Plating Onions

What is Tor? How to use? and how to run a relay

Caio Volpato (caioau)
caioau.keybase.pub → caioauheyuuiavlc.onion
210B C5A4 14FD 9274 6B6A 250E **EFF5 B2E1 80F2 94CE**
All Copylefts are beautiful: licensed under CC BY-SA 4.0

Laboratório Hacker de Campinas – LHC

Summary:

- What's Tor?
- Demystifying the deep/dark web
- How to use Tor Browser
- Cool stuff made with Tor
- Running a relay!

About Casahacker hackerspace:

About the cebolas collective:

Acknowledgments

This presentation would not be possible without the help of this awesome Tor people: **Cybelle**, **ggus** and **egypcio**

Tor

Tor – The onion router, is a libre software that enables communicating anonymously and privately.

Tor

Demystifying Tor: Myths

- Deep web? Dark web? Hackers? Illegal stuff?

Tor

Demystifying Tor: Myths

- Deep web? Dark web? Hackers? Illegal stuff?
- Deep dark web much bigger than Google, Facebook and YouTube?

Tor

Demystifying Tor: Myths

- Deep web? Dark web? Hackers? Illegal stuff?
- Deep dark web much bigger than Google, Facebook and YouTube?
- When you use Tor bad things can happen to you such entering the FBI most wanted?

Tor

Demystifying Tor: Myths

- Deep web? Dark web? Hackers? Illegal stuff?
- Deep dark web much bigger than Google, Facebook and YouTube?
- When you use Tor bad things can happen to you such entering the FBI most wanted?
- Using Tor means entering suspicious sites ?

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?
  - Privacy NGOs: Like EFF and Calyx Institute.

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?
  - Privacy NGOs: Like EFF and Calyx Institute.
  - Humans rights defenders and LGBTQIA+: such Human Rights Watch.

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?
  - Privacy NGOs: Like EFF and Calyx Institute.
  - Humans rights defenders and LGBTQIA+: such Human Rights Watch.
  - Journalists: Such as BuzzFeed, Huffington Post

Demystifying Tor: Facts

So, Whats this Tor is all about?

- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?
  - Privacy NGOs: Like EFF and Calyx Institute.
  - Humans rights defenders and LGBTQIA+: such Human Rights Watch.
  - Journalists: Such as BuzzFeed, Huffington Post
  - Several companies such as facebook and cloudflare.

Demystifying Tor: Facts

So, Whats this Tor is all about?

