

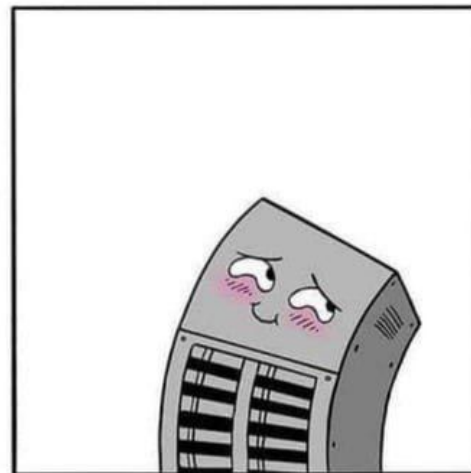
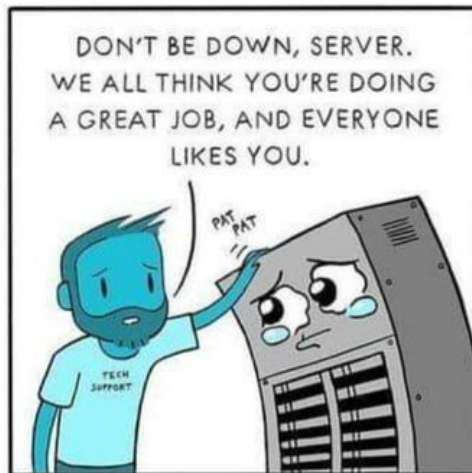
An intro to setting up Linux servers
and maintaining it

—

Caio Volpato (caioau)

Topics

- How [Redacted] platform works
- How to make your own Debian package
- Managing systemd services
- Firewalls
- Security best practices
- Monitoring
- Scheduled Tasks
- Disk failures
- Reliable remote only access principle
- Ansible: infrastructure as code
- Getting started and help on Linux
- Bonus: Raspberry pi specific stuff
- Bonus: “Onionize” your application



How [Redacted] platform works: Overview

How [Redacted] platform works: Details

How to make our own Debian package

Several times we needed to setup a new [Redacted] client machine, the tradicional procedure was **6 PAGES LONG**, which had many problems such as:

- Time consuming (~3 hrs)
- Super susceptible to human errors.
- Every time a new person that will start to run [Redacted] their user needed to created manually on every client machine and configure their environment. Now a shared user is used ([Redacted])
 - The user couldn't edit that machine's devices.txt and trigger executions on new devices.

How to make our own Debian package



How to make our own Debian package

Now with the package in place all the main problems were solved, and deploying a new [Redacted] client machine should be as easy as:

0. makepkg.sh (fetches the [Redacted] code and generates the package)
1. `sudo apt install ./package_1.0-1.deb` (installs the package)
2. InstallMissingDeps (Installs the python dependencies and making the package more generic)

Managing systemd services

All commands below need to be issued via sudo

- `systemctl status [--all] [service]`
 - `systemctl start/stop, enable/disable service`
 - `systemctl restart/reload service`
-
- `journalctl`
 - `journalctl -u service --since 2019-11-25`

Security best practices



OWASP -- Open Web Application Security Project

OWASP is an online community that gives a lot of recommendations on making your web application secure.

Recommended reading: [Threat Modeling Cheat Sheet](#)

Firewalls

- Iptables
 - ufw frontend
- nftables
- OpenBSD's PF

After issuing your firewall rules, keep your current SSH connection open and open a new SSH connection to make sure you won't get locked out.

Passwords

- [OC] My text (PT-BR): [P@ssw0rds: The weakest link of our security](#)
- [Relevant XKCD](#)
- A secure password requires to be: truly random, long enough (14+) and most important easy to remember.
 - Diceware method: [english wordlist](#) and [PT-br wordlist](#).
 - Example: panoramic nectar precut smith banana handclap
- Using a well established open source Password manager such as [KeePassXC](#) (or [lessPass](#) or [bitwarden](#)) is highly encouraged, and enable 2FA or even U2F (aka yubikey)
- Passwords requirements and expiring policies are full of shit, and do more harm than good
- New (2017) [Nist recommendations](#) should be used.
- When storing the passwords your app should always try to use argon2, and validate the password strength using [zxcvbn](#) and verify if it was compromised using for instance the [haveibeenpwned.com API](#)

