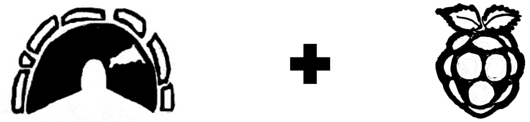
tunnel + backups with a raspberry-pi



A ZINE ABOUT HOW TO CHOOSE A VPN SERVICE
AND HOW TO USE ONE ON A SMARTPHONE

this manual is part of the VPN-zine
project from psaroskalazines.gr

it is made with hand drawn icons by the author
and layout design is done with scribus

font for the cover + colophon + content's tites:
https://www.fontspace.com/casaletwo-nbp-font-f16929

font for the content:
www.fontspace.com/kp-programmer-nbp-font-f16304

many thanks to:
digitaldefenders.org for the financial support
systerserver.net for hosting the project's git repo

find the author on mastodon
systerserver.town/@mara

INTRODUCTION

This guide is about running a VPN with a raspberry-pi for accessing our local homework from outside our home.
It goes over an openVPN installation for Raspbian and the configurations needed for our home router.
Then the raspberry-pi or other device in our local network can be remotely accessible. We can use this VPN to do backups or share the VPN with our peers who would like to access our home base media library, especially handy in times of lockdowns.
Enjoy!

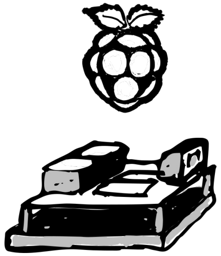
Visit the rest of the VPN-zine project:
Tunnel Up / Tunnel Down
zines.cucu.gr/prints/tunnel-up-tunnel-down-en/

Troubleshooting OpenVPN
zines.cucu.gr/prints/troubleshooting-openvpn-en/

Tunnels and Smarthones
zines.cucu.gr/prints/tunnels-and-smartphones-en/

Support the project by purchasing a physical copy!

INDEX



INVENTORY

pi = raspberry-pi

OS = Operating System, here we mostly refer to the Raspbian OS

ssh = remote connection from our laptop to pi, needs to be enabled during pi´s OS installation, under interfacing options

router = our local network's public interface to the internet, which is also why it's called the default **gateway**.

IP = internet protocol

pc = personal computer

DHCP = Dynamic host configuration protocol

a DHCP server automatically assigns an IP address to each host on the network. DHCP also assigns the subnet mask, the default gateway address, and the domain name system (DNS) address

curl = command for downloading from the internet via the terminal

bash = shell and command language for unix systems

installation wizard = a tool to lead the user through an installation

iptables = program that allows a system administrator to configure the Linux kernel firewall by filtering the incoming and outgoing traffic of the IP packets as well as forward them to other destination

ip routes = command which adds new route in the routing table.

routing table = a data table stored in a router or a network host that lists the routes to particular network destinations

host = a device in the network

nmap = command that scans the network, ports, also used for monitoring, security, discovering available hosts