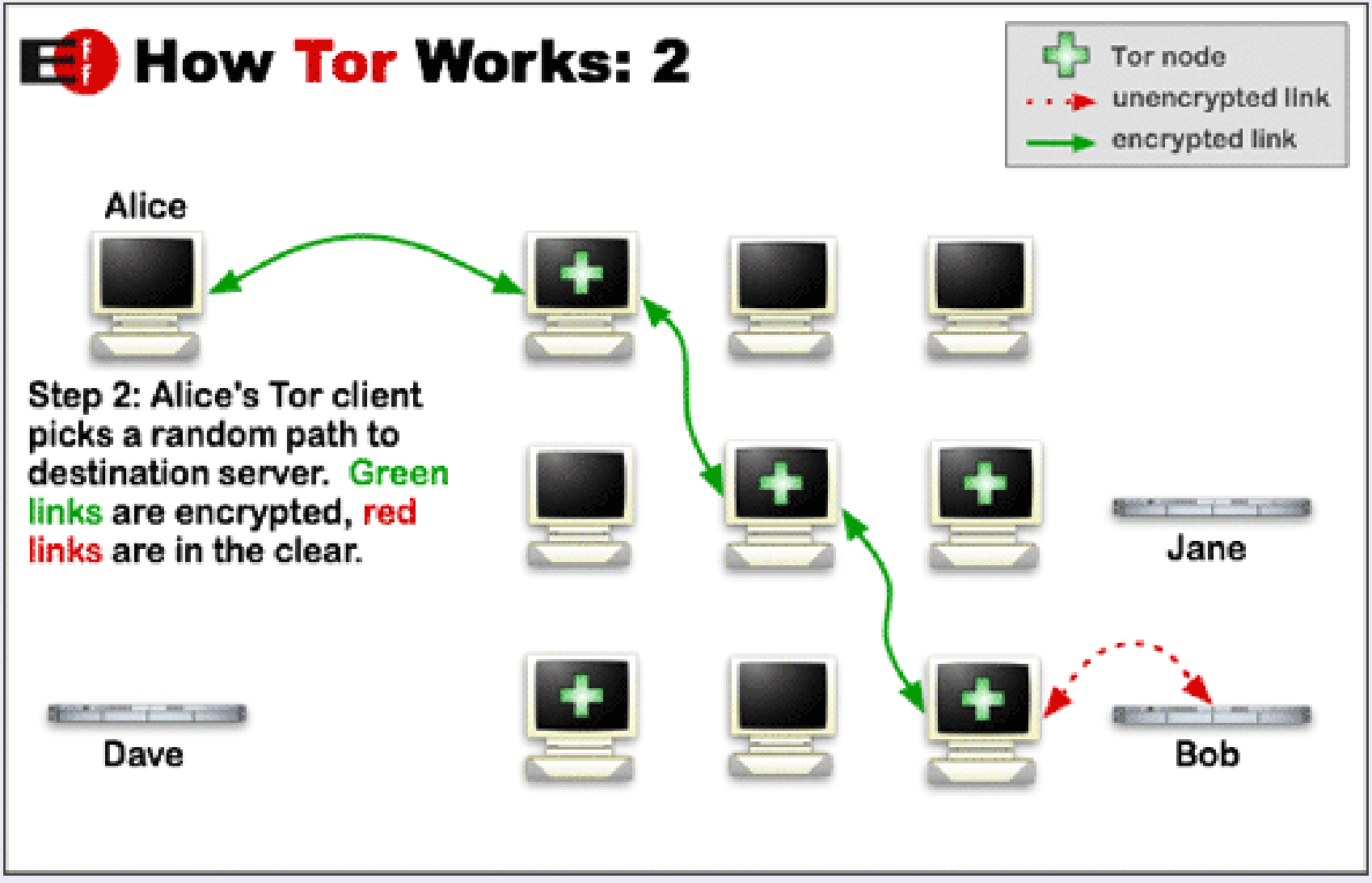
- A libre software used for bypassing censorship, surveillance and tracking on the web.
- Who uses and supports Tor?
  - Privacy NGOs: Like EFF and Calyx Institute.
  - Humans rights defenders and LGBTQIA+: such Human Rights Watch.
  - Journalists: Such as BuzzFeed, Huffington Post
  - Several companies such as facebook and cloudflare.
  - The police and law enforcement.