Security updates

- Have you heard these names?
 - Shellshock
 - Heartbleed
 - dirtyCOW
 - KRACK
 - POODLE and logjam
 - Spectre, meltdown and cpu.fail
- Sign Up for the [Ubuntu security notices](#) , [Debian Security announces](#) and your framework security mailing list such: [Django announce](#).
- [Unattended upgrades](#): it's a native mechanism to automatically install updates.
- Install the latest microcode from your CPU vendor and run [spectre-meltdown-checker](#)

DoS attacks



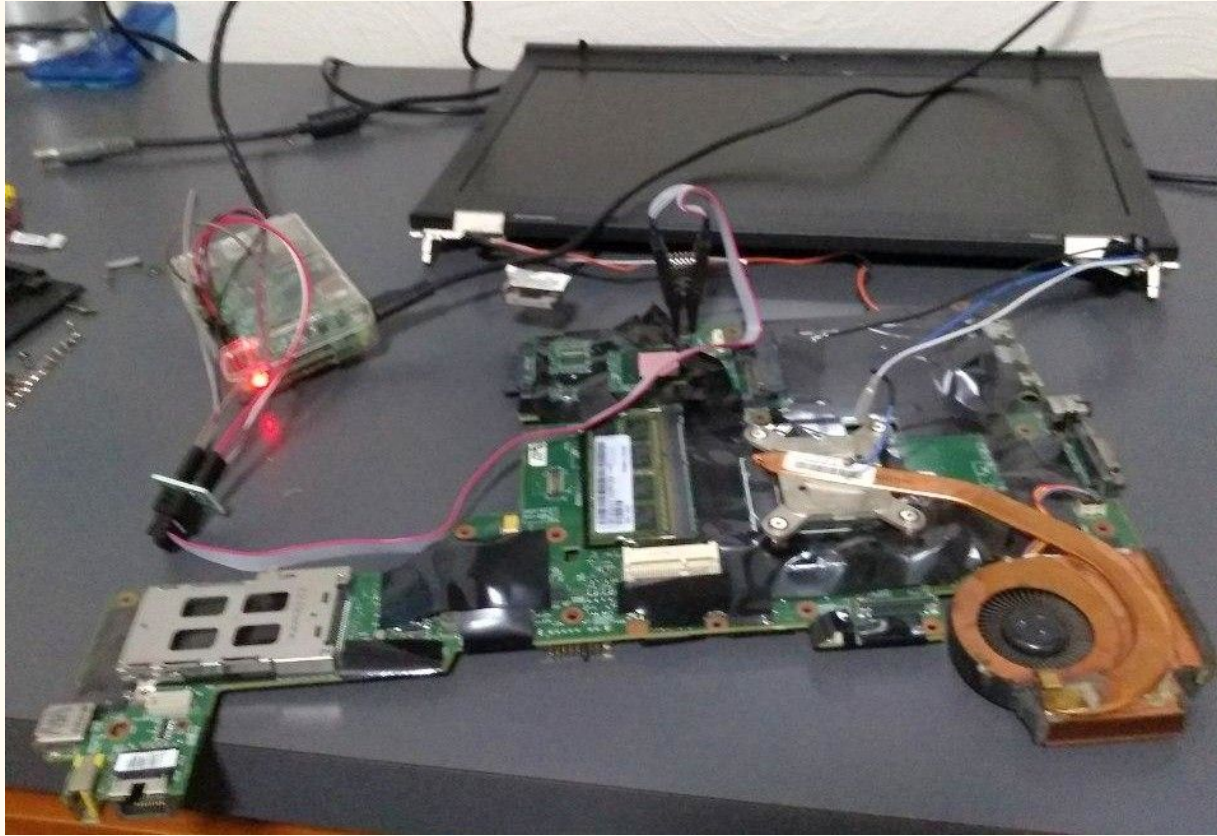
503
Service Unavailable

Source: <http://cat>

DoS attacks

- Rate limiting on the firewall
- Apache: mod_qos and mod_evasive, server_status
- Nginx: limit_req

Firmware security



Flashing coreboot+me_cleaner on
my personal laptop (thinkpad t430)

goto;

Ring 3: User space

Ring 0: Kernel

Ring -1: Hypervisor

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

goto; chicago

2:27 / 31:49

[source](#)

CHICAGO

GOTO 2019 • Why Open Source Firmware is Important • Jessie Frazelle

goto;

Adds up to: 2½ other kernels/OSes...