ISP = internet service provider

internet interface id = the device's connection, e.x wlan0, eth0, tun0

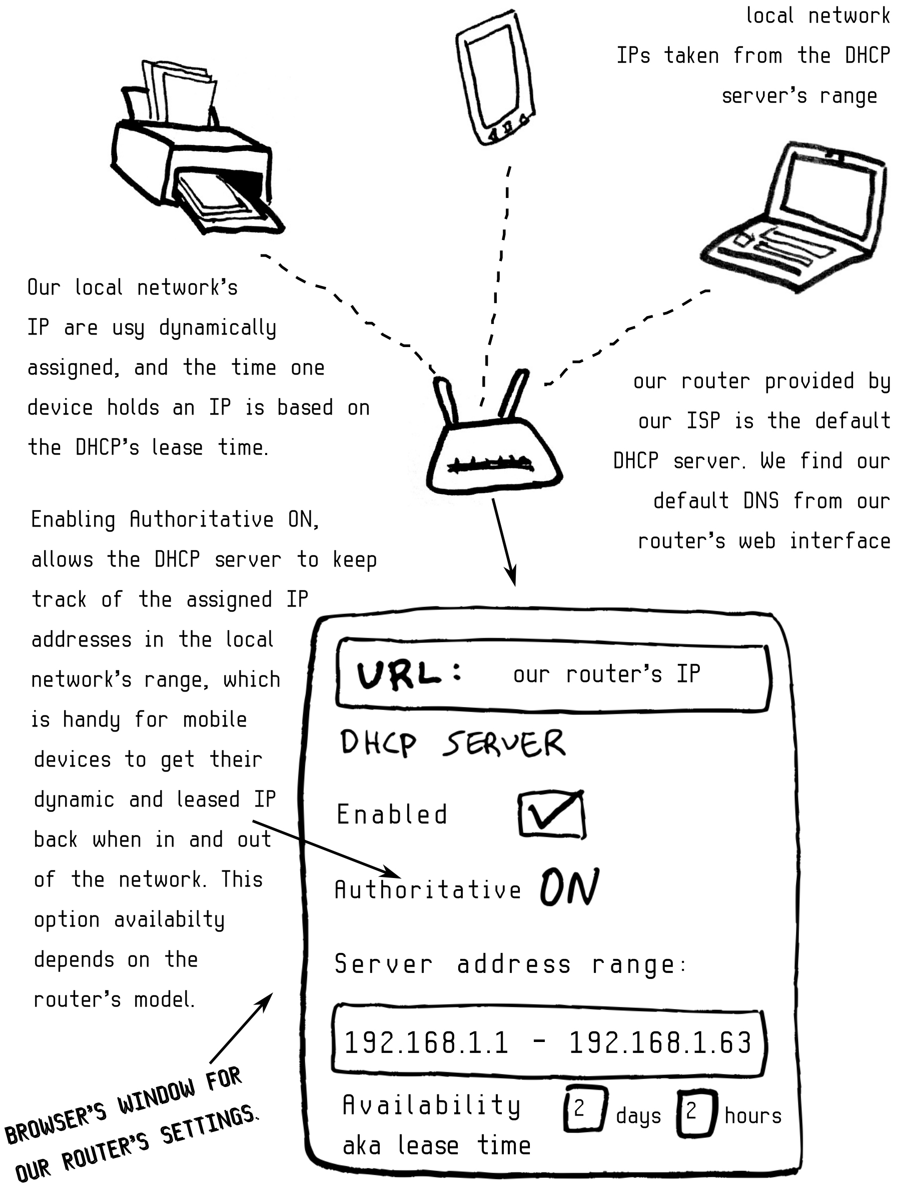
LOCAL NETWORK SETTINGS

We can install a Debian flavor OS on a raspberry-pi, which is called Raspbian (see references in the CHEAT-SHEET). Here we will focus on setting up openVPN, which we may install as we would on a linux server/pc. Check out the zine "Troubleshooting openVPN" for a thorough documentation of that method and how to debug errors. Or we can do it with the tool pivpn which is a wizard installer that guide us through the installation process.

But before we dive into that, let's investigate and prepare our general network setting. What is critical about installing a VPN server in our home network is how our ISP provides internet to our residency. If our home router/gateway is behind a larger area's NAT, then our public IP doesn't reflect our home's gateway but that of a another box in the middle that we cannot conigure. This type of connection is called carrier grade or large scale NAT, and only if a public IPv6 is possible for our router, can we proceed with running a service such as an openVPN server from home.

We can find out how we connect to the internet by checking our public IP on a website (e.x whatismyip.com) and verify that our router's public IP is the same. To do that we login to our router's web interface and look for the WAN settings. If the IPs are not the same, we can ask our ISP if we may get an IPv6 instead. However this guide focuses on IPv4 network setting, but it takes little changes to configure openvpn over IPv6 .

HOW DCHP WORKS



local network IPs taken from the DHCP server's range

Our local network's IP are usy dynamically assigned, and the time one device holds an IP is based on the DHCP's lease time.

our router provided by our ISP is the default DHCP server. We find our default DNS from our router's web interface

Enabling Authoritative ON, allows the DHCP server to keep track of the assigned IP addresses in the local network's range, which is handy for mobile devices to get their dynamic and leased IP back when in and out of the network. This option availabilty depends on the router's model.