Demystifying Tor: Facts

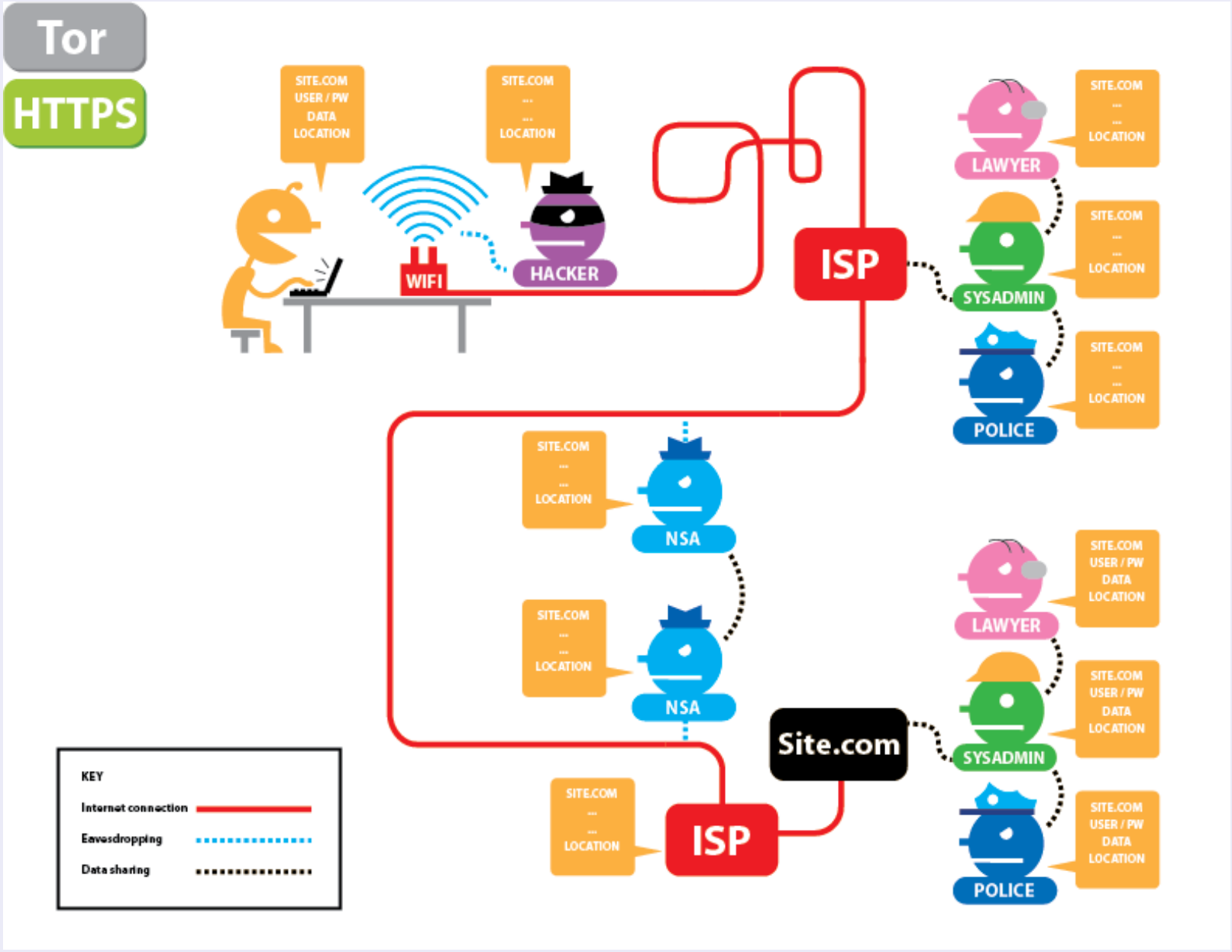
Whats Tor mission?

To advance human rights and freedoms by creating and deploying free and open source anonymity and privacy technologies, supporting their unrestricted availability and use, and furthering their scientific and popular understanding.