- They each have their own networking stacks, web servers (wtf)
- The code can modify itself and persist across power cycles and reinstalls

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

goto
chicago



12:07 / 31:49



Intel ME: A few examples of attacks

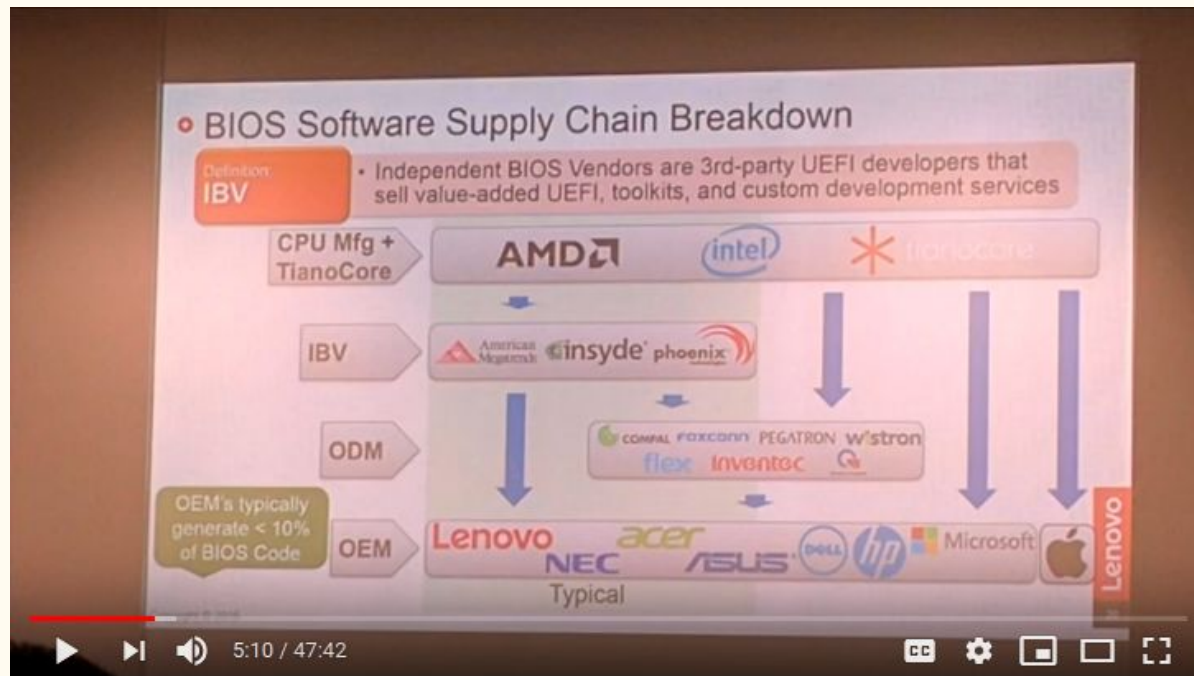
CVE-2017-5689

- An unprivileged network attacker could gain system privileges to provisioned Intel manageability SKUs: Intel Active Management Technology (AMT) and Intel Standard Manageability (ISM).
- This vulnerability could be exploited for 7 years, after intel released a patch, but the patch must be carried out by the OEMs.

CVE-2018-3628

- Buffer overflow in HTTP handler in Intel Active Management Technology in Intel Converged Security Manageability Engine Firmware may allow an attacker to execute arbitrary code via the same subnet.

BIOS SW Supply chain:



[source](#)

Bootstrapping a slightly more secure laptop (33c3)

4,798 views • Dec 27, 2016

76

0

SHARE

SAVE

...

goto;

Through open source,
visibility,
minimalism, and open
communication we can
push computing to a
better, more secure
place from the
hardware up.

goto
chicago



30:48 / 31:49



goto;

We can't keep building
on top of 🍌. We
really need to care
about the base we
build on.