URL: our router's IP

DHCP SERVER

Enabled ☑

Authoritative ON

Server address range:

192.168.1.1 - 192.168.1.63

Availability aka lease time 2 days 2 hours

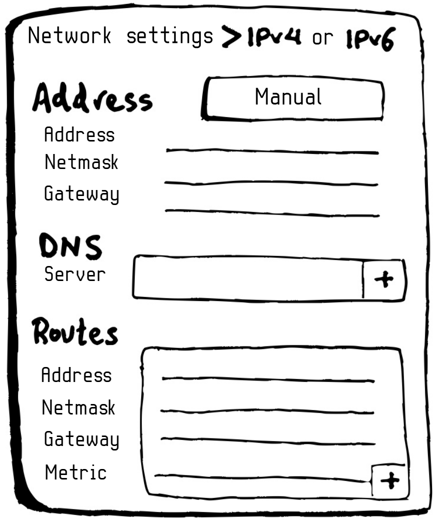
BROWSER'S WINDOW FOR OUR ROUTER'S SETTINGS.

DCHP CLIENT + STATIC IP

.The raspberry-pi needs a local static IP. We can set it from pi's settings or from our router or both. It should be outside the leased range of dynamic IPs.. In the above schema, the range of dynamic IPs are up until 192.168.1.63, so we can choose one outside that range and ensure that it isn't taken by the router's own local IP either. In linux a '**routel**' or a "**ip r | grep default**" command will tell us the local IP of our router, which is marked as the default gateway and our network's interface, called interface ID, e.x eth0 or wlan0.

We ssh into our pi and in the file /etc/dhcpcd.conf we look for "Example static IP configuration', we uncomment the block and we edit the "static_IP" to the one of our choice, e.x 192.168.1.65. We add the gateway and dns' IP. And we also set the network's interface ID, e.x eth0, (a wired pi connection is faster and more stable). We restart pi.

For adding a static IP with a GUI, we go to network settings > ipv4 and pick the manual option, add netmask 255.255.255.0, and fill in our router's IP. Adding DNS and routes is optional.



DOMAIN NAME FOR THE GATEWAY

Next we move to set up a domain name to associate our home network's external/public IP which is provided by our ISP and which periodically changes. If the ISP can set us up with a static IP, then we skip this step. In a home setting our router is also our main gateway to the internet. The goal is to be able to remotely access the pi behind our home gateway's public IP via a VPN tunnel. We will also add a port in our router to forward to pi VPN's open port.

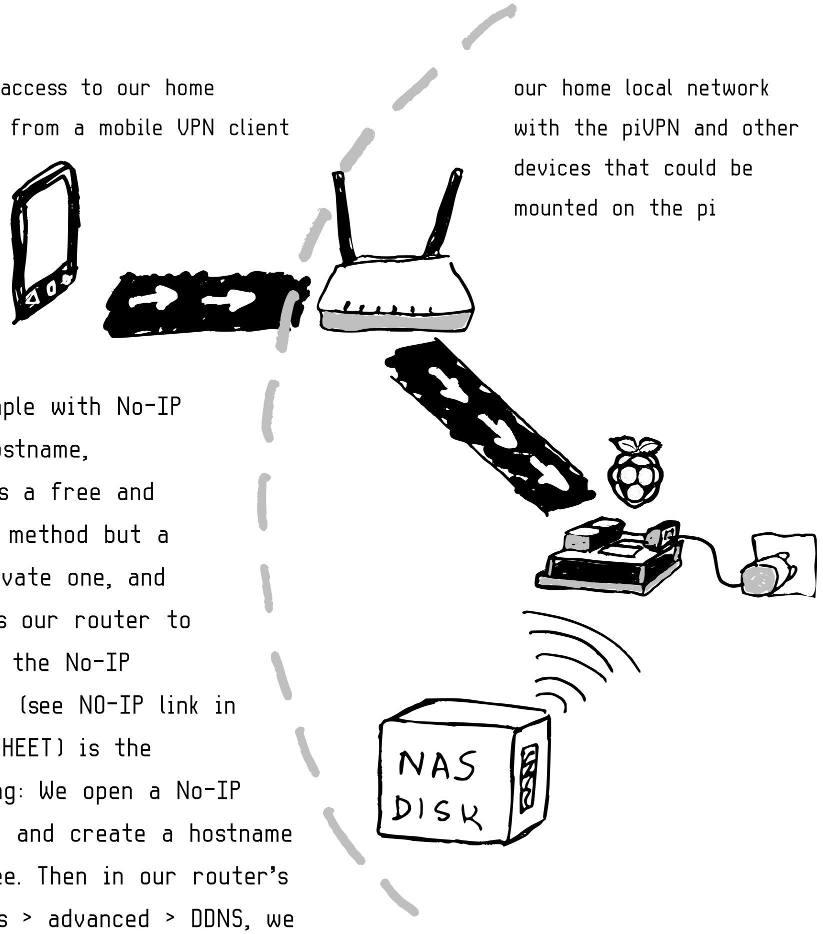
We can purchase a domain name from a hosting company or get one for free from Dynamic DNS services such as No-IP, Dyndns, Duckdns. Here we first give an example with a registered domain name in gandi.net, which provides an API that we can call locally to dynamically change the IP of the domain name's DNS record.