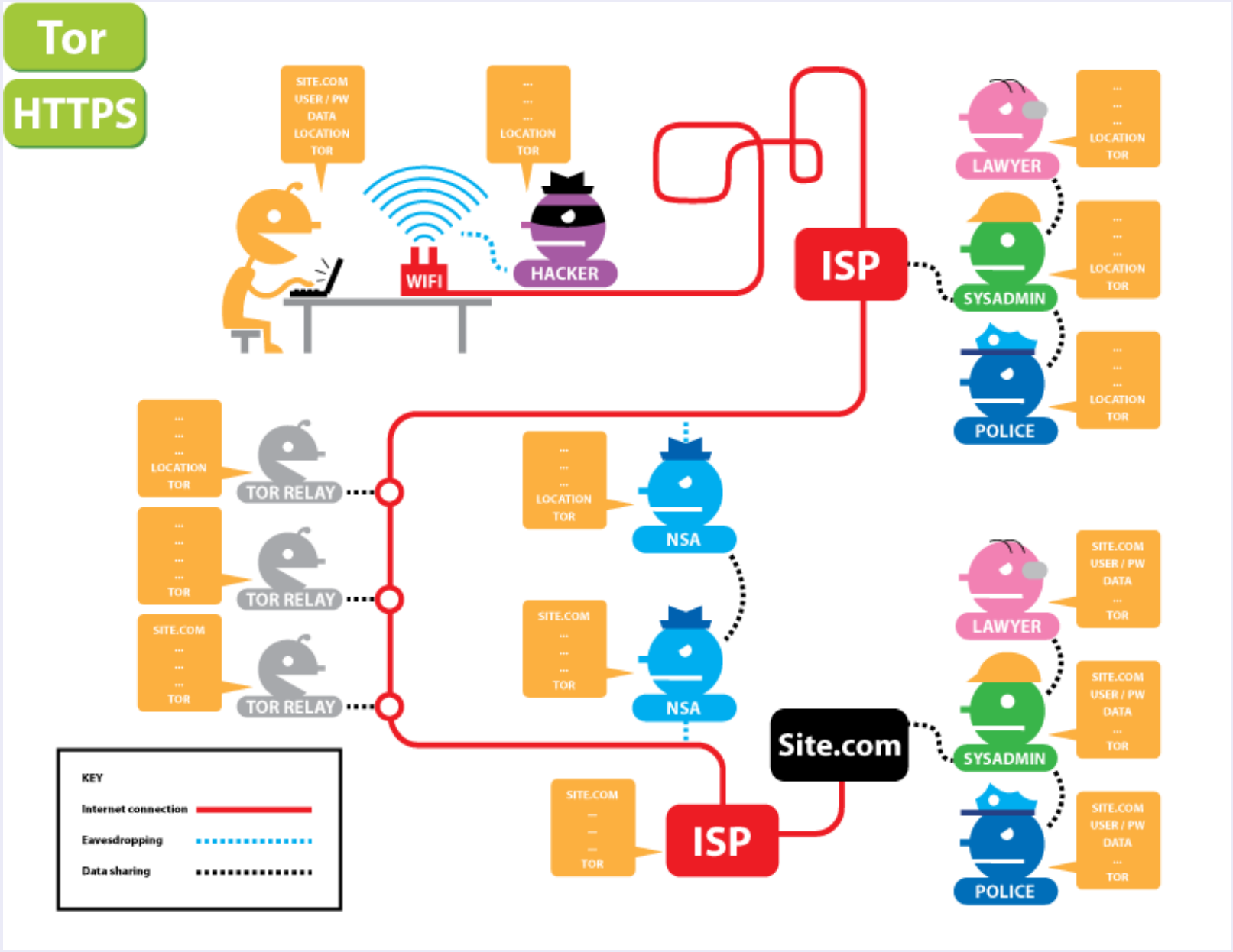
How Tor works:



Without Tor: Tor and SSL



With Tor: Tor and SSL



Getting started: Downloading Tor Browser:

1. Goto : torproject.org and select **Download**
2. Select your Operating System and also select the signature.
3. (**Highly encouraged**) Verify the signature:
    1. gpg --keyserver=hkps://keys.openpgp.org --recv-keys 0xEB774491D9FF06E2
    2. gpg --verify tor-browser-linux64-9.0.1_en-US.tar.xz.asc
4. Extract and Run the Tor Browser.