30:58 / 31:49



Firmware Security: Cool alternatives:

- Me_cleaner: A script that from a BIOS dump it neutralizes the intel ME.
- Coreboot: Fast, secure and flexible open source BIOS available for several models such as ThinkPads, several chromebooks can install the mrchromebox. Some models can run without any proprietary blobs with libreboot.
- Heads: Coreboot based BIOS, with several hardening and security features such as super stronger protection Evil maid attacks.
- safeboot.dev: An alternative for who wants Heads, but your computer model is not supported by coreboot/heads.
- Cool vendors that are doing a great firmware security job:
 - Purism
 - system76


Debian vs Ubuntu

Posted by u/Grevillea_banksii **Glorious OpenSuse** 1 year ago

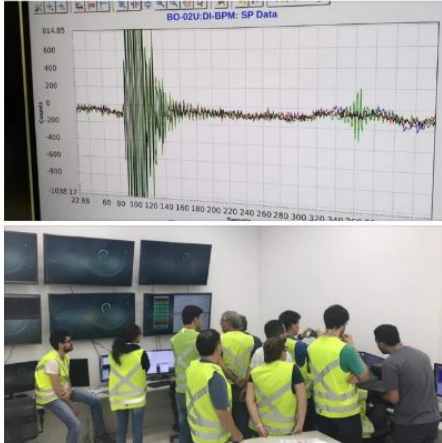
1.1k

The biggest Particle Accelerator in the Southern hemisphere runs on Debian - CNPEN - Campinas, Brazil

Other flair please edit

 **Cnpem - Centro Nacional de Pesquisa em Energia e Materiais** 🇧🇷 feiert diesen besonderen Tag.
4 Std. · 🌐

EXTRA! Tivemos agora há pouco a primeira volta dos elétrons no acelerador injetor do Sirius, conhecido como Booster!
É no Booster que os elétrons circulam para ganhar energia e velocidade, até que atinjam os níveis adequados para que possam gerar a tão desejada luz síncrotron. Quando estão "prontos", eles seguem para o acelerador principal. O Sirius possui três aceleradores, e agora já temos elétrons circulando em dois deles! 😊 #VaiSirius



137 Comments Share Save ...

99% Upvoted

[source](#)

Ubuntu vs Debian

Ubuntu is based on Debian, but it grew so much apart that there's a lot of important differences: [A newbie's newbie guide to Debian, by Helen Koike](#)

- Debian is a non profit project (unlike Ubuntu and Fedora)
- All Debian volunteers sign the Debian social contract.
- Ubuntu has a [spyware/keylogger](#) installed by default (since 2012).
 - And [threatened to sue a website which contains instructions do remove the spyware](#)
- Debian is rock solid Stable.
 - Debian requires reboots every ~4 months while ubuntu ~2 weeks.
 - Ubuntu often breaks the system when updating (most release upgrade break the system)
- Most Ubuntu mirrors doesn't even have HTTPS ([CVE-2019-3462](#))
 - Debian not only have HTTPS but also onion: [onion.debian.org](#)
- Most Debian packages are [build reproducible](#) (fundamental for trust and security)
- Debian doesn't require frequent updates (only bugfix and security)
- Ubuntu has a lot of broken packages (such as usbguard, firejail, munin)

Other “GNU/Linux distros” to look for.

- Tails: The Amnesic Incognito Live System, it's a security focused distro installed on CDs or flash drives, not leaving any trace on the computer, and routing all traffic via Tor.
- Qubes: A reasonably secure operating system. A system that has a bunch of VMs (using Xen) to compartmentalize all the activities.
 - Security Through Distrusting -- Joanna Rutkowska
- GuixSD: A distro made around the super complete and flexible guix package manager, inspired by NixOS
 - Solving the deployment crisis with GNU Guix
- OpenBSD: Not a Linux distro!!! It's a BSD security focused system.
 - In more than **20** years of the project it only had **2** remote holes vulnerabilities in the base install: Why OpenBSD rocks! Runbsd.info

SSH best practices



[source](#)

SSH best practices

- Good practices:
 - Secure cipherlist
 - rate limit in the firewall or fail2ban
 - pubkey auth only (disable password auth)
 - port knocking
 - 2FA
 - Recent versions of OpenSSH supports using any U2F key as smartcard, both the client and server most support it :(
 - Access only via Onion, this way only you who knows the onion addr can connect.
- SSH tarpit
- Securing your ssh keys:
 - Generate the key in a air gapped machine and only use it in a yubikey
 - QubesOS SSH split mode

Monitoring

WHO WOULD WIN?

an army of penguins



a bunch of weird symbols

:(){ :|: & }::

WHO WOULD WIN??