Once we obtained one, we can register for Gandi's API and get the api key. There are plenty of libraries that help with changing the DNS record's IP from a host.
E.x https://github.com/ghitier/gandi-ddns. has a README for all steps involved in installing and running the code.
We can find libraries when searching for "script refresh ip + <name of our hosting's API>" + our preferred coding language.

DDNS + OTHER ALTERNATIVES

remote access to our home network from a mobile VPN client

our home local network with the piVPN and other devices that could be mounted on the pi



An example with No-IP DDNS hostname, which is a free and quicker method but a less private one, and requires our router to support the No-IP feature (see NO-IP link in CHEAT-SHEET) is the following: We open a No-IP account and create a hostname for free. Then in our router's settings > advanced > DDNS, we select NO-IP as the service provider, and we enter our No-IP username, password and the No-IP hostname we created.
Another method that doesn't require any domain name creation, but it involves manual work: run a cron job from pi that fetches our public IP and emails this to us. Before we connect to the piVPN we edit our client's certificate line under "The hostname/IP and port of the server", which starts with "remote" with the IP sent from the email.

INSTALLATION WITH PIVPN

So far we have a static IP for the pi, and a method to retreive/resolve our router's dynamic public IP which is the gateway to our home network. If our pi is connected to a monitor we can do "**ip addr**" to find its IP, or we try to ssh from our device with pi's hostname with "**ssh pi@raspberrypi.local**". If we changed the default username "pi" or the hostname, then our ssh command should reflect these changes. Mosh is an alternative to ssh for uniterrupted logins, which would prove handy to use it during our ssh session installing the openVPN.
Once we are logged in the pi, we do a firewall check, with "**sudo iptables -L**", which should indicate INPUT, FORWARD, OUTPUT have the "ACCEPT" policy (all requests should be accepted for the time being,)

Phew, let's get openVPN now! Download the pivpn installer with:
curl -L https://install.pivpn.io/ | bash
The above gets the tool and runs it with the bash intepreter.
Before the wizard installer pops up, we are asked to give our root password. The wizard first asks weather we want openVPN or WireGuard installed, we go for the first in this example (see CHEAT-SHEET for pros & cons of each). The next steps go over:

- confirming pi's static IP
- DNS we pick custom to add our default gateway/router's IP
- user choice

CLIENT CERTIFICATES + PORT FORWARDING

- adding the public IP or domain name for the clients to reach the pi’s VPN. Depending on what method we chose earlier, either having set a DDNS or a domain name for our router, we enter it here.
- port and TLS encryption. Opting for the default settings (udp 1194, ECDSA, and certificate size of 256bits is fine
- and we shall opt for unattended upgrades too.

The installer configures these options, generates the server keys and config file, and finishes with instructing us to generate a client profile with "pivpn add". For this step we give a name for the client, and once the profile is created we send it to the device which we want to connect to the piVPN. We can scp or sftp to get the profile to our working station/laptop/mobile. The device which uses the client profile also needs to have openvpn installed. See how to get it for various OS in the CHEAT-SHEET.

