2.75 MB/s) - 'wifi-to-eth-route.sh' saved [1246/1246]

pi@raspberrypi:~$ ls
2018-01-20-191856_1680x1050_screenshot.png  Music  Templates
Desktop  Pictures  Videos
Documents  Public  wifi-to-eth-route.sh
Downloads  python_games
pi@raspberrypi:~$
```

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo bash wifi-to-eth-route.sh
pi@raspberrypi:~$
```



Can I use this to create my own wifi group if I connect to a wifi router? I have a boat and sometimes we can be quite a distance from a hotspot. To remedy this I have a high gain antennae and amplifier which plugs into a USB on my Macbook. Works a well, but unfortunately I cannot then share the wifi from my computer to other devices like phones and the wife's iPad! I have RasPi B (I think it is) Can I use this setup plugged into my Pi and the take the wifi signal out via Ethernet into an old router to create a hotspot on the boat?



@Frickinghell, yes this should be possible. I tried it on my home network and was able o connect the RPi to my ISP router with wifi and then connect a TP-LINK router to the RPi with the ethernet cable and the TP-LINK router handled all the wifi connections in my home.



Thank you that's great news. Now to dig out my RPi :)



Eh, sorry, that image ended up upside down, but hopefully you can see everything. The ISP router is in the back, the TP-LINK router is in the foreground and the RPi is on top of the TP-LINK router.



I'm a wannabe geek so I have a few let's call them "will I get fired" questions. I work at a large

college that takes their network security very seriously. No, I mean SERIOUSLY.

So first question, could I use this method to plug in a wi-fi router and create my own network in my office (mainly to bypass the firewalls and safeguards that prevent me from doing things like streaming to a ChromeCast or controlling Keynote from my phone)?

Second, let's pretend I did this (And used a solid WPA with no broadcast SSID). Am I opening my entire school's network up to a security breach because of the presence of this network? I assume not because the network failsafes would lock out my RPi before anything was able to do any damage.

Finally (and here's the important question), would my system administrator easily figure out that something like this is going on, get angry, and make sure I never worked another day in this town?



Umm, this is a complicated question, but overall it's just a really bad idea.

1- Could you do this? Probably, but you'll be under all the same firewall rules for the gateway that you're currently using. This won't enable you to bypass any of those network wide settings.

2- Yes, you're potentially opening up your network to a security breach. Any time you're adding more devices to a network you add the potential for a vulnerability, especially if those devices have poor security mechanisms in place.

3- Yes, most likely they will be able to figure it out. It may take some time and you could get away with it for a while, but a good network admin would be able to track devices on the network.



That's a neat setup :)



Thanks Swansong!



I have worked with Ethernet (beginning at 10 MPS) since the early 1990's and I know how to use a WiFi router, but this interface is brand new and well designed, even for an experienced engineer such as myself. I like your simplicity to the build and I recommend anyone to fully investigate the Ethernet equipped device and see if it has a fixed IP address. then setup the IP range to cover that address only (or a very small range for more devices). WiFi is becoming better each year and soon we will have vehicles using a special WiFi version for V2V communications.



That's a great suggestion @RaymondR6. Locking it down to only the single IP address will reduce the chances of exploitation on your device. Thanks for that suggestion.