Downloading Tor Browsing When censored:

TODO (twitter DM, github, mirrors)

While using Tor Browser pay attention:

- Your ISP (**only**) knows* that you are using Tor.
  - If you're under a hostile environment connect using Bridge (this way your ISP will not even know that you are using Tor).
- Only use Tor with the Tor Browser
- Never BitTorrent on Tor.
- **Leave the Tor Browser as is (Don't install any extensions)**
- Avoid opening downloaded files from Tor Browser (Use Tails)

Any questions? Read the Tor Browser User manual

Tor Browser has a user manual website under:
tb-manual.torproject.org

Tor onion services

When using Tor, the connection have to leave the Tor network (via the Exit node), leaving that connection exposed. Also when using Tor on clearnet only protects the client, not the service.
Briefly: When using Onion service (.onion domains) the client and the service each build a circuit to a rendezvous point.

Onion service properties:

- Self authenticated.
- End to end encrypted.
- NAT punch builtin.
- No need to “leave” the Tor network.

Just some onion services:

- Cryptorave: utw4svtv5ccjastc.onion
- Casa Hacker: casa-hackrd5564weeiw7biseh6oms5jatpfa5klfoobeworbksx4z2qd.onion
- Debian: onion.debian.org
- QuebesOS: qubesos-fasa4zl44o4tws22di6kepyzfeqv3tg4e3ztknltfxqrymdad.onion
- BBC: bbcnewsv2vjtpsuy.onion
- NYTimes: nytimes3xbfgragh.onion
- BuzzFeed: bfnews3u2ox4m4ty.onion
- Facebook: facebookcorewwwi.onion
- DuckDuckGo: 3g2upl4pq6kufc4m.onion
- ProtonMail: protonirockerxow.onion
- invidio.us : axqzx4s6s54s32yentfqojs3x5i7faxza6xo3ehd4bzzsg2ii4fv2iid.onion
- riseup.net: list
- keybase.io: key-base5wmilwokqirssclfnsqrjdsi7jdir5wy7y7iu3tanwmtp6oid.onion

Creating a onion service for your website:

Setting up a onion service for your website is super easy! No excuses for not doing it!

This can be done in two different ways:

1. Installing tor program on your server and pointing the onion service to your webserver:
   - Just follow the setting up your onion service
2. Using eotk – The Enterprise Onion Toolkit to act as a reverse proxy.

- **Bonus points** if you use **alt-svc header** in your website, so when connecting to your website via Tor Browser it'll send its onion address via this header so it will connect to it transparently. Follow the privacytools.io guide
- Also checkout the onion balance that provides load-balancing and redundancy for Tor hidden services

Cool stuff built on Tor:

- ooni.torproject.org – Open Observatory of Network Interference
- secure drop: platform to submit documents.
- onion share: allow sharing files anonymously.
- Tails – The Amnesic Incognito Live System.
- Whonix: Operating system that isolates the Tor program and Browser in separated VMs, can be installed on QuebesOS.
- Briar: p2p messenger.
- Debian/QuebeOS anonymous updates.

Running a Tor relay

The Tor project community created a awesome Tor relay guide, with the following sections:

1. Types of relays.
2. Relay requirements.
3. Technical consideration.
4. Technical setup.
5. Community and legal resources
6. Getting help

Running a Tor relay: Types of relays:

Understating the right type of relay for you is important given the legal implications.

When using Tor the connection goes like this:

You $\rightarrow$ Guard relay $\rightarrow$ middle relay $\rightarrow$ Exit relay $\rightarrow$ site.com

The entire path from you to site.com is called a **circuit**

So the Tor relays types are:

- Exit relay: Does the final connection to the desired websites, **requires huge legal support** since a random person can do illegal stuff and your Exit relay will be blamed for it.
- Non-exit relay: A relay which will be either a middle or guard relay (more on that later).
- Bridge relay: Acts like a Guard relay, but since every Tor relay IP is public it can be easy censored, Bridges IPs are not public (optionally they could be only published to the bridgedb). **Are useful for Tor users under oppressive regimes.**

Relay requirements:

- Bandwidth and Connections: Tor relay should be able to handle a lot of concurrent connections at least 70k~100k. For bandwidth it required at least 16 Mbps for upload and download, if you have less run a Bridge (1Mbps minimum) **This is why you should not run a relay at home, a bridge maybe be okish**
- Monthly Outbound traffic: Tor relays needs a lot of traffic, **Ideally a relay runs on an unmetered plan**