Monitoring

- [How Regular Expressions and a WAF DoS-ed Cloudflare](#)
 - [Official report](#)
- Monitoring solutions: Old school Munin, prometheus and zabbix
- Notifications:
 - mailbot
 - [healthchecks.io](#)
 - [uptimerobot](#)
 - [Gotify](#)

Scheduled Tasks

- cron: user tabs, /etc/cron.{weekly,hourly ...}/, /etc/cron.d/
- systemd timers
- Lock files

Disk Failures



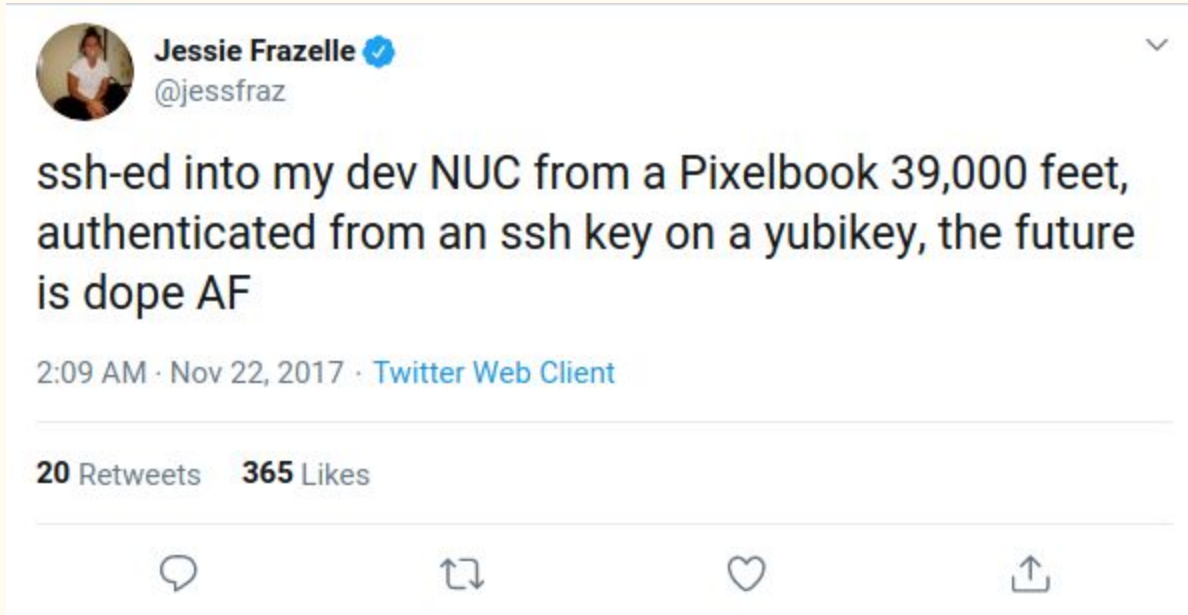
Disk failures

- Smartctl: `sudo apt install smartmontools`
 - `sudo smartctl -l long /dev/sda`
 - `smartctl -H /dev/sda`
- fsck every boot: `tune2fs -c 1 /dev/sda1; adjust sda1 , use lsblk command`
- fsck with badblocks: `fsck.ext4 -vcDfty -C 0 /dev/sda1 ; -c that runs badblocks(read)`
- io latency in the monitoring

Backup solutions

- Tools: rsync (or unison), syncthing, rclone, borgbackup, duplicity.
- Disk level (softRAID): LVM, BTRFS, ZFS
- Cloud: **There's no cloud just other people's computer:** So make sure to securely **encrypt before sending it to the cloud:** Use cryptomator, tarsnap, or enable encryption on backup solution.

Reliable remote only access principle



[source](#)

Reliable remote only access principle

We need to be able to access all the machines reliably and without the need of physical intervention. In order to do that we can:

- Cool project: mosh mobile shell: Better “ssh” for bad connections
- Remotely unlock encrypted disks: so we don't need to type the password into the keyboard (dropbear-initramfs).
- **[Redacted]**:
 - Dummy screen: we don't need a physical monitor to control the screen. (xorg.conf)
 - VNC server: So we can access the dummy screen remotely.
 - Autologin so the user password is not prompted
 - Seahorse: Disable password to encrypt the vault
 - Desktop autostart: skype autostart after boot.
 - Firejail: we can run 2 skypes at the same time, with 2 different accounts. (/home is mounted as different dir)

Ansible: infrastructure as code

Ansible is a powerful server and configuration management tool. It's similar to Chef, Puppet or Saltstack.

The biggest advantage is that ansible is agentless (so the managed machines only need a python interpreter installed, no additional installations), it's super easy to make playbooks/roles it's just a yaml file, and the available modules that do the heavy work.

It can be used to setup a machine to install an application or manage and do all the necessary maintenance in all the servers.

The best way to test your playbook is using vagrant: That creates your environment in a VM, which can be easily re-created or destroyed.

Ansible: Concepts

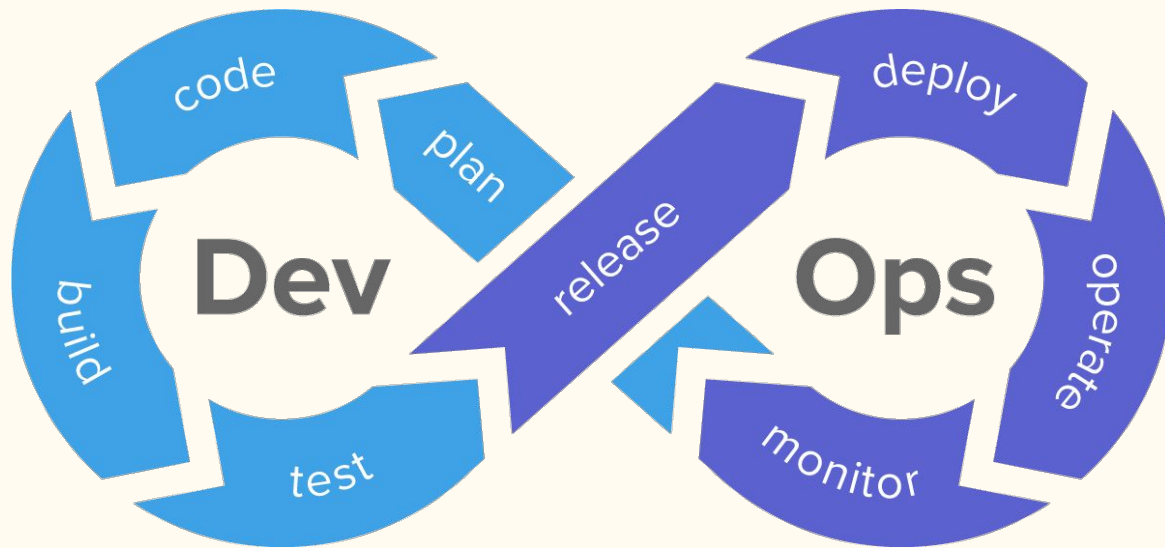
- Inventory
- Playbook
- Roles
- Tasks
- Modules

Ansible: Good reference:

I recommend the book: [Ansible for DevOps](#)

Using ansible-lint helps a lot, because it gives good practices and tips for your playbooks.

DevOps Overview:



Useful commands

- Fish shell: the friendly interactive shell
- Script / scriptreplay: record your terminal session, so you can lookup later.
- Wormhole and onionshare: send files
- hostnamectl: discover everything about the machine: what distro if its a vm ...
- Watch: run a command every x seconds
- netcat (nc): TCP/IP swiss army knife
- Netstat or ss: show several network stuff
- curl cheat.sh/tar ; commands cheat sheets
- Know your shell (bash): HISTTIMEFORMAT and HISTCONTROL
- Arpscan: the arp scanner
- Nmap: Network exploration tool and security / port scanner
- Rssh: restricted secure shell allowing only scp and/or sftp
- tmate : Instant terminal sharing

Getting help



The screenshot shows a GitHub discussion thread with four posts. Each post includes a user profile picture (a grey square with a white arrow), a username, a score, a hidden status, and a timestamp. The posts are: 1. User **brianatwork_** (score hidden, 1 hour ago) asking "Do people still use IRC?". 2. User **heeen** (score hidden, 52 minutes ago) replying "Open source developers do". 3. User **myxomat0sis_** (score hidden, 16 minutes ago) replying "but what about people". 4. User **onthefence928** (score hidden, 13 minutes ago, with a star icon) replying "cries in open-source". Each post has a row of action links: "permalink", "embed", "save", "parent", "report", "give gold", and "reply".

↑ [-] **brianatwork_** [score hidden] 1 hour ago
↓ Do people still use IRC?
permalink embed save parent report give gold reply

↑ [-] **heeen** [score hidden] 52 minutes ago
↓ Open source developers do
permalink embed save parent report give gold reply

↑ [-] **myxomat0sis_** [score hidden] 16 minutes ago
↓ but what about people
permalink embed save parent report give gold reply

↑ [-] **onthefence928** [score hidden] 13 minutes ago ☆
↓ *cries in open-source*
permalink embed save parent report give gold reply

[source](#)

How to get help:

- [/r/linux4noobs](#) is great for beginners.
- DigitalOcean has a lot of great [tutorials](#)
- A great place deeply understand something is the [archlinux wiki](#)

Great communities:

- [casahacker HC](#) and [LHC HC](#)
- DevOps Campinas slack group
- Unicamp groups:
 - [Enigma](#) (cryptography, privacy and security study group)
 - [LKCamp](#) (Linux Kernel study group)
 - Archventure Time (Arch Linux group)

What I can do for learning:

- Host a Tor relay or bridge
- Use Linux in your computer, start with an easy distro such as Linux Mint, then when you're confident use Arch Linux or Gentoo.
 - 10 ways Linux is just better!
- Self Hosting is always fun:
 - Nextcloud instance: so you can host your files like dropbox, calendar, contacts, notes and kanban.
 - Plex server: so you can host your media.
 - Ttrss: Best way to burst out your feed bubble: follow anything RSS.
 - Jitsi web: Host your own video conferences.
 - Wireguard vpn
 - awesome-selfhosted list

Other cool topics to learn:

- [Linux desktop hardening](#)
- Containers and virtualization.
- Disk encryption: LUKS, Veracrypt (awesome hidden volumes), ecryptfs
- Encryption: SSL/TLS (Let's encrypt,FS) , security http headers, PGP, Steganography (Stegosuite)
- Anonymize your users data: differential privacy, homomorphic encryption

Awesome resources to follow:

- Youtube:
 - [Chris Titus Tech](#)
 - [LearnLinuxTV](#)
 - [Level1Linux](#)
 - [Luke Smith](#)
 - [Quidsup](#)
 - [Wolfgang's Channel](#)
 - [LINUXtips](#)
 - [FiqueEmCasaConf](#)
- [Media.ccc.de](#): Chaos Computer conference recorded talks.
- Podcasts:
 - [Jupiterbroadcasting](#): podcast network with dozen podcasts.
 - [The Syscast podcast](#)
- [Julia Evans](#): awesome zines to learn about a lot tech things.
- [Guiafoca.org](#): (PT-BR) awesome resource to learn about computers and GNU/Linux

Summing up:



Summing up:

Your platform absolutely needs:

1. Always have the latest security updates installed.
2. A backup solution, which is automatic, periodic and monitored (but also verify from time to time (Two Generals' Problem)).
3. In case of hardware failure, you need to know how to setup your environment.
4. A monitoring solution, so you can get notified of any problem.

Other things that are nice to have:

1. Deploy method, such as a package, or better yet a CD solution too.
2. Automatic configuration tool, such as ansible.

Bonus: Raspberry pi specific stuff

A lot of people use their pies to host some of their stuff, but raspberry pi is not reliable as ordinary computers, so **don't host anything critical on pies**, because:

- SD card corruption (wearing out)
- Power problems: not enough power, power not stable

In order to mitigate that we can:

- Disable disk swap and it's a good idea to use zram instead ;)
- Logs and journals write to RAM, not to the card.
- Run fstrim periodically, to wear level ...
- `noatime` mount flag, to write less stuff to the card.

Bonus: “Onionize” your application

Unless your application is a bank system, you can consider to allow your users to access your app via onion, so the users have complete anonymity.

For instance, facebook can be accessed via facebookcorewwi.onion, surely all the actions that a user does in the site are logged, like the clearnet version, but having an onion can help for instance journalists to access it without their ISP or gov knowing it. It's accessed by 1 million users monthly as of April/2016 ([source](#)).

Tor can also help with censorship: [Roger Dingledine - The Tor Censorship Arms Race](#), for instance [Most of the popular Brazilian ISPs started blocking safe abortion website](#)