Last step is to do port forwarding from our router to the piVPN’s port we set up earlier during the VPN’s configuration. Back to our router’s admin via the web browser, we look for port forwarding, it might be under advanced or firewall settings, depending on the router model. There we add a new port forward with the udp port number we gave during the pivpn’s installation, and pi’s static IP. Finally, we go to our device terminal and connect to piVPN with:
openvpn --config <client_profile_file>
Or from a smartphone we import the new client profile in the openvpn connect app.

REMOTE BACKUPS

When our pi VPN is tested and running, we can put it in good use. One such use would be for backing up our laptop’s files when we travel or when we work away from home. In the following example we use restic to do just this.

In a terminal of the device we want to backup, we create a new user with: **useradd -m backups**
To restrict restic execution only by root and the user backups check the restic docs which demonstrate the required steps to accomplsh it:
https://restic.readthedocs.io/en/stable/080_examples.html#backing-up-your-system-without-running-restic-as-root

The destination where we keep our backups could be a directory of the pi itself, or an external disk mounted to the pi. We call it a repository, and it should have a dedicated user for ownership and the right permissions to write and read in the repository.

Then we create ssh keys for the backups user and transfer its new public key to the repository owner’s home under .ssh/authorized_keys. From the vpn logs or from running an **’ip addr**’ in the pi, we mark the virtual IP of our pi, which is the IP associated with the interface id tun0.

OTHER USES FOR A PIVPN

Then we initiate the backup repository with:

```
restic -r sftp:pi@<virtual-ip>:/media/pi/backups init
```

and we enter a password and save it for future itirations of the backup command as well as for restoring our backup data. Now if, for example, we want to back up our html files' we run:

```
restic -r sftp:pi@<virtual-ip>:/media/pi/backups --verbose backup /var/www/html
```

What else can we do with a VPN running in our pi? Access our media files, and create client certificates for our peers so that they can access these files too. For instance we can mount a NAS external disk with our movies on the pi and give ssh access to others to reach the mount point via the terminal. Or enable access to the rest of the pi's local network via the VPN. Or use it to tunnel all our traffic when we surf from public WiFi spots.

Depending on what we want to do with the piVPN, we have to adjust pi's routing table. E.x for openvpn clients to reach the internet via the piVPN tunnel we do:

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

where source **'-s'** is our tunnel subnet declared in the server.conf, and output **'-o'** is the pi's internet connection, so should indicate either the wired or wireless interface id.

If we want to reach the local network behind the pivpn, we add "push route <local-network-subnet> netmask" in the server.conf.

CHEAT SHEET

Install openVPN with pivpn wizard:

www.comparitech.com/blog/vpn-privacy/raspberry-pi-vpn/#Installing_Pi_VPN

How DHCP works:

www.networkworld.com/article/3299438/dhcp-defined-and-how-it-works.html

www.lifewire.com/what-is-dhcp-2625848

Add a static IP for the pi:

raspberrypi.stackexchange.com/questions/37920/how-do-i-set-up-networking-wifi-static-ip-address-on-raspbian-raspberry-pi-os/74428#74428

Raspbian is a Debian based OS optimized for the Raspberry Pi www.raspbian.org/

Dynamic update of DNS with Gandi

virtuallytd.com/post/dynamic-dns-using-gandi/

set up NO-IP with the home router:

www.noip.com/support/knowledgebase/how-to-configure-ddns-in-router/

Cloudflare alternative to no-ip

medium.com/@mirrormirage0/configuring-dynamic-ip-auto-update-for-custom-domain-name-alternative-to-dyndns-noip-etc-57a1e100efd5

access the router's settings:

www.pcmag.com/how-to/how-to-access-your-wi-fi-routers-settings

Get openvpn client for various OS: openvpn.net/openvpn-client-for-linux/

Pros and cons of VPN protocols WireGuard and openVPN

restoreprivacy.com/vpn/wireguard-vs-openvpn/

Install restic for backups: restic.readthedocs.io/en/stable/020_installation.html

restic.readthedocs.io/en/stable/040_backup.html

General backup tips and scripts:

www.debian.org/doc/manuals/debian-reference/ch10.en.html#_backup_and_recovery