Tip: calculate bandwidth given traffic quota using wolframAlpha, ie type: 2TB/month

- Public IPv4 Address
- RAM memory requirements: Tor is very lightweight: 512MB should be fine for <= 40Mbps and 1GB for more.
- CPU: 1vCPU should be fine for most relays.
- Uptime: 2hr/day is the least required.
- Tor version: Older version are unsupported (EOL)

Technical consideration:

Choosing a hosting provider:

Look the community doc: Good Bad ISPs where most providers are listed if they allow (or not) hosting Tor.

Avoid as must as possible the following providers:

- OVH SAS (AS16276)
- Online S.a.s. (AS12876)
- Hetzner Online GmbH (AS24940)
- DigitalOcean, LLC (AS14061)

Try hosting in a AS and country that already has a lot of relays.

Look in the metrics: AS and Country

When looking for a provider **the most important thing is traffic quota, ideally look for unmetered.**

(For me) the payment methods are important too: I always prefer bitcoin or at least PayPal.

Technical consideration:

Choosing a Operation System:

Most relays (like 90%) runs on Linux, and most of them are Debian based. This is bad because if a security vulnerability is to be found most of the network would be down.

There's a great effort to bring more diversity to the network, by encouraging BSD: torbsd

I'm most familiar to OpenBSD and it has great security features like:

- Only two remote holes in the default install, ever!
- KARL – kernel address randomized link.
- Immune by design against Spectre & Meltdown.
- W ^ X memory.

why OpenBSD rocks

Given that, **choose a OS that you're most familiar with.**

Why I prefer Debian over other Linux:

- Rock solid stable.
- Not frequent updates (only security updates).
- Very committed to Libre software (unlike Ubuntu, Fedora).
- All volunteers signs the Debian Social Contract.
- No profit organization (unlike Ubuntu, Fedora).
  - Ubuntu had a spyware installed by default.

A (very) brief introduction to systemd

systemd is a init system that manages system and services, It's been the default since Debian Jessie (2015)

Basic usage:

- Show system (or service) status
  sudo systemctl status [service]
- Enabling and disabling a service:
  sudo systemctl enable/disable service
- Starting and stopping a service:
  sudo systemctl start/stop service
- Restarting and reloading a service:
  sudo systemctl restart/reload service

The difference between restart and reload: Some services handle a Unix signal (normally SIGHUP) that tells the program to reload the configuration file without the need to stop and start (restarting) the service. No downtime!

How to read systemd services logs: journalctl

Systemd centralizes all system and services in the journald daemon.

Basic usage:

- Show all the logs:
  sudo journalctl
- Filter by a unit (like a service) and date
  sudo journalctl -u service --since 2019-11-25

The actual setup:

In order to setup our relay we are going to:

- Creating a non root user (and giving it sudo rights).
- Removing unnecessary packages and services.
- ssh hardening.
- Firewall setup (ufw).
- Always having the correct time: ntp.
- Automatic updates (unattended-upgrades).
- Tor setup and install.
- Monitoring: vnstat and optionally munin.
- Misc recommend things and Tips.
- Raspberry pi specific things.

Creating a non root user:

After creating your VM, the provider will send you the credentials to root (using and login as root is highly discouraged), so run:

```
apt update
apt install sudo
adduser USERNAME
usermod -aG sudo USERNAME
```

Copying your ssh key to your new user:

If you don't have ssh key create it by running **on your local machine**:

ssh-keygen -t ed25519 -o -a 300

Should I type a password to my ssh key, or leave it blank? For most cases putting a password is a Security theater

Now copy your key to the server:

ssh-copy-id -i ~/.ssh/id_ed25519 username@server

Removing unnecessary packages and services:

Se what services are started and boot:

systemd-analyze blame

the following services are safe to disable:

- bluetooth
- accounts-daemon
- avahi-daemon
- ModemManager
- pppd-dns
- wpa_supplicant (if not using WiFi)

Also install htop to see running process:

sudo apt install htop

Then look for non-standard process, the remove it with sudo apt remove package.

ssh hardening:

Generate new host keys:

We have no guarantee that your provider is not using the same ssh host keys for everyone.

```
sudo -s
cd /etc/ssh
rm ssh_host_*
ssh-keygen -t rsa -b 4096 -f ssh_host_rsa_key
ssh-keygen -t ed25519 -f ssh_host_ed25519_key
```

sshd config : TODO colocar link

```
# /etc/ssh/sshd_config
Port 123456 # FIXME: change the default port
HostKeyAlgorithms ssh-ed25519-cert-v01@openssh.com,ssh-rsa-
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-g
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.co
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openss
LoginGraceTime 1m
PermitRootLogin no
# TODO: copy your ssh key than disable password login
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
UsePrivilegeSeparation sandbox
ClientAliveInterval 10
```

Bonus: What to do with the default ssh port – 22 free:

As the previous step you really should change the sshd port, but what to do with the 22 port? You can:

- Have a ssh honeypot: cowrie
  - Or use HaaS – Honeypot as a service: haas.nic.cz
- (What I recommend): ssh tarpit: endlessh
  - This creates a herd immunity against ssh bots so protecting vulnerable machines that could be exploited.

Firewall (ufw):

ufw is a easy to use firewall built on netfilter and iptables.

- Installing ufw:
  sudo apt install ufw
- Enabling it:

```
sudo ufw default deny incoming # (1)
sudo ufw limit SSHPORT
sudo ufw allow 443 # ORPORT
sudo ufw allow 80  # DIRPORT
sudo ufw allow 22  # ssh tarpit (endlessh)
sudo ufw enable
```

1. this will set the default policy to incoming packages: deny incoming packages, unless the ports we allowed.

(recommended): keep your current terminal opened, then open a new terminal and try ssh-in , this way if you get lockout you still have a terminal open to get yourself in.

Automatic updates (unattended-upgrades).

No need to say that keeping your system always updated is essential.

- Install unattended-upgrades package:
  sudo apt install unattended-upgrades
- Enable it: run the following and select to enable automatic upgrades:
  sudo dpkg-reconfigure unattended-upgrades
- Configuration: TODO: github link

Change buster to the current stable name at the time!

And the packages origins to whitelist it, to do this list the installed packages origins:

apt-cache policy | grep release

Now your server should auto upgrade and if needed it'll reboot

- Testing:
  sudo unattended-upgrade --debug --dry-run

Actual Tor install and setup

Now we will install the tor from the project repository:
sudo apt install apt-transport-https

if you are running a bridge also install obfs4proxy

- Adding the repository:

Change buster to the current stable name at the time!

```
# /etc/apt/sources.list.d/tor.list


deb https://deb.torproject.org/torproject.org buster main
deb-src https://deb.torproject.org/torproject.org buster ma
```

- Adding and trusting the repository key:
curl https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDE
| sudo apt-key add
- Installing:

```
sudo apt update
sudo apt install tor deb.torproject.org-keyring
```

Tor setup:

Download my torrc template files:

- non-exit torrc template file.
- bridge torrc template file.

Change to suit your needs: Nickname, ContactInfo.

Try as much as possible to keep Orport to 443 and dirport to 80.

Tips on how to deal with traffic quotas on the next slide.

then replace your edited torrc to /etc/tor/torrc and restart Tor:

sudo systemctl restart tor

Tor setup:

Configuring for traffic quota:

It this example we are configuring for a provider that allows 2TB/month for Download + upload.

Monitoring

Misc recommended stuff and Tips

Raspberry pi specific stuff:

Tor relay life cycle

Reaching out: